

C A R T O G R A F Í A
DE LAS IDEAS ACTUALES SOBRE LAS
MÁQUINAS INTELIGENTES

DE LOS **PROBLEMAS REALES** A LA **FICCIÓN**, DE LA **FICCIÓN**
A LA **INTELIGENCIA ARTIFICIAL**, Y DE LA **INTELIGENCIA**
ARTIFICIAL PASEMOS NUEVAMENTE A LA **FICCIÓN**



FRANCISCO JOSÉ SERÓN ARBELOA

PRENSAS DE LA UNIVERSIDAD DE ZARAGOZA



IASAC

Inteligencia Artificial y Sistemas Autónomos Cognitivos



Financiado por
la Unión Europea

<<http://unidigitaliasac.unizar.es/>>

<<http://unidigitaliasac.unizar.es/contenidos-iasac>>

<<http://unidigitaliasac.unizar.es/ficha/una-vision-general-de-la-ia>>

Cartografía de las ideas actuales sobre las máquinas inteligentes

Solo podemos mandar hacer a una máquina aquello que conocemos.

Subtítulo

De los problemas reales a la ficción, de la ficción a la inteligencia artificial, y de la inteligencia artificial pasemos nuevamente a la ficción

Debemos recordar que la ficción no significa falsedad.

Sir Arthur Helps

Autor: Dr. D. Francisco José Serón Arbeloa
Catedrático de Universidad del Área de Lenguajes y Sistemas Informáticos
Universidad de Zaragoza: <<http://www.unizar.es/>>
C. V. completo: <<http://webdiis.unizar.es/~seron/>>
<<http://cgit.unizar.es/>>

El texto está disponible bajo la Licencia Creative Commons Atribución Compartir Igual 3.0; (CC BY-SA 3.0) (<<https://creativecommons.org/licenses/by-sa/3.0/deed.es>>).

El siguiente documento es una copia en formato pdf del material utilizado en clase por el Dr. Francisco José Serón Arbeloa que ha ido recopilándose durante muchos años para la impartición de diferentes asignaturas incluidas en los planes de estudios vigentes de la Universidad de Zaragoza, de la Universidad de la Experiencia de la Universidad de Zaragoza, así como la colaboración en el proyecto UNIDIGITAL_IASAC 2022.

Consideraciones:

- *La estructura lógica es original del profesor Dr. F. J. Serón Arbeloa.*
- *Las referencias explícitas aparecen en el texto o en el apartado de bibliografía.*
- *Los textos o la información recopilada pueden ser:*
 - *Originales del profesor.*
 - *Recogidas de Internet en sitios en los que se hace constar expresamente el permiso de utilización.*
 - *Recogidas de Internet en sitios en los que no se hace constar expresamente el permiso de utilización, pero tampoco se impide.*

Por lo tanto, los derechos de autor de dicho documento pueden considerarse colectivos, aunque de autores muchas veces desconocidos o por el contrario fácilmente detectables.

Realizadas las consideraciones previas, dicho material se entrega con las siguientes condiciones:

- *Es de uso exclusivo para su estudio.*
- *No puede ser transferido a ninguna otra persona, pero puede indicarse la dirección de Internet de la que se puede descargar.*
- *No puede ser utilizado para ninguna actividad comercial.*

Índice

Prólogo	8
Disculpa	13
Prefacio (objetivo del libro)	14
Un mundo con problemas	17
<i>Problemas</i>	17
<i>Decisiones</i>	17
<i>Contextos de decisión</i>	18
<i>Tipos de problemas</i>	21
Según E. F. Schummacher	21
Según los dominios de D. Snowden	22
Según el razonamiento utilizado para resolverlo.....	24
<i>Ejemplos de problemas que hay que resolver</i>	27
<i>Racionalidad</i>	29
¿Cuán racional es el <i>homo sapiens</i> ?	29
Racionalidad e irracionalidad.....	30
¿Qué le pasa a la gente?	31
Por qué es importante la racionalidad	41
<i>Las tres componentes de las posibles soluciones</i>	43
Gente artificial. De la ficción a la realidad.....	49
<i>¿De qué va esto?</i>	49
<i>Un mundo de ficción</i>	49
<i>Bases ideológicas</i>	50
<i>El entorno tecnológico actual</i>	52
Reflexión histórica sobre los artefactos	53
Reflexión histórica sobre las revoluciones tecnológicas	54
Reflexión histórica sobre la tecnología digital	55
Un anhelo.....	59
<i>El marco conceptual</i>	62
La inteligencia artificial (IA).....	62
La vida artificial (VA).....	62
El denominador común	63
<i>Por dónde se prevé que hay que seguir</i>	64
<i>Conclusiones</i>	65

Planteamiento de una pregunta	66
<i>La caída de algunos mitos</i>	66
<i>Definición de sexta caída</i>	66
Qué es un algoritmo inteligente. ¿Estamos en la sexta caída?	68
<i>Primera parte. Qué es un algoritmo inteligente</i>	68
Introducción	68
Qué es un algoritmo	68
Cómo se hace un algoritmo	70
Qué es eso de la inteligencia artificial clásica	71
Qué es un algoritmo inteligente	73
¿Hasta dónde pueden llegar?	73
Pregunta, ¿pero estamos en la sexta caída?.....	73
<i>Segunda parte. Hagamos bien la pregunta</i>	74
¿Todo es algoritmizable?	74
Ejemplos de problemas que no admiten solución algorítmica	82
Ejemplos de problemas que admiten solución algorítmica y son intratables	84
Ejemplos de problemas que admiten solución algorítmica y son tratables	85
La pregunta bien hecha	85
<i>Conclusiones</i>	86
Teoría de la información	88
<i>Velocidad de los avances tecnológicos</i>	88
<i>El alfabeto, la gramática y la lógica</i>	92
<i>Los números</i>	99
<i>La información</i>	102
Mapa cartográfico de la inteligencia artificial	124
<i>Conceptos básicos sobre limitaciones</i>	129
<i>Algoritmos de búsqueda en grafos (modelo simbólico)</i>	132
Un ejemplo completo	141
Algoritmos de búsquedas no informadas o búsquedas a ciegas	160
Algoritmos de búsquedas no informadas con información parcial	164
Algoritmos de búsquedas informadas mediante heurísticas y exploración.....	168
Algoritmos de búsqueda local.....	172
Problemas de satisfacción de restricciones (PSR)	179
<i>Algoritmos basados en el uso de la lógica (modelo simbólico)</i>	193

Lógica.....	194
Lógica proposicional.....	198
Otros tipos de lógica.....	218
Lógica difusa o Lógica Fuzzi.....	226
<i>Planificación.....</i>	<i>238</i>
<i>Algoritmos basados en el uso de la probabilidad (modelo simbólico).....</i>	<i>260</i>
Probabilidad.....	261
Razonamiento probabilístico.....	271
Redes bayesianas.....	275
Razonamiento probabilista en el tiempo.....	287
<i>Toma de decisiones. Teoría de la utilidad.....</i>	<i>318</i>
<i>Algoritmos de aprendizaje.....</i>	<i>352</i>
<i>Problemas de los algoritmos de aprendizaje.....</i>	<i>360</i>
<i>Tipos de algoritmos de aprendizaje.....</i>	<i>365</i>
Algoritmos de aprendizaje supervisado (modelo simbólico).....	365
Algoritmos de aprendizaje no supervisado (modelo simbólico).....	389
Algoritmos de aprendizaje por refuerzo (modelo simbólico).....	400
<i>Tipos de algoritmos bio-inspirados.....</i>	<i>413</i>
Redes neuronales artificiales (modelo conexionista).....	413
Redes neuronales artificiales.....	415
Redes neuronales profundas.....	424
Comentario final sobre las redes neuronales artificiales.....	455
Optimización y algoritmos genéticos (modelo evolutivo).....	456
Algoritmos de comportamiento colectivo.....	466
Sistemas cognitivos (modelo corpóreo).....	467
<i>La inteligencia.....</i>	<i>469</i>
<i>Principios de diseño de sistemas corpóreos.....</i>	<i>472</i>
Los problemas fundamentales de la IA clásica y de las ciencias cognitivas.....	476
Escritos de Santiago Sánchez-Migallón.....	483
Categorías de definiciones sobre IA.....	484
Las leyes de la robótica de Asimov.....	485
Sentido común.....	487
Historia de la inteligencia artificial.....	490
Epílogo.....	504

Vuelta a la ficción.....	508
Referencias bibliográficas recomendadas para saber mucho más.....	509
Sobre el autor	511

Prólogo

No hay nada más humano que una máquina.

Anónimo

Durante la primera parte de mi juventud (1972-1977) [17-22] años, siendo estudiante de físicas, cuando curioseaba entre las estanterías de las librerías de mi ciudad en busca de un libro nuevo que leer, buscaba fundamentalmente obras de ciencia ficción, de terror, de divulgación científica relacionada con la naturaleza (Geología, Zoología, Botánica), del universo, sus leyes y su evolución. Por suerte para mí, siempre encontré libros interesantes desde mi incipiente punto de vista.

En este momento y por el motivo de este libro, voy a rescatar las tres leyes relacionadas con el avance científico que fueron propuestas por el escritor británico de ciencia ficción Arthur C. Clarke (1917-2008) que en la edición revisada del año 1973 del ensayo «Hazards of prophecy: the failure of imagination» («Peligros de la profecía: la falta de imaginación») que se encuentra en el libro *Profiles of the future* (*Perfiles del futuro*, de 1962):

1. Cuando un científico eminente pero anciano afirma que algo es posible, es casi seguro que tiene razón. Cuando afirma que algo es imposible, muy probablemente está equivocado.
2. La única manera de descubrir los límites de lo posible es aventurarse un poco más allá, hacia lo imposible.
3. Cualquier tecnología lo suficientemente avanzada es totalmente indistinguible de la magia.

En noviembre de 1980 entré como profesor ayudante del aquel entonces Departamento de Matemática Aplicada de la Universidad de Zaragoza en el que permanecí hasta el año 1990. Durante esa estancia realicé mi tesis doctoral en Ciencias Físicas simulando la propagación de ondas sísmicas utilizando métodos de elementos finitos y trabajando con supercomputadores vectoriales y paralelos.

Posteriormente se fundó el Departamento de Informática e Ingeniería de Sistemas y para allí que me fui con el objetivo de poner en marcha un grupo de investigación especializado en Informática Gráfica. Personalmente en el grupo trabajé en simulación de la propagación de la luz, con objeto de generar imágenes fotorrealistas hasta que en un momento determinado leí el artículo «Cognitive Modeling: Knowledge, Reasoning and Planning for Intelligent Characters» («Modelado cognitivo: conocimiento, razonamiento y planificación para personajes (animados) inteligentes») de los autores John Funge, Xiaoyuan Tu y Demetri Terzopoulos, publicado en 1999.

La figura siguiente es una adaptación que yo hice posteriormente de la figura 1 de dicho artículo.

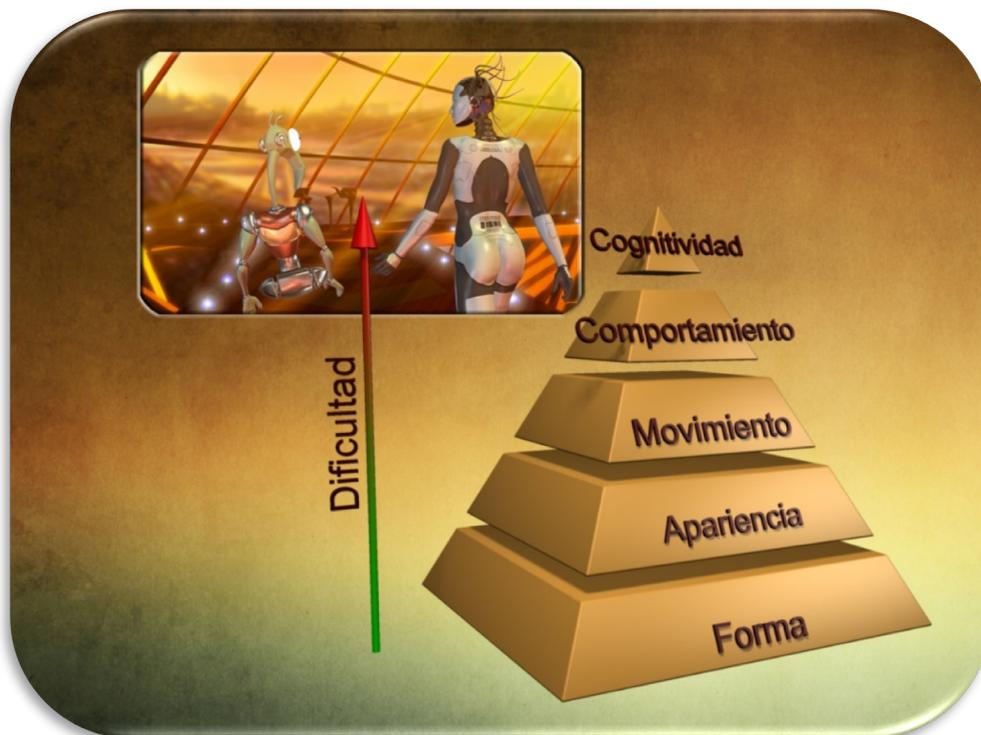


Figura. Jerarquía de los diferentes tipos de modelado que intervienen a la hora de abordar el desafío de automatizar un personaje inteligente.

En ella se desgranar los diferentes problemas que hay que ir resolviendo jerárquicamente para contarle a un computador qué es un personaje inteligente capaz de deambular por un mundo virtual. Una descripción rápida de cada uno de ellos podría ser la siguiente:

- **Forma:** Hay que describirle la geometría del personaje, es decir, cómo ocupa el espacio. Problema de definición basado en la rama de la Geometría que en el momento actual está completamente resuelto.
- **Apariencia:** Hay que definir el comportamiento óptico de los diferentes materiales que forman la superficie del personaje y simular la interacción de la luz con ellos, todo ello desde algún punto de vista determinado. Problema de simulación basado en la rama de la Óptica que en el momento actual está completamente resuelto.
- **Movimiento:** Hay que modelar los movimientos de cada una de las partes del cuerpo y su interacción con el medio que lo rodea. Problema de simulación basado en la rama de la Mecánica que en el momento actual está completamente resuelto.

- **Comportamiento:** En ese momento consistía en conseguir personajes que se movieran de forma autónoma reaccionando de manera apropiada a los estímulos percibidos a través de sus sentidos en su interacción con el entorno que los rodeaba. Problema de modelado sensorial muy simplificado basado en técnicas de «soft computing» rama de la inteligencia artificial desarrollada durante los noventa para solucionar problemas que manejan información incompleta, con incertidumbre y/o inexacta. En este caso el problema estaba adecuadamente resuelto, aunque hoy en día sigue en desarrollo.
- **Cognitividad:** Y por fin, llegamos al vértice de la pirámide. En el momento de su publicación, lo que se pretendía presentar era una propuesta que fuera más allá del modelado comportamental que permitiera modelar lo que un personaje sabe, cómo adquiere ese conocimiento y cómo puede utilizarlo para planificar sus acciones en el mundo en el que está inmerso. Dicha propuesta pretendía crear una nueva raza de personajes autónomos, casi-inteligentes que estaban empezando a ser demandados por los diseñadores de juegos, de producciones de animación y cinematográficas.

En el año 2003 propuse a algunos miembros del grupo que había fundado, trabajar en la definición de algún tema novedoso para investigar e innovar durante los siguientes años. Recuerdo que acababa de ver la película *The Time Machine* del año 2002. Del director Simon Wells y del productor Arnold Leibovit, basada en el libro de H. G. Wells *The Time Machine* de 1895 y en la película de David Duncan de 1960, con el mismo título. Conseguí una secuencia de la película en la que Alexander, el viajero en el tiempo, se dirige en el futuro a la biblioteca pública de la Quinta Avenida. Allí conoce a Vox 114, un bibliotecario creado gracias a la inteligencia artificial, el cual le informa sobre las aplicaciones prácticas de los viajes en el tiempo y le asegura que la idea de viajar al pasado es imposible. El personaje del bibliotecario es un holograma con capacidad visual y verbal conectado a todas las bases de datos del planeta. En YouTube <<https://www.youtube.com/watch?v=CQbkhYg2DzM>>, pueden ver la impresionante escena que me atrapó y definió los siguientes años de mi dedicación a la I+D+i.

Durante el año 2003 sopesamos lo que sabíamos, lo que teníamos que aprender y los problemas que teníamos que resolver, y decidimos que el tema era factible, interesante y parecía adecuado para conseguir fondos de manera sucesiva tanto de Aragón, como de España y de la Unión Europea.

El camino fue largo, pero muy productivo e interesante. Dedicamos el 2004 al tema de la forma, el 2005 a la de la apariencia, el 2006 al del movimiento, el 2007 y 2008 al tema del comportamiento y el 2009 y el 2010 a algunos temas concretos de cognitividad. A partir de esas fechas nos pusimos manos a la obra y durante el periodo 2011-2015 conseguimos fondos de la Unión Europea, del Ministerio español de Economía y Competitividad, del Ministerio de Industria, Energía y Turismo, del

Departamento de Industria e Innovación del Gobierno de Aragón, y otra vez del Ministerio de Economía y Competitividad.

Uno de nuestros artículos que habla un poco de todo lo conseguido es «VOX system: a semantic embodied conversational agent exploiting linked data», de los autores F. Serón y C. Bobed, publicado en la revista *Multimedia Tools and Applications*, del año 2016, vol. 75 (1), pp. 381-404.

La experiencia global fue personalmente interesante, leí y aprendí sobre inteligencia artificial y sobre sistemas cognitivos y todavía tenemos interesantes temas pendientes.

Durante el mismo periodo, en todos los medios de comunicación se ha hablado y se habla mucho de «inteligencia artificial» y un poco menos de «sistemas cognitivos», ambos como producto de la inteligencia natural. Por otro lado, es verdad que lo que ya hay en la actualidad está transformando el mundo y también hay mucha gente que nos avisa que puede llegar a superar a la humana. Hay que tener en cuenta que la inteligencia artificial, aparte de una ciencia o una tecnología es, ante todo, una industria. Al parecer, ahora todo lo que nos rodea se va volviendo inteligente, como los móviles, las televisiones, las casas, las aplicaciones informáticas... De hecho, muchas de las aplicaciones más conocidas, se han ido haciendo expertos en ti. Ya existen esos asistentes virtuales especializados en servirte y cuya IA reside en aprender más y más sobre tus gustos y costumbres. En la actualidad, la actividad se ha redireccionado hacia las demandas de unos usuarios que quieren máquinas que les faciliten más su vida.

Además, por mi cargo de vicerrector de Prospectiva, Sostenibilidad e Infraestructuras durante el periodo 2016-2020 me he podido mover por muchos campus universitarios y he dado conferencias en algunas titulaciones. Y es curioso analizar lo que las personas tienen en la cabeza cuando les interrogas sobre qué entienden por las siglas IA y SC, salvo si hablas con especialistas, el resto lo que tiene es una verdadera empanada mental. Es verdad lo que decía Clarke en su tercera ley, «Cualquier tecnología lo suficientemente avanzada es totalmente indistinguible de la magia».

Por ello, uno de los mayores problemas a los que se enfrenta el desarrollo de la inteligencia artificial, según todos los expertos es que, una vez identificada como área estratégica, su irrupción requiere de una formación más adaptada a la sociedad digital en la que nos encontramos, así como el desarrollo de competencias particulares en nuestro sistema educativo. No está tan claro que nuestras propuestas académicas actuales estén en sintonía con estas necesidades. Resultado de todo ello es que opino que una sociedad que aplica confusamente esas palabras debería hacérselo mirar, por ello voy a intentar hacerles entender de qué estamos hablando.

En calidad de profesor universitario y educador, mi inclinación natural es querer compartir lo que sé. Escribir este libro me ha dado la oportunidad de adentrarme en una serie de áreas de manera minuciosa. No pretendo hacer una revisión exhaustiva de la IA. Mis metas son más modestas, simplemente pretendo presentarles a algunas de las criaturas más interesantes que han propiciado el avance de la IA.

Este libro va dirigido a todas aquellas personas de entre 17 y 75 años que no son especialistas en nada de lo que voy a hablar, pero consideran lúcidamente que no se puede avanzar por el siglo XXI sin tener unas nociones mínimas de por qué el mundo se está transformando. Confío en ejercer de algo parecido a un guía mientras emprendemos juntos un viaje, a un mundo al parecer misterioso que solo estamos empezando a comprender. Si se consiguen esas nociones, podremos mejorar nuestro propio criterio a la hora de entender el escenario en el que estamos inmersos. Como iremos viendo a lo largo de sus páginas, los seres humanos somos una gente brillante pero particularmente negada para razonar sobre problemas complejos grandes, por ello necesitamos del apoyo de las máquinas.

Con este libro, me pasa lo que expresó muy bien George Lucas hablando de su actividad creadora cinematográfica. Le pasa a todo el mundo. Te viene una idea de algo que crees que podría molar y luego luchas por hacerla realidad. Es real en tu cabeza, vagamente, pero cuando lo pones por escrito, dices: «Vaya, en mi sueño no era para nada así».

Disculpa

Utilizo las palabras leídas al escritor Arturo Pérez Reverte que expresa con su particular estilo lo que a mí me gustaría expresar en estos momentos.

Si ustedes escriben, les habrá pasado alguna vez. A mí me pasa. Trabajas un montón un texto, lo corriges, lo maquetas, y cuando te lo echas a la cara, en la primera página que abres salta el gazapo, es decir, una metida de gamba: planchazos que a veces te hacen decir, tierra trágame. Además, siempre hay lectores que saben más, y nunca falta el espabilado que te dice: en esta lo he pillado amigo. Y tú te lo zampas estoico, das las gracias y lo corriges en la siguiente edición. En lo que a mí respecta, será bienvenida cualquier corrección, sugerencia o mejora.

Con respecto a las referencias he intentado citar todas aquellas que han ido sobreviviendo a lo largo de los años y no he perdido. Si alguien echa en falta alguna, por favor que me la envíe.

Prefacio (objetivo del libro)

Con el arranque de la carrera espacial tras la Segunda Guerra Mundial, la idea de viajar al espacio pasó de ser un sueño disparatado a convertirse en una posibilidad real. Una parte de ese cambio de actitud se le puede atribuir a los artículos publicados en la prensa popular, que especulaban sobre la posibilidad de realizar dichos viajes. Los artículos explicaban a grandes rasgos los conceptos del trabajo del ingeniero sobre los viajes espaciales, sin embargo, las ilustraciones de las lanzaderas eran tan sugerentes como una tira cómica.

Para los estadounidenses de mediados del siglo pasado, el futuro de la humanidad en el espacio estaba tan solo a un par de décadas de distancia. Las series de televisión, las películas y los libros sobre inteligencia artificial (IA) se encargaron de generar esa sensación.

Las películas realistas sobre el espacio alcanzaron su apogeo en 1968 con *2001: Una odisea del espacio*. Las películas combinaban la estética con una historia que hacía pensar al espectador sobre la evolución y sobre el lugar de la humanidad en el universo. A pesar de la genialidad del largometraje, el entusiasmo por los mundos que están más allá de la Tierra decayó durante esa época. La realidad de la exploración espacial resultó ser lenta, rutinaria y, en cierta forma, aburrida, al menos para la mayoría de la audiencia.

Por supuesto *La guerra de las galaxias* (1977) se encargó de demostrar lo contrario. Nunca una película de ciencia ficción había conseguido un éxito semejante, tanto en la taquilla como en la imaginación del público. Una de sus características principalmente innovadora era lo real que parecían los acontecimientos. Todos los países querían formar parte de la acción e invirtieron mucho dinero en la ciencia ficción y las historias del espacio. Lo que permitió que el género se expandiese más allá de sus límites, hacia nuevos territorios, demostrando la madurez suficiente para abordar temas complejos. Esas películas presentan a la IA como un lugar de horror intenso psicosexual, o como la lucha entre la avaricia y la lealtad, donde el espacio se convirtió en una palestra geopolítica compleja y en un crisol en el que los enemigos tenían que aprender a confiar los unos en los otros. Los últimos años de los setenta y toda la década de los ochenta supusieron una nueva edad de oro para las películas del espacio con mucho presupuesto.

La literatura también expandió sus horizontes, pues las editoriales añadían nuevos títulos de ciencia ficción a su catálogo para aprovechar el aumento de popularidad del género. Las historias demuestran que las representaciones realistas de los desafíos a los que tendrán que enfrentarse las personas en el espacio exterior levantan tanta fascinación como los exóticos paisajes y los campos de acción y aventura. A la vez, los relatos ambientados en los lugares más alejados del espacio satisfacen nuestra necesidad de pensar a lo grande sobre lo que podríamos ver y

hacer allí fuera. El espacio, tan cerca y a la vez tan lejos, sigue interesándonos, y siempre será el corazón de la ciencia ficción.

En las historias de ciencia ficción, tanto en la Tierra como en galaxias lejanas, suelen aparecer máquinas inteligentes, a menudo como los protagonistas. Pueden ser inteligencias artificiales incorpóreas, como el ordenador Hal 9000 que dirigía la nave espacial hacia Júpiter en *2001: Una odisea del espacio* (1968), con cuerpos mecanizados como Gort, el enorme robot de *Ultimátum a la Tierra* (1951, 2008), o el elegante y encantador C-3PO de *La guerra de las galaxias* (1977), o androides con aspecto humano como T-800 de *Terminator* (1984), el comandante Data de *Star Trek: La nueva generación* (1987-1994) y los replicantes de *Blade Runner* (1982) y *Blade Runner 2049* (2017). *A. I. Inteligencia Artificial* (2001) y el thriller de androides *EX Machina* (2015), *Yo robot* (2004), *ED-209 Robocop* (1987).

Estas máquinas aparecen con diferentes nombres (robots, seres artificiales, androides, inteligencias artificiales, droides, ciborgs, replicantes), pero independientemente de su nomenclatura y su forma, todos son inteligentes, autónomos y físicamente capacitados, aunque a veces tengan ciertas restricciones estructurales. A menudo son aterradores, pero también pueden tener un lado atractivo y siempre son intrigantes.

¿Por qué sentimos esa fascinación cuando vemos versiones artificiales de nuestras mentes y nuestros cuerpos? Tal vez haya tras ello un nostálgico secreto humano: si somos capaces de hacer eso, a lo mejor algún día sepamos cómo mejorarlos a nosotros mismos. O tal vez simplemente queramos imaginar un mundo en el que unos sirvientes mecánicos realicen las tareas que preferiríamos no hacer nosotros mismos, o que nos atiendan con una perfección inhumana. Ya sea que los seres artificiales se muestren dignos de los humanos o como enemigos de la humanidad, todas esas historias consiguen algo típico de la buena ciencia ficción (además de entretener): elaborar posibles futuros, permitiéndonos imaginar hacia donde pueden llevarnos las nuevas tecnologías antes de que lleguen de verdad.

Pero la IA, la robótica y la ingeniería genética, las ciencias que previsiblemente permitan hacer un ser artificial, están todavía en la cuna. A día de hoy no existen seres artificiales tan capacitados como los que aparecen en las películas. De hecho, la IA actual no ha demostrado tener la inteligencia que vemos en la pantalla. Y la realidad es que nos queda un largo camino hasta llegar a crear una inteligencia como cualquier de las citadas.

De todas formas, la IA y los robots están entrando en nuestro mundo cada vez más rápido, y tenemos que aprender a vivir con ellos. Algunos países no tienen humanoides como Terminator para enviar en misiones para matar, pero se están desarrollando armas que podrían tomar decisiones mortales en batalla. La creación de armas completamente autónomas que decidan ellas mismas cuál es el blanco y cuándo disparar es el siguiente paso que estamos capacitados para dar. El rápido

desarrollo de la IA fomentará una creciente carrera armamentística basada en IA que se caracterice por no necesitar el control humano.

Por ello, necesitamos responder previamente en el desarrollo de una posición moral hacia las máquinas inteligentes. La creación de máquinas inteligentes sin la previsión suficiente hay que pensárselo.

El problema de la inteligencia artificial no es fabricar agentes inteligentes (lo cual ya lo hace con mucho éxito en muchos casos), sino agentes conscientes o sintientes. La IA ya ha fabricado inteligencia, lo que hace falta es que fabrique mentes.

Ante estos retos siempre encontraremos tres tipos de personas:

- El optimista, que siempre ve que la botella está medio llena.
- El pesimista, para el que la botella está medio vacía.
- El ingeniero, la botella es el doble de grande de lo que debería de ser.

Pero, desgraciadamente, en lo que respecta a fabricar una mente artificial similar o superior a la humana, las cosas siguen bastante glaciares. Aunque se han dado ciertos avances, no se ha solucionado el problema del sentido común y los computadores siguen sin poder enfrentarse a entornos poco formalizados. Hemos descubierto que la inteligencia no se reduce a mero cálculo (en lo cual las máquinas son las campeonas), sino que hay muchos tipos de inteligencias muy interconectadas: hay inteligencia social, emocional, creativa... ¿Cómo implementar en una máquina algo parecido a una emoción? Podemos simular conducta emocional, pero hacer que un ordenador sienta realmente dolor o pena, cero patatero. Y ya ni hablar de consciencia, de que una máquina se dé cuenta realmente de lo que hace.

Lo siento amigos, tendremos que esperar muchos, muchos años, para poder jugar al ajedrez con una máquina, mientras hablamos con ella del sentido de la vida.

Un mundo con problemas

Esta parte está formada por los siguientes subapartados:

- Problemas.
- Decisiones.
- Contextos de decisión.
- Tipos de problemas.
- Ejemplos de problemas que hay que resolver.
- Las tres componentes de las posibles soluciones.

Problemas

Un problema es una situación cuya respuesta desconocida debe obtenerse a través de algún método.

En ciencias, es común asumir la existencia de un continuo progresivamente complejo, integrado por los datos, la información, el conocimiento y la sabiduría, cuyas definiciones podrían ser las siguientes:

- Un dato es una representación simbólica de un atributo o variable cuantitativa o cualitativa. Los datos describen hechos empíricos, estado de las cosas, sucesos y entidades. En esta línea, un metadato es un dato que hace referencia a otro dato. Por ejemplo, un dato podría ser el título de un libro. Un metadato podría ser la ficha de una biblioteca sobre un libro.
- La información es lo que surge de los Datos + la Semántica asociada a los Datos. Por ejemplo, fulanita pesa 30 kilos.
- El conocimiento se define como el conjunto organizado de datos e información que permiten resolver un determinado problema o tomar una decisión (conocimiento «accionable»).

C sabe K «si y solo si» [(K es cierto) y (C acepta K) y (K es evidente para C)].

- La sabiduría es la forma correcta de aplicar el conocimiento. No debe confundirse con acumulación de información. La sabiduría es una habilidad que se desarrolla con la aplicación de la inteligencia en la experiencia, obteniendo como resultado un mayor entendimiento.

Decisiones

La toma de decisiones es el proceso mediante el cual se realiza una elección entre las opciones o formas que se tienen para resolver diferentes situaciones de la vida en diferentes contextos. La elección la realiza una persona o personas haciendo uso

de su razonamiento y pensamiento. En el proceso de toma de decisiones se utiliza como materia prima la información, sin ella no resultaría posible evaluar las opciones existentes o desarrollar opciones nuevas.

A veces se busca la elección de la mejor opción, entonces los siguientes términos pueden ayudar a tomar la decisión según el resultado que se busque:

- Maximizar: Tomar la mejor decisión posible.
- Satisfacer: Elegir la primera opción que sea mínimamente aceptable satisfaciendo de esta forma una meta u objetivo buscado.
- Optimizar: La que genere el mejor equilibrio posible entre distintas metas.

Contextos de decisión

A la hora de tomar decisiones nos influyen la razón y la emoción, sí, pero hay otro elemento menos obvio e igual de importante: el contexto. Conocer sus secretos nos puede ayudar en la vida.

Conocer cómo tomamos decisiones nos ayuda a tener más éxito en nuestros objetivos. Solemos pensar que decidimos por motivos lógicos o que, al menos, somos capaces de explicar las razones. Pero somos muy complejos a la hora de tomar decisiones. No solo nos mueve la razón o la emoción, sino que también somos vulnerables a otros factores que actúan en nosotros y de los que no somos ni tan siquiera conscientes. Y uno de los más importantes es el contexto. Puede ser el contexto físico, lo que nos ha sucedido antes de tomar la decisión. Somos seres sociales y no es de extrañar que nos veamos influidos por quienes nos rodean, aunque sea de manera consciente.

Los contextos también son amplificadores de las emociones, piensen en la música de las películas de terror, y en la misma película sin la música.

En definitiva, el contexto influye en nuestras decisiones, en nuestras emociones y en cómo percibimos la realidad. Es una buena información para tener en cuenta cuando queremos conseguir un objetivo. Muchas veces nos centramos en qué hemos de hacer, cuáles son los pasos para conseguirlo, y no siempre reparamos si el contexto es el más adecuado o cómo podemos influir en él.

Identificar si tenemos el contexto adecuado antes de una decisión o una conversación importante nos ayuda a reducir impactos no deseados y a veces inconscientes.

El contexto es el marco sobre el que nos movemos, sentimos, pensamos y decidimos, e influye directamente en nosotros, aunque no siempre nos demos cuenta.

O como escribió José Ortega y Gasset en *Meditaciones del Quijote*: «Yo soy yo y mi circunstancia y si no la salvo a ella no me salvo yo».

Las situaciones, los ambientes o los contextos en los cuales se toman las decisiones, se pueden clasificar según el conocimiento y el control que se tenga sobre las variables que intervienen o influyen en el problema, ya que la decisión final o la solución que se tome va a estar condicionada por dichas variables.

Tipos de contextos:

- En función de la observabilidad:

- Totalmente observable o ambiente de certeza:

Se tiene conocimiento total sobre el problema, las opciones de solución que se planteen van a causar siempre resultados conocidos e invariables. Al tomar la decisión solo se debe pensar en la opción que genere mayor beneficio.

- Parcialmente observable:

La información con la que se cuenta para solucionar el problema es completa, es decir, se conoce el problema, se conocen las posibles soluciones, pero no se conocen con certeza los resultados que pueden arrojar.

En este tipo de decisiones, las posibles opciones de solución tienen cierta probabilidad conocida de generar un resultado. En estos casos se pueden usar modelos matemáticos o también el decisor puede hacer uso de la probabilidad objetiva o subjetiva para estimar el posible resultado. Entendiendo por:

- La probabilidad objetiva es la posibilidad de que ocurra un resultado basándose en hechos concretos.
 - La probabilidad subjetiva determina el resultado basándose en opiniones y juicios personales.

- Con ambiente de incertidumbre:

En este contexto, se posee información deficiente para tomar la decisión, no se tiene ningún control sobre la situación, no se conoce cómo puede variar o la interacción de las variables del problema, se pueden plantear diferentes opciones de solución, pero no se le puede asignar probabilidad a los resultados que arrojen.

Con base en lo anterior, hay dos clases de incertidumbre:

- Estructurada: No se sabe qué puede pasar entre diferentes opciones, pero sí se conoce que lo que puede ocurrir está entre varias posibilidades.
- No estructurada: No se sabe qué puede ocurrir ni las probabilidades para las posibles soluciones, es decir, no se tiene ni idea de qué pueda pasar.

- Determinista o estocástico:

Cuando se actúa en un entorno determinista, se sabe que siempre que se realiza la misma acción, se obtiene el mismo resultado. En un entorno estocástico, el resultado de una acción es incierto debido a la incertidumbre o al azar.

- Episódico o secuencial:

Se considera que el entorno es episódico cuando una acción no influye en las acciones posteriores. La experiencia se puede explicar mediante un conjunto de experiencias independientes. En un entorno secuencial, una acción realizada afectará a las decisiones futuras.

- Estático o dinámico:

Estático cuando el entorno no cambia con el tiempo y dinámico cuando se producen cambios posibles en el proceso evolutivo.

- Discreto o continuo:

Se denomina *entorno discreto* cuando las variables de las que dependen el resto de las variables, que son normalmente el espacio o el tiempo, solo pueden tomar ciertos valores. Si los datos del espacio y del tiempo pueden tomar cualquier valor, entonces el entorno se denomina *continuo*.

Ejemplos de contextos:

- Juego del ajedrez: Totalmente observable. Determinista. Secuencial. Estático. Discreto.
- Juego del póker: Parcialmente observable. Determinista. Secuencial. Estático. Discreto.

- Diagnóstico médico: Parcialmente observable. Estocástico. Secuencial. Dinámico. Continuo.
- Análisis de imagen: Totalmente observable. Determinista. Episódico. Dinámico. Continuo.

Tipos de problemas

Según E. F. Schummacher

- Problemas convergentes:

También llamados problemas lógicos o estructurados, ya que tienen respuestas únicas y definidas. Para resolverlos se necesita rigor de pensamiento y gran capacidad para extraer deducciones válidas. A un problema específico se le plantean varias soluciones que convergen poco a poco de manera creciente hasta que surge la respuesta. Esta solución resulta ser estable a lo largo del tiempo porque cumple todos los requisitos, cuanto más inteligencia se aplique a estudiarlo más se acercan las respuestas a una solución ideal, es decir, más convergen. Las respuestas cada vez se hacen más precisas para considerarse como definitivas y los podemos encontrar en los campos de la física, la química, la astronomía, la geometría, las matemáticas, el juego del ajedrez, la ingeniería.

- Problemas divergentes:

Se presentan cuando varias personas competentes se ponen a estudiar un mismo problema y encuentran soluciones que pueden incluso llegar a contradecirse entre sí, es decir, no convergen. La lógica ordinaria y lineal no sirve.

Es imposible resolver un problema divergente mediante lógica o estadística. No es útil establecer una fórmula perfecta que permita operar mecánicamente. Se puede decir que los problemas no se resuelven ni establecen con una fórmula correcta, solo pueden superarse tomando como elemento decisivo algo muy fuera de él, es decir, trascendiéndolo. Para esto se deben desarrollar las facultades supraglóticas del ser humano como son la creatividad y el control, la innovación y el orden..., la integración de ese tipo de parejas se experimenta de manera existencial gracias al aprendizaje de la vida.

Por ejemplo: El anuncio publicitario para un producto concreto realizado por diferentes agencias de publicidad.

Según los dominios de D. Snowden

- Dominio simple:

En este tipo de casos, es muy fácil identificar las causas y sus efectos. Por lo general, la respuesta correcta es clara, indiscutible y está al alcance de todos los que tengan los conocimientos adecuados. Ya que la obtención de la respuesta es en cierto modo axiomática, los procesos que se deben seguir son eficientes y quedan especificados por una serie lógica de pasos. Es el dominio en el que se pueden aplicar **las mejores** prácticas. Normalmente el número de variables que intervienen en el problema suelen ser pocas. En resumidas cuentas, para resolver estos problemas hay que seguir las reglas. La razón para ello es que la relación causa y efecto es repetible y predecible.

Ejemplos de este dominio son la construcción en serie de un mismo producto, la producción en cadena o la identificación de causas elementales de enfermedades que dan lugar a través de la acumulación de evidencia a tratamientos eficaces.

- Dominio complicado:

Estos problemas pueden tener múltiples soluciones correctas para una misma problemática. Es el dominio en el que se pueden aplicar **buenas prácticas** seleccionadas por diferentes perfiles de expertos. Normalmente el número de variables que intervienen en el problema suelen ser muchas. Resumiendo, para resolver estos problemas hay que preguntar al experto. En este caso, la relación causa y efecto no es tan evidente, pero pueden estudiarse.

Un ejemplo típico de este escenario sería enviar un cohete a la Luna, para ello las ecuaciones de la física juegan un papel crítico y necesario. El envío de un cohete incrementa las posibilidades de éxito con el siguiente. Se precisa coordinación y alto nivel de conocimiento experto en muchos campos especializados. Las familias de cohetes guardan similitud, y con la experiencia acumulada hay un elevado grado de certidumbre de alcanzar el éxito.

- Dominio complejo:

Cuando nos enfrentamos a problemas complejos, los resultados se vuelven más impredecibles. No existen ni mejores ni buenas prácticas catalogadas para las situaciones frente a las cuales nos podemos encontrar. Simplemente, no sabemos con anticipación si una determinada solución va a funcionar. Solo podemos examinar los resultados y adaptarnos. Este es el dominio de las **prácticas emergentes**. Las soluciones encontradas rara vez son replicables, con los mismos resultados, a otros problemas similares. Normalmente el

número de variables que intervienen en el problema suelen ser muchas. Se requieren niveles altos de creatividad, innovación, interacción y comunicación. En estos casos lo adecuado es ensayar y probar. La razón es que las relaciones de causa y efecto no se repiten y son impredecibles.

Un ejemplo típico sería la educación de un niño o de una niña. Los conocimientos históricos tienen una aplicación limitada. La experiencia acumulada previamente no garantiza el éxito en la siguiente vez. El conocimiento especializado puede ayudar, pero no es suficiente para asegurar el éxito. Cada niño o niña son únicos y su entorno diferente. La incertidumbre se mantiene con los resultados.

- Dominio caótico:

Los problemas caóticos son los más amenazantes y se dan cuando surgen periodos de crisis en los que los hechos son inciertos, los datos se discuten, las opiniones son diversas y las decisiones hay que tomarlas de manera urgente. Se suelen requerir grupos de trabajo multidisciplinar que necesitan tomar el control e intentar mover la situación fuera del caos. Normalmente el número de variables que intervienen en el problema suelen ser pocas. Este es el dominio de las **prácticas novedosas**. Lo que prima es actuar, percibir y luego responder. Las relaciones causa y efecto no pueden percibirse.

Es lo que sucede en los países en guerra, en situaciones revolucionarias o en periodos de pandemias.

- Dominio desordenado:

Nos movemos en el espacio desordenado cuando no sabemos en qué dominio estamos. Se considera que se está en una zona peligrosa, ya que no podemos caracterizar de alguna manera las situaciones ni determinar la forma de actuar. En este caso las personas involucradas tienen que interpretar las situaciones y actuar según preferencias personales, confiando en que la suerte nos conduzca hacia alguno de los otros dominios citados previamente.

Situaciones en las que se hacen presentes el pánico, el estrés y la incertidumbre en grandes muchedumbres, desconciertan a cualquiera.

Según el razonamiento utilizado para resolverlo

Entendemos como razonamiento al producto de un conjunto de habilidades cognitivas complejas y superiores a través de las cuales somos capaces de relacionar y vincular diferentes informaciones de forma estructurada, una vinculación que permite establecer diferentes estrategias, argumentos y conclusiones en función de dicha estructuración de la información. Razonar permite elaborar nuevas informaciones e ideas. Asimismo, nos permite encontrar la resolución de los problemas o situaciones con las que nos encontremos.

Si bien el concepto de *razonamiento* puede parecer simple, lo cierto es que al igual que ocurre con la inteligencia definirlo de forma clara y delimitada (sin mezclarla con otros conceptos) reviste gran complejidad. La verdad es que el razonamiento en sí es difícil de estudiar como un todo, dividiéndose a menudo en diferentes procesos que dan lugar a distintos tipos de razonamiento. Entre ellos destacan los siguientes.

Todos los razonamientos llevan implícita la afirmación de que sus premisas ofrecen algún razonamiento para la verdad de su conclusión.

- Razonamiento deductivo:

Consiste en la aplicación correcta de las relaciones lógicas entre enunciados que llevan a conclusiones válidas. Aunque hay que distinguir que el hecho de que la conclusión sea válida no quiere decir que sea cierta (véase el ejemplo). Este tipo de razonamiento está influido por los conocimientos específicos que uno posee y que considera como ciertos acerca del mundo, así como por los recursos de representación que puede utilizar en un problema de razonamiento específico. Se va de lo general a lo particular. A menudo emplea la **lógica**. Este tipo de pensamiento puede ser categórico (a partir de premisas consideradas válidas se extrae una conclusión) o puede ser disyuntivo (se confrontan dos premisas opuestas con el fin de extraer una conclusión que elimine una de ellas). Es habitual que la mera deducción pueda desencadenar juicios, argumentos y creencias que no se ajustan a la realidad. Solo los razonamientos deductivos pretenden a través de sus premisas ofrecer fundamentos concluyentes.

Ejemplo 1. Razonamiento válido y conclusión verdadera:

- Regla general: Todas las canicas de ese saco son blancas.
- Caso: Esta canica proviene de ese saco.
- Resultado: Esta canica es blanca.

Ejemplo 2. Razonamiento válido y conclusión no verdadera:

- El agua hidrata.
 - El mar es agua.
 - El agua de mar hidrata (no es verdad, de hecho, deshidrata).
- Razonamiento inductivo:

El razonamiento inductivo es aquel proceso de pensamiento en el cual se parte de la información particular para llegar a una conclusión general. En este tipo de razonamiento las premisas no necesariamente deben ser válidas, sino que deben ofrecer un fundamento para ello (véase el ejemplo), se estiman como mejores o peores de acuerdo con el grado de verosimilitud o probabilidad que sus premisas confieran a sus conclusiones. Se trataría del proceso inverso al de la deducción. Se observa un caso particular tras otro para a través de la experiencia poder determinar una conclusión más generalizada. Sus conclusiones se basan en probabilidades más que en certezas lógicas, ya que se van tomando las pruebas disponibles para llegar a conclusiones probables, pero no seguras. Los razonamientos inductivos no son válidos, ni inválidos.

Ejemplo 1

- Caso: Estas canicas proviene de ese saco.
- Resultado: Todas estas canicas son blancas.
- Regla general: Todas las canicas de ese saco son blancas.

Ejemplo 2

El primer cisne negro que se trajo a Europa fue en 1679 cuando los colonos ingleses que volvieron de Australia trajeron consigo en sus barcos los primeros ejemplares. Hasta ese momento, todos los europeos solo veían cisnes blancos, por lo tanto, todos los cisnes eran blancos.

- Razonamiento abductivo:

En muchos casos las abducciones no son sino las conjeturas espontáneas de la razón. Para que esas hipótesis surjan se requiere el concurso de la imaginación y del instinto. La abducción es como un destello de comprensión, un saltar por encima de lo sabido.

Ejemplo 1

- Resultado: Todas estas canicas son blancas.
- Regla general: Todas las canicas de ese saco son blancas.
- Caso: Estas canicas proviene de ese saco.

- Razonamiento hipotético-deductivo:

Este tipo de razonamiento es la base del conocimiento científico. Se parte de la observación de la realidad analizando una serie de casos particulares para generar una hipótesis, de la cual a su vez se deducirán posibles consecuencias o interpretaciones de lo observado. Estas, a su vez, deberán contrastarse empíricamente para comprobar su veracidad.

Ello no quiere decir necesariamente que siempre se alcancen resultados válidos, siendo un tipo de razonamiento que también es sensible a los sesgos.

Un ejemplo es la mecánica de Newton que está comprobada para objetos cuyas velocidades son menores que la luz. Esta mecánica queda corregida y mejorada por la teoría de la relatividad de Einstein.

- Razonamiento por analogía:

Su resolución consiste en traer a la memoria casos del pasado, estableciendo una analogía entre las características de la situación actual y las características de situaciones anteriores. En este caso las experiencias pasadas conllevan a establecer una generalización que permite recordar métodos para resolver problemas actuales.

- Razonamiento...

Ejemplos de problemas que hay que resolver

No hay más que prestar atención a muchos de los dispositivos o de las instalaciones o de los servicios que nos rodean para asombrarse. Vean las imágenes relacionadas con los flujos de información de cualquier red social; los dispositivos médicos relacionados con la imagen médica digital; los teléfonos móviles que utilizamos y el sinfín de servicios de comunicación que nos ofrecen; las cadenas de montaje robotizadas; las transformaciones paulatinas que van surgiendo en los coches y en los vehículos de cualquier tipo que nos llevan de manera inexorable a la conducción automática; los desafíos cada vez más insalvables que los modernos videojuegos nos ofrecen, y las capacidades de aprendizaje que muestran algunos de ellos, las herramientas cada vez más efectivas de ayuda a la toma de decisiones que van apareciendo en el mercado... Pero a pesar del enorme potencial del conocimiento científico y tecnológico actual, aún quedan muchos problemas del mundo con los que hay que lidiar:

Problemas globales:

- El hambre.
- El acceso al agua potable, desalinización energéticamente eficiente.
- La contaminación, aérea, marina, de los ríos...
- La reducción del CO₂ generado y almacenamiento del existente.
- La desigualdad.
- Los conflictos.
- La generación de energía barata y su almacenamiento eficiente.
- El acceso a la energía, con distribución eficiente y segura.
- La mejora de la salud, vacunas, la lucha contra la demencia.
- El conocimiento del funcionamiento del cerebro.
- ...

Más problemas:

- La predicción del tiempo.
- La predicción de los terremotos, el vulcanismo...
- La computación cuántica.
- ...
- La fabricación avanzada.
- Los nuevos materiales.
- Los vuelos rápidos y cómodos para todos.
- Los coches autónomos seguros.
- Los agentes autónomos, los robots o los sistemas informáticos.
- Los entornos inteligentes.
- La traducción simultánea.
- La telepresencia.
- ...

Muchos de estos problemas reales se caracterizan porque no se pueden plantear bien, ya que el dominio en el que hay que resolverlos es parcialmente observable en el mejor de los casos, y en el peor existe incertidumbre. Muchas veces no se conoce la solución o en otras existen múltiples soluciones. Además, dependen de un gran número de parámetros, de interrelaciones y no basta con tomar una sola decisión. Y por si fuera poco son dinámicos. Sin embargo, nuestra calidad de vida y de lo que producimos, hacemos o consumimos dependen, precisamente, de la calidad de nuestro pensamiento.

Y por si esto no fuera poco, la capacidad de la mente humana para formular y resolver problemas categorizables en los distintos dominios citados (complejos, caóticos o desordenados) es muy pequeña en comparación con el tamaño de los problemas cuya solución se requiere. Y esto a pesar de que todo el mundo piensa; es parte de nuestra naturaleza. Pero mucho de nuestro pensar por sí solo es arbitrario, distorsionado, parcializado, desinformado o prejuiciado. La causa de ello es que la estructura de nuestra mente se ha ido formando a lo largo de millones de años durante los cuales lo importante era sobrevivir y se optó fundamentalmente por el uso de las emociones para obtener soluciones simples y rápidas en un mundo peligroso y cambiante.

Todo esto muestra que todavía existe un gran número de problemas que se desea resolver. Las preguntas que pueden surgir en estos momentos serían:

- ¿Cómo podemos enfrentarnos a problemas de los tipos complejos, caóticos o desordenados?
- ¿Se plantea un naufragio social causado por la polución de ese tipo de problemas a la hora de encontrar soluciones a los problemas que tenemos pendientes?
- ...

Personalmente creo que las repuestas a ese tipo de preguntas son que podríamos enfrentarnos a ellas utilizando la capacidad racional del ser humano ayudado por máquinas como los computadores.

Racionalidad

Párrafos entresacados del libro de Steven Pinker titulado *Racionalidad*, Editorial Paidós, 2021.

¿Por qué somos tan inteligentes?, ¿cómo hemos descubierto las leyes de la naturaleza, transformado el planeta, prolongado y enriquecido nuestras vidas y, en particular, articulado las reglas de la racionalidad que incumplimos con tanta frecuencia?

Lo cierto es que hoy en día disponemos de refinados instrumentos de la razón y que, tanto individual como colectivamente, salimos ganando cuando los entendemos y los aplicamos.

Las herramientas intelectuales del razonamiento certero son: la lógica, el pensamiento crítico, la probabilidad, la correlación y la causalidad, las maneras óptimas de ajustar nuestras creencias y comprometernos con decisiones con pruebas inciertas, y los criterios para tomar decisiones racionales tanto en solitario como con otras personas.

Estas herramientas del razonamiento resultan indispensables a la hora de evitar la estupidez tanto en nuestra vida personal como en las políticas públicas. Nos ayudan a calibrar las decisiones arriesgadas, a evaluar las afirmaciones dudosas, a entender las paradojas desconcertantes y a comprender mejor las vicisitudes y las tragedias de la vida.

Los nuevos territorios que están convirtiendo la irracionalidad humana en una cuestión apremiante en nuestros días son: la naturaleza del rumor, la sabiduría popular y el pensamiento conspirativo; el contraste entre la racionalidad del individuo y de la comunidad; y la distinción entre dos modalidades de creencia; la mentalidad realista y la mentalidad mitológica.

¿Cuán racional es el homo sapiens?

Los recursos cognitivos para entender el mundo y someterlo a nuestra voluntad no son un trofeo de la civilización occidental; son el patrimonio de nuestra especie.

Los modelos normativos provienen de la lógica, la filosofía, las matemáticas y la inteligencia artificial, y suponen la mejor comprensión de la solución correcta de un problema y del modo de hallarla por nuestra parte. Y sirven de aspiración a aquellos que desean ser racionales, que deberíamos ser todos.

Por muy excelentes que sean nuestros sistemas cognitivos, en el mundo moderno hemos de saber cuándo dejarlos de lado y dirigir nuestro razonamiento hacia los instrumentos: las herramientas de la lógica, la probabilidad y el pensamiento crítico,

que amplían nuestras capacidades de razonar más allá de las que la naturaleza nos ha otorgado. Porque en el siglo XXI, cuando pensamos guiándonos por nuestra intuición, cualquier corrección puede empeorar las cosas y puede enviar nuestra democracia a una espiral de cementerio.

Aunque las explicaciones de la irracionalidad pueden absolver a las personas del cargo de absoluta estupidez, comprender no equivale a perdonar.

Racionalidad e irracionalidad

Nerd, wonk, geek o brainiac.

¿Qué es la racionalidad? Como sucede con la mayoría de las palabras de uso común, ninguna definición puede estipular su significado con exactitud:

- *El diccionario solo nos conduce a un círculo: la mayoría define racional como dotado de razón, pero la razón misma proviene del latín ratio-, definida con frecuencia como razón.*
- *La capacidad de utilizar el conocimiento para alcanzar objetivos:*
 - *Conocimiento: creencia verdadera y justificada.*
 - *Las creencias se profesan al servicio de un objetivo.*

Un agente racional debe tener un objetivo, ya sea este determinar la verdad de una idea digna de ser tenida en cuenta, lo cual se conoce como razón teórica, o ya sea lograr un resultado notable en el mundo, lo cual se conoce como razón práctica («qué es verdad» y «qué hacer»). Y para alcanzar ese objetivo debe emplear algún conocimiento relevante para las circunstancias.

Teniendo en cuenta que la racionalidad perfecta y la verdad objetiva son aspiraciones que ningún mortal puede afirmar jamás haber alcanzado. Pero la convicción de que estas existen ahí afuera nos autoriza a desarrollar reglas que todos podemos cumplir y que nos permiten aproximarnos colectivamente a la verdad de maneras que resultan imposibles para cualquiera de nosotros individualmente.

Las reglas se diseñan para dejar de lado sesgos que se interponen en el camino de la racionalidad: las ilusiones cognitivas incorporadas en la naturaleza humana, así como el fanatismo, los prejuicios, las fobias y los ismos que infectan a los miembros de una raza, una clase, un género, una sexualidad o una civilización.

Por otro lado, la belleza, el amor y la bondad no son literalmente racionales, tampoco son exactamente irracionales. Podemos aplicar la razón a nuestras emociones y a nuestra moralidad, y existe incluso una racionalidad de orden superior que nos dice cuándo puede ser racional ser irracional.

La razón es el medio para un fin y no puede decirnos cuál debería ser el fin, ni siquiera que hemos de perseguirlo. Las pasiones son la fuente de esos fines: los gustos, los deseos, los impulsos, las emociones y los sentimientos programados en nosotros, sin los cuales la diferencia, entre pensar y querer, entre creer algo que consideras verdadero y desear algo que quieres conseguir. No es ni racional, ni irracional preferir la tarta de chocolate a la de queso. Y no es irracional en modo alguno cuidar un jardín, enamorarse, ir de fiesta o bailar.

Es un error lógico pensar que la razón puede oponerse a las emociones.

¿Qué le pasa a la gente?

*Dile a la gente que hay un hombre invisible en el cielo
que creó el universo y la inmensa mayoría te creerá.
Dile que la pintura está húmeda y tendrá que tocar para asegurarse.*

George Carlin

Seguro que habrá oído decir que la humanidad parece estar perdiendo la cabeza!

Analicen la situación proporcionada por las COVID-19, el magnífico avance científico de la obtención de las vacunas frente a la absurda reacción de ciertas personas antivacunas. Un ejemplo de rechazo de gran envergadura de las reglas de la razón y de la ciencia. Otros ejemplos, la negación del cambio climático, las teorías de la conspiración, las fake-news, las creencias en los espíritus benignos o malignos, la magia negra y otras supersticiones, la astrología, la negación del holocausto... Lo que se denomina crisis epistemológica y era de la posverdad. ¿Cómo se puede explicar esta pandemia de necedades?

Explicaciones simples:

- *Las falacias lógicas y estadísticas. Muchas supersticiones se originan en la sobre interpretación de las coincidencias, en la incapacidad de calibrar las evidencias en función de las probabilidades previas, en la generalización excesiva a partir de anécdotas y en el salto de la correlación a la causalidad.*
- *Los medios sociales y la popularidad de las teorías conspiranoicas.*
- *Jamás es una explicación satisfactoria decir que las personas abrazan una creencia falsa porque esta las consuela o las ayuda a dar sentido al mundo, porque eso no hace más que plantear la pregunta de por qué los individuos habrían de hallar consuelo y sentido de finalidad en creencias que no podrían hacerles ningún bien.*

- *Tampoco basta con tachar a los humanos de irremediabilmente irracionales.*

Al igual que nuestros antepasados cazadores recolectores vivían de su ingenio en ecosistemas implacables, hoy día los teóricos de la conspiración y los creyentes en los milagros superan las exigentes pruebas de sus propios mundos: mantienen trabajos, educan a sus hijos, y tienen un techo sobre su cabeza y comida en el frigorífico.

Hay que reconocer que la mayoría de los miembros de nuestra especie poseen la capacidad para descubrir y aceptar los cánones de la racionalidad. A fin de comprender los delirios populares y la locura de las multitudes, tenemos que examinar las facultades cognitivas que funcionan bien en algunos entornos y para algunos propósitos, pero fracasan cuando se aplican a gran escala, en circunstancias novedosas o al servicio de otras metas.

Razonamiento motivado

La racionalidad es la misma para todo el mundo en todas partes.

El acopio de recursos retóricos para conducir un argumento hacia una conclusión predilecta se denomina razonamiento motivado.

Muchos de los sesgos que pueblan las listas de enfermedades cognitivas son tácticas de razonamiento motivado.

- *La asimilación sesgada (o exposición selectiva), los individuos buscan argumentos que ratifiquen sus creencias y se protegen de aquellos que podrían refutarlas.*
- *En la evaluación sesgada, desplegamos nuestro ingenio para apoyar los argumentos que respaldan nuestra posición y somos quisquillosos con aquellos que la refutan.*
- ...

Al parecer no hemos evolucionado como científicos intuitivos, sino como intuitivos abogados. Por fortuna, esta hipocresía puede mobilizarse para hacernos más racionales colectivamente de lo que cualquiera de nosotros lo es a nivel individual. Cuando las personas evalúan una idea en pequeños grupos con la química adecuada, que consiste en que no están de acuerdo en todo, pero tienen un interés común en hallar la verdad, detectan sus respectivas falacias y puntos ciegos, y suele vencer la verdad.

Sesgo de mi lado

Los individuos razonan para conseguir llegar a una conclusión o huir de ella, incluso cuando ello no les ofrezca ninguna ventaja personal. Basta con que la conclusión realce la corrección o la nobleza de una tribu política, religiosa, étnica o cultural.

Se sabe desde hace mucho tiempo que los humanos están muy interesados en dividirse en equipos competitivos, pero no está claro por qué es hoy la división entre izquierda y derecha la que está tirando de la racionalidad de cada lado en direcciones diferentes, en lugar de las líneas de falla habituales de la religión, la raza y la clase.

El eje derecha-izquierda se alinea con varias dimensiones morales e ideológicas:

- *Jerárquico frente a igualitario.*
- *Libertario frente a comunitarista.*
- *Trono y altar frente a visiones utópicas.*
- *Cultura del honor frente a cultura de la dignidad.*
- *Morales vinculantes frente a morales individualizadoras.*
- *...*

Pero recientemente debido a la inmigración, el comercio... sugieren que los bandos políticos se han convertido en tribus socioculturales más que en ideologías coherentes.

Los bandos se asemejan menos a tribus en sentido literal, que mantienen su unidad mediante el parentesco, que, a las sectas religiosas, que se mantienen unidas mediante la fe en su superioridad moral y su desprecio hacia las sectas opuestas.

Las razones:

- *Fraccionamiento y polarización de los medios audiovisuales.*
- *Los fraudes electorales y otras distorsiones geográficas de la representación política, que incentivan a los políticos a atender a las camarillas más que a las coaliciones.*
- *La dependencia de los políticos y de los laboratorios de ideas de donantes ideológicamente comprometidos.*

- *La autosegregación de los profesionales liberales educados en enclaves urbanos.*
- *El declive de organizaciones interclasistas de la sociedad civil, tales como iglesias, clubes de servicio y grupos de voluntariado.*

Nada garantiza que las posiciones favoritas de la izquierda y de la derecha en cualquier momento histórico se alineen con la verdad. Incluso si ambos bandos interpretan la realidad a través de sus respectivas creencias, el bando cuyas creencias estén justificadas actuará racionalmente. Tal vez, el bien documentado desequilibrio del mundo académico entre izquierda y derecha no sea un sesgo irracional, sino una calibración acertada de sus probabilidades previas bayesianas al hecho de que la izquierda siempre tiene razón.

Aunque puede que sea cierto que las posiciones izquierdistas estén justificadas con más frecuencia que las de derechas (especialmente si, por la razón que sea, la izquierda congenia más con la ciencia que la derecha).

Desgraciadamente, lo que es racional para cada uno de los que buscamos la aceptación en una camarilla no es tan racional para todos nosotros en una democracia que busca la mejor comprensión del mundo.

Dos clases de creencia: realidad y mitología

A pesar de todo, como diría Philip K. Dick, la realidad es aquello que, aun cuando dejes de creer en ello, no desaparece.

Pero ¿por qué la realidad no obliga a retroceder e impide que la gente crea absurdidades o recompense a aquellos que las afirman y las comparten?

La respuesta es que depende de lo que queramos decir con creer. Mercier observa que quienes profesan creencias extrañas a menudo carecen del valor de sus convicciones. Este propone que las teorías de la conspiración y las creencias extrañas son reflexivas, fruto de la meditación y la teorización consciente, más que intuitivas, que son las convicciones que sentimos en nuestros huesos.

Mentalidad realista y mentalidad mitológica

Los individuos dividen su mundo en dos zonas. Una consta de los objetos físicos que los rodean, las otras personas con las que tratan cara a cara, el recuerdo de sus interacciones, y las reglas y normas que regulan sus vidas. Las personas poseen creencias básicamente precisas acerca de esta zona y razonan racionalmente dentro de ella. En esta zona, creen que existe un mundo real y que las creencias acerca de

este son verdaderas o falsas. No tienen elección: esa es la única manera de tener gasolina en el coche, dinero en el banco y a los hijos vestidos y alimentados. Podemos llamarla mentalidad realista.

La otra zona es el mundo más allá de la experiencia inmediata: el pasado distante, el futuro incognoscible, los pueblos y lugares lejanos, los corredores remotos del poder, lo microscópico, lo contrafactual, lo metafísico. Las personas pueden abrigar ideas acerca de lo que sucede en estas zonas, pero no tienen manera de comprobarlas y, de todos modos, ello no marca ninguna diferencia discernible en sus vidas. Las creencias en estas zonas son relatos, que pueden ser entretenidos, inspiradores o moralmente edificantes. Si son verdaderas o falsas no es la pregunta adecuada. La función de estas creencias es construir una realidad social que cohesione a la tribu o secta y le confiere un propósito moral. Podemos llamarla mentalidad mitológica.

Es célebre la sentencia de Bertrand Russell: Es indeseable creer una proposición cuando no hay fundamento alguno para suponer que sea cierta.

La máxima de Russell es el lujo de una sociedad tecnológicamente avanzada con ciencia, historia, periodismo y su infraestructura de búsqueda de la verdad, incluidos los registros de archivos, los conjuntos de datos digitales, los instrumentos de alta tecnología y las comunidades de edición, verificación de datos y revisión por pares. Como hijos de la Ilustración, abrazamos el credo radical del realismo universal: sostenemos que todas nuestras creencias deberían caer dentro de la mentalidad realista. Nos preocupa que nuestra historia de la creación, nuestras leyendas fundacionales, nuestras teorías de los nutrientes, los gérmenes y las fuerzas invisibles, nuestras concepciones de los poderosos, nuestras sospechas acerca de nuestros amigos, sean verdaderas o falsas. Ello se debe al hecho de que disponemos de las herramientas para obtener respuestas a estas preguntas, o al menos para asignarles grados justificados de creencia. Y contamos con un estado tecnocrático que, en teoría, debería poner en práctica estas creencias.

Ahora bien, por muy deseable que sea ese credo, no es la manera natural humana de creer. Al otorgar un mandato imperialista a la mentalidad realista para conquistar el universo de las creencias y empujar la mitología hacia los márgenes, somos nosotros los extraños [los WEIRD (raros, extraños): occidentales, educados, industrializados, ricos y democráticos]. Al menos lo somos los muy instruidos de entre nosotros, en nuestros mejores momentos.

La mente humana está adaptada para comprender las esferas remotas de la existencia mediante una mentalidad mitológica. Lo que se debe a que descendemos de personas que no suscribían el ideal ilustrado del realismo universal. El sometimiento de todas nuestras creencias a los juicios de la razón y las evidencias es una destreza antinatural, como la alfabetización y el cálculo, y ha de ser inculcada y cultivada. Y a pesar de todas las conquistas de la mentalidad realista, la mentalidad

mitológica continúa ocupando franjas de territorio en el paisaje de las creencias establecidas.

- *El ejemplo evidente es la religión. Sus seguidores han olvidado lo inapropiado o burdo de considerar la existencia de Dios como una cuestión de verdad o falsedad, ya que la creencia en Dios es una idea que cae fuera de la esfera de la realidad comprobable.*
- *Otra zona de irrealidad convencional son los mitos nacionales. La mayoría de los países consagran un relato fundacional como parte de su conciencia colectiva. Entre los temas comunes de nuestra época figuran la antigua esencia de la nación definida mediante una lengua, una cultura y una patria; un sueño prolongado y un glorioso despertar; una larga historia de victimización y opresión; y una generación de libertadores y fundadores sobrehumanos. Los guardianes del patrimonio mítico no sienten la necesidad de llegar hasta el fondo de lo realmente acaecido, y pueden estar molestos con los historiadores que lo ubican en la zona realista y sacan a la luz su historia trivial, su identidad construida, sus provocaciones recíprocas con los vecinos y los pies de barro de los padres fundadores. Recordando que hay una zona de creencias ni ciertas del todo ni completamente falsas.*

La frontera entre la zona realista y la mitológica puede variar con las épocas y la cultura. Desde la Ilustración, las mareas en el Occidente moderno han erosionado la zona mitológica, pero en las fronteras continúa habiendo escaramuzas, tiras y aflojas.

La psicología de los apócrifos

Una vez que apreciamos que los humanos pueden tener creencias que no consideren fácticamente verdaderas, podemos comenzar a comprender la paradoja de la racionalidad: cómo es posible que un animal racional abrace tantos disparates. Comentemos algunos:

- *La pseudociencia, los fenómenos paranormales y el curanderismo movilizan nuestras intuiciones cognitivas más profundas. Somos intuitivamente dualistas y sentimos que las mentes pueden existir aparte de los cuerpos. Eso es algo natural en nosotros, y no solo porque no veamos las redes neuronales que subyacen a las creencias y los deseos de nosotros mismos y de los demás. Muchas de nuestras experiencias sugieren realmente que la mente no está amarrada al cuerpo, incluidos los sueños, los trances, las experiencias extracorpóreas y la muerte. Por lo tanto, no supone un gran salto concluir que las mentes pueden estar en comunión con la realidad y las unas con las otras sin necesidad de un soporte físico. Y así surgen la telepatía, la clarividencia, las almas, los fantasmas, la reencarnación, y los mensajes desde el más allá.*

- *Somos, asimismo, intuitivamente esencialistas, y sentimos que los seres vivos contienen sustancias invisibles que les confieren su forma y sus poderes. Estas intuiciones inspiran a la gente a investigar en busca de semillas, drogas y venenos. Pero esta mentalidad también hace creer a la gente en la homeopatía, los remedios herbales, las purgas y sangrías, y el rechazo de adulterantes extraños como las vacunas y los alimentos genéticamente modificados.*
- *Y somos intuitivamente teleólogos. Al igual que nuestros planes y artefactos están diseñados con un propósito, propendemos a pensar que también lo está la complejidad del mundo viviente y no viviente. Así pues, somos receptivos al creacionismo, la astrología, la sincronicidad y la creencia mística en que todo sucede por una razón.*

Supuestamente, una educación científica ha de sofocar estas intuiciones primitivas, pero su alcance es limitado por varios motivos. Uno de ellos es que no resulta fácil renunciar a las creencias que son sagradas para una facción religiosa o cultural, como el creacionismo, el alma y un propósito divino, y estas pueden guardarse dentro de la zona mitológica de los individuos. Otro es que, incluso entre la gente muy culta, la comprensión científica es superficial. Los individuos instruidos confían en la comunidad científica y la pseudociencia borrosa. Para la mayoría de las personas, lo más cercano a la ciencia en su vida es el médico, y muchos médicos son más prácticos que expertos en ensayos clínicos aleatorizados. Los documentales y los informativos de la televisión convencional también pueden difuminar las líneas y dramatizar crédulamente teorías especulativas como los antiguos astronautas y los videntes que luchan contra la delincuencia.

Los principios fundacionales, como que el universo no tiene objetivos relacionados con las preocupaciones humanas, que todas las interacciones físicas están gobernadas por unas cuantas fuerzas fundamentales, que los cuerpos vivientes son intrincadas máquinas moleculares y que la mente es la actividad de procesamiento de la información del cerebro, nunca se expresan con claridad, quizá porque se ofenderían las sensibilidades religiosas y morales. No debería sorprendernos que lo que la gente saque en limpio de la educación científica sea un batiburrillo sincrético, donde la gravedad y el electromagnetismo coexisten con la psi, el qi, el karma y la sanción con cristales.

Para entender patrañas virales como las leyendas urbanas, los titulares de los tabloides, y las noticias falsas, hemos de recordar que son extraordinariamente entretenidas. Tratan temas de sexo, violencia, venganza, peligro, fama, magia y tabú, que siempre han despertado el interés de los mecenas de las artes, tanto altas como bajas.

No es de extrañar que la gente busque toda suerte de entretenimientos. Lo que nos sorprende es que esas obras de arte se conviertan en una afirmación fáctica. Del mismo modo que los mitos religiosos y nacionales se arraigan en la corriente

dominante cuando se siente que proporcionan elevación moral, las noticias falsas pueden volverse virales cuando sus propagadores piensan que está en juego un valor superior, como reforzar la solidaridad en el seno de su bando y recordar a los camaradas la perfidia del bando contrario. A veces la moraleja no es siquiera una estrategia política coherente, sino un sentido de superioridad moral: la impresión de que las clases sociales rivales y las poderosas instituciones respecto de las cuales se sienten alienados los propagadores son decadentes y corruptas.

Las teorías de la conspiración

Florecen porque los humanos siempre han sido vulnerables a las conspiraciones reales. Los complotos de las coaliciones enemigas son diferentes de otros peligros, como los depredadores y los rayos, porque estas despliegan su ingenio para atravesar las defensas de sus objetivos y cubrir sus propios rastros. La única salvaguarda contra este subterfugio de intriga y misterio consiste en superar al otro en astucia con carácter preventivo, lo cual puede conducir a intrincadas series de conjeturas y a la negativa a creer a pies juntillas los hechos evidentes. En términos de detección de señales, el coste de pasar por alto una conspiración real es más elevado que el de una falsa alarma ante una sospecha. Lo que nos hace intentar enterarnos de las posibles conspiraciones incluso con evidencias endebles.

Las teorías de la conspiración, al igual que las leyendas urbanas y las noticias falsas, logran convertirse en rumores, y los rumores son la materia prima de la conversación. Los estudios sobre los rumores muestran que estos tienden a transmitir amenazas y peligros, y que confieren un aura de sabiduría a quien los difunde. Y tal vez sorprendentemente, cuando circulan entre personas con un interés creado en su contenido, como en el caso de los lugares de trabajo, suelen ser correctos. Aunque los rumores en los medios sociales, a diferencia de los rumores en los lugares de trabajo, son con frecuencia incorrectos.

La clave restante para comprender el atractivo de las creencias extrañas es examinar las creencias mismas con microscopio. La evolución no solo funciona en los cuerpos y en los cerebros, sino también en las ideas. Un meme, tal como lo definió Richard Dawkins cuando acuñó el término, no es una fotografía con un pie que circula por internet, sino una idea que ha sido moldeada por generaciones que la han ido compartiendo hasta tornarse muy compartible.

Al igual que los organismos desarrollan adaptaciones que los protegen de ser devorados, las ideas pueden desarrollar adaptaciones que las protegen de ser refutadas. El ecosistema intelectual está repleto de estas ideas invasivas. «Los designios de Dios son inescrutables», «La negación es un mecanismo de defensa del ego», «Los poderes psíquicos son inhibidos por la exploración escéptica», «Si no denuncias a esta persona por racista, eso demuestra que tú eres racista», «Todo el mundo es egotista, porque ayudar a los demás hace que te sientas bien», y, por supuesto, «La falta de pruebas de esta conspiración demuestra lo diabólica que es».

Las teorías conspiratorias, por su propia naturaleza, están adaptadas para ser propagadas.

Reafirmar la racionalidad

Comprender no implica perdonar. Podemos ver por qué los humanos dirigen sus razonamientos hacia conclusiones que redundan en beneficio de sí mismos o de sus sectas, y porque distinguen una realidad en la que las ideas son verdaderas o falsas de una mitología en la que las ideas son entretenidas o inspiradoras, sin conceder que estas cosas sean buenas. No son buenas. La realidad es aquello que, cuando le aplicas razonamientos motivados, o de mi lado, o mitológicos, no desaparece. Las creencias falsas sobre las vacunas, las medidas de salud pública y el cambio climático amenazan el bienestar de miles de millones de personas. Las teorías conspirativas incitan al terrorismo, los pogromos [Un pogromo consiste en el linchamiento multitudinario, espontáneo o premeditado, hacia un grupo particular, étnico, religioso u otro, acompañado de la destrucción o el expolio de sus bienes], las guerras y el genocidio. La corrosión de los estándares de la verdad socava la democracia y despeja el terreno para la tiranía.

No obstante, pese a todas las vulnerabilidades de la razón humana, nuestra imagen del futuro avanza con el arco del conocimiento doblándose hacia la racionalidad. No deberíamos perder de vista cuánto prolifera la racionalidad ahí afuera.

Hoy en día son pocas las personas de los países desarrollados que creen en los hombres lobo, los sacrificios animales, las sangrías, los miasmas, el derecho divino de los líderes o los augurios en los eclipses y en los cometas, aunque todas estas creencias estaban a la orden del día en siglos pasados. Aunque unos cuantos asuntos científicos caldean los ánimos religiosos o políticos, la mayoría no: hay facciones que desconfían de las vacunas, pero no de los antibióticos; del cambio climático, pero no de la erosión costera. A pesar de sus sesgos partidistas, la mayoría de las personas juzgan con muy buen criterio la veracidad de los titulares y, cuando se les presentan correcciones claras y fiables de una afirmación falsa, cambian de opinión, independientemente de sus simpatías o antipatías políticas.

Tenemos asimismo una cabeza de puente de la racionalidad en el estilo cognitivo llamado apertura mental activa, especialmente en el subtipo denominado apertura a las evidencias. Este es el credo en virtud del cual las creencias deberían estar basadas en buenas razones. Es un rechazo del razonamiento motivado; un compromiso para ubicar todas las creencias dentro de la zona realista; un respaldo a la declaración atribuida a John Maynard Keynes: «Cuando los hechos cambian, cambio de opinión. ¿Y qué hace usted, señor?».

Dado que nadie puede saberlo todo, y la mayoría de las personas no saben casi nada, la racionalidad consiste en externalizar el conocimiento a instituciones especializadas en crearlo y compartirlo, principalmente el mundo académico, las

unidades de investigación públicas y privadas, y la prensa. Aunque la confianza en la ciencia haya permanecido estable durante décadas, la confianza en las universidades se está hundiendo. Esta es la razón por la que las universidades tienen la responsabilidad de garantizar la credibilidad de la ciencia y la erudición, comprometiéndose con la diversidad de puntos de vista, la libre investigación, el pensamiento crítico y la apertura mental activa.

La alfabetización y el cálculo ocupan un lugar preferente en la escolarización porque son un prerrequisito para todo lo demás, entonces las herramientas de la lógica, la probabilidad y la inferencia causal recorren toda clase de conocimiento humano. La racionalidad debería convertir en cuarteto el trío clásico de la lectura, la escritura y la aritmética. De tal manera que ante la vida los estudiantes se animen a detectar lo no racional en escenarios realistas, sean capaces de reformular los problemas en formatos amigables para la mente y tengan la capacidad de retroalimentación inmediata sobre sus errores.

El razonamiento humano tiene sus falacias, sus sesgos y su complacencia en la mitología. Pero la explicación última de la paradoja de cómo puede nuestra especie ser tan racional y tan irracional no es un error en nuestra programación cognitiva. Reside en la dualidad del yo y el otro; nuestra capacidad de razonamiento está orientada por nuestros motivos y limitada por nuestros puntos de vista. La imparcialidad es el núcleo de la racionalidad: una reconciliación de nuestras ideas sesgadas e incompletas con una comprensión de la realidad que trascienda a cualquiera de nosotros. La racionalidad no es solo una virtud cognitiva, sino también moral.

Por qué es importante la racionalidad

El progreso humano es un hecho empírico, si se observan las líneas de tendencia históricas, constatamos que la humanidad es en conjunto más saludable, más rica, más longeva, está mejor alimentada, más instruida y más a salvo de las guerras, los asesinatos y los accidentes que en las décadas y los siglos pasados.

Por el contrario, la naturaleza no tiene consideración alguna por nuestro bienestar y, con frecuencia, como en el caso de las pandemias y los desastres naturales, se diría que está tratando de destrozarnos.

Progreso es la abreviatura de un conjunto de retrocesos y victorias cosechados en un universo implacable, y es un fenómeno que precisa explicación.

La explicación es la racionalidad. Cuando los humanos se fijan la meta de mejorar el bienestar de sus congéneres (frente a otras aspiraciones dudosas, como la gloria o la redención) y aplican su ingenio a instituciones que lo comparten con el de los demás, de vez en cuando triunfan. Cuando retienen los éxitos y toman nota de los fracasos, los beneficios pueden acumularse y designamos el panorama general como progreso.

El progreso consiste en algo más que en avances en la seguridad y el bienestar material. Consiste asimismo en avances en la forma de tratarnos unos a otros, en igualdad, benevolencia y derechos. Muchas prácticas crueles e injustas se han reducido en el transcurso de la historia.

Los argumentos sólidos, que imponen la coherencia de nuestras prácticas con nuestros principios y con la meta del florecimiento humano, no pueden mejorar el mundo por sí mismos. Pero han guiado y deberían guiar los movimientos transformadores. Marcan la diferencia entre la fuerza moral y la fuerza bruta, entre las marchas por la justicia y las turbas de linchamiento, entre el progreso humano y la destrucción. Y serán argumentos sólidos tanto para revelar plagas morales como para descubrir remedios viables, lo que necesitamos para garantizar que continúe el progreso moral, que las prácticas abominables de hoy lleguen a resultarles tan increíbles a nuestros descendientes como lo son para nosotros la quema de herejes y la subasta de esclavos.

El poder de la racionalidad para guiar el progreso moral va de la mano de su poder para orientar el progreso material y las decisiones sabias en nuestras vidas. Nuestra capacidad para lograr incrementar el bienestar en un cosmos implacable y para ser buenos con los demás a pesar de nuestra naturaleza imperfecta depende de que captemos principios imparciales que trasciendan nuestra experiencia provinciana. Somos una especie que ha sido dotada de una facultad elemental de razonar y que ha descubierto fórmulas e instituciones que amplían

su alcance. Estas nos despiertan a ideas y nos exponen a realidades desconcertantes para nuestras intuiciones, pero que son verdaderas a pesar de todo.

Las tres componentes de las posibles soluciones

Personalmente creo que las repuestas a ese tipo de preguntas que se realizaron en el apartado *ejemplos de problemas que hay que resolver*, son que podríamos enfrentarnos a ellas siempre y cuando consigamos avanzar en tres aspectos básicos:

- La primera componente tiene que ver con la formación futura que debe aumentar la visión relacionada con las técnicas de resolución de problemas, de una forma inter-trans-multi-disciplinar. No se trata de que todo el mundo deba saber todo de todo en profundidad, se trata de conseguir que cada persona tenga una visión global, del tipo mapamundi, de las diferentes aproximaciones que el ser humano utiliza para resolver problemas. Como dijo Blaise pascal (1632-1662): *«Es mucho mejor conocer algo de todo que todo de una sola cosa. Lo universal es siempre mejor»*. Si reflexiona sobre lo que está pasando en el mundo, vamos hacia una formación super especializada. Justo todo lo contrario.
- La segunda componente tiene que ver con el hecho de que en estos momentos los computadores ya ofrecen velocidad, capacidad de almacenamiento, escalabilidad de los datos y posibilidad de tratar con problemas repetitivos muy complejos dependientes de un gran conjunto de parámetros independientes. En esto superan a las posibilidades de resolución de problemas por parte de los seres humanos. Pero debemos enseñar a las máquinas, recordemos que ellas siempre nos han servido para complementar nuestras deficiencias. Por lo tanto, para aumentar nuestra capacidad de resolver los difíciles problemas a los que nos enfrentamos, necesitamos utilizar herramientas informáticas que sean cada vez más autónomas y capaces. Por decirlo de alguna manera, debemos aumentar su inteligencia y capacidad de aprendizaje.
- La tercera componente está relacionada con que hay que usar y abusar de lo que se conoce con el nombre de *fertilización cruzada*. Vamos a introducir este concepto tal y como lo explicó George Bernard Shaw: *«Si tú tienes una manzana y yo tengo otra manzana, y nos las intercambiamos, al final tendremos una manzana cada uno. Pero si tú tienes una idea y yo tengo otra idea, y nos las intercambiamos, al final tendremos dos ideas cada uno»*.

La fertilización cruzada de ideas entre la gente conduce a que emerjan las mejores ideas. Por lo tanto, es necesario crear de manera adecuada entornos humanos de trabajo colaborativo, sin olvidar que esos entornos de trabajo deben tender a ser inter-trans-multi-disciplinarios.

Empecemos analizando mejor la primera componente citada, que de manera resumida podría expresarla como «Hay que enseñar a las mentes». De cara a resolver problemas complejos lo que necesitamos es:

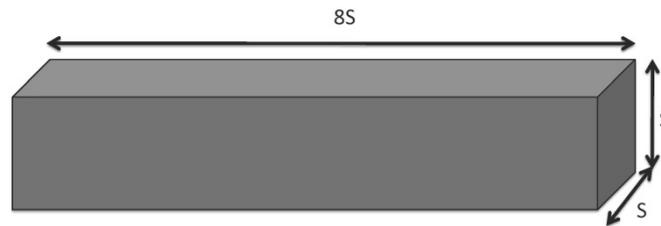
- Datos e información.
- Comprender las relaciones y las conexiones entre los diferentes trozos de información (conocimiento).
- Comprender qué tipo de conocimiento hay que aplicar (sabiduría).
- Saber construir modelos. Los modelos proporcionan marcos aproximados en los cuales se embeben todos los datos e información existente que se considere importante. De manera que dentro de esos marcos se pueden deducir de manera lógica las implicaciones de nuestras suposiciones y comprobar su veracidad. De hecho, cuando se crea un modelo, este:
 - Proporciona un medio para transformar datos de información en conocimiento.
 - Impone lógica a nuestro pensamiento y proporcionan condiciones bajo las cuales la intuición puede navegar.
 - Permite que dicho conocimiento pueda ser interpretado y entendido por seres humanos e incluso por máquinas.
 - Se puede ir mejorando por refinamientos sucesivos.

Con lo cual, el proceso de creación y de mejora de modelos nos hace mejores pensadores.

El proceso de calcular el comportamiento de un modelo, se denomina *simulación*. La gran ventaja que nos proporciona el modelado y su simulación es que es el procedimiento que tenemos para analizar el comportamiento de la solución de un problema real, sin examinar realmente la vida real.

A modo de ejemplo, veamos la solución que encontró el magnate y multimillonario griego Stauros Niarchos (1909-1996) para el problema de obtener más beneficio con el transporte de petróleo decidiendo cómo mejorar su flota. La solución que encontró, y que nadie antes había aplicado, fue que a partir de 1952 hizo construir, para la época, los petroleros más grandes que jamás habían existido en el mundo. De esta manera se benefició tanto de la crisis de Suez como de la creciente demanda de hidrocarburos, y se convirtió en uno de los gigantes del transporte mundial del petróleo, junto con su compatriota y rival Aristóteles Onassis. Al modelo utilizado para tomar esa decisión se le denominó el modelo de los 10 millones de dólares.

- Modeló un barco mediante un paralelepípedo como el de la figura



Fíjese en que el modelo es una notable simplificación de un barco.

- Recordemos que:
 - El área de un rectángulo es el producto de la longitud de la base por la altura.
 - El volumen de un paralelepípedo es igual al producto de la longitud de base por la altura y por la anchura.
- Se sabe que el coste de un navío es proporcional a la cantidad de acero necesario y este es proporcional a la superficie del navío. En el caso del modelo propuesto la superficie es de $34S^2$. (Observe que la superficie va con el cuadrado).
- También se sabe que la ganancia que proporciona un navío de transporte es proporcional al volumen. En el caso del modelo propuesto el volumen es de $8S^3$. (Observe que el volumen va con el cubo).

Ahora supongamos dos navíos con $S_1 = 30$ y $S_2 = 150$ (5 veces más).

- Las áreas, los volúmenes y sus relaciones respectivas serían:
 - $\text{Área}_1 = 030.600 \text{ m}^2$ $\text{Área}_2 = 765.000 \text{ m}^2$
cociente de las áreas = 25
 - $\text{Volumen}_1 = 00.216.000 \text{ m}^3$ $\text{Volumen}_2 = 27.000.000 \text{ m}^3$
cociente de los volúmenes = 125
- Resumiendo. Los superpetroleros son capaces de transportar 125 veces más petróleo, pero solo requieren un gasto de 25 veces más acero.

¡Una ganga!

Por cierto, los primeros superpetroleros se amortizaban por sí mismos después de unas pocas cargas útiles.

En ausencia de más conocimientos sobre barcos y la industria del transporte marítimo, uno podría preguntarse de manera lógica que por qué los actuales superpetroleros no son más grandes todavía. La razón es el tamaño del canal de Suez. Mide tan solo 16 metros de ancho en su parte más estrecha. Cualquier barco más ancho no podría pasar.

Pero si el barco de transporte no tiene por qué pasar por dicho canal, vean como ejemplo el portacontenedores CSLS Globe chino, de 400 m de eslora, 73 m de altura y 59 m de manga. Puede transportar 19 100 contenedores del tamaño aproximado (6,1 m x 2,4 m x 2,6 m) con un peso máximo de 26 000 kg. Véase

<https://www.youtube.com/watch?v=YLuhUI1C8Ds>

Esta historia de los superpetroleros proporciona un ejemplo brillante y simple de cómo se puede emplear un modelo para resolver un problema. También muestra de manera pedagógica los límites de un modelo.

Las razones para construir un modelo son las siguientes:

- Explican (diferente a predecir).
- Guían a través de la colección de datos.
- Iluminan comportamientos dinámicos.
- Sugieren analogías dinámicas.
- Descubren nuevas preguntas.
- Promueven un hábito mental científico.
- Acotan los resultados en rangos plausibles.
- Ofrecen análisis de opciones críticas casi en tiempo real.
- Demuestran disyuntivas y sugieren mejoras.
- Desafían la robustez de la teoría frente a perturbaciones.
- Exponen incompatibilidades en los datos.
- Forman de manera práctica.
- Disciplinan el método de dialogar.
- Educan al público en general.
- Revelan que lo aparentemente simple puede ser complejo y que lo aparentemente complejo puede ser simple.
- ...

Las consideraciones que pueden hacerse sobre los modelos son:

- Los modelos son simplificaciones necesarias.
- Los modeladores afirman que «todos los modelos son incorrectos», pero a pesar de ello, «algunos» son útiles. Lo que es una gran verdad.
- Las inexactitudes de los modelos obligan a dos tipos de comportamientos:
 - El primero es constantemente deben refinarse y mejorarse, si se desea obtener resultados más exactos.

- El segundo, debido al hecho de que los modelos no son perfectos, se deben conseguir colecciones de modelos diferentes.
- En sistemas complejos, los modelos simples no funcionan. Hacer modelos no es fácil.
- Para que tenga sentido el modelo, una persona necesita una multitud de modelos para conseguir crear un diálogo interno entre la lógica del modelo y el comportamiento del mundo.
- Cuantos más modelos hay disponibles, más posibilidades haya para recombinar sus partes y producir mejores modelos, lo que va a ser necesario para enfrentarnos a la creciente complejidad del mundo.
- Aunque cada tipo de problema tiene su conjunto de modelos típicos asociados que se originan en disciplinas concretas. Muchos de ellos pueden aplicarse en otras disciplinas con mucho aprovechamiento.

La utilización de modelos nos permite ser pensadores más profundos, generar mejores soluciones a problemas complejos y avanzar en una comprensión más coherente de nuestro mundo. La evidencia muestra que las personas que piensan con modelos de manera consistente superan a aquellos que no lo hacen. Y, además, aquellas personas que utilizan muchos modelos superan a las que utilizan menos.

La pregunta que queda es la siguiente:

¿Se puede transformar un modelo en 10 millones de dólares?

La respuesta es que sí, tan solo hay que conocer icuándo, cómo y dónde aplicarlo! Tomando prestada la elegante frase de Duncan Watts «todo es obvio: una vez que se sabe la respuesta».

Sigamos analizando mejor la segunda componente citada al principio, que de manera resumida podría expresarla como «Hay que enseñar a las máquinas». De cara a resolver problemas complejos lo que necesitamos es introducirnos en el campo de la inteligencia artificial. Este campo de estudio se basa en la afirmación de que una propiedad fundamental de los seres humanos, la sabiduría del *Homo sapiens* «puede ser descrita de manera tan precisa como para que una máquina puede hacerse pasar por inteligente». Este va a ser el objetivo que veremos con detalle más adelante en este texto.

Y finalicemos ampliando la tercera componente que hemos denominado «fertilización cruzada y creación de grupos». Hablando ahora del tema de grupos de trabajo. Las sociedades están formadas por individuos con diversas capacidades cognitivas. Las causas de esas diferencias son:

- La formación adquirida según sus experiencias vitales. (Causas directas).

- Los genes que se consideran causas indirectas y que quizá sean de una importancia menor en este contexto.

Imagínese ahora dos conjuntos de personas, un conjunto formado por personas diversas y otro conjunto formado por personas homogéneas, ambas desde el punto de vista de sus experiencias vitales. Supongamos también que la «habilidad» (capacidad para trabajar bien en soledad sobre un problema concreto) es la misma en todas las personas. Entonces, se sabe que el conjunto formado por los resolutores de problemas, del tipo diverso, en promedio, supera a la colección de resolutores de problemas de tipo homogéneo. También se sabe que, dadas las condiciones suficientes (no se dice nada de ser necesarias), una colección de resolutores de problemas seleccionada aleatoriamente, supera a una colección de resolutores de problemas formada por los mejores resolutores de problemas.

Gente artificial. De la ficción a la realidad

¿De qué va esto?

De cómo hemos llegado al actual entorno tecnológico digital. Del marco conceptual actual, y por donde se prevé que hay que seguir.

La estructura de este capítulo está formada por los siguientes subapartados:

- Un mundo de ficción.
- Bases ideológicas.
- El entorno tecnológico actual.
- El marco conceptual.
- Por dónde se prevé que hay que seguir.
- Conclusiones.
- Vuelta a la ficción.

Un mundo de ficción

No sé si ustedes han visto alguna vez la película de 1982 titulada *Blade Runner*, del director Ridley Scott. Con los actores principales Harrison Ford, Rutger Hauer, Sean Young, Edward James Olmos. Música de Vangelis y guion basado en la novela *¿Sueñan los androides con ovejas eléctricas?* de Philip K. Dick. Es una película de esas que recomiendo porque a mí me impactó cuando la vi por primera vez con 27 años, después la he vuelto a ver y he leído sobre ella, incluida la novela.

Una sinopsis sería la siguiente: «*A principios del siglo XXI, la poderosa Tyrell Corporation desarrolló un nuevo tipo de robot llamado Nexus, un ser virtualmente idéntico al hombre y conocido como Replicante. Los Replicantes Nexus-6 eran superiores en fuerza, agilidad, y al menos igual en inteligencia a los ingenieros de genética que los crearon. En el espacio exterior, los Replicantes fueron usados como trabajadores esclavos en la arriesgada exploración y colonización de otros planetas. Después de la sangrienta rebelión de un equipo de combate de Nexus-6 en una colonia sideral, los Replicantes fueron declarados proscritos en la Tierra bajo pena de muerte. Brigadas de policías especiales, con el nombre de Unidades de Blade Runners, tenían órdenes de tirar a matar al ver a cualquier Replicante invasor. A esto no se le llamaba ejecución, se le llamaba retiro»». Un aspecto crucial de la película es que los Replicantes tenían una fecha concreta de muerte su vida era de unos cuatro años para evitar que desarrollen una cognición empática y, por lo tanto, inmunidad a la máquina Voight-Kampff que servía para detectarlos. Por ello, vuelven a la Tierra con objeto de que su creador les amplíe el plazo.*

Hay una escena que me impactó especialmente, hacia el final de la película el modelo Nexus-6 Roy Batty (Rutger Hauer) que es el último Nexus-6 vivo y además el líder de la revuelta, está envuelto y de hecho ganando la lucha con Rick Deckard

(Harrison Ford) que está colgando del alfeizar de una casa. Roy, que siente el final de su vida, decide sujetarlo, elevarlo y situarlo sobre el tejado. En sus últimos instantes se sienta frente a él y le recita el siguiente texto. *«Yo he visto cosas que vosotros no creeríais: Atacar naves en llamas más allá de Orión. He visto rayos C brillar en la oscuridad cerca de la Puerta de Tannhäuser. Todos esos momentos se perderán... en el tiempo... como lágrimas en la lluvia. Es hora de morir».*

¡Una maravilla!... En realidad, una ficción.

Sigamos con otra película, esta vez de 2002, titulada *The Time Machine*. Del director, Simon Wells nieto del famoso escritor H. G. Wells. Producida por Arnold Leibovitz, basada en la novela *The Time Machine* de H. G. Wells (1895) y la película de 1960 *The Time Machine* de David Duncan. Esta película está protagonizada por Guy Pearce, Jeremy Irons, Orlando Jones, Samantha Mumba, Mark Addy, Sienna Guillory y Phyllida Law. Es una película que recomiendo en mis conferencias de IA porque, como ya he dicho previamente, con 47 años me sugirió una nueva línea de investigación.

Una sinopsis sería la siguiente: El científico e inventor Alexander Hartdegen pretende demostrar que los viajes a través del tiempo son posibles. Una tragedia personal que lo ha sumido en la desesperación explica su deseo de volver al pasado para alterar el curso de los acontecimientos y salvar así a la chica de su destino fatal. Pero la máquina de su invención, en contra de sus deseos, lo lleva al futuro, concretamente a 800 000 años después, donde descubre que la humanidad se ha dividido en dos bandos: los depredadores y las presas.

La secuencia de la película que me impactó es aquella en la que Alexander, el viajero en el tiempo, se dirige en el futuro a la biblioteca pública de la Quinta Avenida. Allí conoce a Vox 114, un bibliotecario creado gracias a la inteligencia artificial, el cual le informa sobre las aplicaciones prácticas de los viajes en el tiempo y le asegura que la idea de viajar al pasado es imposible. El personaje del bibliotecario es un holograma con capacidad visual y verbal conectado a todas las bases de datos del planeta.

¡Una maravilla!... De partida una ficción, pero a la que nos acercamos realmente mucho.

Bases ideológicas

Usted se preguntará ¿por qué hablar de gente artificial? Personalmente le puedo responder que tanto la parte de los efectos especiales como la relacionada con la inteligencia artificial (IA) son parte de mis objetivos de trabajo. De hecho, me atrae la búsqueda de sistemas artificiales que hagan lo que se necesita, sin decirles exactamente cómo hacerlo. Otra parte de la respuesta es que es un tema abierto de investigación, desarrollo e innovación (I+D+i) que está en la frontera del

conocimiento y, por lo tanto, es interesante. El horizonte de mi trabajo sería realizar una aproximación tecnológica al problema de la mente. Debo aclarar de entrada el hecho de que estamos empezando a ser capaces de concebir cómo podría ser esta respuesta en función de las leyes conocidas de la física, la biología, ... o la información. ¡Pero eso es lo interesante! ¿No?

La primera pregunta que uno se hace es ¿se puede modelar la mente? Para responder hay que aceptar un hecho innegable, la mente es una magnífica herramienta de adaptación, fruto de la evolución biológica basada en el azar y la necesidad, que ha producido como resultado un procesador neuronal material, que está capacitado por la selección natural para manejar algoritmos combinatorios del razonamiento causal y probabilístico que le ha servido al *homo sapiens* para alcanzar sus objetivos de supervivencia y de felicidad.

El sentido común nos sugiere que la mente y el cuerpo deben interactuar. Nuestras percepciones, pensamientos, intenciones, deseos y emociones afectan directamente a nuestros cuerpos y a nuestras acciones. Parece claro que el sistema nervioso completo del ser humano forma parte del mundo físico: tangible, visible, público extenso en el espacio, cuyo análisis se basa en la observación. Sin embargo, los pensamientos, los sentimientos, las consciencias y otros, se nos presentan como estados de la mente: intangibles, invisibles, privados, ordenados en el tiempo, pero no en el espacio, cuyo análisis se basa en la introspección. La pregunta que surge es si, por un lado, el encéfalo y los estados físicos, y, por el otro, la mente y los estados mentales son cosas fundamentalmente diferentes. ¿Es lícito el dualismo? Opino que en esa pregunta hay un error lógico: que dos cosas tengan un acceso epistémico diferente no implica que sean dos cosas diferentes. Hasta que no sepamos lo suficiente de cómo funcionan los estados físicos del cerebro. Podría ser que solo estemos contemplando dos perspectivas de un mismo fenómeno.

Para mí, lo que es muy evidente es que la mente es un fenómeno emergente de la materia. La mente humana no es siempre el precioso recipiente espiritual que tanto valoramos. Ciertas afecciones cerebrales desbaratan su funcionamiento, como por ejemplo el alzhéimer. Sea esa la explicación u otra diferente, se está en camino de empezar a entender por qué es entendible! Y, por lo tanto, tarde o temprano acabará siendo modelable.

Si la respuesta a la pregunta de si se puede modelar la mente, fuera sí, la segunda pregunta que interesa en el marco de este libro sería: ¿se podría simular con un computador?

El concepto de *cognición* (del latín: *cognoscere*, 'conocer') hace referencia a la facultad de los seres de procesar información a partir de la percepción, el conocimiento adquirido y las características subjetivas que permiten valorar y considerar ciertos aspectos en detrimento de otros.

Existe un modelo simple del comportamiento de un ser humano como un procesador de la información que recoge del mundo exterior a través de sus sentidos (sensores de entrada), esta información cuando llega al cerebro estimula las neuronas y se interpreta para generar las sensaciones, posteriormente se procesa para tomar las decisiones adecuadas, que se reflejan en los controladores cerebrales de nuestros actuadores y estos a través de nuestros actuadores (elementos de salida) modifican el mundo exterior. Y de paso la información informa a «alguna parte de mi cerebro» para que esta produzca, de una manera totalmente desconocida a día de hoy por la ciencia, un estado mental (que ya no es información, solo es una sensación consciente).

Si ese ser se puede explicar como un sistema que procesa información, entonces podría simularse mediante un computador, que es la máquina universal de procesamiento de la información que el ser humano ha construido.

Este enfoque en términos de la teoría de la información ha sido uno de los grandes descubrimientos del siglo XX y ha propiciado e impulsado la utilización de la informática y de la automática, entre otros tipos de conocimiento para la construcción de sistemas artificiales inteligentes.

Entonces se preguntará, ¿ha venido a decirnos cómo la información produce estados mentales? Me temo que no. Gente como yo, pensamos que debe existir algún tipo de proceso físico, químico, biológico todavía desconocido que produzca esas sensaciones. Descubrir qué es ese algo es todavía una línea fundamental de investigación con resultado completamente desconocido. Por ahora, solo quiero insistir en que es una buena línea de investigación, contarles qué es eso de la inteligencia artificial, hasta donde hemos llegado y qué se vislumbra con el paso del tiempo.

Pero para poder avanzar y alcanzar los objetivos que me he planteado, son necesarios los siguientes pasos previos. Primero, voy a describir el entorno tecnológico actual en el que hay que interpretar todo. Segundo, voy a contarles lo que deseo transmitir bajo la perspectiva de ese entorno. Y, finalmente, por donde se prevé que hay que seguir.

El entorno tecnológico actual

Esta parte está formada por los siguientes subapartados:

- Reflexión histórica sobre los artefactos.
- Reflexión histórica sobre las revoluciones tecnológicas.
- Reflexión histórica sobre la tecnología digital.
- Un anhelo.

Reflexión histórica sobre los artefactos

Empecemos con una idea básica. La evolución de la especie humana está ligada al progreso de los artefactos. El hombre desde su comienzo ha tenido la capacidad de imaginarlos, diseñarlos, construirlos y utilizarlos. El *homo sapiens* es el *homo faber*. De hecho, sin ciencia y tecnología el ser humano probablemente no habría sobrevivido, actualmente seríamos un fósil más. La aparición de la técnica fue posible por el desarrollo de la facultad racional, y desde entonces el tándem (racionalidad, técnica / tecnología) han ido apoyándose el uno en el otro y el otro en el uno, consiguiendo tanto la mejora de la capacidad de resolver los problemas del humano como la mejora imparable de los procesos técnico / tecnológicos. Su historia es la historia de la invención de artefactos y procedimientos con un propósito práctico en un entorno económico.

Un artefacto, ya sea una herramienta o una máquina, es un objeto elaborado con el fin de facilitar la realización de una tarea mecánica. Que requiere de una aplicación correcta de energía y que precisa de un elemento de control para su correcto funcionamiento. Por ejemplo, un arco, es un objeto elaborado por un ser humano con el fin de cazar o defenderse. Para que funcione hay que suministrarle una energía que permita tensar la cuerda o cable. Y para que su funcionamiento sea correcto, es necesario que el arquero sepa controlar el dispositivo apuntando bien.

Desde este punto de vista, el progreso humano visto a través de los artefactos no es más que la consecución a lo largo del tiempo de artefactos mejores y más potentes. Si hacemos un recorrido rápido por la evolución histórica de los artefactos, podríamos decir que estos han sido sus etapas principales.

- Primera. La aparición de la herramienta.

Entendiendo que una herramienta es un artefacto, alimentado por la energía humana o animal, que requiere de un operador humano para su función de control. Surgieron a través de los cazadores-recolectores del Paleolítico, principalmente para la adquisición de alimentos. El propio ser humano era el que proporcionaba la energía. Ejemplos podrían ser el canto tallado, la lasca y el bifaz, o la raedera, la lanza, la flecha o el martillo.

Posteriormente durante la época del Neolítico, hace unos 10 000 años, la humanidad dejó de ser nómada y pasó a convertirse en sedentaria. Conocieron la agricultura y la ganadería, desarrollaron técnicas artesanales y comenzaron a trabajar los metales. En esta época, los animales de tiro o carga, como los caballos, los bueyes, los camellos... proporcionaron la energía para herramientas como el arado o el carro.

- Segunda. La aparición de la máquina.

Una máquina es un artefacto alimentado por energía no humana o animal, es decir, la energía la proporciona la naturaleza. Que requiere de un operador humano para su función de control. Ejemplos de máquinas, a lo largo de la historia son: los barcos de vela, los molinos de viento, el ferrocarril de vapor, los molinos eólicos para producir energía, los coches, los aviones, las centrales nucleares.

Si hacemos un pequeño resumen de lo dicho hasta ahora, observemos que, hasta hace poco, las herramientas y las máquinas permitían acelerar el trabajo y liberar a las personas de las tareas arduas, permitiendo a los seres humanos superar tremendamente los límites de sus cuerpos. Piense en una grúa portuaria controlada por un solo hombre que le permite mover varias toneladas de peso desde un cómodo asiento. Pero la historia continua con la aparición de unas máquinas basadas en una idea singular.

- Tercera. La emergencia del autómeta.

Esta etapa surge cuando hacen su aparición ciertas máquinas basadas en una idea antigua pero singular, «el autómeta». El autómeta es una máquina que se caracteriza porque elimina el elemento de control humano mediante un procedimiento intrínseco a ella misma. En sus inicios esta eliminación se basaba en algún algoritmo físico y en la actualidad además de ellos se completa con algoritmos de *software*. Ejemplos, los diferentes autómetas del Renacimiento, el regulador centrífugo de Watt de 1788, el telar mecánico controlado por tarjetas perforadas de Jacquard de 1801, el reloj automático Maurice Lacroix Masterpiece Skeleton Chronograph de los noventa del siglo pasado, o los coches autónomos de nuestros días.

Ahora bien, dentro de los adelantos surgidos en la tercera etapa, y después de la Segunda Guerra Mundial, surgen los autómetas por antonomasia que son los artefactos informáticos que son mediadores simbólicos que amplifican el intelecto más que el músculo de quienes los utilizan.

Reflexión histórica sobre las revoluciones tecnológicas

Los conocimientos científicos y tecnológicos siempre han producido un gran impacto social y han conducido a revoluciones. Las principales revoluciones que nos interesan en este contexto son las siguientes:

- La industria 1.0, o la era de la revolución basada en la mecanización.

Surge a finales del siglo XVIII, impulsada por el vapor de agua.

- La industria 2.0, o la era de la revolución basada en la electricidad

Emerge al inicio del siglo XX, con ella se genera una producción masiva impulsada por la energía eléctrica.

- La industria 3.0, o la era de la revolución basada en la automatización.

Surge alrededor de los setenta del siglo XX, la producción se automatizó mediante el uso de los dispositivos electrónicos.

- La industria 4.0, o la era de la revolución basada en las redes de humanos, máquinas y cosas.

Se desarrolla durante el primer cuarto del siglo XXI, la producción está basada en el uso de sistemas ciber físicos.

Reflexión histórica sobre la tecnología digital

Como ya he dicho, los artefactos informáticos son mediadores simbólicos que amplifican el intelecto, más que el músculo de quienes los utilizan.

Del mismo modo que hemos descrito rápidamente la evolución histórica de los artefactos, vamos a ir recorriendo sus etapas principales, pero en este caso en vez de usar la palabra *etapa*, se suele denominar a los hitos principales como *eras*.

- Era 1.0. Surge el concepto de *información* y de su procesado automatizable.

Referentes de esta era son: Charles Babbage (1822), Herman Hollerith (1900), Claude Shannon (1948), Alan Turing (1950).

En esta época, la revolución de la industria (1.0) se basa en la tecnología mecánica y la automatización asociada.

La relación o cociente promedio entre el número de computadores partido por el número de personas es de 0, ya que no existían los computadores.

- Era 2.0. Surge el mundo de los sistemas programables.

Referentes de esta época son: el ENIAC (1945), computador con una arquitectura física basada en válvulas; el EDVAC (1951), computador con una arquitectura física basado en válvulas y en una arquitectura lógica del tipo Von Newman; el IBM 650 (1953), computador con una arquitectura física basada en transistores. En 1956, se acuña la expresión *inteligencia artificial*. El IBM 360, computador con una arquitectura física basada en circuitos integrados.

La revolución de la industria (2.0) se basa en la tecnología eléctrica y la automatización asociada.

La relación o cociente promedio entre el número de computadores partido por el número de personas era de un computador para muchísimos usuarios en centros o empresas o instituciones muy específicos.

- Era 3.0. Surge el mundo del procesado de la información con datos estructurados en entornos con certeza. Más adelante entenderá con mayor precisión esa frase.

Referentes de esa época son: a partir de 1965, simulación exitosa de sistemas basados en las leyes que proporcionan las ciencias naturales; en 1993 Internet se hace de dominio público; en 1995 aparece en el cielo de los Balcanes el «General Atomics MQ-1 Predator», inicialmente llamado RQ-1 Predator; es un vehículo aéreo no tripulado clasificado por la Fuerza Aérea de los Estados Unidos como de altitud media y largo alcance; en 1997 se realiza la primera confrontación entre una máquina y un hombre jugando al ajedrez, resultando exitoso para la máquina; en 1999 se empieza a hablar de la Internet de las Cosas; en el 2000, ya hay 600 millones de usuarios de Internet; en el 2003 se habla del *Big Data*, el mundo produce alrededor de 5 exabytes (10^{18}) de información nueva al año. Apostillemos que la IA sigue avanzando en la panoplia de sus conocimientos y aplicaciones.

En esta época, la revolución de la industria (3.0) se basa en la tecnología electrónica y la automatización asociada.

La relación o cociente promedio entre el número de computadores partido por el número de personas alcanza el valor aproximado de 1.

- Era 4.0. Surge el mal llamado «Smart world» o «mundo inteligente». Es un término que relaciona las modernas técnicas de automatización, el intercambio de datos y las tecnologías de fabricación. Que asocia a la cadena de valor, el uso de sistemas ciber físicos, Internet de las Cosas e Internet de los Servicios.

Referentes de esa época son: en el 2006 surge la idea del *Cloud computing*, o computación en la nube; en el mismo año se describe lo que podría llegar a ser la Web 3.0, o Web semántica; en el 2011 se acuña la referencia «industria 4.0» a la que se le asocia el adjetivo de inteligente; en el 2011 se presenta el sistema IBM Watson, en el 2014 la Internet de las Cosas ya conecta 10^9 dispositivos con diferentes niveles de procesamiento de la información. Además, se dirige la atención hacia los datos no estructurados como las

imágenes, el audio, el vídeo..., que muestran un crecimiento en su uso galopante, y la IA sigue avanzando.

En esta época, la revolución de la industria se basa en la tecnología electrónica y la automatización asociada.

La relación o cociente promedio entre el número de computadores partido por el número de personas alcanza valores de cientos por cada persona.

Como comentarios aclaratorios de lo dicho hasta ese momento, indicaré que:

- Un sistema ciber físico no es más que la «nueva palabra» que se aplica a la unión de la robótica + redes de sensores + la IA, de manera que se pretende conseguir sistemas más adaptables, autónomos, eficientes, funcionales, fiables, seguros y usables.
- Que el significado de la palabra *cosas* en la Internet para las cosas, normalmente se asocia a cosas físicas, y eso es muchísimo menos de la mitad de las «cosas» que existen en el mundo de los *homo sapiens*.
- Del mismo modo, el concepto de *robot* suele ir asociado al concepto de *artefacto autónomo tangible* que ocupa una parte del espacio físico existente. Otro error conceptual más común de lo que sería deseable. El mundo de los agentes autónomos abarca también el de los agentes computacionales o virtuales (bot's). Este otro tipo de agentes están llegando para quedarse y van a ser, en un corto plazo, importantes económicamente en muchas situaciones empresariales.
- El concepto de *nube* debe ampliarse con el concepto de *niebla*. La computación en la nube es remota y la computación en la niebla es local. Como siempre los procesos de información son más eficaces y eficientes cuando no se desprecia ninguna de las posibilidades existentes.

¿Qué es lo que se espera a continuación? Para muchas personas, no hay diferencia entre «Smart» e «Intelligence», debido a la creencia de que son sinónimas. Sin embargo, hay una diferencia importante en su significado y, por lo tanto, en el uso correcto de estas palabras.

«Smart» se aplica a las inferencias aprendidas. Cuando estudiamos algo y aprendemos, nosotros mejoramos nuestra inteligencia en relación con la materia estudiada. En cambio, con la «Intelligence» se nace, es algo inherente a nuestra carga genética. Es la «capacidad» mayor o menor que una persona tienen para pensar, resolver problemas, razonar, pensar, comprender... Por decirlo de una manera más simple, la «Intelligence» es una medida de su habilidad para llegar a ser más «Smart» a través del aprendizaje.

Volviendo a la pregunta que encabeza este apartado, empezaré diciendo que lo que se pretende actualmente y de manera clara y sin discusiones (salvo filosóficas) es ir en la búsqueda de máquinas (5.0) verdaderamente calificables como «smart». Previamente cuando he hablado de la era 4.0 la he adjetivado como «la mal llamada era del mundo smart». ¿Por qué? Me imagino que la razón es que, para los tecnólogos, al hacer uso de las técnicas de inteligencia artificial en muchas de sus aplicaciones, ha sido cómodo y muy interesante desde el punto de vista del *marketing* de la I+D+i, utilizar la palabra *smart*.

- Qué se espera que sea la era 5.0. Se empieza a hacer referencia a ella con el nombre de «Cognitive computing» o computación cognitiva. En ella, surgirá el mundo del procesado eficiente de la información con datos no estructurados en entornos con incertidumbre. Además, se va buscar la integración de cuatro tipos de tecnologías, sistemas que aprendan basados en el conocimiento neurocientífico, gestión rápida de *Big Data* que provendrá de miles de sitios, computadores paralelos con capacidades de supercomputación y el uso de nanodispositivos.
- En cuanto a la «Conexión», los dispositivos se deberían diseñar para auto conectarse y auto monitorizar su comportamiento.
- Con respecto a la «Conversión», a partir de los datos provenientes de los dispositivos auto conectados y de los sensores, se deberían identificar las características de aquellos aspectos más sensibles del sistema utilizando dispositivos de auto reparación; el sistema podría utilizar esta información como mecanismo de predicción de las situaciones potencialmente críticas.
- A nivel «cyber», cada máquina debería poder crear su propio «modelo-representación» lo que le permitirá tener un control de la seguridad del sistema. Este (ciber) modelo que se crease permitiría hacer autoajustes del sistema y síntesis de datos con posible interés en el futuro.
- A nivel «cognitivo», los resultados de las fases de autoevaluación y autovaloración se deberían presentar a los usuarios mediante recursos infográficos para hacer uso de la vista, que es el sentido humano que tienen el mayor ancho de banda a la hora de analizar información. De esta manera es factible mostrar el contenido y el contexto de los datos de cara a analizar potenciales situaciones futuras (de oportunidad y/o riesgo).
- Con respecto a la «Configuración», el sistema debería reconfigurarse basándose en criterios de prioridad y/o riesgo con el fin de alcanzar comportamientos «resilientes».
- ... (la lista es más extensa).

Todo ello para integrar lo mejor de los computadores y de los humanos a la hora de resolver problemas. Más adelante entenderá con mayor precisión este párrafo.

La relación o cociente promedio entre el número de computadores partido por el número de personas alcanzará valores de varias magnitudes por cada persona.

Un anhelo

Con esos antecedentes ¿es de extrañar que el ser humano también haya intentado concebir máquinas con algún tipo de talento? La pluralidad de individuos que se han diseñado mentalmente o se han intentado hacer por el ingenio, mano o arte del hombre se pueden clasificar en dos grandes grupos:

- Los individuos mejorados, actualmente denominados *ciborgs* (organismos cibernéticos).

La idea subyacente en todo ellos es que, si no te conformas con lo que te dio la Madre Naturaleza, usa la tecnología para crear una versión mejorada de tu persona. El resultado es un ser que en parte es humano, y en parte es máquinas.

El concepto se basa en el desarrollo de partes biónicas del cuerpo, prótesis y órganos artificiales, y en la actualidad órganos naturales que hacen concebir grandes esperanzas.

Ejemplos reales que podrían servir:

- Siglo V a. de C. Herodoto ya nos habla de un hombre con una pierna de madera.
- Siglo XIX. Los primeros relojes de pulsera inventados por Patek Philippe y diseñados para mujeres.
- (1961) implantes cocleares.
- (2000) Homayoon Kazerooni, director del Laboratorio de Ingeniería robótica y humana de la Universidad de Berkeley, crea BLEEX un exoesqueleto que permite a un hombre cargar con 90 kg como si fueran solo 4,5 kg.
- (2001) Brack Hattler de la Universidad de Pittsburgh presentó un pulmón artificial que puede implantarse en un paciente durante como mucho dos semanas.
- (2003) Miguel Nicollis implantó un chip en el cerebro de un mono con el que el animal dirige un brazo robotizado. En el 2009 pretendía implantarlo en un ser humano.
- (2006) un conejo recibe un pene artificial cultivado a partir de sus propias células.
- ...

Visiones de la ciencia ficción

- Humanos con atributos mecánicos:
 - L'Horlogere (*mechanical mistress*).
 - Number 18 (*Dragonball Z*).
 - Robocop.
 - The Bionic Woman.
 - Jax (*Mortal Kombat*).
 - 6 Million Dollar Man.
 - Molly and Dixie Flatline (*Neuromancer*).
 - Seven of Nine (*Star Trek*).
 - ...
- Máquinas con atributos humanos:
 - Data (*Star Trek*).
 - Terminator (Arnold Schwarzenegger).
 - Vicky (*Small Wonder*).
 - T-1000 (*Terminator 2*).
 - Andrew (*Bicentennial Man*).
 - DARYL.
 - ...
- Humanos con atributos mágicos:
 - Harry Potter.
 - Fibi (*Charmed*).
 - ...
- Los individuos artificiales, denominados robots o personajes sintéticos.

En nuestro mundo cada vez más virtual, donde pueden realizarse con facilidad negocios y establecer relaciones sin contacto humano, el paso siguiente es el de las máquinas capaces de conducta inteligente, de aprendizaje y de adaptación.

Ejemplos reales que podrían servir:

- (1495) Leonardo da Vinci diseña un caballero mecánico.
- (1936) Alan Turing. Se publica el artículo «Los números computables».
- (1966) Joseph Weizenbaum, del MIT, crea ELIZA, el primer chatbot.
- (1999) La NASA confía el control de Deep Space 1 a un sistema de IA.
- (2001) El Global Hawk vuela de California a Australia sin tripulación.

- (2005) El coche-robot Stanley, de Stanford Racing Team, ganó el grand Challenge 2005 de DARPA al completar una carrera de 229 km a través del desierto en menos de 7 horas.
- (2005) Replicante Q1, robot actroide que parpadea, habla y respira. Puede mostrar expresiones de alegría, enfado y tristeza.
- ...

Mitos y visiones de la ciencia ficción:

- Talos el gigante de bronce.
- Pigmalión y la estatua.
- El Golem.
- Paracelso y su hombre artificial.
- El homúnculo de Fausto (Goethe).
- Frankenstein (Mary Shelley)
- Robots universales de Rossum Karel Čapek.
- El ajedrecista de Maelzel (Edgar Allan Poe).
- Pinocho.
- Data (*Star Trek*).
- David (*Artificial Intelligence*).
- Gort (*The Day the Earth Stood Still*).
- Robbie (*Forbidden Planet*).
- Hal 9000 (*2001: A Space Odyssey*).
- C3PO y R2D2 (*Star Wars*).
- Roy, Pris, Zhora, Leon (*Blade Runner*).
- Rachel (*Blade Runner*).
- T800 (*Terminator*).
- Sonny (*I Robot*).
- María I y María II (*Metrópolis*).
- Aki Ross (*Final Fantasy*).
- Rosie (*Los Jetsons*).
- ...

El marco conceptual

Esta parte está formada por los siguientes subapartados:

- La inteligencia artificial (IA).
- La vida artificial (VA).
- El denominador común.

La inteligencia artificial (IA)

Alan Turing publicaba en 1950 su artículo «Computing Machinery and Intelligence» en el que establecía el test que hoy lleva su nombre. Decía que si una máquina se comportase en todos los aspectos como inteligente, entonces tendría que ser considerada inteligente o, al menos, deberíamos atribuirle tanta inteligencia como a su interlocutor humano. Tras las primeras reflexiones de Turing, la siguiente gran cita histórica hacia la formación de una nueva disciplina y una comunidad investigadora en «Inteligencia Artificial» (IA) se produjo alrededor de la Conferencia en el Dartmouth College (Estados Unidos), organizada por J. McCarthy en 1956 que definió la inteligencia artificial como «la ciencia e ingeniería de hacer máquinas que se comporten de una forma que llamaríamos inteligente si el humano tuviese ese comportamiento».

Para muchas personas las palabras *inteligencia* y *artificial* son contrapuestas. Para mí no lo son, si entiende lo que esas dos palabras juntas pretenden indicar.

Desde entonces la IA ha ido evolucionando, con más o menos altibajos, hasta llegar a nuestros días. En estos momentos es un área multidisciplinaria, que a través de ramas de conocimiento como las ciencias de la computación, la matemática, la lógica, la neurociencia, la filosofía, las ciencias sociales... estudia la creación y el diseño de sistemas capaces de resolver problemas cotidianos por sí mismos, utilizando como paradigma la inteligencia humana. Es decir, nos fijamos como actuamos y nos elegimos como modelo, patrón o ejemplo.

Formalmente la inteligencia artificial propone soluciones ingenieriles potentes y eficaces normalmente **disjuntas** para problemas concretos. Por ejemplo, visión artificial, *big data*, búsquedas, aprendizaje...

La vida artificial (VA)

Es el estudio de los sistemas construidos por el hombre que exhiben comportamientos característicos de los sistemas vivos naturales. También es un área multidisciplinar en las que intervienen las ciencias de la vida, las ciencias del conocimiento, las ciencias de la computación, las ciencias de la naturaleza...

El objetivo prioritario es tratar de capturar la fenomenología informacional que ocurre en el interior de los seres vivos. Lo que pretende es simular la lógica de la

vida y no la propia vida, es decir, cómo realizan el procesamiento de la información interpretada de manera algorítmica obviando el sustrato físico que siempre es diferente entre los seres vivos y las máquinas. La razón de este tipo de trabajos es que los animales son agentes autónomos naturales que se adaptan con éxito a entornos tan complejos como los naturales.

La diferencia de la aproximación (VA) con respecto a la IA, es que los seres vivos utilizan marcos **unificados** para resolver cualquier tipo de situación tanto conocida como desconocida. Esos marcos unificados se denominan en el lenguaje coloquial «mentes» y algunos de esos seres vivos a los que hacemos referencia se caracterizan por que «saben lo que están haciendo».

Hay razones de dos tipos para intentar conseguirlo: Construir máquinas más eficaces que pudieran autoconfigurarse, autooptimizarse, autorrepararse y autoprotgerse. Modelar sistemas naturales y, por último, construir artefactos como nosotros. Durante el proceso de avance, el beneficio es doble. Mejor comprensión del comportamiento animal y su inteligencia. Y el desarrollo de herramientas y de técnicas inspiradas en la biología animal para construir sistemas autónomos que se adapten a entornos complejos.

El denominador común

Sin intentar ser exhaustivo empezaré enunciando algunas características o capacidades que por ejemplo un sistema, ya sea de IA o de VA debería tener para poder ser bien adjetivado verdaderamente como sistema «Smart» capaz de interactuar en su dominio de definición o entorno:

- Capacidad de percibir el entorno y capacidad de actuar sobre el entorno. Es decir, capacidad de comportarse de manera interactiva en tiempo real desde la perspectiva del sistema.
- Capacidad de comunicación con otros sistemas.
- Capacidad de procesado en paralelo.
- Capacidad de aprender y recordar.
- Capacidad de adaptación. Es decir, capacidad de reaccionar a los cambios del entorno intentando alcanzar los objetivos definidos (intencionalidad).
- Capaz de generar comportamientos diferentes, para situaciones diferentes (robustez y generalidad).
- Capacidad de mostrar comportamientos emergentes, es decir, comportamientos no programados por el diseñador.

Desde el punto de vista perceptivo:

- Capacidad de detectar Gestalts. El sistema debería ser capaz de detectar las «estructuras de lo que se percibe», por ejemplo, cuando un ser humano escucha música, no solo escucha secuencias de sonido (lo que se percibe), sino que es capaz de extraer la melodía (la estructura de lo que se percibe).
- Capacidad de detectar propiedades funcionales («affordances»). La función esencial de los sistemas perceptivos es la de ser capaces no solo de recoger información, sino la de obtener propiedades funcionales de lo que se percibe respecto al sujeto perceptor. Por ejemplo, una taza con su asa podría ser lo detectado, pero la propiedad funcional de su «agarrabilidad», usando el asa o sin usarla, es solo percibible por un sistema que tenga posibilidades de intervención sobre el mundo basadas en la existencia o no, de manos.
- Capacidad de diferenciar entre leyes y correlaciones. Las leyes detectan relaciones «causales» y pueden servir, por lo tanto, para dar explicaciones, las correlaciones tan solo muestran la existencia de posibles relaciones.
- El problema de la relevancia de la información o del «framework problem». Entender una situación para dotarla de sentido, pasa por distinguir entre la información que se recibe y es relevante para alcanzar el objetivo, y aquella información que se recibe que no es útil en ese momento, para alcanzar ese objetivo.
- El problema del significado de los símbolos «Symbol-Grounding Problem». Cuando se utilizan sistemas informáticos, es completamente inevitable tener que codificar la información que se procesa mediante símbolos. Símbolos definidos por el diseñador que normalmente el sistema utiliza sin conocer en realidad su significado en el mundo. Si se desea crear sistemas «autónomos» que interaccionan con su entorno, el sistema debe ser capaz de entender el significado de dicho símbolo en el mundo en el que se encuentra enraizado, es decir, su semántica.
- ... (la lista es más extensa).

Por dónde se prevé que hay que seguir

En estos momentos, la IA y la VA intentan aproximarse a solucionar los siguientes desafíos:

- Construir sistemas capaces de aprender y mejorar según la experiencia temporal, de modo que el sistema pueda modificar su comportamiento en función de la interacción, con el entorno, con otras máquinas o con personas. Se sabe que para ello se deberán aplicar múltiples aproximaciones integradas

de manera efectiva, al modo de lo que hacen los seres humanos en muchas ocasiones.

- Diseñar artefactos capaces de «sentir», «comprender» y «actuar», en entornos dinámicos cambiantes, cada vez más generales, en los que puedan surgir situaciones tanto previstas como imprevistas. Dicho de otra manera, icreando máquinas conscientes!

Y una vez dicho, vamos a dejarlo aquí por el momento. Necesitamos avanzar un poco más en la lectura de este libro para poder seguir hablando de ello.

Conclusiones

Esto que acabo de escribir y de la manera que lo he escrito, puede parecer descabellado. Es evidente que en el momento del conocimiento actual de cómo procesan la información los sistemas vivos puede parecer un desafío demasiado temprano, demasiado complejo o incluso según para quienes, imposible de alcanzar. Pero lo que sí es verdad es que lo enunciado previamente es el objetivo de personas que combinan conocimientos relacionados con la neurociencia, el procesamiento de la información, supercomputación, ingeniería, biología... a la hora de realizar actividades de investigación para avanzar en el conocimiento o de innovación a la hora de aplicar industrialmente el conocimiento ya adquirido.

Tal y como dije al principio, solo quería insistir en que esta temática es un buen tema de investigación.

La evolución no solo ha ocurrido y eventualmente ha conducido a que haya seres capaces de comprender el proceso, sino que seguirá ocurriendo y quizá se llegue incluso a comprender el proceso por el cual ellos lo comprenden. Todo lo que sigue no es más que un recorrido por algunos de los aspectos del conocimiento humano que persiguen esa meta.

Debo aclarar que «estamos empezando» a ser capaces de concebir cómo podrían ser las respuestas. ¿Ficción en estos momentos? No lo sé, pero se estima que el impacto económico de este tipo de actividades será muy elevado y, por lo tanto, interesante.

Planteamiento de una pregunta

Esta parte está formada por los siguientes subapartados:

- La caída de algunos mitos.
- Definición de sexta caída.

La caída de algunos mitos

Si bien al ser humano se le considera el culmen de la creación, en los últimos siglos ha ido perdiendo posiciones. En el siglo II el astrónomo y matemático griego Tolomeo propuso la teoría científica de que la Tierra ocupaba el centro del sistema solar. En el siglo XVII, el astrónomo polaco Copérnico colocó al Sol en el centro y no a la Tierra. En el mismo siglo XVII, el físico italiano Galileo lo confirmó. Por lo tanto, es lícito afirmar que Copérnico nos expulsó del centro del universo. *Primera caída.*

En 1704 se acuña el término *sistema solar*, y durante los siguientes ciento cuarenta años, siglos XVIII-XIX, se terminaron de descubrir los planetas. Viendo la distribución de planetas, distancias al Sol y tamaños relativos, es evidente que perdemos el lugar preponderante. *Segunda caída.*

En el siglo XIX, el naturalista inglés Charles Darwin devuelve al hombre al mundo animal, con lo que nuestro ego volvió a sufrir otra embestida. *Tercera caída.*

Hasta bien entrado el siglo XIX, las galaxias no fueron reconocidas como un tipo de objeto y en menos de cien años hemos pasado de creer que en el universo solo había una galaxia, la nuestra, a una cuenta que ya va por los millones de millones y crece diariamente. *Cuarta caída.*

Hoy sabemos que nuestra Vía Láctea, es una galaxia común. Y conforme la hemos ido estudiando, nos hemos visto arrinconados a una posición bastante irrelevante de la misma. *Quinta caída.*

Comentario: Si seguimos así, nadie diría que somos demasiado especiales, pero no ser especiales, tampoco indica que seamos escoria o basura. Simplemente, somos. Somos y ya está.

Definición de sexta caída

La inteligencia artificial fuerte o IAF, también conocida como inteligencia artificial general o IAG, es la inteligencia artificial que iguala o excede la inteligencia humana promedio, es decir, la inteligencia de una máquina que puede realizar con éxito cualquier tarea intelectual de cualquier ser humano. Es un objetivo importante para la investigación sobre inteligencia artificial y un tema interesante para

la ciencia ficción. ¿Llegará a alcanzarse la IAG?, o dicho en el contexto de este capítulo, ¿estamos predestinados a una sexta caída?

Qué es un algoritmo inteligente. ¿Estamos en la sexta caída?

Este capítulo tiene un doble objetivo, explicar «Qué es un algoritmo inteligente» y responder a la pregunta ¿Estamos en la sexta caída? La estructura que he decidido seguir está formada por dos partes y unas conclusiones. La primera parte la he titulado «Qué es un algoritmo inteligente». La segunda parte se titula «Hagamos bien la pregunta». Se finaliza con el apartado de «Conclusiones».

Primera parte. *Qué es un algoritmo inteligente*

Esta parte está formada por los siguientes subapartados:

- Introducción.
- Qué es un algoritmo.
- Qué es eso de la inteligencia artificial.
- Qué es un algoritmo inteligente.
- Hasta donde pueden llegar.

Introducción

Piense qué respondería si alguien le preguntase *¿Qué sabe usted hacer?*, seguro que le respondería con «algo concreto».

Normalmente, en esa respuesta van implícitas al menos dos cosas: La primera es que siempre que hacemos «ese algo», es porque sabemos lo que estamos haciendo, es decir, somos **conscientes** desde el principio del proceso que vamos a seguir para llegar al final que deseamos obtener. La segunda es que la seguridad en la respuesta dada tiene que ver con el conocimiento profundo o superficial que se tiene de lo que se «sabe hacer».

Y en este momento, creo que estaremos de acuerdo en *que no podemos hacer aquello que no sabemos hacer*.

Qué es un algoritmo

Cuando se mira el mundo tecnológico digitalizado en el que estamos inmersos, uno se puede preguntar ¿cómo es posible todo esto? La respuesta no es única, la evolución de la potencia de cálculo y de almacenamiento ofrecida por los diferentes microchips, la existencia de sensores cada vez más precisos y sofisticados, Internet, el GPS... Pero desde mi particular punto de vista, el elemento clave son los **algoritmos**. ¿Qué es un algoritmo? Tomemos una definición simple. Un **algoritmo** es un conjunto prescrito de instrucciones, bien definidas, ordenadas de la manera adecuada, y finitas. Que, si se ejecutan, convierten los **datos** de un **problema** en una **solución**, es decir, los algoritmos resuelven problemas. Pero para ello son necesarias dos cosas: la primera, que exista un ser o un artefacto que sea capaz de construir el algoritmo;

la segunda, que exista un ser o un dispositivo capaz de entender y ejecutar las instrucciones.

Por ejemplo, resolver un **problema** de electromagnetismo algorítmicamente requiere conocer las leyes de Maxwell (**algoritmo**) y los conceptos matemáticos que las expresan. Eso está en cualquier libro de texto de ese campo de la Física. Pero para resolver ese problema, se ha requerido del señor Maxwell que fue capaz de expresar matemáticamente las leyes del comportamiento de la naturaleza (construir el **algoritmo**) y se requiere de un físico o de un ingeniero, que sea capaz de entenderlas y resolverlas para hallar la **solución** buscada.

Otro ejemplo, cocinar un pollo al chilindrón (**problema**). Si no se sabe cómo hacerlo, se busca una detallada receta (el **algoritmo**) que algún buen cocinero habrá construido y cualquier persona relativamente aficionada a la cocina que entienda cada una de las instrucciones que aparece en la receta, con los ingredientes adecuados (los **datos**) y siguiendo las indicaciones de la receta, conseguirá el plato que deseaba (el **resultado**).

¡Pero! Una característica de los algoritmos es que el que los sigue puede no tener ni idea de lo que está haciendo, con tal de que entienda y sepa ejecutar las instrucciones va a llegar al resultado correcto. Hacer algo y ser conscientes de lo que se está haciendo son dos cosas completamente diferentes.

Veamos otro ejemplo, esta vez supongamos que existe un robot (el dispositivo) que es capaz de doblar cualquier tipo de material maleable. Me imagino que habrá hecho alguna vez en su vida una pajarita de papiroflexia a partir de un trozo cuadrado de papel. En este caso, el dato de entrada sería el trozo cuadrado de papel, la salida sería la pajarita, el algoritmo los pasos que le enseñaron a seguir doblando sucesivamente el papel (el algoritmo) y cada paso o tipo de doblez sería una instrucción.

En este caso, si existiera ese dispositivo que entendiera cada instrucción y la supiera ejecutar, podría llegar al mismo resultado que usted, partiendo de un trozo de papel con las mismas características.

Recordemos ahora un párrafo anterior que decía, *«Una característica de los algoritmos, es que el que los sigue puede no tener ni idea de lo que está haciendo, pero con tal de que entienda y sepa ejecutar las instrucciones va a llegar al resultado correcto»*.

¡Pero habíamos afirmado en un párrafo anterior! *«... y todos sabemos que no podemos hacer lo que no sabemos hacer»*.

Comentario: La gracia de los algoritmos es que nos permiten obtener resultados sin entender lo que estamos haciendo, y nos permiten realizar tareas complejas sin conocer / explicar por qué hay que hacerlo así.

Soy consciente de que, llegado a este punto, en su cabeza puede haber surgido la idea de que estoy utilizando una falacia lógica en mi razonamiento. Pero la verdad es que no es así.

Sigamos con otro ejemplo, supongo que usted sabe caminar, que aprendió en su infancia y que desde entonces camina todo el día con destreza salvo que le haya ocurrido algún tipo de accidente. Ahora, escriba en una hoja cuál es el proceso anatómico detallado que usted sigue cuando anda. Verá que es completamente imposible salvo que usted sea un experto en locomoción.

Esto que usted sabe hacer y no sabe cómo se hace es un magnífico ejemplo de que usted sigue un algoritmo, sabe ejecutar cada una de las instrucciones necesarias, pero usted no entiende lo que está haciendo. Realiza una tarea compleja sin conocer o saber explicar por qué hay que hacerlo así.

Comentario: Esto ha sido así históricamente con muchas de las actividades, otro ejemplo con el que seguro que está familiarizado es el algoritmo que sigue para realizar multiplicaciones o divisiones. Hay muchos diferentes, pero salvo que sea aficionado al cálculo, me imagino que no sabría explicar por qué el algoritmo es como es. Sencillamente usted lo conoce, sabe ejecutar las instrucciones y consigue el resultado buscado si no se ha confundido en su ejecución.

Lo que ocurre en estos momentos de nuestra historia, es que los algoritmos que nos son particularmente útiles, también lo son para un interlocutor especial que no entiende lo que hace, ni es consciente de lo que está haciendo, pero sabe ejecutar las instrucciones de muchos de nuestros algoritmos, con la particularidad de que es capaz de seguir y ejecutar millones de instrucciones por segundo. Recibe el nombre de *ordenador*, aunque a mí me gusta mucho más la palabra *computador*.

Cómo se hace un algoritmo

Algoritmizar es el arte de convertir las cosas que hacemos o las que quisiéramos hacer en una secuencia de instrucciones que una persona o un computador sepa ejecutar, es decir, mejor dicho, algoritmizar es la técnica de construir algoritmos. Entendiendo como técnica el conjunto de procedimientos o recursos que se usan en una actividad determinada que se adquieren por medio de su práctica y requieren experiencia y habilidad.

Algoritmizar es una tarea muy compleja que requiere una labor inmensamente creativa y una comprensión muy profunda del problema a resolver (lo que no se conoce, mal se puede explicar a nadie, y menos a una máquina).

Los algoritmos no tienen que ver solo con tareas repetitivas, sino con procedimientos que sirvan para resolver tareas de muchos tipos. Dado un problema, cada solución algorítmica buena que encontremos, tarea que hasta ahora siempre requiere un ser humano inteligente, es una solución que se puede replicar tantas veces como sea necesario y mandarla ejecutar a máquinas específicas.

Comentario: A día de hoy, tenemos ya una inmensa acumulación de conocimiento algoritmizado que crece de forma exponencial.

Problemas: Si el algoritmo no está bien hecho porque tiene errores de cualquier tipo, normalmente pueden surgir problemas. Por desgracia, generar errores durante el proceso de algoritmizar la solución de un problema es habitual en lugar de ser una excepción. Muchos errores no pasan de producir algunas molestias o inconvenientes, pero en muchas ocasiones producen consecuencias extremadamente serias.

He aquí un ejemplo. El Mars Climate Orbiter tenía como misión fotografiar Marte durante años. Pero no llegó a enviar ni una. Y todo por un fallo como un error de conversión. El sistema de control de la nave en la Tierra usaba el sistema métrico anglosajón, mientras que el sistema de navegación de la nave esperaba valores en el sistema métrico decimal. Esto hizo que la trayectoria de la nave se acercara demasiado a Marte y acabará desintegrada por la fuerza de fricción atmosférica del planeta.

Una lista de errores dinámica y exhaustiva se puede encontrar en la siguiente dirección que hace referencia a una entrada de la Wikipedia

<https://en.wikipedia.org/wiki/List_of_software_bugs>

Qué es eso de la inteligencia artificial clásica

Desde un punto de vista computacional simple, cada investigador podría dar su propia definición de lo que se entiende por inteligencia artificial clásica, en adelante IAc. Esto es debido a que esas palabras que deberían tener un peso específico en sí mismas con una interpretación unívoca, van perdiendo potencia frente al uso indiscriminado por la política, la tecnología con sus sorprendentes aplicaciones, los medios de comunicación, los actores sociales, la publicidad y el *márketing*, y toda la industria que rodea las múltiples facetas de la generación y divulgación de mensajes.

En el contexto de este libro, y por el momento, voy a utilizar una definición dada por Raymond Kurzweil (1990) que dice: «La IAc es el arte de desarrollar máquinas con capacidad para realizar funciones que cuando son realizadas por personas requieren de inteligencia». Esta definición piensa en la IAc como aquellos sistemas artificiales que actúan como los humanos. Por ejemplo, piense en las máquinas que son capaces de jugar al ajedrez, conducir... Teniendo siempre en cuenta que, por el momento, los sistemas artificiales y los humanos somos diferentes. Para entender

mejor de qué estoy hablando voy a hacer uso de una analogía, es decir, voy a establecer una relación de semejanza entre cosas distintas.

El hombre siempre ha sentido el deseo de volar, si nos fijamos en cómo se ha resuelto este problema en la naturaleza, vemos que una posible respuesta es tener alas con plumas y batirlas a una frecuencia determinada. ¿Cómo lo hemos resuelto nosotros? Hemos dado diferentes soluciones, pero la más potente son los aviones que sí tienen alas, pero no plumas y normalmente usan motores a reacción en vez de batir sus alas.

La Comisión Europea se ha referido recientemente a la IA

<https://publications.jrc.ec.europa.eu/repository/bitstream/JRC118163/jrc118163_ai_watch_defining_artificial_intelligence_1.pdf>

como «sistemas de *software* (y posiblemente también de *hardware*) diseñados por humanos que, ante un objetivo complejo, actúan en la dimensión física o digital: percibiendo su entorno, a través de la adquisición e interpretación de datos estructurados o no estructurados, razonando sobre el conocimiento, procesando la información derivada de estos datos y decidiendo las mejores acciones para lograr el objetivo dado. Los sistemas de IA pueden usar reglas simbólicas o aprender un modelo numérico, y también pueden adaptar su comportamiento al analizar cómo el medio ambiente se ve afectado por sus acciones previas».

Y en la Estrategia Nacional de Inteligencia Artificial española (ENIA2021) se dice textualmente sobre las áreas de aplicación de la IA en la actualidad:

Además de su incidencia en actividades cotidianas (buscadores de Internet, asistentes personales, electrodomésticos, recomendaciones de comercio electrónico, la inteligencia artificial en la robotización de procesos informáticos o físicos, etc.), así como en los ámbitos de investigación multidisciplinar, la IA tiene un alto potencial de aplicación en diferentes áreas de la actividad profesional y de servicios. En el ámbito sanitario en el diseño de nuevos fármacos y reduciendo los tiempos y costes de su producción, reduciendo errores de diagnóstico, mejorando la prevención y el tratamiento personalizado de las enfermedades más frecuentes; en las industrias de materiales creando nuevos biomateriales que ofrezcan mejores aplicaciones a la ingeniería, en la productividad empresarial y de la administración, optimizando recursos y automatizando procesos, lo que permite la predicción de la demanda y la mejora de la productividad; en el ámbito financiero, mejorando la eficiencia de los sistemas de gestión de riesgos; en la educación, permitiendo la adaptación del aprendizaje a las necesidades personales; en el ámbito del transporte, la movilidad y la logística, mejorando la gestión, la eficiencia y la seguridad; o en el impacto medioambiental, permitiendo una mejor gestión de las redes energéticas y la eficiencia climática de edificios, la mitigación y adaptación al cambio climático, la predicción meteorológica y climática, entre otras aplicaciones.

La IA tiene, por tanto, un gran potencial de transformación desde el punto de vista tecnológico, económico, ambiental y social dada su penetración intersectorial, elevado impacto, rápido crecimiento y contribución a la mejora de la competitividad.

Qué es un algoritmo inteligente

Sigamos con definiciones simples. Un **algoritmo inteligente** es un conjunto prescrito de instrucciones, bien definidas, ordenadas de la manera adecuada, y finitas. Que, si se ejecutan, convierten los **datos** de un **problema** en una **solución**, es decir, recordemos que los algoritmos resuelven problemas. Dicho algoritmo debe ser ejecutado por un sistema artificial capaz de seguir las instrucciones. Y se comporta de tal manera que esa tarea de ser realizada por humanos, se diría que se requiere inteligencia.

Con objeto de ayudarle a entender lo que estoy diciendo, piense en el acto de la conducción. Para conducir un coche hace falta ser inteligente, aunque a veces la experiencia diaria ponga en duda esta afirmación 😊. La prueba es que no hay ningún ser vivo animal en la tierra capaz de conducir salvo los seres humanos. Por lo tanto, para conducir un coche hace falta ser inteligente. En estos momentos se sabe que hay numerosos prototipos, incluso coches autónomos a la venta. La definición que he utilizado de algoritmo inteligente me permite afirmar que, si ese coche es capaz de funcionar solo, controlado por un complejo sistema de IA, ese coche es tan inteligente para llevar a cabo esa tarea como un ser humano.

La paradoja es que a pesar de que lo que lo ejecuta es un sistema artificial y no es consciente de lo que está haciendo, como sabe ejecutar las instrucciones, podrá llegar siempre al resultado correcto que es conducir de manera autónoma.

¿Hasta dónde pueden llegar?

Basta con seguir las noticias de los periódicos, revistas de divulgación, documentales de televisión de ciencia y tecnología... La conclusión es que el ser humano ha empezado y sus aplicaciones se están propagando por todos los sitios.

Pregunta, ¿pero estamos en la sexta caída?

Para poder responder a esa pregunta, hay que hacerla de otra manera mejor, cambiémosla inicialmente por esta otra ¿todo es algoritmizable?

¡Fin de la primera parte! *To be continued*

Segunda parte. Hagamos bien la pregunta

Esta parte está formada por los siguientes subapartados:

- ¿Todo es algoritmizable?
- Ejemplos de problemas que no admiten solución algorítmica.
- Ejemplos de problemas que admiten solución algorítmica y son intratables.
- Ejemplos de problemas que admiten solución algorítmica y son tratables.
- La pregunta bien hecha.

¿Todo es algoritmizable?

Para encontrar la respuesta a esta pregunta, recordemos que existe una rama teórica del conocimiento humano que está relacionada con el estudio de los problemas, los algoritmos y su complejidad. Esa rama se denomina *Teoría de la Complejidad Computacional*. Esta estudia las necesidades de almacenamiento de información, de tiempo y de otros posibles recursos que son necesarios para resolver problemas, como velocidad de cálculo del dispositivo utilizado... Siendo los dos primeros los más importantes con diferencia. Cuando se ignoran el resto de otros posibles recursos, se dice que se utiliza un criterio de análisis asintótico.

Por otro lado, esos parámetros es evidente que van a depender del tamaño del problema, es decir, de los datos de entrada que se van a utilizar para resolver el problema. Por ejemplo: no es lo mismo ordenar los 1000 números naturales desordenados, que ordenar 100 000 000 000 números naturales. Esa variable que nos va a caracterizar el tamaño del problema, lo denotaremos con la variable «n». En el primer caso n es 1000 y en el segundo n es 100 000 000 000.

Para calcular el recurso en tiempo que requiere un algoritmo, se realiza una aproximación al número de pasos de ejecución que un algoritmo emplea para resolver un problema. Luego el tiempo se obtiene en función de la rapidez del dispositivo de cálculo. Ese tiempo lo vamos a representar con una función denominada $T(n)$, que nos va a permitir calcular el tiempo asintótico de un algoritmo determinado con una entrada de tamaño «n».

Por ejemplo, el tiempo requerido por un algoritmo determinado con un problema de tamaño «n» podría ser $T(n) = 9n^2 + 12n + 36$, eso quiere decir que si $n = 10$,

$$T(10) = 9 \times 10 \times 10 + 12 \times 10 + 36 = 1056 \text{ unidades de tiempo.}$$

Para calcular el recurso en necesidad de almacenamiento, se realiza una aproximación a la cantidad de memoria utilizada para resolver el problema. Esa cantidad la vamos a representar con una función denominada $M(n)$, que nos va a permitir calcular la cantidad de almacenamiento asintótica de un algoritmo determinado con una entrada de tamaño (n).

Por ejemplo, la cantidad de memoria requerida por un algoritmo determinado con un problema de tamaño «n» podría ser $M(n) = 2^n + n^2$ eso quiere decir que si $n = 10$,

$$M(10) = 2^{10} + 10^2 = (2 \times 2 \times 2) + 100 = 1124 \text{ unidades de memoria.}$$

Muchas veces para caracterizar o comparar la complejidad de los algoritmos, no nos preocupan los coeficientes de las fórmulas, ni las constantes, ni los términos más pequeños, ya que no influyen para categorizarlos. A eso se llama trabajar con los límites superiores de esas funciones, y en ese caso se utiliza lo que se denomina «gran notación O».

Por ejemplo, en los dos ejemplos anteriores, si se utilizase esa notación tendríamos

$$T(n) = 9n^2 + 12n + 36 \text{ pasará a ser representada como } O(n^2)$$

$$M(n) = 2^n + n^2 \text{ pasará a ser representada como } O(2^n)$$

Para tener una idea mejor de lo que estamos diciendo, observe la siguiente tabla en la que se representa el comportamiento real de diferentes algoritmos utilizando la gran notación O, dependiendo del tamaño del problema.

		n = 10	n = 100	n = 1000
Algoritmo 1	$O(5 \times n)$	50 Valor con 2 dígitos	500 Valor con 3 dígitos	5000 Valor con 4 dígitos
Algoritmo 2	$O(n \times \log_2 n)$	33 Valor con 2 dígitos	665 Valor con 3 dígitos	9966 Valor con 4 dígitos
Algoritmo 3	n^2	100 Valor con 3 dígitos	10 000 Valor con 4 dígitos	1 000 000 Valor con 7 dígitos
Algoritmo 4	n^3	1.000 Valor con 4 dígitos	1 000 000 Valor con 7 dígitos	1 000 000 000 Valor con 10 dígitos

Algoritmo 5	2^n	Valor con 4 dígitos	Valor con 31 dígitos	Valor con 302 dígitos
Algoritmo 6	$n!$	Valor con 7 dígitos	Valor con 161 dígitos	inconmensurable
Algoritmo 7	n^n	Valor con 11 dígitos	Valor con 201 dígitos	inconmensurable

Para que se haga una idea de qué estamos hablando, el número de protones que se estima que hay en el universo es un valor de 79 dígitos, el número de microsegundos transcurridos desde el BIG Bang es de 24 dígitos, y el latido del procesador Intel edición especial Core i9-9900KS es de 5GHz, es decir, un valor de 5 000 000 000 de pulsaciones por segundo, un valor de 10 dígitos.

Viendo esa tabla, es meridiano que alguno de los algoritmos indicados, para los tamaños de problema indicados (que en todos los casos pueden considerarse enanos), requieren más tiempo o espacio de almacenamiento del que podemos llegar a alcanzar nunca, por increíblemente rápidas que los dispositivos de cálculo del futuro puedan llegar a ser.

Los nombres que reciben los diferentes comportamientos mostrados son:

Algoritmo 1	$O(5xn)$	lineal
Algoritmo 2	$O(nx\log_2n)$	n-logarítmico
Algoritmo 3	n^2	polinomial cuadrático
Algoritmo 4	n^3	polinomial cúbico
Algoritmo 5	2^n	exponencial
Algoritmo 6	$n!$	factorial
Algoritmo 7	n^n	potencial-exponencial

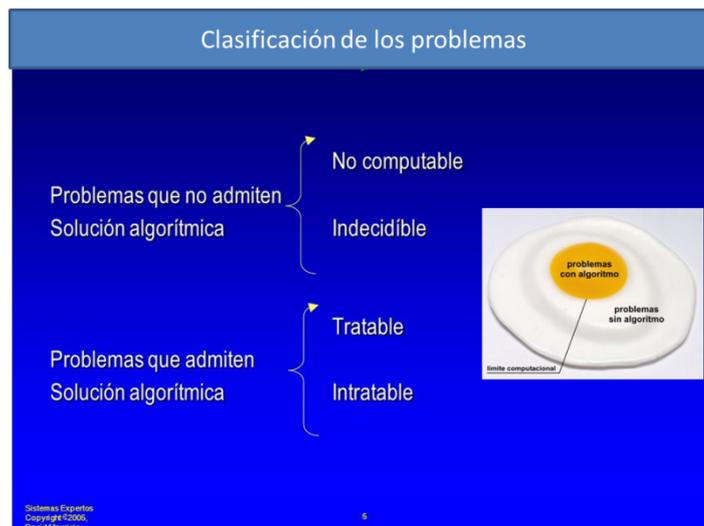
Ese análisis que usted mismo ha podido hacer ya permite clasificar los problemas en dos categorías. Los ejemplos **tratables** (en azul), los ejemplos **intratables** (en rojo), teniendo en cuenta que dentro de esta tratabilidad y de esta intratabilidad es posible distinguir varios grados. La evidencia empírica es que si un problema admite una solución lineal, n-logarítmica o polinomial, entonces se trata de un problema **tratable**; en caso contrario se le considera **intratable**.

A pesar de todo, el límite de tratabilidad entre algoritmos polinomiales y superpolinomiales es arbitrario, pero mayoritariamente aceptado. Por ejemplo, $(1,00001^n)$ o $(n^{\log_2 n})$ serían intratables, mientras que (n) o $(100\ 000; 000\ 000xn)$ serían tratables.

También se debe tener en cuenta que lo dicho se aplica directamente al tiempo de cálculo, pero en el caso de la cantidad de información necesaria, se tiene que tener más cuidado, porque ya un espacio polinomial muchas veces es intratable.

A todo lo anterior hay que decir que hemos hablado de problemas que tienen un algoritmo que lo resuelve, sea este de la categoría tratable o de categoría intratable. Pero existen problemas que no poseen un algoritmo que los pueda resolver, sin importar que tengan o no solución. No es que no se conozca aún ese algoritmo, sino que no existe. A este tipo de problemas se les llama *no computables* o *indecidibles*.

Resumiendo, la Teoría de la Complejidad Computacional nos clasifica los problemas de acuerdo con su dificultad intrínseca como se indica en la siguiente figura.



Véase <<https://geeks.ms/etomas/2019/02/02/que-significa-un-problema-p-o-np/>>

Es la explicación más clara que he encontrado hasta el momento sobre el problema de la decibilidad, doy su referencia y copio el texto que direcciona el *link* indicado. Mi más cordial enhorabuena al autor <<https://geeks.ms/etomas/sobre-mi/>>

Hasta principios del siglo XX se creía que todo problema matemático se podía resolver siguiendo una serie de pasos determinados. No se cuestionaba la dificultad de encontrar dichos pasos, pero su existencia se daba por supuesta fuese cual fuese el problema.

Fue Leibniz quien planteó la cuestión por primera vez (allá por el siglo XVII):

¿Es posible encontrar una manera de decidir
si un problema matemático cualquiera tiene solución?

No se trata de encontrar un método que nos permita encontrar la solución a un problema matemático. Se trata de encontrar un método que nos permita comprobar solo si un problema matemático tiene solución o bien es un problema indecible. Es decir, una afirmación X que hay que determinar si es cierta o falsa o bien que hay afirmaciones que son indemostrables.

No fue hasta bien entrado el siglo XX cuando David Hilbert formalizó ese problema y planteó tres cuestiones fundamentales:

¿Son las matemáticas completas? Es decir, ¿se puede probar que cualquier sentencia (matemática) es cierta o falsa?

¿Son las matemáticas consistentes? Es decir, ¿no se puede probar como cierto algo falso?

¿Son las matemáticas decidibles (o computables)? Es decir, ¿se puede probar que cualquier sentencia (matemática) es cierta o falsa siguiendo un número finito de pasos?

Cuando Hilbert formalizó las tres preguntas su intuición era que la respuesta a las tres era Sí. Es decir, las matemáticas, el lenguaje universal, eran completas, consistentes y decidibles.

Poco duró esa esperanza, porque las dos primeras suposiciones las desmontó Gödel, cuando formuló su teorema de incompletitud donde venía a decir que las matemáticas no podían ser a la vez consistentes y completas. Fue un doloroso mazazo al orgullo de las matemáticas, pero al menos todavía quedaba el punto 3. Es decir, antes de Gödel se esperaba que se pudiera demostrar que cualquier cosa era cierta o falsa siguiendo un algoritmo con un número finito de pasos. Después de Gödel lo que se esperaba era que en todas aquellas sentencias que se podían demostrar ciertas o falsas, se pudieran hacerlo siguiendo un algoritmo finito.

Los algoritmos deterministas son con los que normalmente trabajamos: les das una entrada, realizan ciertas cosas y devuelven un resultado.

Pero existen algoritmos no deterministas, que pueden devolver distintos resultados ante la misma entrada. Si imaginas que un algoritmo determinista es una lista de pasos que se siguen para, a partir de una entrada, llegar a una salida, un algoritmo no determinista sería como un árbol: a partir de una entrada se puede llegar a varias salidas distintas, y no puede saberse *a priori* cuál de esas salidas se devolverá.

Para los propósitos de este documento solo nos interesa saber que es posible convertir cualquier algoritmo no determinista en uno equivalente determinista, aunque la complejidad temporal no tiene que ser equivalente.

Por último, Turing se cargó la tercera suposición de Hilbert. La demostración de la indecidibilidad del problema de la parada. Un problema indecidible es aquel para el cual no podemos construir un algoritmo (finito) que nos lleve a un resultado. Pero ojo, no podemos no porque no sepamos, sino porque no existe. A día de hoy sabemos que hay infinitos problemas indecidibles.

Resumiendo:

- Los problemas indecidibles son los que no poseen un algoritmo. No es que no se conozca aun, sino que no existe.
- Los problemas decidibles son los que poseen algoritmos, aunque no los conozcamos. Estos a su vez se pueden distinguir en tres tipos:

- Problemas tratables (Clase P), son de orden polinómico o inferior.
- Problemas intratables (Clase NP), son de orden exponencial o superior.
- Clase NP-completa (NP-duro), no se conoce solución polinómica. Son los problemas más difíciles de NP y muy probablemente no formen parte de la clase P.

Como he dicho antes P y NP son dos grupos (formalmente dos clases de complejidad) que agrupan problemas decidibles distintos. Es decir, tanto para cualquier problema P como para cualquier NP hay un algoritmo que lo resuelve en un tiempo finito.

Ante un problema podemos hacer dos cosas: calcular los resultados o verificarlos. P. ej. ante el problema «dado x obtener su cuadrado», podemos hacer dos cosas: crear un algoritmo que encuentre el cuadrado de cualquier « x » o bien hacer un algoritmo que verifique si un « y » determinado es el cuadrado de un « x ».

Hay problemas que son fáciles de solucionar y de verificar. En este contexto «fáciles» significa mediante un algoritmo de orden polinomial. A esos problemas los llamamos problemas de tipo P. Los de orden polinómico los consideramos «sencillos» en el sentido de que los podemos computar y el tiempo no nos explota literalmente de las manos cuando los tamaños de datos aumentan.

Por ejemplo, calcular el cuadrado de un número es un problema de tipo P (hay un algoritmo en tiempo polinomial para hacerlo: multiplicar el número por sí mismo). La verificación también sencilla. ¿Es « y » el cuadrado de « x »? Nos basta con hacer la raíz cuadrada de « y » verificando que su valor es « x ».

Por lo tanto: aquellos problemas que podemos solucionar y verificar con un algoritmo de orden polinómico, los llamamos problemas P. Esos son los buenos, los que podemos solucionar «fácilmente».

Por otra parte, tenemos el grupo de problemas NP. Este grupo está formado por aquellos problemas que podemos verificar en tiempo polinomial pero que para solucionarlos puede ser que requiramos un algoritmo de orden superior.

Un ejemplo informal es un *puzzle*: difícil de solucionar, pero basta un vistazo para saber si el *puzzle* está bien resuelto o no. A esos tipos de problemas los llamamos NP. Observe que no queremos tratar con ese tipo de problemas.

Además: todos los problemas P son a su vez problemas NP. Es decir, P es un subconjunto de NP. ¿Que por qué? Muy sencillo: recuerda que un problema P es aquel que tiene un algoritmo de orden polinómico que lo resuelve. Mientras que un problema NP tiene un algoritmo de orden polinómico que lo verifica. Pues bien, entonces dado un problema de tipo P puedo verificarlo... corriendo el algoritmo que lo soluciona y

comprobando la solución. Por lo tanto, un problema de tipo P es, por definición, un NP.

Ahora la duda de si P es igual a NP ya está más clara, ¿verdad? Si resulta que eso es así, cosa que nadie ha demostrado, eso significa que cualquier problema que se pueda verificar en tiempo polinomial se puede resolver también en tiempo polinomial (otra cosa es que descubramos el algoritmo que lo hace, pero se sabrá que existe). Por otro lado, si P no es igual a NP, cosa que nadie ha demostrado tampoco, eso implicará que hay problemas para los cuales nunca existirá una solución en tiempo polinómico. Eso, aunque parece una desgracia, tampoco cambiaría gran cosa: actualmente estamos dando por sentado que P es distinto de NP, por ej., la criptografía se basa, precisamente, en eso.

Como digo la creencia «popular» hoy en día es que P es distinto de NP. Eso implica que hay problemas irresolubles en un tiempo polinómico, aunque ¡ojo! puede ser que haya problemas que pensemos que sean NP y sean P (solo que aún no hayamos descubierto el algoritmo).

Vale, ahora que tenemos claro lo que son los problemas P y los NP, ha llegado el momento de introducir otro tipo de problemas: los temidos NP-Completo.

En la década de los setenta se soltó una bomba termonuclear cuyos efectos todavía estamos padeciendo: se descubrieron un «tipo» nuevo de problemas, los llamados NP-Completo.

La primera parte de la bomba es que se demostró que cualquier problema NP se puede reducir a otro tipo de problemas NP. Esos otros problemas NP son los que llamamos NP-Completo. Decimos que un problema X reduce a X2 si X2 es más complejo de resolver que X, de forma que el algoritmo (o parte de él) usado para resolver X2 también resuelve X.

Por lo tanto, y esa es la clave, un problema C es NP-Completo si cualquier problema NP se reduce a él, lo que significa que el algoritmo usado para resolver C puede resolver cualquier NP. Por lo tanto, si encontramos un algoritmo para resolver C en un tiempo polinomial este mismo algoritmo (o parte de él) se podrá usar para resolver en tiempo polinomial cualquier NP.

Pero... la segunda parte de la bomba: también se demostró que todos los problemas NP-Completo eran equivalentes entre sí. Eso significa que si un solo problema NP-Completo tiene una solución polinómica entonces por definición todos la tienen. Y viceversa: si se demuestra que tan solo un problema NP-Completo no tiene solución en tiempo polinómico, entonces ninguno la tiene.

Veamos un ejemplo clásico de un problema NP-Completo: el problema del viajante. Dicho problema nos dice que debemos obtener el camino más corto que pase una vez

por cada una de las N ciudades y que vuelva al punto de origen. Sabes la distancia entre todas las ciudades y puedes ir de cualquier ciudad a cualquier otra en cualquier orden.

Hoy en día se cree que la manera de solucionar dicho problema es calculando todas las posibilidades una a una, lo que nos deja un algoritmo de orden exponencial (concretamente orden factorial). Si tu algoritmo para 20 ciudades tarda 1 segundo en tu ordenador, en el mismo ordenador dicho algoritmo tardará 21 segundos para 21 ciudades (solo una más). Agregar otra ciudad hará que tarde más de 7 minutos y si pasas de 20 a 30 ciudades, lo que antes tardaba 1 segundo ahora tardará unos 3 millones de años.

Otro ejemplo clásico: el problema de la suma de subconjuntos. Consiste en, dado un conjunto de enteros, determinar si existe algún subconjunto suyo tal que la suma de sus elementos sea cero. Este problema es claramente NP (fácil de verificar si una solución x es cierta o no [basta con comprobar que x es un subconjunto del conjunto original y que sus elementos suman 0], pero difícil de solucionar). Pues se demostró que ese problema es también NP-Completo.

¿Por qué son tan importantes los problemas NP-Completo? Pues bien, precisamente porque cualquier NP se puede reducir a ellos y también porque son equivalentes entre sí. Por lo tanto, si se demuestra que un NP-Completo tiene solución polinómica, entonces todos los NP-Completo la tienen. Y dado que todos los NP se pueden reducir a ellos, eso significa que todos los NP tienen solución polinómica. O, dicho de otro modo: $P = NP$. Y lo mismo es cierto a la inversa: si se demuestra que un problema NP-Completo no tiene solución polinómica (ojo, debe demostrarse, no basta con no encontrarla) entonces eso significa que P es distinto de NP.

A día de hoy de todos los problemas que se sabe son NP-Completo no se ha encontrado la solución polinómica de ninguno de ellos, aunque no se ha demostrado formalmente que no la posean.

Ejemplos de problemas que no admiten solución algorítmica

Si tienen curiosidad podrán encontrar ejemplos de problemas indecidibles en el siguiente *link* a la Wikipedia:

[<https://es.wikipedia.org/wiki/Problema_indecidible>](https://es.wikipedia.org/wiki/Problema_indecidible)

A modo de curiosidad les presento un problema de este tipo que es fácil de entender. Se trata del problema de los «números pedrisco», también conocido como la *conjetura de Collatz*, en la cual se propone un algoritmo que básicamente trabaja del modo siguiente:

- Pida que alguien que le dé un número n entero natural cualquiera.
- Si n es un número par divídalo por 2 hasta llegar a un número impar o 1.
- Si n es un número impar diferente de 1, o ha alcanzado en el paso anterior un número impar, multiplíquelo por 3 y súmele 1 e inicie otra vez el proceso de dividirlo por 2.
- Continúe este procedimiento hasta alcanzar el valor uno.

Ejemplo 1:

- Empecemos con el número 10
- $10/2 = 5$
- $5 \times 3 + 1 = 16$
- $16/2 = 8$
- $8/2 = 4$
- $4/2 = 2$
- $2/2 = 1$

Ejemplo 2:

- Una secuencia algo más larga, pero que también termina en 1 se obtiene con $n = 27$, un número ciertamente pequeño, que genera una secuencia considerablemente grande: 111 pasos
- 27, 82, 41, 124, 62, 31, 94, 47, 142, 71, 214, 107, 322, 161, 484, 242, 121, 364, 182, 91, 274, 137, 412, 206, 103, 310, 155, 466, 233, 700, 350, 175, 526, 263, 790, 395, 1186, 593, 1780, 890, 445, 1336, 668, 334, 167, 502, 251, 754, 377, 1132, 566, 283, 850, 425, 1276, 638, 319, 958, 479, 1438, 719, 2158, 1079, 3238, 1619, 4858, 2429, 7288, 3644, 1822, 911, 2734, 1367, 4102, 2051, 6154, 3077, 9232, 4616, 2308, 1154, 577, 1732, 866, 433, 1300, 650, 325, 976, 488, 244, 122, 61, 184, 92, 46, 23, 70, 35, 106, 53, 160, 80, 40, 20, 10, 5, 16, 8, 4, 2, 1

Ejemplo 3:

- Supercomputadoras lo han comprobado con los números que van más o menos hasta

$5_3764\ 607_2500\ 000_1000\ 000$

El problema que se plantea y por el que se pregunta si existe un algoritmo para responder es el siguiente, dado un número de partida cualquiera, en cuántos pasos se alcanza el valor 1.

La solución es que no es posible determinar lógicamente y en general cuántos pasos se necesitan para alcanzar el valor 1 sin aplicar el algoritmo de partida en cada caso particular.

Ejemplos de problemas que admiten solución algorítmica y son intratables

Dada una lista de ciudades y las distancias por carretera entre cada par de ellas, la pregunta es, ¿cuál es la ruta más corta posible que consigue visitar cada ciudad exactamente una vez y al finalizar regresa a la ciudad de origen? Este problema se planteó por primera vez en 1930 y es uno de los problemas de optimización más estudiados. Es probable que en el caso peor el tiempo de ejecución para cualquier algoritmo que lo resuelva aumente de forma exponencial con respecto al número de ciudades.

Dado un problema con «n» ciudades, la complejidad es $T(n) = (n-1)! / 2$, es decir

$$T(n) = 0,5 \times [(n-1) \times (n-2) \times (n-3) \times \dots \times 3 \times 2 \times 1]$$

n = 5	12 rutas que examinar
n = 10	181 440 rutas que examinar
n = 30	4×10^{30} rutas que examinar

En el caso $n = 30$, si se hubiera comenzado a calcular al comienzo de la creación del universo, hace 13 400 millones de años, todavía no se habría acabado el cálculo.

Usted se preguntará que ese problema debe estar resuelto, ya que hay flotas de camiones o autobuses o aviones que aparentemente lo realizan.

Lo que le puedo decir es que:

- Si n es pequeño, se puede encontrar la solución óptima.
- Si n aumenta, existen procedimientos que obtienen soluciones diferentes, pero no aseguran que las soluciones ofrecidas son óptimas. A este tipo de algoritmos se les denomina *algorítmicos heurísticos*, ya que pueden producir una solución buena e incluso óptima, pero también pueden producir una mala solución o incluso ninguna. Es decir, su funcionamiento no está asegurado.
- Puede ocurrir que existan configuraciones geométricas especiales para las que se puede encontrar una solución óptima.

Para conocer mejor este problema y las soluciones obtenidas puede consultar el siguiente *link* a la Wikipedia:

<https://es.wikipedia.org/wiki/Problema_del_viajante>

Otros problemas famosos pueden encontrarse en los siguientes *links* a la Wikipedia:

<https://es.wikipedia.org/wiki/Torres_de_Hanói>

<https://es.wikipedia.org/wiki/Problema_de_la_mochila>

<https://es.wikipedia.org/wiki/Lista_de_21_problemas_NP-completos_de_Karp>

Ejemplos de problemas que admiten solución algorítmica y son tratables

Se puede afirmar que la mayoría de los problemas que nos rodean son tratables: o sea, tienen solución para tamaños grandes de dichos problemas...

La pregunta bien hecha

Habíamos dicho que para poder responder a la pregunta ¿Estamos en la sexta caída?, había que hacerla bien, e inicialmente la habíamos cambiado por esta otra pregunta que estaba mejor hecha ¿todo es algoritmizable?

Y ahora vamos a poder plantear la pregunta todavía de manera más correcta. Creo que coincidirá conmigo en esta afirmación. El ser humano ha demostrado que es un sistema que tiene inteligencia y la manifiesta a la hora de resolver problemas. Por lo tanto, la pregunta debería ser *¿El ser humano es algoritmizable?*

La respuesta la desconocemos, hay científicos, tecnólogos que desean que la respuesta sea no y los hay que desean que la respuesta sea sí.

Comentario: La razón para que exista esta dicotomía es que cuando se habla de inteligencia nadie sabe exactamente de qué estamos hablando, lo que sí es cierto es que cuanto más avanzan los desarrollos de la IAc parece ser que lo que todavía no sabe hacer la máquina y sí el ser humano, eso es inteligencia, pero a ojos vista, esa distancia parece que va reduciéndose paulatinamente.

¡Fin de la segunda parte! *To be continued*

Conclusiones

Concluir es terminar un razonamiento, una argumentación, una discusión, una prueba en virtud de relaciones necesarias o demostradas con las proposiciones anteriores. El que concluye se apoya en principios demostrados o que por tales tiene, y cuyo enlace con la consecuencia es o parece ser necesario.

El título de este capítulo es: «Qué es un algoritmo inteligente. ¿Estamos en la sexta caída?» El objetivo ha sido intentar explicar «Qué es un algoritmo inteligente» y responder a la pregunta de si ¿Estamos en la sexta caída? El discurso se ha estructurado en las dos partes siguientes:

- Explicación de «Qué es un algoritmo inteligente».
- Hagamos bien la pregunta.

En realidad, es que hemos llegado a una conclusión que es que la pregunta debería ser *¿El ser humano es algoritmizable?*, cuya respuesta ha sido *que la desconocemos, hay científicos, tecnólogos que desean que la respuesta sea no y los hay que desean que la respuesta sea sí.* Por lo tanto, el título de este apartado debería ser «**In conclusiones**».

Debo aclarar que «estamos empezando» a ser capaces de concebir cómo podrían ser las posibles respuestas a ese tipo de preguntas. Partiendo de dónde estamos, podríamos decir hacia dónde vamos y cuál será el futuro. Para no entrar en argumentaciones demasiado técnicas, prefiero sugerirle una metáfora relacionada con la belleza. Para ello necesito de su colaboración. Piense en tres personas en función de su grado subjetivo de belleza, de tal manera que una de ellas sea normal, la segunda atractiva y la tercera el nivel más alto de belleza que usted pueda atribuir. Entonces la analogía consiste en:

- La respuesta a la pregunta ¿Dónde estamos? La encontrará por analogía en la persona normal.
- La respuesta a la pregunta ¿Hacia dónde vamos? La encontrará por analogía en la persona atractiva.
- ¿La respuesta a la pregunta ¿Cuál es el futuro? La encontrará por analogía en la persona más bella.

Hay esperanza en llegar a alcanzar inteligencia artificial general equivalente a la del ser humano, desde mi punto de vista la prueba somos nosotros, por lo tanto, el desafío es planteable. Pero *¿es factible encontrar la respuesta?*, una minoría piensa que es imposible, muchos creen que es posible pero no en los próximos cincuenta años, otra minoría piensa que será posible antes de veinte años.

Siempre que en mis conferencias sobre esta temática llego a este punto, me gusta recordar que, a lo largo de la historia, el ser humano ha sido incapaz de predecir con exactitud cómo será el futuro, de hecho, el patrón que se ha seguido es que «el futuro nunca es como nos lo hemos imaginado».

De cualquier modo, por el momento no se ha creado inteligencia artificial general, tal cual la puede entender a partir de esas dos palabras un universitario no especializado en estas lides. Ni tampoco estupidez artificial. Tan solo agentes con trastorno autista.

Por lo tanto, la respuesta a la pregunta inicial de si ¿Estamos en la sexta caída? Recordemos qué transformó a la especie humana en dominante. El verdadero momento de la transformación genial de la inteligencia humana no fue el pensamiento individual, fue la construcción de una inteligencia cooperadora y compartida entre humanos, buscando la supervivencia, y construyendo una red de intereses y conocimientos común.

Yo soy de ese subgrupo de personas que no se quedan con el enunciado del sí o del no de acuerdo con el deseo interno de que la respuesta sea su apetencia. La única forma de obtener la respuesta es buscándola trabajando, cualquier otra aproximación no es adecuada, ni fiable, ni racional. Y recuerden, pertenecemos a la humanidad «sapiens». Y si la respuesta fuera sí, no estaríamos en el sexto derrumbe, del mismo modo que no ha habido ningún derrumbe de los citados previamente salvo para aquellos humanos que sin saber y sin buscar se instalaron en la soberbia mental de basar sus afirmaciones en sus creencias o ansias mitómanas. De cualquier modo, a pesar de todo... y por lo contado estoy convencido de que ihay que asomarse a una de las nuevas fronteras del conocimiento y por lo menos, hay que curiosear! Recordando siempre que lo menos que podemos hacer es intentar comprender el mundo que nos rodea.

Teoría de la información

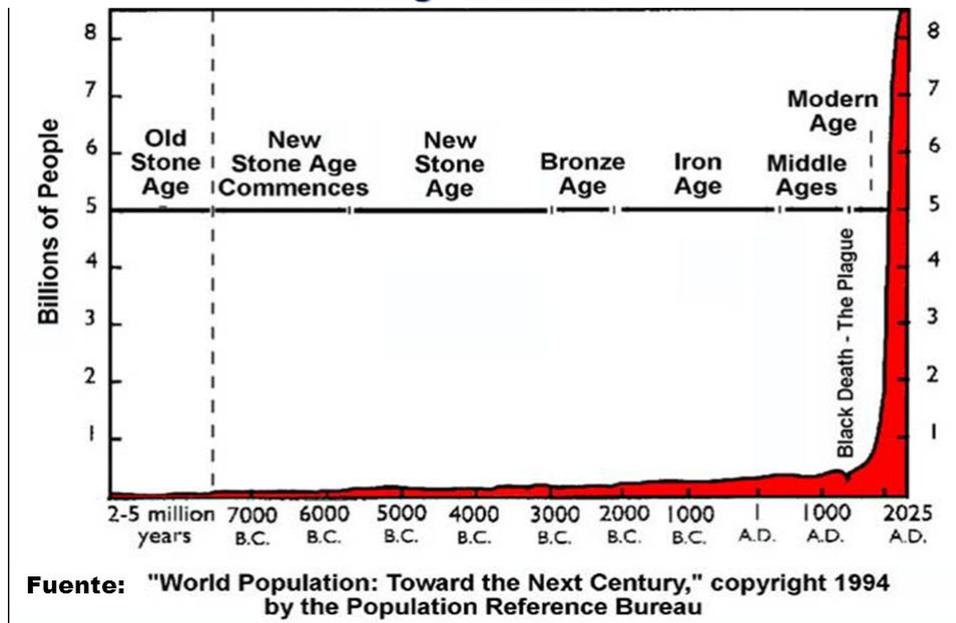
Velocidad de los avances tecnológicos

La tecnologización de cualquier actividad económica siempre ha producido y sigue produciendo una expansión espectacular en ella y en la sociedad en la que está inmersa.

Observen el crecimiento de la población a lo largo de los siglos.

Siglo	Habitantes (en millones)	Siglo	Habitantes (en millones)
0	170	XI	265
I	180	XII	320
II	180	XIII	360
III	180	XIV	360
IV	180	XV	350
V	190	XVI	425
VI	190	XVII	545
VII	200	XVIII	680
VIII	210	XIX	980
IX	220	XX	5000
X	240	XXI	7000

Crecimiento de la Población Mundial a lo largo de la historia



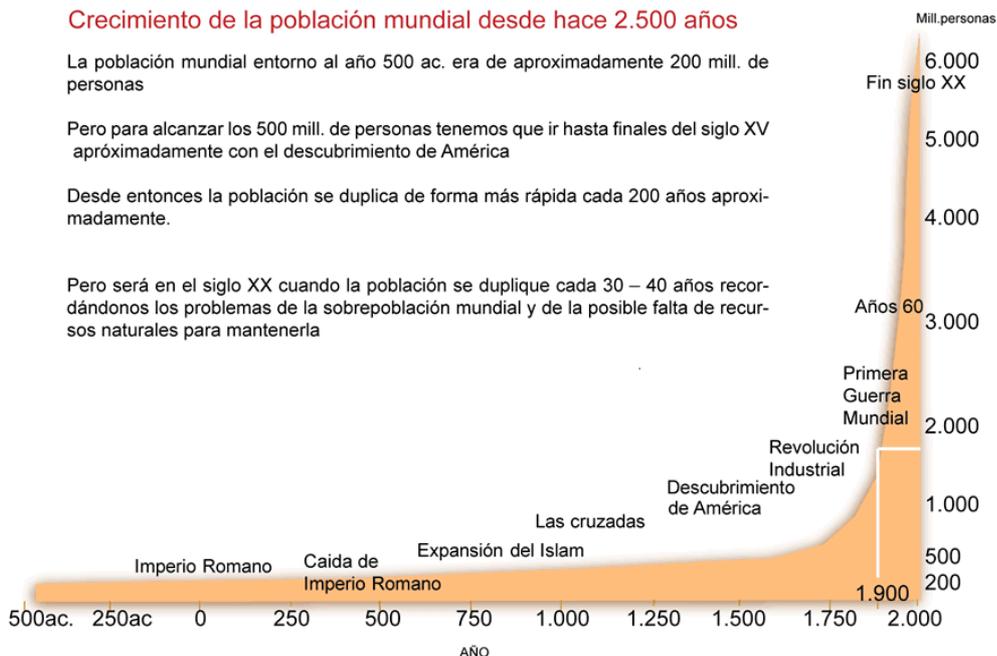
Crecimiento de la población mundial desde hace 2.500 años

La población mundial entorno al año 500 ac. era de aproximadamente 200 mill. de personas

Pero para alcanzar los 500 mill. de personas tenemos que ir hasta finales del siglo XV aproximadamente con el descubrimiento de América

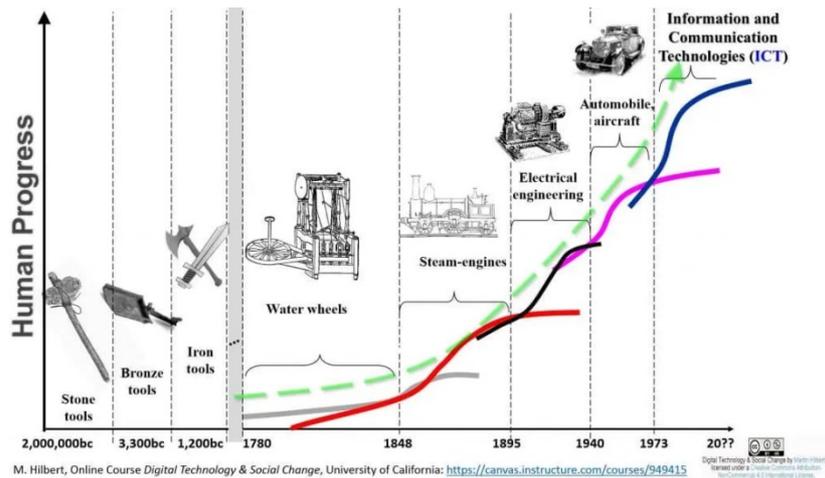
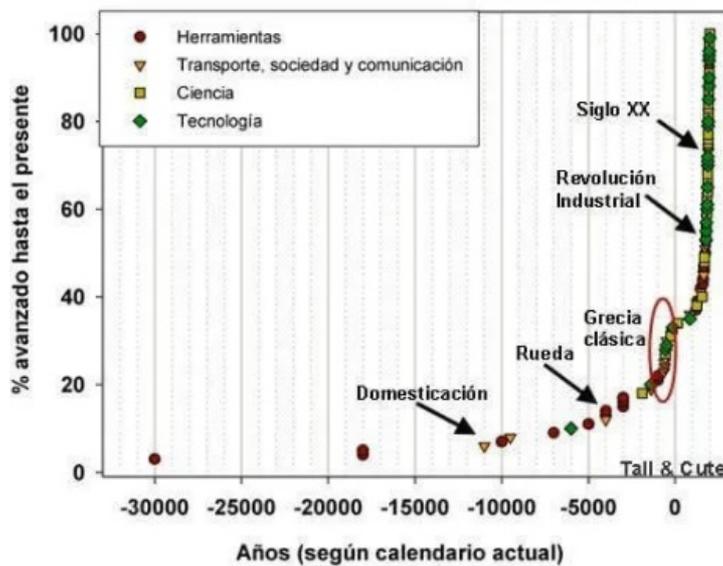
Desde entonces la población se duplica de forma más rápida cada 200 años aproximadamente.

Pero será en el siglo XX cuando la población se duplique cada 30 – 40 años recordándonos los problemas de la sobrepoblación mundial y de la posible falta de recursos naturales para mantenerla



En un análisis histórico del avance de las tecnologías desde el inicio de la civilización, se puede observar cómo a partir de finales del siglo XIX el ritmo de aparición de nuevas tecnologías con impacto transformador en la sociedad y en la economía ha sido creciente, alcanzando en la actualidad un ritmo aparentemente exponencial.

En los últimos años han aparecido numerosos estudios que reflejan el modo en que han surgido los avances científico-técnicos a lo largo de la historia de la humanidad y el modo en que han impactado y transformado a la sociedad. De dichos estudios se extrae que la evolución tecnológica se comporta de un modo exponencial, de lo que se deduce que los cambios tecnológicos, y su impacto, serán cada vez más rápidos y más disruptivos, tal y como observamos en las últimas décadas.



Véase: <<https://www.iti.es/blog/utopia-distopia-digital-consecuencias-eticas-irrupcion-ia/>>

Es inevitable la comparación entre los gráficos de evolución humana y de evolución tecnológica, es evidente que correlan.

Pero antes de seguir avanzando en esa línea, hagámonos la siguiente pregunta. ¿Se avanza a la misma velocidad en todos los campos de la actividad humana? La respuesta se la adelanto: ¡No!

Veamos algunos valores de los ritmos de los avances tecnológicos.

Área	Antigüedad	1870	1950	1970	2020
Transporte	65 km/día	325 km/día	10 000 km/día	58 500 km/día	27 750 km/h (*)
Expectativa de vida	22 años	45 años	66 años	68 años	84 años
Grandes obras	Gran Muralla	Canal de Suez	Presa de Fort-Peck	Presa de Mangla	Ampliación del canal de Panamá
Educación	Ninguna	1 año	6 años	10 años	18 años
Potencia explosiva	0,5 kg de TNT	500 kg de TNT	10 ⁵ ton. TNT	10 ⁸ ton. TNT	10 ¹¹ ton. TNT
Gasto de energía / persona	0,5 C.V.	1,6 C. V.	10 C. V.	12 C. V.	16 C. V.
Velocidad de cálculo. Operaciones por segundo	0,005	0,005	40	10 ⁷	10 ¹⁴

(*) Estación Espacial Internacional

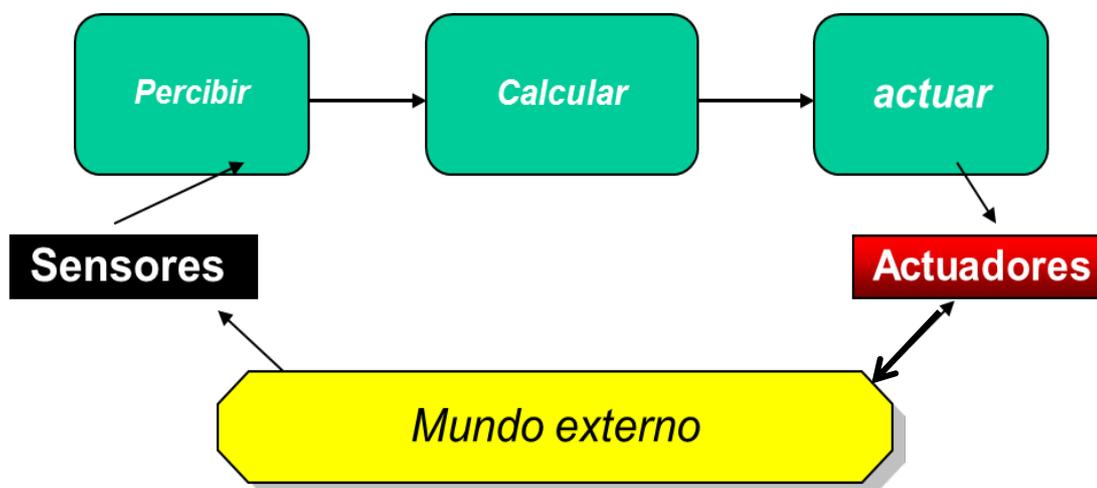
A partir de los datos de este cuadro comparativo, observamos cómo la potencia explosiva y la velocidad de cálculo han crecido a lo largo de los siglos a un ritmo muy superior que el mantenido por otros campos de la tecnología. Por ejemplo, si los medios de transporte hubieran evolucionado con la misma progresión, hubiéramos superado ya la velocidad de la luz.

El alfabeto, la gramática y la lógica

Existe una metáfora simple sobre el comportamiento de todos los seres vivos en relación con el mundo exterior en el que están inmersos.

El ser vivo recibe información del mundo exterior, la procesa de algún modo y con posterioridad interactúa con el mundo exterior.

En el caso de los seres vivos con un sistema nervioso, la metáfora se expresa de la siguiente forma. El sistema cognitivo recibe información del mundo exterior, la procesa y con posterioridad interactúa con el mundo exterior.



En el caso de los seres vivos con un sistema nervioso central, los sentidos son los receptores a través de los cuales recibimos señales externas. Estas las transforman en señales eléctricas internas que se propagan por el sistema nervioso. El sistema nervioso central las traduce en sensaciones. Estas se procesan. A partir de lo cual se actúa sobre el mundo exterior.

Según la metáfora planteada, se puede decir, por lo tanto, que todos los seres vivos son procesadores de información biológicos. Aunque hay que tener en cuenta que no todos los seres vivos tienen las mismas características sensoriales, ni el mismo sistema nervioso, ni procesan a la misma velocidad, ni extraen la misma información, ni tienen la misma capacidad cognitiva, ni actúan sobre el mundo de la misma manera.

Cuando los seres vivos actúan sobre el mundo externo, la mayoría suelen utilizar algún tipo de comunicación con el resto de los seres vivos. En el caso de nuestra especie, parece bastante lógico pensar que la capacidad de fabricación de utensilios (por toscos que fueran) por parte del *Homo hábilis* de hace unos 3 000 000 de años, sugiere que en estos ya existía un lenguaje oral articulado muy rudimentario pero lo

suficientemente eficaz como para transmitir información o enseñanza con objeto de confeccionar los artefactos.

Estudios realizados en la sierra de Atapuerca (España) evidencian que el *Homo antecesor*, de hace unos 800 000 años, ya tenía la capacidad, al menos en su aparato fonador, para emitir un lenguaje oral lo suficientemente articulado como para ser considerado simbólico.

En este momento, se hablan entre 3000 y 5000 lenguas en todo el mundo, con tendencia a la desaparición de muchas de ellas. Los expertos coinciden en que las lenguas que existen en la actualidad proceden de siete grandes familias. Además, algunos lingüistas creen que todas ellas proceden de un tronco común, una lengua madre todavía desconocida.

Los primeros humanos se transmitían el conocimiento (información) de generación en generación mediante la palabra. Cada nueva generación aprendía los conocimientos acumulados por la anterior gracias a los relatos orales que les contaban sus ancestros durante la infancia. Pero eso hacía que la información fuera evanescente y local. Los sonidos tienen un alcance de pocos metros y caen en el olvido. Lo hablado parece al instante sin dejar huella material, y si llega a pervivir, lo hace solo en las mentes de quienes lo han escuchado (Samuel Butler).

Sin embargo, con el tiempo apareció un tipo especial de palabras, sujeto a normas especiales de uso: ¡Los dibujos!

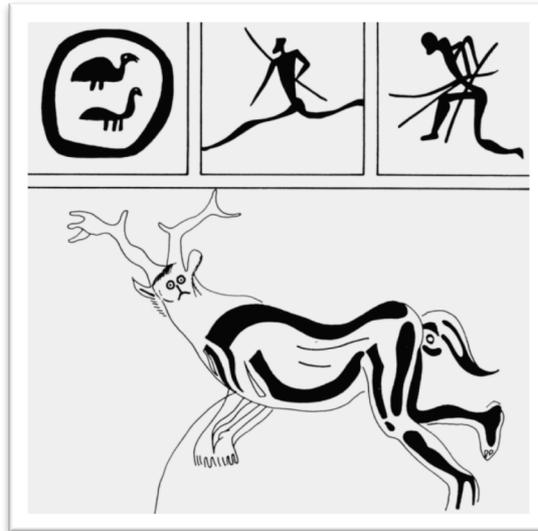
Las pinturas rupestres son las manifestaciones artísticas más antiguas de las que se tiene constancia, ya que al menos existen testimonios datados hasta los 40 000 años de antigüedad, es decir, durante la última glaciación (Paleolítico). Por aquel entonces, los habitantes de las cavernas comenzaban a almacenar otro tipo de información: hicieron dibujos o pinturas realistas de animales en las paredes de las cavernas. ¡Nadie sabe cómo y por qué!

Varios miles de años después, y de forma inevitable, las pequeñas aldeas del Neolítico situadas en zonas cultivables, crecieron convirtiéndose en ciudades donde se produjo una mayor diversificación y especialización de la actividad: más y mejores productos, materiales e inmateriales; más intercambio (de necesidades) y en ese entorno nació la escritura. La fuerza motriz que impulsó el desarrollo de la escritura fue siempre la misma: el proceso de urbanización y la aparejada exigencia de una organización administrativa capaz de coordinar eficazmente el trabajo de miles de personas.

Por ello, los seres humanos idearon un sistema para representar el lenguaje con figuras. Denominemos a eso «habla visual». El arte de escribir ha sufrido numerosos cambios a lo largo de su dilatada historia hasta alcanzar el grado de perfección que tiene en la actualidad, contribuyendo como parte esencial a nuestra vida cotidiana.

Los primeros sistemas de escritura se establecieron hace casi 5500 años. Consistían en dibujos de personas, animales y objetos que hoy conocemos como pictogramas. Posteriormente, las civilizaciones inventaron los ideogramas que, mediante símbolos, representaban ideas más complejas. Esta escritura se denomina *cuneiforme*, constaba de 2000 caracteres, de los cuales únicamente 250 se utilizaban habitualmente.

Un pictograma es un signo que representa esquemáticamente un objeto real, por ejemplo, los pictogramas rupestres.



- a) Avutardas o cisnes dentro de un círculo mágico. Castellón, España.
- b) Arquero tras la presa. África del Sur.
- c) Arquero preparándose para la caza. Castellón, España.
- d) El «hechicero disfrazado». Ariège, Francia.

Un ideograma es una representación gráfica de una idea. Los ideogramas suelen formarse por combinación de pictogramas.



Ideograma Mujer. Un triángulo pubiano con la raya de la vulva representa a la mujer.



Ideograma Esclava. Si se añade al triángulo pubiano signos que designaban montañas, se representa a "mujeres extranjeras" procedentes del otro lado de las montañas, es decir, esclavas de sexo femenino.

Antes del alfabeto actual, era necesario un símbolo especial (pictograma o ideograma) para cada palabra o para cada sílaba (logografías), de modo que para aprender a escribir se requería memorizar enormes cantidades de símbolos.

La base del sistema de escritura más rápido y eficaz que existe fue inventado hace casi tres mil años en las orillas orientales del Mediterráneo. El primer alfabeto establecía letras para todos los sonidos, menos para las vocales. Después al descomponer los elementos del lenguaje en los sonidos fundamentales, el número de símbolos quedó reducido a menos de treinta. ¡Ahora sí, cualquier persona con tiempo podía aprender a leer!

El sistema actual que consiste en formar palabras a base de combinar caracteres o letras que representan diferentes sonidos surgió por primera vez en una región que se convirtió en un cruce de culturas políticamente inestables y que comprendía Palestina, Fenicia y Asiria hará unos tres mil setecientos años. Al este estaba la gran civilización mesopotámica, con su escritura cuneiforme de mil años de antigüedad. Al sudoeste, estaba Egipto, donde se desarrollaron los jeroglíficos (pictogramas), la escritura hierática (ideográfica) y la demótica (fonética).

Es muy probable que los primeros elementos fonéticos que se expresaron en forma escrita se inventaran para describir nombres de lugares y personas relacionadas con operaciones comerciales.

El alfabeto se inventó una sola vez, en Oriente Medio, y de allí se extendió a todo el mundo por contagio. Pasó por Arabia y la India, constituyendo además la base del griego, que a su vez dio origen a las grafías latina y cirílica que aún perviven en Europa. Los hitos más importantes son hacia el 1050 a. C. la escritura fenicia (21 letras) sistema consonántico, en vigor hasta el siglo III d. C. con sentido de la escritura de derecha a izquierda. A partir del 700 a. C. la escritura griega alfabética (24 letras), con sentido de la escritura de derecha a izquierda o en bustrófedon. El acto de transcripción de la *Iliada* y de la *Odisea* (pensados para la oralidad) sentó las bases para la destrucción del sistema vía oral de pensamiento. A partir del 500

a. C. la escritura latina (24 letras) con sentido de la escritura horizontal, siendo al principio de derecha a izquierda o en bustrófedon, aunque después se afianzó de izquierda a derecha.

De alguna manera, se puede afirmar que la evolución de la tecnología del lenguaje escrito comenzó con imágenes, progresó a los pictogramas, más tarde se convirtieron en ideogramas y posteriormente se pasó a lo logográfico, es decir, primero a representar las unidades fonéticas y finalmente al alfabeto. Cada nuevo paso adelante fue sin duda un progreso hacia una comunicación más eficiente.

Hoy se conocen unas setecientas formas distintas de expresión escrita. Sus orígenes parten de cuatro troncos básicos: los jeroglíficos egipcios y las paleo escrituras mesopotámica, china y zapoteca.

La escritura permite la conservación y la transmisión del conocimiento. Lo escrito se extiende sin limitaciones, en lo que concierne a tiempo y a espacio, ampliando el ámbito en el que una mente puede comunicarse con otra. Proporciona a la mente del que escribe una existencia limitada únicamente por la duración del soporte utilizado.

En tanto alguien invente otra cosa mejor, la escritura sigue siendo el sistema humano de almacenamiento de la información. Las primeras escrituras se realizaron en materiales tan distintos como la cerámica, la madera o el yeso cristalizado, pero el más común y barato era la piedra. La madera, el bambú y los huesos de animales fueron los primeros soportes de los escritos chinos. Los egipcios fueron los primeros que comenzaron a escribir en papiros, que era un material barato y abundante. El pergamino fue el soporte normal de escritura en Occidente hasta que apareció el papel.

Por otra parte, la tarea de escribir requirió de la aparición de la profesión del ser capaz de escribir. Los escribas en el Imperio egipcio. Los escribientes, en principio monjes, de la Edad Media. El nacimiento de la imprenta que surgió en el lejano Oriente, aunque en Occidente no se utilizó hasta que el orfebre alemán Johannes Gutenberg ideó en el siglo XV la imprenta con letras intercambiables, que dio un nuevo sentido a la escritura. La aparición de la imprenta ocasionó la proliferación de maestros calígrafos que enseñaban a escribir de manera elegante y artística.

Todo ese avance en la generación de información produjo la aparición de un nuevo problema. Una vez producida y almacenada, ¿cómo se podía encontrar cuando se necesitaba?

En China lo más parecido a un diccionario fue durante muchos años el Erya (siglo III a. C.). Ordenaba sus dos mil entradas por significados, por categorías temáticas: monarquía, edificios, instrumentos y armas, los cielos, la tierra, las plantas y animales. En Egipto se compilaron listas de palabras organizadas según principios filosóficos o didácticos. Lo mismo ocurría en el mundo árabe. El procedimiento no

era ordenar palabras, sino el mundo. Pero quedó demostrado que no es un procedimiento adecuado.

En algún momento a alguien se le ocurrió inventarse el orden alfabético. Y se hizo la luz. Semejante sistema no es natural. Obliga al usuario a disociar la información del significado; a tratar las palabras estrictamente como cadenas de caracteres; a centrarse de manera abstracta en la configuración de las palabras.

El orden alfabético supone un par de procedimientos, inversos entre sí: organizar (clasificar) y buscar (examinar). Ambos procedimientos son recursivos y la operación básica es una decisión binaria ("> que" o "< que").

En el mundo antiguo no aparecieron listas alfabéticas hasta aproximadamente el 250 a. C. en textos papiráceos alejandrinos. En 1587, Thomas Thomas presentó el *Dictionary* latino-inglés para traducir entre ambas lenguas. En 1604, Robert Cawdrey, un maestro de pueblo y cura no conformista, escribió un libro con el título *Tabla alfabética que contiene y enseña la verdadera escritura, y explica los vocablos ingleses arduos, pero usuales* (2000 entradas). Hasta 1613 no se elaboró el primer catálogo alfabético a mano para la Biblioteca Bodleriana de Oxford...

Quedó demostrado con el paso del tiempo que las listas temáticas eran irritantes, imperfectas y creativas. Mientras que las listas alfabéticas eran mecánicas, eficaces y automáticas.

Otro aspecto importante asociado al lenguaje, ya sea oral o escrito es la necesidad de la gramática. Esta es el estudio de las reglas y de los principios que regulan el uso de las lenguas y la organización de las palabras dentro de una oración. También se denomina así al conjunto de reglas y principios que gobiernan el uso de un lenguaje muy determinado; así, cada lengua tiene su propia gramática.

La teoría gramatical ha evolucionado a través del uso y la división de las poblaciones humanas y las reglas sobre el uso del lenguaje tendieron a aparecer con el advenimiento de la escritura. La gramática más antigua que se conoce es el *Astadhiaia*, un estudio sobre el sánscrito, escrito por Pánini, en la India, hacia el año 480 a. C. Sócrates, Aristóteles y otros sabios de la antigüedad disertaron sobre gramática... Por otra parte, la *Ars Grammatica* de Elio Donato (siglo IV) dominó los estudios gramaticales durante la Edad Media.

Resumiendo: La persistencia de la escritura permitió poner una estructura a lo que se sabía del mundo y, a continuación, a lo que se sabía del saber. Cuando se pudo escribir palabras, examinarlas, volver a examinar esas mismas palabras al día siguiente y considerar su significado, aparecieron los filósofos, y los filósofos comenzaron a emprender a partir de cero un vasto proyecto de definición. El conocimiento triunfó por sí mismo (Eric Havelock).

El proceso de convertir la «prosa narrativa oral» en una «prosa escrita, o prosa de ideas», es decir, de abrazar la disciplina de la abstracción, Havelock lo denominó «pensar». Era el descubrimiento, no solo del yo, sino del yo pensante, en efecto, el verdadero comienzo de la conciencia.

En nuestro mundo arraigado en el conocimiento de las letras, pensar y escribir parecen actividades apenas relacionadas entre sí. Incluso podemos imaginar que la segunda depende de la primera, pero no al revés. Pero la palabra escrita (la palabra persistente) es un requisito del pensamiento consciente como lo entendemos actualmente. Fue el detonador de un cambio irreversible y radical en la psique humana.

Platón confirmó y afianzó las conjeturas de una generación anterior que había tenido la sensación de dirigirse hacia la idea de que podía «pensar», y que «pensar» era un tipo de actividad psíquica muy especial, muy incómoda, pero también muy apasionante; una actividad que requería la escritura.

El siguiente paso por el camino de la abstracción aparece con Aristóteles que utilizó categorías y relaciones en un orden sistematizado para desarrollar un simbolismo del razonamiento: «la lógica».

Tal vez imaginemos que la lógica existía independientemente de la palabra escrita, ya que los silogismos pueden expresarse tanto oralmente como por escrito, pero no es así. El lenguaje hablado es demasiado efímero para poder someterlo a análisis. La lógica deriva de la palabra escrita, tanto en Grecia como en la India o en China, donde se desarrolló de manera independiente.

Sabemos que la lógica formal fue creación de la cultura griega después de haber asimilado la tecnología de la escritura alfabética (Walter Ong). Ocurrió de forma análoga en la India y en China.

La lógica implica directamente el simbolismo: las cosas forman parte de unas clases; poseen cualidades, que son abstractas y generales. Los individuos de una cultura oral carecen de las categorías que son habituales para las personas que viven en medio de una cultura alfabetizada.

No deja de ser un viaje tortuoso pasar de cosas a palabras, de palabras a categorías, y de categorías a la metáfora y a la lógica.

Lenguaje y razonamiento encajan tan bien que los usuarios no siempre apreciamos sus defectos y lagunas. No obstante, en cuanto una civilización inventaba una lógica, surgían las paradojas, lo ilógico del lenguaje natural, las normas de la ortografía, las excepciones de los verbos... pero esto ya es otro cantar.

Los números

A la vez que la humanidad avanzaba en el uso del lenguaje, con el tiempo apareció un tipo especial de palabras, sujeto a normas especiales... los números!

En el principio de la humanidad, cuando la gente apenas pronunciaba algunas palabras, antes de que existieran palabras para describir los números, estos ya se conocían. Aunque no se tuvieran nombres para ellos, se utilizaban. ¿Cómo? Mediante el recuento. Se utilizaban piedras. Se colocaban haciendo montones, se ponían, se quitaban y de esa manera se podía realizar acuerdos y transacciones. Aunque en vez de piedras también se pueden utilizar ramitas o dedos.

El hueso de Lebombo tiene una antigüedad de 35 000 años y fue hallado en Border Cave, en las montañas de Lebombo, entre Sudáfrica y Suazilandia. Ese hueso (peroné de babuino) está fracturado, pero sabemos que tiene al menos 29 marcas incisas, por lo que se podría relacionar con el ciclo lunar o menstrual femenino, es decir, un instrumento para contar el tiempo de algún ciclo natural. Existen otros huesos con incisiones, correspondientes todos ellos a diferentes épocas del Paleolítico superior.

El proceso de contar empezó siendo digital (dedos), pero tenía el pequeño problema de que tenemos un número finito de dedos. Por otra parte, una vez contado ¿hay alguna forma de conservar o almacenar el resultado? El problema a la hora de utilizar ramitas o dedos o un hueso con rallas, es el espacio que ocupan, la posibilidad de perderlos y su baja utilidad.

En algún momento de la historia, entre el uso del hueso marcado y la civilización incipiente, la gente se acostumbró a contar en grupos de cinco y de diez, por una razón muy evidente, era un método muy manual, pero de corto alcance en algunas situaciones.

Es probable que cuando el hombre vivía en las cavernas se encontraran con el problema de emitir sonidos relacionados con los números. La solución era sencilla: realizar un ruido diferente para cada número relacionado con los dedos y las manos. Sorprendentemente, las palabras que crearon estos cazadores y agricultores constituyen la base de los términos que se utilizan hoy en día en las lenguas de Europa y de otras partes del mundo cuyas lenguas son indoeuropeas.

El nacimiento del número escrito surgió en la civilización babilónica en Mesopotamia (Irak actual) del 2000 a. C. hasta el 300 d. C. En ella se han encontrado tabletas de barro cocido con símbolos cuneiformes que contienen textos matemáticos relacionados con tablas para multiplicar, dividir, elevar al cuadrado, elevar al cubo, raíces cuadradas, medir longitudes y áreas, calcular soluciones a ecuaciones lineales y cuadráticas. Utilizaron la notación sexagesimal (base 60, sin el símbolo para el cero, sin la coma decimal), pero fue el primero en emplear notación posicional. Hacia el 700 a. C., utilizan el (.) para denotar el espacio vacío. Inicialmente

contaban con los dedos de las manos, posteriormente utilizaron las mesas de contar (predecesoras del ábaco).

La civilización egipcia perduró desde el 3150 a. C. hasta el 31 a. C. El papiro Rhind (o Ahmes) es uno de los más famosos que trata sobre matemáticas. Es un documento escrito en un papiro de unos 6 metros de longitud y 33 centímetros de anchura, está en un buen estado de conservación, con escritura hierática. Se data del 2000 al 1800 a. C. Contiene 87 problemas matemáticos con cuestiones aritméticas básicas, fracciones, cálculo de áreas, volúmenes, progresiones, repartos proporcionales, reglas de tres, ecuaciones lineales y trigonometría básica. Su sistema numérico era de base 10.

Los antiguos encontraron dos formas básicas de poner eso por escrito. En el sistema egipcio se empleó un símbolo diferente para cada cantidad, lo que podría tener un cierto atractivo artístico, pero esos símbolos numerales son de muy difícil lectura y realizar operaciones de esa manera resultaba una tarea ardua.

El desarrollo de las matemáticas en China fue independiente de los desarrollos en otros países, debido a su localización geográfica. Parece que el desarrollo se inició hacia el año 1300 a. C. Su libro de referencia es el *Mathematical Treatise in Nine Sections* (XIII a. C.) utilizado durante cientos de años.

El sistema chino de cálculo se basaba en la utilización de palillos. Cada pila de ellos se colocaba en posiciones diferentes indicando, unidades, decenas, centenas... (análogo al procedimiento actual denominado *notación posicional*) dejando un hueco vacío cuando se representaba lo que ahora conocemos como cero. Para facilitar se utilizaba un tablero cuadrículado.

Mientras en el Mediterráneo apareció un gran invento, que también apareció en algunas otras civilizaciones: ¡el ábaco! Consiste en una disposición en columnas de pequeñas «cuentas». Cada una de ellas en una columna dada tiene el valor de diez de las que están en la columna situada inmediatamente a su derecha. Los números se registran empujando las cuentas de modo que queden alineadas en grupos.

La civilización romana inventó un sistema de numeración propio para representar números, heredaron el conocimiento griego y calculaban con el ábaco. El suyo consistía en cuentas de mármol que deslizaban sobre ranuras hechas en una placa de bronce. Esas cuentas se denominaban *cal/x* ('caliza') y de ahí proviene el nombre de *calcular* y de *calculus*, que ahora expresamos como calcular y cálculo.

Por decirlo de alguna manera la Antigüedad fue realmente la era de las calculadoras.

Dirijamos ahora nuestra atención hacia la India, más o menos hacia el siglo I o II. El símbolo para el cero se inventó aproximadamente hace mil ochocientos años.

Aunque ya los babilonios, los griegos, los mayas y los chinos se habían dado cuenta de la necesidad de utilizar un símbolo especial como marcador de posición para asegurar que el resto de números permanecían en su posición correcta, fue en la India donde se percataron de que el cero significaba algo más que eso. Comprendieron que el cero era en realidad un número. Y esto abrió la posibilidad de realizar operaciones aritméticas decimales en una hoja de papel.

Otras de las contribuciones hechas por los matemáticos indios a las matemáticas son:

- El sistema decimal y los guarismos actuales.
- El cero.
- Los números negativos.
- ...

Sus textos más famosos se recogen en lo que se conoce como *suvasutras*. Una colección de documentos religiosos que fueron escritos entre el 600 y el 300 a. C.

Durante el siglo VIII, la matemática de la India fue absorbida por los árabes que la llevaron en sus conquistas a todo el Occidente, en particular hasta España.

Hacia el año 830, un erudito persa escribió el texto «estándar» acerca de la materia de calcular con papel. Su nombre era Mohammed Ibn Musa Abu Djefar, más conocido como Al-khwarismi, o en español Aljuarismi. El título del libro se denominó *Hisab al-jabr wál-muqabala (Libro de la reducción)* o bien *Álgebra*, en cristiano.

Los árabes hicieron aportaciones en todas las ramas del conocimiento matemático y en la época del florecimiento de los árabes en España, muchos textos de las matemáticas se tradujeron del árabe al latín.

El primer europeo que usó numeración arábica en los años setena del siglo X fue el monje Geribert D'Aurillac, que posteriormente llegó a ser el papa n.º 139 con el nombre de Silvestre II (938-1003).

En 1202 Leonardo de Pisa (1170-1250) también llamado Fibonacci, publicó el *Liber Abaci*, explicación del sistema indoárabe, que hacia el año 1400 se generalizó y hacia el 1550 acabó siendo de uso popular.

A propósito, observen que no se pueden hacer operaciones aritméticas sin papel, de modo que los europeos también aprendieron de los árabes la fabricación de papel, que a su vez lo habían obtenido de los chinos que fueron sus inventores.

El conocimiento transmitido por los árabes se introdujo por primera vez en España y Sicilia en el siglo X. El primer taller fue fundado en Córdoba en 1036, seguido por otro en 1144, en el pueblo de Xátiva.

La información

El motivo de esta presentación es hacerle llegar la idea de que la información es «algo» que para empezar se puede medir y, por lo tanto, existe. Por ello, la informática definida como el tratamiento de la información tiene su lugar dentro del campo de los conocimientos humanos.

El Sistema Internacional de Unidades (abreviado SI), también denominado Sistema Internacional de Medidas, es el nombre que recibe el sistema de unidades que se usa en casi todos los países y es la forma actual del sistema métrico decimal. Consta de siete unidades, que son las que se utilizan para expresar magnitudes físicas definidas como básicas, a partir de las cuales se definen las demás.

Como curiosidad se puede afirmar que solo hay tres países que no lo han adoptado como sistema prioritario: Birmania, Liberia y Estados Unidos

Magnitudes básicas del SI

Magnitud física básica	Símbolo dimensional	Unidad básica	Símbolo de la Unidad	Observaciones
Longitud	L	metro	m	Se define fijando el valor de la velocidad de la luz en el vacío
Tiempo	T	segundo	s	Se define fijando el valor de la frecuencia de la transición hiperfina del átomo de cesio.
Masa	M	kilogramo	kg	Es la masa del «cilindro patrón» custodiado en la Oficina Internacional de Pesos y Medidas, en Sèvres (Francia).
Intensidad de corriente eléctrica	I	amperio	A	Se define fijando el valor de constante magnética.
Temperatura	Θ	kelvin	K	Se define fijando el valor de la temperatura termodinámica del punto triple del agua.
Cantidad de sustancia	N	mol	mol	Se define fijando el valor de la masa molar del átomo de carbono-12 a 12 gramos/mol. Véase también número de Avogadro
Intensidad luminosa	J	candela	cd	Véase también conceptos relacionados: lumen, lux e iluminación física

Dicha tabla se acompaña de una tabla de nombres que simbolizan los múltiplos y submúltiplos de las unidades básicas

1000 ⁿ	10 ⁿ	Prefijo	Símbolo	Escala Corta	Escala Larga	Equivalencia decimal en los Prefijos del SI	Asignación
1000 ⁸	10 ²⁴	yotta	Y	Septillón	Cuatrillón	1 000 000 000 000 000 000 000 000	1991
1000 ⁷	10 ²¹	zetta	Z	Sextillón	Mil trillones	1 000 000 000 000 000 000 000	1991
1000 ⁶	10 ¹⁸	exa	E	Quintillón	Trillón	1 000 000 000 000 000 000	1975
1000 ⁵	10 ¹⁵	peta	P	Cuatrillón	Mil billones	1 000 000 000 000 000	1975
1000 ⁴	10 ¹²	tera	T	Trillón	Billón	1 000 000 000 000	1960
1000 ³	10 ⁹	giga	G	Billón	Mil millones / Millardo	1 000 000 000	1960
1000 ²	10 ⁶	mega	M	Millón		1 000 000	1960
1000 ¹	10 ³	kilo	k	Mil / Millar		1 000	1795
1000 ^{2/3}	10 ²	hecto	h	Cien / Centena		100	1795
1000 ^{1/3}	10 ¹	deca	da	Diez / Decena		10	1795
1000 ⁰	10 ⁰	ninguno		Uno / Unidad		1	
1000 ^{-1/3}	10 ⁻¹	deci	d	Décimo		0,1	1795
1000 ^{-2/3}	10 ⁻²	centi	c	Centésimo		0,01	1795
1000 ⁻¹	10 ⁻³	mili	m	Milésimo		0,001	1795
1000 ⁻²	10 ⁻⁶	micro	μ	Millonésimo		0,000 001	1960
1000 ⁻³	10 ⁻⁹	nano	n	Billonésimo	Milmillonésimo	0,000 000 001	1960
1000 ⁻⁴	10 ⁻¹²	pico	p	Trillonésimo	Billonésimo	0,000 000 000 001	1960
1000 ⁻⁵	10 ⁻¹⁵	femto	f	Cuatrillonésimo	Milbillonésimo	0,000 000 000 000 001	1964
1000 ⁻⁶	10 ⁻¹⁸	atto	a	Quintillonésimo	Trillonésimo	0,000 000 000 000 000 001	1964
1000 ⁻⁷	10 ⁻²¹	zepto	z	Sextillonésimo	Miltrillonésimo	0,000 000 000 000 000 000 001	1991
1000 ⁻⁸	10 ⁻²⁴	yocto	y	Septillonésimo	Cuatrillonésimo	0,000 000 000 000 000 000 000 001	1991

Vivimos en la era de la información excesiva. Gracias a los avances tecnológicos de los siglos XX y XXI, los ciudadanos del mundo gozamos de acceso instantáneo a más información que la que cualquiera de nosotros puede asimilar.

Hay información acerca del tiempo atmosférico, de la política, de los personajes célebres, de los deportes, de la ciencia, de la tecnología, del arte, de la banca, de las religiones, de las redes sociales, de páginas web, de blogs, del mercado de valores, de publicidad, de historia, de libros, de películas, de música, de asociaciones, de empresas, de educación..., incluso de información acerca de la información.

Es evidente que nuestra era actual requiere de un dispositivo tecnológico dedicado exclusivamente a almacenar, clasificar, seleccionar, comparar, combinar, enviar y presentar información a alta velocidad. Tal equipo es lo que conocemos como computadora u ordenador.

¿Qué es la información?

En el sentido en que se la emplea cotidianamente, según el diccionario de la Real Academia Española es: «Comunicación o adquisición de conocimientos que permiten ampliar o precisar los que se poseen sobre una materia determinada». Sin embargo, en el mundo de las computadoras este término tiene un significado mucho más amplio. La definición moderna proviene de Claude Shannon, ingeniero de la empresa Bell Laboratories. A él se le considera fundador de la teoría de la información. En 1948 aparece publicado un libro titulado *Una teoría matemática de la comunicación*. En ese trabajo se demostró que todas las fuentes de información (telégrafo eléctrico, teléfono, radio, la gente que habla, las cámaras de televisión, etc.) se pueden medir, o lo que es lo mismo, cualquier tipo de dato se puede convertir a dígitos binarios.

Mostró también que la información se puede transmitir sobre un canal si, y solamente si, la magnitud de la fuente no excede la capacidad de transmisión del canal que la conduce, y sentó las bases para la corrección de errores, supresión de ruidos y redundancia.

Según Shannon, la información está presente siempre que se transmite una **señal** de un sitio a otro, partiendo de un **emisor** y llegando a un **receptor**. Y no importa de qué tipo de señal se trate.

Ejemplo 1, suponga dos seres humanos que hablan entre sí. En un momento determinado, uno habla (el **emisor**) y el otro escucha (el **receptor**). La señal tiene forma de **palabras** que físicamente son ondas mecánicas que se propagan por el aire.

Ejemplo 2, suponga un ser humano que mira un cuadro. En un momento determinado la luz (energía electromagnética) rebota en el cuadro (el **emisor**) y este refleja ondas luminosas que llevan la información sobre los colores y su distribución (la **señal**) hasta el ojo del ser humano (el **receptor**).

Ejemplo 3, además la retina de nuestros ojos (el **emisor**) transforma las señales anteriores en impulsos eléctricos (la **señal**) que van a ciertas partes del cerebro (el **receptor**).

Ejemplo 4, un puñetazo entre dos seres humanos, el que da (el **emisor**), el que recibe (el **receptor**), y el golpe (la **señal**), también tiene valor informativo.

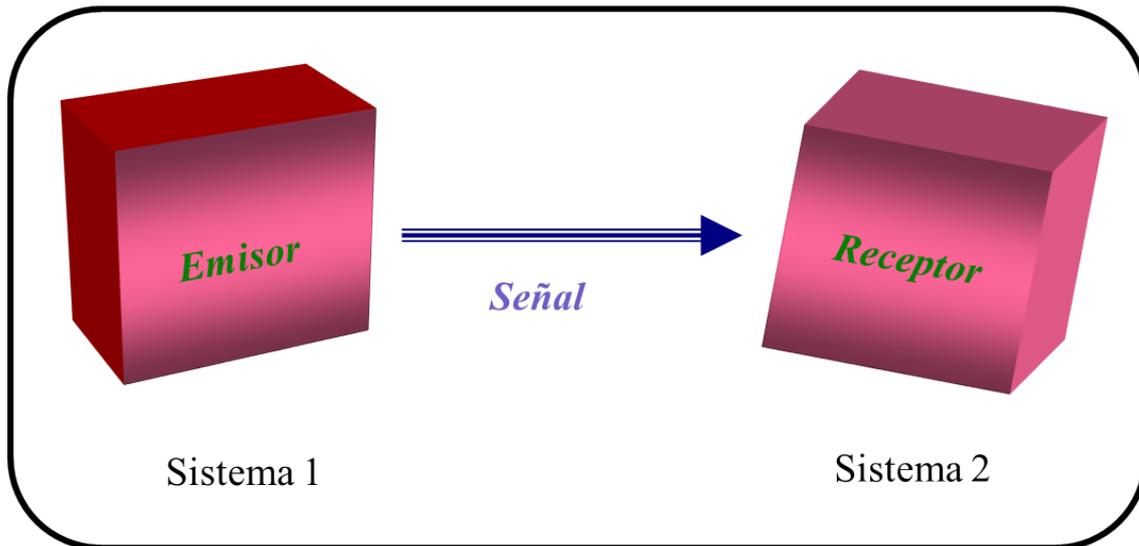
Como ve, la información aparece en muchas formas, verbal, visual, sonora, musical... Y toda ella puede manejarse mediante una computadora.

La teoría de la información

Para adentrarnos en ella vamos a necesitar definir un pequeño número de conceptos.

- Proceso de comunicación:
 - Es la interacción que tiene lugar entre dos sistemas, tal que los aspectos relevantes del resultado de la misma no pueden explicarse exclusivamente en términos de un intercambio energético.
 - Ello presupone que la descripción pertinente de los sistemas que interaccionan incluye aspectos no descriptibles usando el lenguaje de los modelos físicos. En el proceso de interacción, desde luego, hay un intercambio de energía, pero este intercambio no es quien determina los efectos que se producen en la interacción.

Proceso de comunicación



- Esta situación se describirá indicando que en el proceso se intercambia un «mensaje».
- Mensaje:
 - Desde un punto de vista formal, el mensaje viene caracterizado por su contenido en información.
 - El estudio y la caracterización de los mensajes se hace con la teoría de la información.
 - Dependiendo del aspecto bajo el que se estudie el problema de la comunicación, se desemboca en un nivel distinto de la teoría.
- Mensaje analizado a nivel sintáctico:
 - Es el más bajo y se ocupa de los **símbolos** del mensaje, su **forma**, su **duración** y sus **propiedades** estadísticas.
 - Es el nivel que, hasta ahora ha sido matemáticamente formalizado y donde existen definiciones rigurosas y experimentalmente operacionales.

- Mensaje analizado a nivel semántico:
 - Es el nivel intermedio, se ocupa del **significado** de los mensajes para los sistemas que interaccionan. Es decir, trata de asegurar que lo que el emisor quiere decir y lo que el receptor va a entender, es lo mismo.
 - Este nivel ha sido algorítmicamente formalizado y donde existen definiciones rigurosas y experimentalmente operacionales.
- Mensaje analizado a nivel pragmático:
 - Es el nivel de comunicación más alto, se ocupa de la posible utilidad o el valor del mensaje recibido por el receptor
 - Se desconoce por el momento cómo aproximarse a él.

La medida de la información

Cojamos el *Quijote*, y vamos a hacer un análisis sintáctico. Los datos que necesitaremos son, el número de letras que utilizó Miguel de Cervantes, la cantidad de cada uno de los símbolos alfabéticos y su porcentaje.

Aunque le parezca mentira, hay alguien que lo ha hecho y el resultado obtenido ha sido el siguiente

Total: 1 640 502 letras

E	229 188	14,0 %	A	200 492	12,2 %	O	162 512	9,9 %	S	125 726	7,7 %
N	108 440	6,6 %	R	100 953	6,2 %	I	90 070	5,5 %	L	89 141	5,4 %
D	87 237	5,3 %	U	79 471	4,8 %	T	61 749	3,8 %	C	59 435	3,6 %
M	44 658	2,7 %	P	35 464	2,2 %	Q	32 483	2,0 %	Y	25 115	1,5 %
B	24 146	1,5 %	H	19 920	1,2 %	V	17 855	1,1 %	G	17 225	1,0 %
J	10 530	0,6 %	F	7581	0,5 %	Z	6491	0,4 %	Ñ	4241	0,3 %
X	377	0,0 %	W	2	0,0 %	K	0,0	0,0 %			

Lo que se puede comprobar es que coincide con las frecuencias de uso de las letras en español:

- Si se ordenan de mayor a menor aparición se obtiene
 - EAOSRNIDLCTUMPBGVYQHFZJXWK
- Las vocales ocupan alrededor del 45 % del texto.
- La E y la A son las que con más frecuencia aparecen.
- Las consonantes más frecuentes son S, R, N, D, L, C...
- Las seis letras menos frecuentes son F, Z, J, X, W, K.

Se define la probabilidad (p_i) de cada letra, como su frecuencia de uso

Conceptos de probabilidad necesarios:

- Experimentos aleatorios:
 - Son aquellos en los que no se puede predecir el resultado, ya que este depende del azar (sin orden).
 - Ejemplos de experimentos aleatorios:
 - Si lanzamos una moneda no sabemos de antemano si saldrá cara o cruz.
 - Si lanzamos un dado tampoco podemos determinar el resultado que vamos a obtener.
- Espacio muestral:
 - Es el conjunto de todos los posibles resultados de una experiencia aleatoria, lo representaremos por E .
 - Ejemplos de espacios muestrales:
 - De una moneda es $E = \{C, X\}$
 - De un dado es $E = \{1, 2, 3, 4, 5, 6\}$
- Suceso seguro:
 - Algo que sabemos que va ocurrir.
 - Ejemplos de sucesos seguros:
 - De una moneda, que salga cara o que salga cruz.
 - De un dado, que salga un número menor de siete.
- Suceso imposible:
 - Es algo que no puede ocurrir.
 - Ejemplos de suceso imposible:
 - De una moneda, que salga un cubo.
 - De un dado, que salga un siete.
- Sucesos independientes:
 - Dos sucesos, A y B , son independientes cuando la probabilidad de que suceda A no se ve afectada porque haya sucedido o no B .
 - Ejemplo de sucesos independientes:
 - Lanzar dos dados, los resultados de cada uno son independientes

- Sucesos dependientes:
 - Dos sucesos, A y B, son dependientes cuando la probabilidad de que suceda A se ve afectada porque haya sucedido o no B.
 - Ejemplo de sucesos independientes:
 - Extraer dos cartas de una baraja sin reposición.
- Definición de probabilidad:

Si realizamos un experimento aleatorio en el que hay «n» sucesos elementales, todos igualmente probables (equiprobables), entonces si A es un suceso, la probabilidad de que ocurra el suceso A viene dada por la expresión:

$$P(A) = \frac{\text{número de casos favorables a A}}{\text{número total de casos posibles}}$$

- Ejemplo: Calcular la probabilidad de que, al tirar un dado, salga un número par.
 - Casos favorables {2, 4, 6}
 - Casos posibles {1, 2, 3, 4, 5, 6}

$$P(\text{par}) = \frac{3}{6} = \frac{1}{2} = 0,5$$

- Axiomas de la probabilidad:
 - La probabilidad es positiva y menor o igual que 1
 $0 \leq p(A) \leq 1$
 - La probabilidad del suceso seguro es 1
 $p(E) = 1$
 - Si A y B son incompatibles, entonces
 $P(A, B) = P(A) + P(B)$

- Sobre los mensajes:

La forma de aparición de los símbolos para formar mensajes depende de la naturaleza de la fuente.

- Ejemplo. Observe cómo se expresa un libro, una pintura, una partitura, una canción...

- Medida de la información contenida en un mensaje:
 - En general se asocia la producción de un mensaje a un «emisor» que lo genera.
 - Ese emisor o fuente se caracteriza por los «símbolos» con los que puede formar mensajes.
 - A ese conjunto de símbolos se le denomina *alfabeto* y se denotará utilizando la teoría de conjuntos $\{S\} = \{s_1, s_2, s_3, s_4, \dots, s_n\}$
 - Ejemplos:
 - ✓ El alfabeto para los textos escrito $\{a, b, c, \dots, z\}$
 - ✓ Las notas musicales $\{do, re, mi, \dots, si, \dots\}$
 - ✓ Los píxeles de un monitor $\{r, g, b\}$
 - ✓ Los niveles de las señales digitales $\{0, 1\}$
 - ✓ El punto y la raya para el código morse $\{., -\}$
 - ✓ ...
 - La forma de aparición de los símbolos para formar mensajes depende de la naturaleza de la fuente.
 - Se supondrá que está determinada satisfactoriamente indicando la probabilidad de aparición de cada símbolo.
 - A cada símbolo « s_i » se le asocia una probabilidad de aparición « p_i ».
 - Por sencillez, inicialmente se supondrá que la aparición de un símbolo no influye en la aparición de otro (normalmente no suele ser así).
 - Las probabilidades « p_i » se consideran independientes, es decir

$$p_i \neq f(p_j)$$
 - Se entiende por mensaje una sucesión finita de símbolos que pueden repetirse del conjunto $\{S\}$.
- Axiomas de la información:
 - La información aportada por cierto resultado es inversamente proporcional a la probabilidad de ser obtenido

$$I(s_i) = f(1/p_i)$$

- La información aportada por varios resultados es la suma de cada una de sus informaciones, y la probabilidad de que aparezcan varios resultados es el producto de sus probabilidades

$$I(s_i \text{ y } s_j) = I(s_i) + I(s_j) = f\left(\frac{1}{p_i} * \frac{1}{p_j}\right)$$

- El resultado con probabilidad de aparición 1, no aporta ninguna información

$$0 = f(1)$$

- La función información I:

- Las propiedades anteriores configuran que la función que mide la información sea algo proporcional a la función logarítmica (L_b) en base (b), siendo (b) cualquier número natural mayor que 0, ya que verifica las tres propiedades anteriores. Por lo tanto

$$I(s_i) = K_b L_b(1/p_i) = -K_b L_b(p_i)$$

- K_b permite que el resultado sea el mismo independiente de la base de numeración utilizada para la función logarítmica. Lo habitual es que si selecciona como (b) el valor (2), entonces $K_2 = 1$ y las unidades de medida se llaman (bits).
- Por lo tanto, en base 2, un bit solo puede tener dos valores {0, 1}

LA INFORMACIÓN DE UN BIT

El bit es la unidad más pequeña con que se puede medir la información. Algo que proporciona un bit de información, como un interruptor, se puede encontrar en dos estados complementarios: encendido o apagado.

1 BIT = 2 ESTADOS (2¹)

Dos interruptores proporcionan dos bits y cuatro estados posibles.

2 BITS = 4 ESTADOS (2²)

Un tercer bit duplica las alternativas posibles a ocho.

3 BITS = 8 ESTADOS (2³)

El byte empleado en informática contiene 8 bits y contempla 256 estados. Para el almacenamiento, procesado y transmisión de datos en informática, a cada uno de estos estados se le asigna un carácter (una letra, un número, un símbolo).

1 BYTE = 8 BITS = 256 ESTADOS (2⁸)

- Comentario. Tal y como se ha definido la función que mide la información aportada por un símbolo, $I(s_i) = f(1/p_i)$, esta depende de la inversa de la probabilidad de aparición:
 - Si p_i es grande, es decir, su aparición es muy frecuente, la información asociada es baja.
 - Si p_i es pequeña, es decir, su aparición es inesperada o poco frecuente, la información asociada es alta.

Esto se puede resumir diciendo que cuanto más inesperado es el símbolo, más alta es la información que aporta.

Ejemplo para pensar. Suponga que usted ve en un periódico en días diferentes las dos noticias siguientes:

- El sr. Fulanito ha sido mordido por un perro al pasar por la calle menganita.
- El sr. Fulanito ha mordido a un perro al pasar por la calle menganita.

¿Cuál de las dos noticias le llamaría más la atención? Se supone que la segunda, porque es más inesperada. Desde el punto de vista que estamos tratando, la información aportada por un suceso es inversamente proporcional a su probabilidad de aparición.

- Notación (recordatorio):

$$\sum_{i=1}^{i=5} i = 1 + 2 + 3 + 4 + 5$$

$$\sum_{i=m}^n x_i = x_m + x_{m+1} + x_{m+2} + \dots + x_n$$

$$\mathbf{u} = c_1 x_1 + c_2 x_2 + c_3 x_3 + \dots + c_n x_n$$

$$\mathbf{u} = \sum_{i=1}^n c_i x_i$$

- Cantidad de información contenida en un mensaje:
 - Si el mensaje está formado por «m» símbolos de un alfabeto «S» que tiene en total «N» símbolos equiprobables con probabilidad «p», entonces « $p_i = 1/N$ » y la cantidad de información que contiene es la suma de las informaciones que aporta cada uno de los «m» símbolos, es decir,

$$I = \sum_{i=1}^m L_2\left(\frac{1}{p_i}\right) = \sum_{i=1}^m L_2(N) = m L_2(N) \text{ bits}$$

- Si el mensaje está formado por «m» símbolos de un alfabeto «S» que tiene en total «N» símbolos no equiprobables, cada uno con probabilidad «p_i», entonces la cantidad de información que contiene es la suma de las informaciones que aporta cada uno de los «m» símbolos, es decir,

$$I = \sum_{i=1}^m L_2 \left(\frac{1}{p_i} \right)$$

- Recordemos que las unidades de medida en el sistema decimal eran

Nombre	Abreviatura	Factor
Kilo	K	10 ³ = 1000
Mega	M	10 ⁶ = 1 000 000
Giga	G	10 ⁹ = 1 000 000 000
Tera	T	10 ¹² = 1 000 000 000 000
Peta	P	10 ¹⁵ = 1 000 000 000 000 000
Exa	E	10 ¹⁸ = 1 000 000 000 000 000 000
Zetta	Z	10 ²¹ = 1 000 000 000 000 000 000 000
Yotta	Y	10 ²⁴ = 1 000 000 000 000 000 000 000 000
Bronto	B	10 ²⁷ = 1 000 000 000 000 000 000 000 000 000

- Al ser 2 los valores que puede tener un bit, por analogía con el sistema decimal los múltiplos de los bits se denominan

- 1 bit = 1 bit
- 1 nibble = 4 bits
- 1 byte = 8 bits

Nombre	Abreviatura	Factor en bytes
Kilo	K	2 ¹⁰ = 1024
Mega	M	2 ²⁰ = 1 048 576
Giga	G	2 ³⁰ = 1 073 741 824 109
Tera	T	2 ⁴⁰ = 1 099 511 627 776
Peta	P	2 ⁵⁰ = 1 125 899 906 842 624
Exa	E	2 ⁶⁰ = 1 152 921 504 606 846 976
Zetta	Z	2 ⁷⁰ = 1 180 591 620 717 411 303 424
Yotta	Y	2 ⁸⁰ = 1 208 925 819 614 629 174 706 176
Bronto	B	2 ⁹⁰ = 1 024 Yottabytes

○ Ejemplo 1:

- Sea el mensaje contenido en una imagen de TV antigua, que por sencillez supondremos que está formada por $m = 500 \times 500 = 250\,000$ píxeles.
- Cada píxel supondremos que puede estar en uno de los 32 niveles de grises.
- Supóngase equiprobabilidad, es decir, $p_i = 1/32$.
- Por lo tanto, la información que presenta es

$$I = m L_2(N) \text{ bits} = 250\,000 L_2(32) = 125\,000 \text{ bits}$$

- Si además se transmite a razón de 25 cuadros por segundo, la capacidad de esa televisión «equiprobable» será de

$$C = I \times 25 = 31\,250\,000 \text{ bits/s} \sim 32 \text{ Mbits/seg}$$

○ Ejemplo 2:

- Mida la información contenida en el *Quijote*, observe que las probabilidades de cada símbolo son diferentes.

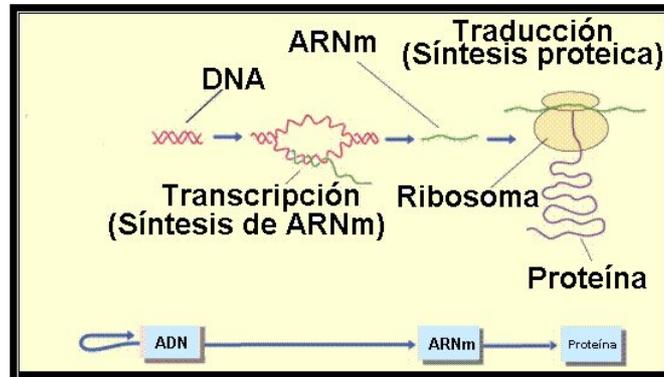
• Otras propiedades de la información:

Todos los tipos de señales que transportan información tienen otras propiedades además de que se pueda medir su cantidad de información. Esas propiedades a las que nos estamos refiriendo son:

- Capacidad de almacenamiento, en libros, discos, cuadros, memoria humana, cintas magnéticas, diagramas... lo que permite la transmisión del mismo mensaje muchas veces.
- Capacidad de transporte y transmisión. Lo que permite la deslocalización de la información con respecto al emisor.
- Capacidad de reproducción en sincronía con el emisor o de manera asíncrona con el emisor. Es decir, en sincronía, el emisor y el receptor se escuchan, se leen, se ven... en el mismo momento, independiente de que se encuentren en espacios físicos diferentes.
- Se puede comprimir aprovechándonos de la redundancia implícita existente en cualquier mensaje.
- ...

- Ejemplo 3: Un caso especial, la elaboración de una proteína conforme a la información almacenada en un gen.

Véase <<http://ac-biologia.blogspot.com/2016/05/sintesis-de-proteinas.html>>



- El DNA o ADN es una larga sucesión de unidades moleculares a las que se les da el nombre de pares de nucleótidos. Actúa de «emisor».
 - Esa secuencia es copiada en una molécula mensajera de ARN. Actúa como «mensaje».
 - El mensaje es recibido por una planta química, «una célula» que utiliza al ARN como diagrama para fabricar una molécula de proteína. Actúa como «receptor».
 - Algunas proteínas ayudan al ADN a reproducirse.
 - Si el ADN codifica proteínas que lo ayudan en su reproducción, entonces producirá más de esas proteínas, se copiará más ADN, etc.
 - Además, el ADN codifica otras proteínas que lo protegen en diversas formas, y otras que atacan y destruyen el ADN...
 - Entonces el sistema ADN-Proteínas se reproducirá una y otra vez, y esto es lo que denominamos una *forma de vida*.
 - Así la «vida», o todo organismo viviente, es un procesador de información molecular que ha funcionado automáticamente durante más de tres mil millones de años.
- Definición de informática

El almacenamiento, la transmisión, la combinación y la comparación de mensajes se conoce como *procesamiento de la información*. Y se define como informática a la ciencia / tecnología del tratamiento automático de la información.

- Definición de entropía:

- Definición de entropía (S) para un físico:

En termodinámica, la entropía (simbolizada como S) es una magnitud física definida para un sistema termodinámico en equilibrio. Mide el número de microestados compatibles con el macro estado de equilibrio, también se puede decir que mide el grado de organización del sistema o también se puede decir que es una medida de la incertidumbre acerca del estado de un sistema físico. Estos microestados no tienen por qué ser igualmente probables, por lo que el físico escribe

$$S = - \sum_i p_i \text{Log} (p_i)$$

- Definición de entropía (H) para un informático:

La entropía también se puede considerar como la cantidad de información promedio que contienen los símbolos usados. Los símbolos con menor probabilidad son los que aportan mayor información; por ejemplo, si se considera como sistema de símbolos a las palabras en un texto, palabras frecuentes como «que», «el», «a» aportan poca información, mientras que palabras menos frecuentes como «corren», «niño», «perro» aportan más información. Si de un texto dado borramos un «que», seguramente no afectará a la comprensión y se sobreentenderá, no siendo así si borramos la palabra «niño» del mismo texto original. También se puede decir que es una medida de la incertidumbre (o sorpresa) acerca de un mensaje entre todos los mensajes posibles que puede producir una fuente de comunicación. Los mensajes posibles no tienen por qué ser igualmente probables, por lo que Shannon escribe:

$$H = - \sum_i p_i \text{Log} (p_i)$$

- Comentario, no se trata de una mera coincidencia de formalismos, de que la naturaleza ofrezca unas respuestas similares a unos problemas similares. ¡Se trata de un mismo problema!

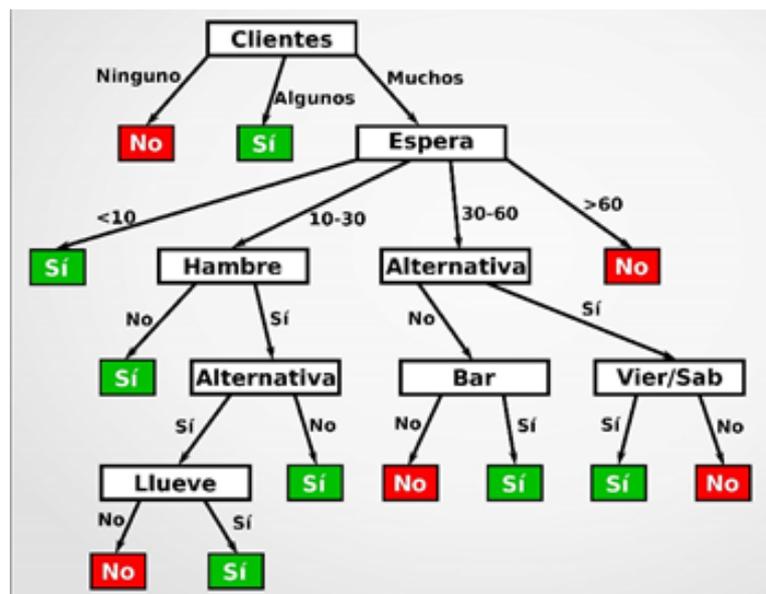
- Ejemplo, veamos cómo se podría tomar una decisión utilizando el análisis estadístico y la teoría de la información.

Hay que tomar la decisión de comer en un restaurante, las clases que son posibles respuestas a la pregunta son las decisiones (Sí o NO) y existen las siguientes posibilidades o atributos, denominemos A a dicho conjunto, al lado de cada uno se muestran sus valores posibles:

- ALT ¿Hay alternativas? (Sí / No)
- BAR Tiene bar (Sí / No)
- VIER Abierto en viernes / sábado (Sí / No)
- HAM Tenemos hambre (Sí / No)
- CLIE Hay clientes (Muchos / Algunos / Ninguno)
- PREC Precio (Barato, Normal, Caro)
- LLU ¿Llueve? (Sí / No)
- TIP Tipo de comida (española, italiana, francesa...)
- ESP Tiempo de espera estimada ([0-10), [10-30), [30, 60), [60-...])

Utilicemos para encontrar la respuesta lo que se conoce como un árbol de decisión, estos son representaciones gráficas de posibles decisiones. Los árboles de decisión tienen un primer nodo llamado *raíz* y luego se descomponen el resto de los atributos en ramas. Cada nodo interno es una pregunta sobre alguna característica y cada nodo hoja es una clase.

Un posible árbol basado en la subjetividad de una persona podría ser la siguiente propuesta



En ella, alguien ha decidido que el atributo «clientes» es el más importante, por ello está en la cabeza. Los otros atributos se van encontrando conforme se va recorriendo el árbol. El orden de aparición se supone que es más arriba cuanto más importante. Luego observe que hay recorridos que finalizan en las hojas de tipo clase, que en este caso es «Se toma la decisión Sí (Verde) o NO (Rojo)».

Ahora bien, si se dispusiera del comportamiento de muchas personas diferentes en una base de datos, se podría construir un árbol de decisión basado en esos datos. ¡Cuanto más mejor!, la idea sería ir seleccionando los atributos en orden de importancia, en este caso lo que vamos a utilizar es la teoría de la información para ir seleccionando el orden de importancia de cada dato.

Supóngase que se tienen un conjunto de ejemplos (E) que son las respuestas que darían 121 clientes diferentes a los atributos y la decisión final que tomarían. Vea la siguiente tabla etiquetada

	ALT	BAR	VIER	HAM	CLIE	PREC	LLU	TIP	...	Respuesta
X1	Sí	No	No	Sí	M	B	Sí	E	...	Sí
X2	Sí	No	No	No	A	N	No	E	...	No
X3	No	Sí	Sí	Sí	N	N	No	I	...	Sí
X4	No	Sí	Sí	No	N	C	Sí	M	...	Sí
X5	No	No	No	Sí	A	B	No	E	...	No
X6	Sí	Sí	No	No	M	C	Sí	E	...	Sí
X...
X121

Introduzcamos algunas definiciones:

- Si tenemos un atributo A y le añadimos la información de un atributo B, la ganancia de entropía es

$$\text{Ganancia (A + B)} = \text{Entropía (A)} - \text{Entropía (A+B)}$$
- La entropía de un atributo Z = $p(Z) \times \text{Información (Z)}$
- La información (Z) = $-L_2(p(Z))$ medida en bits
- La entropía (Z) = $-p(Z) \times L_2(p(Z))$
- Si Z tiene asociados N valores, entropía (Z) = $-\sum_{i=1}^N p(Z) \times L_2(p(Z))$
- $P(Z=z) = \frac{\text{número de casos favorables a } Z=z}{\text{número total de casos posibles Z}}$ siendo z uno de los valores posibles del atributo Z
- La entropía de (Z = z), si tiene N valores el atributo Z

$$\left(-\sum_{i=1}^N \frac{\text{número de casos favorables a } Z=z}{\text{número total de casos posibles Z}} \times L_2\left(\frac{\text{número de casos favorables a } Z=z}{\text{número total de casos posibles Z}}\right)\right)$$

Si denominamos:

- E al conjunto de ejemplos disponibles.
 - E_x al ejemplo X.
 - A al conjunto de atributos.
 - A_y al atributo Y.
 - V_y al conjunto de valores del atributo A_y .
 - C al conjunto de clases.
 - A_{yx} al atributo Y en el ejemplo X.
 - V_{yx} al valor del atributo Y en el ejemplo X.
 - C_x clase del ejemplo X.
 - n_z número de ejemplos que cumplen z.
- Tal y como ya se ha comentado, se va a tener en cuenta E para construir un árbol de decisión, y se van a ir eligiendo los atributos en el orden que vayan aportando la máxima ganancia de información, es decir,

$$\text{Ganancia}(E, A_y) = \text{Entropía}(E) - \text{Entropía}(E, A_y)$$

Pero Entropía(E) es un valor constante dado el conjunto E, luego maximizar la Ganancia(E, A_y) es lo mismo que minimizar Entropía(E, A_y).

Entonces las expresiones anteriores se pueden expresar con la notación indicada del modo siguiente:

Entropía (E, A_y , $V_y = v$) = entropía del valor «v» del atributo «A» en E, viene dada por

$$\text{Entropía}(E, A_y, V_y=v) = - \sum_{k=1}^C \frac{n_{yvk}}{n_{yv}} \chi L_2 \left(\frac{n_{yvk}}{n_{yv}} \right)$$

Entropía (E, A_y) = entropía del atributo «A» en E, viene dada por

$$\text{Entropía}(E, A_y) = \sum_{v=1}^{V_y} \frac{n_{yv}}{N} \left[- \sum_{k=1}^C \frac{n_{yvk}}{n_{yv}} \chi L_2 \left(\frac{n_{yvk}}{n_{yv}} \right) \right]$$

Entropía(E) = entropía del conjunto de ejemplos E, viene dada por

$$\text{Entropía}(E) = - \sum_{k=1}^C \frac{n_k}{N} L_2 \left(\frac{n_k}{N} \right)$$

Volvamos al ejemplo inicialmente propuesto:

- ALT ¿Hay alternativas? (Sí / No)
- BAR Tiene bar (Sí / No)
- VIER Abierto en viernes / sábado (Sí / No)
- HAM Tenemos hambre (Sí / No)
- CLIE Hay clientes (Muchos / Algunos / Ninguno)
- PREC Precio (Barato, Normal, Caro)
- LLU ¿Llueve? (Sí / No)
- TIP Tipo de comida (española, italiana, francesa...)
- ESP Tiempo de espera estimada ([0-10), [10-30), [30, 60), [60-...])

Y supongamos que tenemos la siguiente información en la base de datos, a partir de la que se calculan las magnitudes indicadas.

Valores «Clientes» (CLIE) (Muchos / Algunos / Ninguno)

Valor Muchos

Clase	n_{yvk}	n_{yv}	p_{iv}	$L_2(p_{iv})$	Ent
Sí	20	60	0,333	1,585	0,528
No	40	60	0,667	0,585	0,390
					0,918

Valor Algunos

Clase	n_{yvk}	n_{yv}	p_{iv}	$L_2(p_{iv})$	Ent
Sí	39	40	0,975	0,037	0,036
No	1	40	0,025	5,322	0,133
					0,169

Valor Ninguno

Clase	n_{yvk}	n_{yv}	p_{iv}	$L_2(p_{iv})$	Ent
Sí	1	21	0,048	4,381	0,210
No	20	21	0,952	0,071	0,068
					0,278

Atributo «Clientes» (CLIE)

Entropía

Valores	n_{yv}	N	p_y	Ent	Ent_Atributo
Ninguno	21	121	0,174	0,278	0,048
Algunos	40	121	0,331	0,169	0,056
Muchos	60	121	0,496	0,918	0,455
					0,559

Valores «Hambre» (HAM) (Sí / No)

Valor Sí

Clase	n_{yvk}	n_{yv}	p_{iv}	$L_2(p_{iv})$	Ent
Sí	51	71	0,718	0,477	0,343
No	20	71	0,282	1,828	0,515
					0,858

Valor No

Clase	n_{yvk}	n_{yv}	p_{iv}	$L_2(p_{iv})$	Ent
Sí	10	50	0,200	2,322	0,464
No	40	50	0,800	0,322	0,258
					0,722

Atributo «Hambre» (HAM)

Entropía

Valores	n_{yv}	N	p_y	Ent	Ent_Atributo
Sí	71	121	0,587	0,858	0,503
No	50	121	0,413	0,722	0,298
					0,802

...

Atributo «XXXXXX» (XXX)

Entropía

Valores	n_{yv}	N	p_y	Ent	Ent_Atributo
...	...	121
...	...	121
					...

Entropía de ejemplos

Clase	n_k	N	P_k	$L_2(p_k)$	Ent
Sí	61	121	0,504	0,988	0,498
No	60	121	0,496	1,012	0,502
					1,000

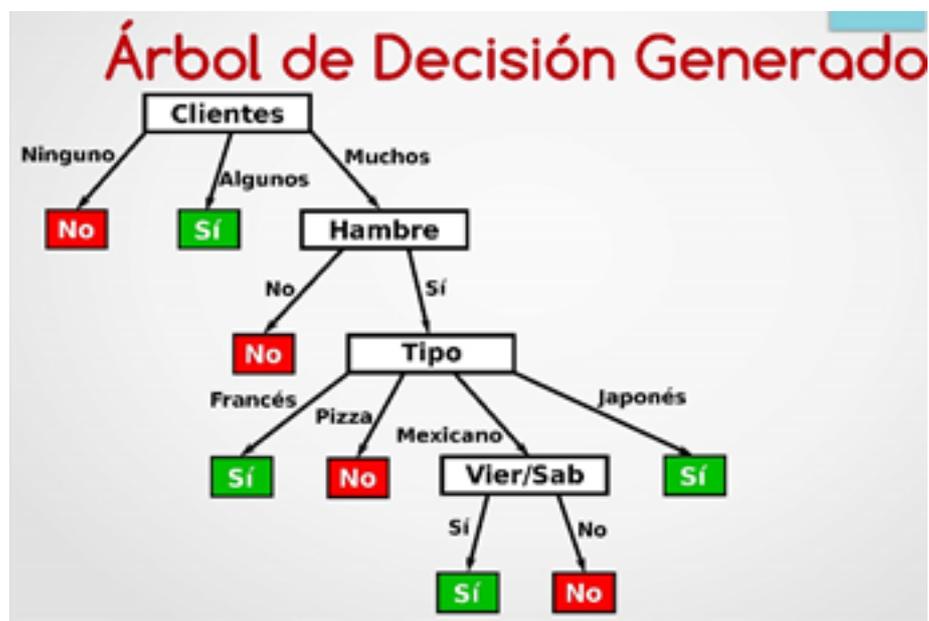
Ganancia de atributos

Atributos	Ent_Ejemplos	Ent_Atributo	Ganancia
Clientes	1,000	0,559	0,441
Hambre	1,000	0,802	0,198
...	1,000

Una vez calculada la ganancia de cada uno de los atributos, se selecciona aquel atributo que en esa tabla da el valor más alto de la ganancia. Supóngase que en este caso es el atributo «Clientes».

Una vez seleccionado este, se vuelve a empezar eliminando este atributo y rehaciendo todas las cuentas, hasta llegar a la nueva tabla de ganancia de atributos y se vuelve a seleccionar aquel que da la ganancia más alta.

Se sigue así hasta que lleguemos a ganancias que no añaden prácticamente información. Es decir, puede ocurrir, como en este caso, que el resto de los atributos no sean necesarios para tomar la decisión que andamos buscando.



- Definición de meme_

Es la unidad teórica de información cultural (es una información reproducida por la sociedad) transmisible de un individuo a otro, o de una mente a otra, o de una generación a la siguiente.

- Ejemplo 1. El cumpleaños:

El hecho de celebrarlo y que se le dé importancia es del todo cultural. Esta celebración implica un preciso código verbal y de comportamiento dentro de un tipo de sociedad.

- Hay que felicitar a la persona que cumple los años en la fecha exacta y se hace con frases típicas, distintas en cada idioma: ¡Feliz cumpleaños!; ¡Happy Birthday!; ¡Tanti auguril!...
- Es muy común organizar una fiesta para celebrar el cumpleaños y normalmente los invitados traen regalos.
- El cumpleaños suele apagar las velas de una tarta y mientras tanto pide un deseo, sin contárselo a nadie.
- Todos los invitados le cantan una canción típica (que, otra vez, varía de lengua a lengua).

El meme es un neologismo acuñado por Richard Dawkins en el libro *El gen egoísta*, por la semejanza fonética con «gene», y para señalar la similitud con «memoria» y «mímesis».

Un meme es una idea que puede evolucionar y que es tremendamente viral y propagable. Puede tener el formato de vídeo, foto, *collage*, melodía, idea, frase ingeniosa, moda, receta...

Los memes pueden sobrevivir y difundirse gracias a las soluciones que encuentran a los problemas que no tienen solución.

De esta forma han nacido las leyendas y las religiones, las cuales siendo un conjunto de memes suelen calificarse como macromemes.

- Un meme del cristianismo es la vida después de la muerte.
- Después de la muerte dos son los destinos principales:
 - El paraíso para los buenos.
 - El infierno para los malos.
- Ejemplo 2. Los niños los trae la cigüeña:

Cuando comienza la fase del ¿por qué? De los niños, muy a menudo los padres temen que llegue la fatídica pregunta, ¿Cómo nacen los niños? Viendo la situación con los ojos de los padres, eso representa un problema

aparentemente sin solución para muchas parejas, la verdad no es una opción cuando sus hijos son aún pequeños. El meme de la cigüeña es una salida salvadora en muchas culturas.

- Otros ejemplos supersticiosos:
 - El gato negro.
 - Viernes 17 / martes 13.
 - Los aviones no suelen tener la fila 13.
 - Algunos edificios omiten el piso 13.
 - No existe el modelo Renault 13.
 - No se suele asignar el número 13 a ningún participante.
 - Microsoft Office pasó de la versión 12 (Office 2007) a la 14 (Office 2010).
 - ...

Para el conjunto de los memes se dan las características propias de todo proceso evolutivo:

- Fecundidad (algunas ideas son especialmente efectivas).
- Longevidad (persisten durante mucho tiempo).
- Fidelidad de replicación (conservadurismo tradicional, especialmente el enseñado como parte de la educación infantil).

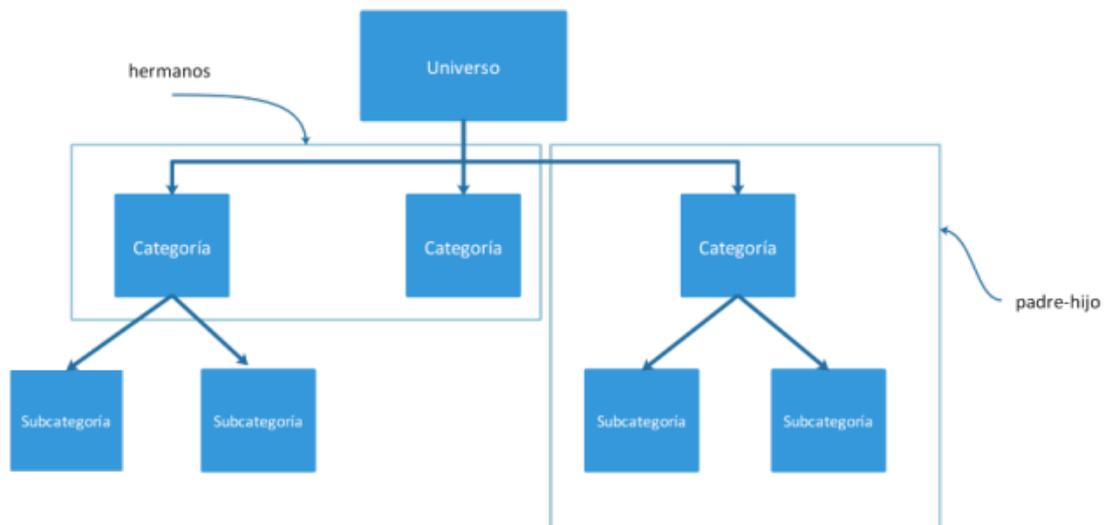
La utilización de la teoría de los memes se ha extendido por varias ramas del conocimiento y del pensamiento, pero no es aceptada universalmente, ni siquiera en el contexto de los estudios evolucionistas. Para algunos es una simple ocurrencia de Dawkins, un paralelismo innecesario que intenta extrapolar al mundo de la cultura su teoría de los genes egoístas.

Mapa cartográfico de la inteligencia artificial

La metáfora se puede usar con la finalidad de explicar y hacer comprender o visualizar un objeto desconocido mediante su reemplazo por otro conocido (que, en el contexto de esta presentación, ese objeto conocido viene a ser el mapa). Nuestro mapa metáfora pretende operar como un método de espacialización del pensamiento y su visualización mental. En este caso concreto, lo que se pretende es conseguir que el lector obtenga una representación tipo atlas, del mundo de los métodos o algoritmos propios de la inteligencia artificial agrupándolos en función del tipo de paradigma que los soporta.

La taxonomía o clasificación que vamos a utilizar es una estructura de organización jerárquica de la información formada por un conjunto de categorías y subcategorías, gracias a las cuales vamos a poder unir entidades que compartan alguna característica común. La segunda idea intrínseca es que todos los términos están relacionados o conectados entre sí. De este modo, se puede predecir dónde van a estar las cosas que buscamos, de esta manera se evitan exploraciones secuenciales o se reduce el número de interacciones necesarias para encontrar algo. La principal virtud de una taxonomía es la claridad, y esta se deriva principalmente de la coherencia lógico-semántica establecida.

La estructura propia de una taxonomía es el árbol que muestra de manera visual la jerarquía de categorías y subcategorías, con las categorías del mismo nivel en una relación de hermanos y las categorías y subcategorías en una relación padre-hijo.



No está predeterminado el número de niveles de una taxonomía, es decir, una subcategoría puede tener a su vez, términos hijos.

La clave en cualquier taxonomía es la consistencia. Los términos de las taxonomías deberían ser autoexcluyentes y, a poder ser, utilizar un mismo principio de organización que esté basado en algún principio razonable. Las categorías tienen una justificación muy clara, teniendo cuidado de que no existan más de un término para el mismo concepto.

Mediante las categorías se pretende facilitar una clasificación jerárquica de las entidades del mundo. Entidades muy parecidas y con características comunes formarán una categoría, y a su vez, varias categorías con características afines formarán una categoría superior.

En el mundo de la inteligencia artificial se pueden encontrar diferentes formas de categorizar el conjunto genérico de tecnologías que forman su cuerpo conceptual. Por ejemplo:

- Atendiendo a sus aplicaciones:
 - Reconocimiento automático del habla, cuyo objetivo es el reconocimiento de fonemas en una señal de voz.
 - Procesamiento del lenguaje natural, su objetivo es comprender los patrones de comunicación hombre máquina y viceversa.
 - Reconocimiento visual, pretende reconocer patrones, formas y elementos de una imagen.
 - Reconocimiento de texto, reconoce e identifica texto en formatos de imagen.
 - *Big Data*, búsqueda de estructuras y relaciones en grandes volúmenes de datos no homogéneos.
 - Sistemas expertos, repositorios de conocimiento humano acerca de una determinada área de conocimiento.
 - Robótica y personajes virtuales, busca sistemas autónomos que sepan adaptarse a un entorno determinado y resolver algún tipo de problema general.
 - Aprendizaje automático, trata de conseguir que un sistema aprenda y relacione información del modo en que lo haría una persona.
 - Inteligencia cognitiva y servicios cognitivos, combinación de tecnologías con el objetivo de crear servicios capaces de tener comprensión humana.
 - ...

- Atendiendo a su enfoque (véase: Stuart Russell y Peter Norvig):
 - Sistemas que piensan como humanos:
 - «El nuevo y excitante esfuerzo de hacer que los computadores piensen... máquinas con mentes, en el más amplio sentido literal». (Haugeland, 1985)
 - «La automatización de actividades que vinculamos con procesos de pensamiento humano, actividades como la toma de decisiones, resolución de problemas, aprendizaje...». (Bellman, 1978)
 - Sistemas que piensan racionalmente:
 - «El estudio de las facultades mentales mediante el uso de modelos computacionales». (Charniak y McDermont, 1985)
 - «El estudio de los cálculos que hacen posible percibir, razonar y actuar». (Winston, 1992)
 - Sistemas que actúan como humanos:
 - «El arte de desarrollar máquinas con capacidad para realizar funciones que cuando son realizadas por personas requieren de inteligencia». (Kurzweil, 1990)
 - «El estudio de cómo lograr que los computadores realicen tareas que, por el momento, los humanos hacen mejor». (Rich y Knight, 1991)
 - Sistemas que actúan racionalmente:
 - «La inteligencia computacional es el estudio del diseño de agentes inteligentes». (Poole *et alii*, 1998)
 - «IA..., está relacionada con conductas inteligentes en artefactos». (Nilsson, 1998)

- Atendiendo al modelo utilizado:
 - Simbólico:
 - Este programa intenta crear una mente artificial de arriba-abajo.
 - Allen Newell y Herbert Simon formularon en 1975 la hipótesis del sistema de símbolos físicos (SSF), según el cual «todo sistema de símbolos físicos posee los medios necesarios y suficientes para llevar a cabo acciones inteligentes».
 - Un SSF consiste en un conjunto de entidades denominadas *símbolos* que, mediante relaciones, pueden combinarse formando estructuras más grandes, como los átomos que forman moléculas, y que pueden ser transformados aplicando un conjunto de procesos. Estos pueden crear nuevos símbolos, crear y modificar relaciones entre símbolos, almacenarlos, comparar si dos son iguales o distintos, etc. Estos símbolos son físicos en tanto que tienen un sustrato físico-electrónico [circuitos electrónicos digitales (en el caso de los ordenadores)] o físico-biológico [neuronas (en el caso de los seres humanos)]. La naturaleza del sustrato carece de importancia siempre y cuando dicho sustrato permita procesar símbolos.
 - Es un modelo arriba-abajo que se basa en el razonamiento lógico y la búsqueda heurística como pilares para la resolución de problemas, sin que el sistema inteligente necesite formar parte de un cuerpo real situado en un entorno real. Opera con representaciones abstractas del mundo real que se modelan mediante lenguajes de representación basados principalmente en la lógica matemática y sus extensiones. Actualmente también se utiliza para aplicaciones que requieren percibir el entorno y actuar sobre él como en el caso de los sistemas autónomos.
 - Conexionista (modelización bioinspirada):
 - Este programa intenta crear una mente artificial de abajo-arriba.
 - Aproximación abajo-arriba, se basa en la hipótesis de que la inteligencia emerge a partir de la actividad distribuida de un gran número de unidades interconectadas que procesan información de forma paralela. Estas unidades son modelos ligeramente aproximados de la actividad eléctrica de las neuronas biológicas.
 - Las neuronas reales poseen complejas arborizaciones dendríticas con propiedades no solo eléctricas, sino también químicas nada triviales. Pueden contener conductancias iónicas que producen efectos no lineales.
 - Las neuronas reales pueden recibir decenas de millares de sinapsis variando en posición, polaridad y magnitud.

- En el cerebro hay unas células llamadas *gliales* que regulan el funcionamiento de las neuronas, siendo incluso más numerosas que estas.
 - Con respecto al proyecto de crear una mente, las neuronas artificiales tienen el problema de que, o bien no son modelos fieles a una neurona, o bien solo son un modelo muy simplificado de una neurona. Las neuronas reales no solo lanzan impulsos eléctricos en función de pesos, sino que también realizan una compleja gama de interacciones químicas que han sido totalmente pasadas por alto en estos modelos. El trabajo reside ahora en intentar entender mejor cómo funciona nuestro sistema nervioso, campo de investigación en el que, a pesar de los últimos avances, va todavía muy despacito.
- Evolutivo (modelización bioinspirada):
 - Se basa en la biología evolutiva seguida por los organismos complejos.
 - La idea es que, mediante un proceso evolutivo, un programa mejore automáticamente la solución a los problemas para los que ha sido programado, gracias a operadores de mutación y cruce de cromosomas que permiten crear nuevas generaciones de programas modificados cuyas soluciones son mejores que las de las generaciones anteriores.
 - Corpóreo:
 - La idea básica es que el cuerpo da forma a la inteligencia y, por lo tanto, sin cuerpo no puede haber inteligencia de tipo general.
 - Consiste en diseñar un sistema capaz de interactuar de forma directa con un entorno y puede relacionar las señales que percibe mediante sensores con representaciones internas simbólicas generadas a partir de lo percibido.
 - Esas interacciones conforman las habilidades cognitivas de los agentes, dando lugar a lo que se conoce como cognición situada.
 - La representación corpórea con representación interna ha ido ganando terreno en la IA.
- Atendiendo a su alcance:
 - IA débil (IAD) o estrecha. Engloba a toda la IA desarrollada hasta el momento. Está dedicada a resolver un problema específico o un conjunto de ellos de forma optimizada, pero sin posibilidad de extenderse a problemas generales. A este objetivo se ha dedicado la mayor parte de los objetivos de investigación. De hecho, si algo se ha aprendido de todo ese esfuerzo, es que los problemas que parecían más difíciles, como jugar al ajedrez, se han podido resolver. Y lo que parecía más fácil, como

traducir un texto o desplazarse por cualquier terreno, han resultado ser de lo más difícil.

- IA fuerte (IAF) o general (IAG). Es aquella inteligencia artificial capaz de igualar o superar la inteligencia humana en capacidad de razonamiento y deducción. A día de hoy es una utopía solo existente en la ciencia ficción porque a pesar de que las máquinas ya nos superan en multitud de capacidades, no poseen conciencia propia ni adaptabilidad a cualquier escenario o situación.

En un texto como este, en el que me he centrado en el concepto de *algoritmo*, opino que resulta mejor práctica categorizar la IA en función de los algoritmos básicos que se han ido desarrollando a lo largo de la historia y que actúan como los ladrillos que subyacen en las diferentes aplicaciones.

No pretendo describir técnicamente cada uno de ellos, para eso existen muchos buenos libros especializados. Como ya he dicho lo que pretendo es conseguir que el lector obtenga una representación tipo atlas, de las ideas básicas que subyacen en mundo de los métodos o algoritmos propios de la inteligencia artificial agrupándolos taxonómicamente según las siguientes categorías:

- Algoritmos simbólicos:
 - Algoritmos de búsqueda.
 - Algoritmos lógicos.
 - Algoritmos probabilísticos.
- ...
- Algoritmos evolutivos:
 - Algoritmos genéticos.
- ...
- Algoritmos conexionistas:
 - Redes neuronales.
 - Redes neuronales profundas.
- ...
- Algoritmos corpóreos.

Conceptos básicos sobre limitaciones

La explosión combinatoria surge cuando el espacio de estados, es decir, el espacio que forman las posibles configuraciones de un problema puede ser enorme y al intentar explorarlo no haya ordenadores que lo puedan hacer en un tiempo razonable. Esto ocurre con mucha frecuencia en problemas reales.

Esta situación aparece cuando para resolver un problema, son necesarias las combinaciones de los valores individuales que pueden tomar sus elementos, y la

solución del problema se plantea como la búsqueda de la solución en el espacio de estados. Pero existen cuatro estrategias para limitarla o controlarla:

- Aproximaciones. Hay ocasiones que, en vez de buscar la solución exacta, nos conformamos con la búsqueda de una solución aproximada, lo cual puede reducir el problema de la explosión combinatoria.
- Heurísticas o trucos de programación. Son criterios que generalmente van bien y provocan mejoras en el rendimiento de los algoritmos, pero no hay garantías de que eso sea siempre así. Las heurísticas se justifican por resultados experimentales y suelen estar explicadas con argumentos de carácter estadístico. Se puede utilizar en formulaciones exactas y aproximadas. El programa de ajedrez de IBM Deep Blue que venció a Gary Kasparov en 1997, utilizaba chips de *hardware* dedicado que procesaban doscientos millones de posiciones por segundo para generar todos los movimientos posibles para las siguientes ocho jugadas. No obstante, tenía que usar heurísticas para seleccionar el «mejor movimiento» entre ellos. Y como sus heurísticas no eran fiables, ni siquiera Deep Blue fue capaz de vencer a Kasparov siempre.
- Poda. Esta idea consiste en eliminar zonas del espacio de búsqueda para las que podemos deducir que no contienen soluciones. La poda se suele utilizar en formulaciones exactas.
- Crear un espacio de búsqueda distinto y más pequeño, representando el problema de una forma nueva.

Recordemos lo dicho sobre la relación existente entre problemas y algoritmos:

- Un problema es decidible cuando existe un algoritmo que siempre termina con una solución o, si esta no existe, retorna porque no hay solución.
- Un problema es semidecidible cuando existe un algoritmo que termina cuando encuentra una solución, pero en caso contrario no termina.

Limitaciones en el razonamiento basado en lógica matemática

Uno de los pilares de la IA simbólica es la lógica matemática, pues se usa intensamente para representar conocimientos y razonar a partir de ellos de acuerdo con las reglas de inferencia de la lógica clásica, tanto proposicional como de primer grado.

Aunque uno de los aspectos críticos de este planteamiento es que no está nada clara la hipótesis de que el razonamiento humano puede ser modelado mediante inferencia deductiva, o incluso aproximadamente deductiva, Además, uno de los

aspectos centrales de esa controversia reside en el problema de representar conocimientos de sentido común y razonar a partir de ellos.

Esta controversia, motivó el desarrollo de nuevos paradigmas que extienden la lógica clásica, como, por ejemplo, las lógicas no monótonas, las lógicas capaces de representar las incertidumbres y la imprecisión, y las lógicas descriptivas.

Finalmente, también se propusieron otros que se apartaban completamente de la lógica matemática, como, por ejemplo, los paradigmas conexionistas, los evolutivos, los corpóreos.

De cualquier modo, el ámbito de la lógica sigue siendo uno de los campos activos dentro de la representación del conocimiento.

Algoritmos de búsqueda en grafos (modelo simbólico)

La búsqueda constituye una importante parte de la mayoría de los procesos de resolución de problemas.

El primer requisito para una búsqueda eficaz es que la colección de lugares en los que hay que buscar, denominado *espacio de búsqueda*, contenga de hecho una solución, tema que no es trivial de asegurar para muchos problemas, como, por ejemplo, la prevención de las guerras. Otro de los aspectos cruciales es que en el caso de que al menos haya una solución aceptable, esta sea detectable en un tiempo razonable y con recursos razonables.

La estructura del espacio de búsqueda debe ser tal que permita que la búsqueda se pueda realizar de forma sistemática. Es decir, es preciso ordenar los elementos del espacio de búsqueda. Algunos espacios de búsqueda, por ejemplo, los números enteros, gozan por sí mismos de una ordenación natural. Además, la búsqueda de una solución a un problema no debe consistir en una consideración a ciegas de los elementos del espacio hasta que se tropiece con la solución. De hecho, la búsqueda debe dirigirse hacia la solución mediante el examen de cada uno de los elementos que ayuden a decidir qué otros elementos hay que examinar a continuación.

Un enfoque general usado con frecuencia que es útil en muchas situaciones de resolución de problemas son dos elementos clave:

- Que el espacio de búsqueda esté estructurado, por ejemplo, que la estructura del espacio de búsqueda consista en un árbol cuyas ramas representen hipótesis o conjeturas y cuyos nudos representen algunas de las conclusiones extraídas de esas hipótesis.
- Utilización del conocimiento extraído del problema original.

El primero de los elementos mencionados ha sido objeto de gran cantidad de análisis y constituye la materia de una disciplina matemática denominada *teoría de grafos*.

El segundo de los elementos es bastante más complejo, el problema de cómo suministrar al computador un mejor conocimiento del «sentido común» que caracteriza a los procedimientos de resolución de problemas hechos por humanos, se ha convertido en un importante objeto de investigación.

Veamos los elementos de una teoría formal de la búsqueda en espacios de búsqueda abstractos y con posterioridad cómo es posible que un computador pueda utilizar conocimiento adicional para perfeccionar los procedimientos de búsqueda abstractos.

Dado un problema, formado por unos constituyentes elementales y unos operadores, a veces se puede formalizar como un problema de búsqueda: la idea básica consiste en considerar una configuración concreta del problema como un estado. Aplicando todos los operadores del problema a ese estado, se construyen todos los estados sucesores. Aplicando de nuevo todos los operadores a esos sucesores, generaremos los sucesores de los sucesores, etc. De esta manera, se construye un enorme grafo dirigido, en donde sus nodos forman el conjunto de todos los estados posibles del problema, también conocido como su espacio de estados. Este grafo es prohibitivamente grande y nunca se desarrolla en la memoria del ordenador. En ocasiones se denomina *grafo implícito*, que subraya su carácter conceptual. A lo más que se llega es a desarrollar alguna pequeña parte de él, denominado *subgrafo explícito*. En muchos casos, resolver un problema es equivalente a encontrar un camino en ese grafo dirigido, entre el nodo inicial, que representa el estado inicial del sistema, y un nodo objetivo que representa el estado deseado.

Por lo dicho, los espacios de búsqueda abstractos toman a menudo la forma de un grafo. Un grafo consiste en una colección de puntos llamados *nudos* y de conexiones llamadas *arcos* que van de un nudo a otro. Si pensamos que los nudos representan posiciones en un espacio de búsqueda, entonces los arcos son conexiones entre cada posición y sus sucesores, cualesquiera que sean las posiciones consideradas como inmediatas a medida que se recorre el espacio. Los arcos son frecuentemente de una sola vía o dirigidos. El nudo que figura en la cola de la flecha se denomina *predecesor del nudo* que figura en la cabeza de la flecha. También los arcos pueden ser bidireccionales, dependiendo del tipo de problema que se esté modelando.

Aunque un grafo puede a veces proporcionar una representación más natural de un problema, suelen ser preferibles las representaciones en árbol. De hecho, si se parte de un grafo y se indica un nudo inicial de ese grafo para utilizarlo en un caso determinado como raíz, se puede construir un árbol correspondiente.

Un árbol es un tipo especial de grafo que se caracteriza por:

- Un árbol tiene una raíz única, o nudo inicial, que no tiene predecesores.
- Cualquier otro nudo del árbol tiene exactamente un predecesor.
- Cada nudo tiene solamente un número finito de sucesores que puede variar de nudo a nudo.
- La profundidad de un árbol puede ser infinita.
- A lo sumo hay un camino o ruta capaz de conducir desde un nudo a cualquier otro nudo, y no admite bucles, caso habitual en los grafos.
- Los arcos de un árbol pueden tener costes asociados con ellos. En este caso el coste de una ruta es la suma de los costes de los arcos que hay en esa ruta.

El espacio de estados se puede explorar mediante un árbol de búsqueda —que se construye a partir del estado inicial considerando todos los estados que se pueden

alcanzar por la aplicación de un operador, de dos operadores, de tres, etc.— con las estrategias de recorrido en profundidad o en anchura. Estas estrategias se denominan de *búsqueda ciega* (fuerza bruta). En problemas no triviales el espacio de estados es muy grande y la búsqueda ciega no alcanza soluciones en tiempos razonables.

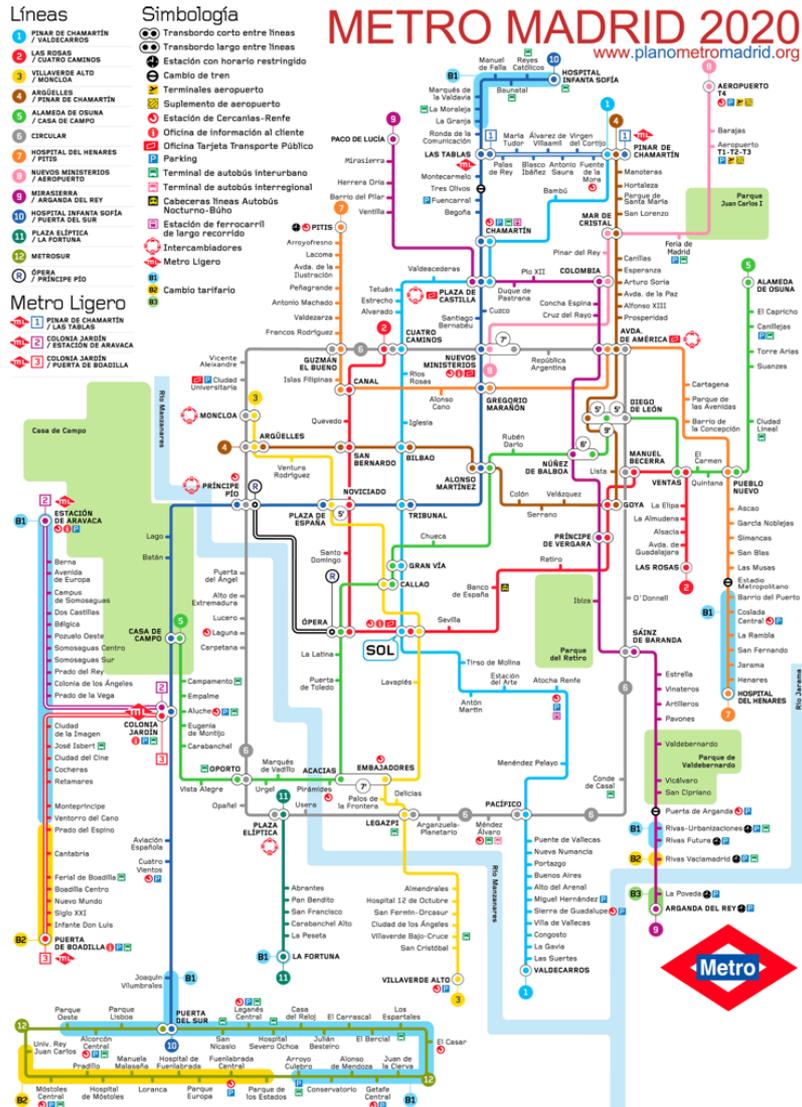
En búsqueda heurística, esta considerable dificultad se ha intentado eludir, con más o menos éxito, mediante el uso de heurísticas: se trata de criterios para la resolución de problemas que suelen dar buen resultado en la mayoría de los casos. Sin embargo, la explosión combinatoria que involucra el crecimiento exponencial del árbol de búsqueda (en cada nuevo nivel hay un número de estados que es b , factor de ramificación, multiplicado por todos los estados del nivel anterior) dificulta enormemente la aplicación de estas ideas a problemas reales.

Lo explicado hasta ahora cae dentro de una búsqueda unidireccional hacia delante. Una estrategia diferente es la búsqueda bidireccional, que requiere una especificación exacta del estado final. La idea básica es buscar hacia delante desde el nodo inicial y hacia atrás desde el nodo final. Cuando las fronteras de las dos búsquedas se encuentran, se ha resuelto el problema. El camino solución es la simple concatenación de los caminos de ambas búsquedas.

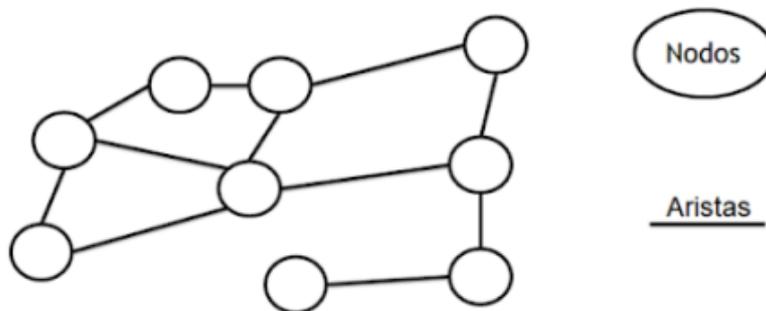
Este tipo de algoritmos se han diseñado para localizar un elemento concreto dentro de una estructura de datos. En este apartado nos centraremos en una estructura de datos concreta, los grafos. Se supone un entorno totalmente observable y acciones deterministas.

Empecemos definiendo conceptos.

- Un dato es una representación simbólica, por ejemplo, numérica, alfabética... de un atributo o variable cuantitativa o cualitativa. Los datos representan la información que un computador manipula durante la ejecución de un algoritmo.
- Una estructura de datos es una forma particular de organizar los datos para que su procesamiento sea más eficiente. Dados un conjunto de datos, se pueden organizar en diferentes tipos de estructuras, dependiendo del tipo de algoritmos que vayan a actuar sobre ellos.
- Un grafo es una forma de representar relaciones binarias entre elementos de un conjunto de datos.

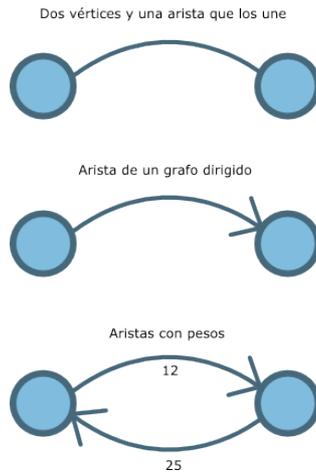


En esencia, un grafo es un conjunto de objetos llamados *vértices* o *nodos*, unidos por unos enlaces llamados *aristas* o *arcos* que permiten representar las relaciones binarias entre los elementos de un conjunto de datos.



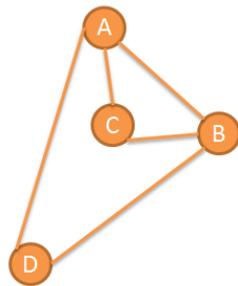
Si dos nodos son los extremos de una arista se dice que son adyacentes.

Las aristas de un grafo pueden ser

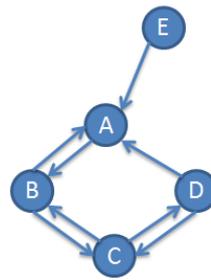


Los grafos pueden ser dirigidos cuando los extremos de los enlaces poseen un orden.

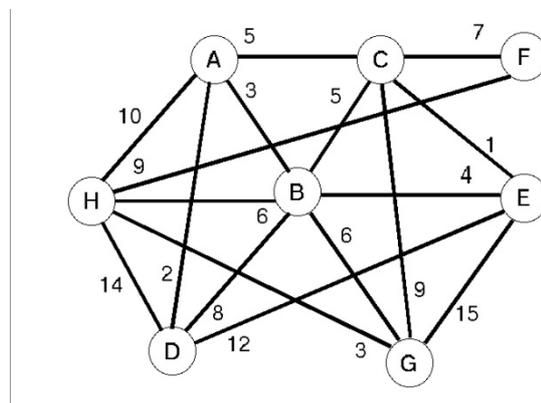
Grafo No Dirigido

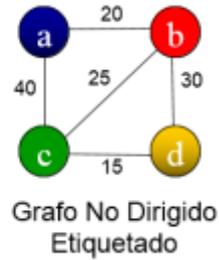
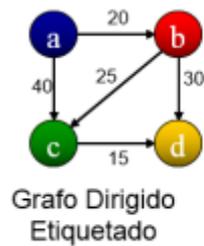


Grafo Dirigido



Los grafos pueden estar etiquetados, esto ocurre cuando cada arista tiene asociada una etiqueta simbólica.



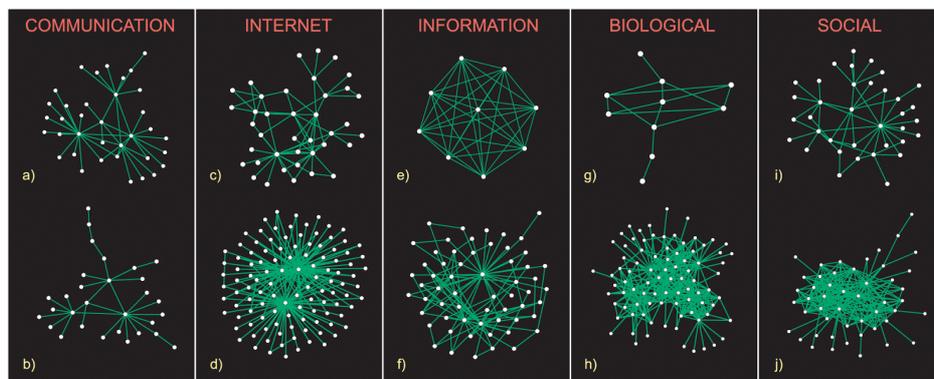


Ejemplo: Grafo de carreteras entre ciudades.



En muchas aplicaciones es necesario visitar los vértices de un grafo a partir de un nodo dado. Se denomina *camino en un grafo* a una secuencia de vértices, tal que exista una arista entre cada vértice y el siguiente.

- Prácticamente cualquier problema puede representarse mediante un grafo. En la medida en que esto es así, se pueden resolver mediante algoritmos específicos de la teoría de grafos. Su sencillez es lo que hace que puedan utilizarse para crear modelos matemáticos en temas de lo más dispares.
 - Existen varios problemas clásicos, cuyos aportes a la teoría de grafos han sido fundamentales:
 - El problema de la ciudad de Königsberg y Leonhard Euler (1707-1783).
 - El problema de los cuatro colores, planteado en 1852 por un estudiante inglés, Francis Guthrie (1831-1899). La prueba aceptada por ahora fue formulada en 1976 por Wolfgang Haken y Kenneth Appel en la Universidad de Illinois, con ayuda de un computador.
 - El problema del cartero chino, Kwan Mei-Ko (1962).



- Representar la dinámica de una red biológica:
 - Redes de regulación génica (Gene Regulatory Networks):
 - Nodos los genes.
 - Aristas las regulaciones.
 - Grafo no dirigido.
 - Redes de asociación micorbiómica (Microbiome Networks):
 - Nodos los microorganismos.
 - Aristas las asociaciones.
 - Grafo no dirigido.
 - Redes transicionales:
 - Nodos los estados.
 - Aristas las transiciones.
 - Grafo dirigido.
 - Redes metabólicas:
 - Nodos los metabolitos.
 - Aristas las reacciones.
 - Grafo dirigido.

- Epidemiología o propagación de enfermedades.
- La caracterización de las estructuras funcionales del cerebro.
- Análisis de redes sociales, redes de proveedores y clientes:

A medida que han crecido las redes sociales, ha crecido la influencia de una persona en otra para comprar, y debido a la globalización de la economía y la interconexión ha crecido el interés por estudiar qué patrones pueden descubrirse para incrementar la inteligencia del negocio.

- El impacto de una persona involucrada en una red social.
- El experimento de Stanley Milgran y la teoría de los seis grados de libertad.
- El modelo de la totalidad de la web obtuvo como patrón el hecho de que está parcializada, es decir, muy pocas páginas están altamente conectadas. La mayoría de las páginas (80 %) están enlazadas como máximo a otras cuatro.
- Google y su algoritmo PageRank, que clasifica las páginas web dándoles un orden de importancia.
- Difusión de información, opiniones o rumores.
- Al visitar una página web y hacer clic en un enlace, si escogemos dos páginas al azar: ¿En cuántos clics se puede pasar de la primera a la segunda?
- Optimización de rutas de distribución:
 - Redes eléctricas:
 - Se generan nodos con un alto número de conexiones.
 - Redes de transporte:
 - La importancia de una carretera en una red urbana.
 - Los componentes esenciales de una red de computadoras según su estudio topológico.
 - Redes de distribución logística para llevar una mercancía a su consumidor final.
 - Organizar las ventas de billetes de una compañía aérea, teniendo en cuenta las escalas, los horarios de vuelo, los enlaces.
 - Rutas óptimas a partir de redes viales reales. Dada una red de transporte, hallar la ruta óptima entre dos elementos.
 - Redes de transporte con trasbordo.
 - El cálculo del flujo máximo de agua, petróleo, gas, etc., que se puede transportar a través de cada arco de un grafo que modele un transporte.

- Redes de evacuación de grandes ciudades en caso de catástrofes causadas por tsunamis, terremotos o accidentes. Para el diseño automático de planes de evacuación y gestión del tráfico adaptado a cada ciudad y cada instante.

Un ejemplo completo

Para entender la utilidad de un grafo a la hora de representar y resolver un problema, veamos con detalle la solución del siguiente problema y su representación mediante un grafo.

Problema: Tres señores y tres criados deben cruzar el río en una barca en la que se permite que viajen una o dos personas. Solo hay un problema, los caballeros tienen motivos para sospechar que los sirvientes han conspirado entre ellos para robar a los señores y matarlos. Por lo tanto, los caballeros son conscientes de que en ningún momento puede haber más criados que señores en alguna de las orillas, puesto que los matarían. ¿Cómo podrían cruzar el río de la manera más eficiente?

Construyamos el grafo asociado a este problema. Si «c» es el número de criados que hay en la primera orilla, y «s» el de señores, entonces si un par (c, s) describe la situación en cada momento en la rivera 1, el par complementario $(3-c, 3-s)$ describe la situación en la rivera 2. Al par (c, s) se le denominará *estado*.

En total hay 16 posibles estados (4×4) en una orilla y sus 16 complementarios correspondientes en la otra. A ese conjunto finito y completo se le denominará *espacio de estados*. Véase la siguiente tabla:

	(c, s)	(c, s)	(c, s)	(c, s)
Orilla 1	(0,3)	(1,3)	(2,3)	(3,3)
Orilla 2	(3,0)	(2,0)	(1,0)	(0,0)
Orilla 1	(0,2)	(1,2)	(2,2)	(3,2)
Orilla 2	(3,0)	(2,1)	(1,1)	(0,1)
Orilla 1	(0,1)	(1,1)	(2,1)	(3,1)
Orilla 2	(3,1)	(2,2)	(1,2)	(0,2)
Orilla 1	(0,0)	(1,0)	(2,0)	(3,0)
Orilla 2	(3,3)	(2,3)	(1,3)	(0,3)

De los 32 posibles estados, hay que eliminar aquellos que no están permitidos en alguna orilla, que son todos los casos en que c es mayor que s , admitiéndose que, si así ocurre, cuando «s» es cero el estado se considera válido. Se indica en marrón el estado que descarta esa configuración.

	(c, s)	(c, s)	(c, s)	(c, s)
Orilla 1	(0,3)	(1,3)	(2,3)	(3,3)
Orilla 2	(3,0)	(2,0)	(1,0)	(0,0)
Orilla 1	(0,2)	(1,2)	(2,2)	(3,2)
Orilla 2	(3,1)	(2,1)	(1,1)	(0,1)
Orilla 1	(0,1)	(1,1)	(2,1)	(3,1)
Orilla 2	(3,1)	(2,2)	(1,2)	(0,2)
Orilla 1	(0,0)	(1,0)	(2,0)	(3,0)
Orilla 2	(3,3)	(2,3)	(1,3)	(0,3)

Las **acciones** permitidas, en cualquiera de los dos sentidos de cruce del río, son los siguientes viajes de personas:

- Acción 1: Dos criados.
- Acción 2: Dos señores.
- Acción 3: Un criado y un señor.
- Acción 4: Un criado.
- Acción 5: Un señor.

Las acciones actúan sobre un estado (c,s) y lo lleva a otro estado (c', s') , o dicho de otra manera, cada acción transita del estado (c,s) al estado (c', s')

En la siguiente descripción se van a aplicar todas las acciones sobre cada estado y se va a utilizar la siguiente nomenclatura caracterizando al estado resultante:

- Rivera 1, es la rivera de partida.
- Rivera 2, es la de enfrente.
- (c,s) posible, es que ese estado es admisible.
- (c,s) no aceptable, es que ese estado no existe.
- (c,s) sin sentido, es que viola la regla de cómo tiene que ser «c» frente a «s».

Para facilitar el entendimiento de la descripción pormenorizada que se va a realizar, vamos a hacer uso de un código de colores, cuando el par (c,s) represente un estado existente en la rivera 1, lo representaremos así (c,s) , y cuando represente un estado existente en la rivera 2, lo representaremos así (c,s) .

	(c, s)	(c, s)	(c, s)	(c, s)
Rivera 1	$(0,3)$	$(1,3)$	$(2,3)$	$(3,3)$
Rivera 2	$(3,0)$	$(2,0)$	$(1,0)$	$(0,0)$
Rivera 1	■	■	$(2,2)$	■
Rivera 2	■	■	$(1,1)$	■
Rivera 1	■	$(1,1)$	■	■
Rivera 2	■	$(2,2)$	■	■
Rivera 1	$(0,0)$	$(1,0)$	$(2,0)$	$(3,0)$
Rivera 2	$(3,3)$	$(2,3)$	$(1,3)$	$(0,3)$

Por ejemplo, si partimos de que en algún momento estamos en la situación (3,3), y aplicamos las acciones en cualquiera de los estados de las dos riveras, lo que se obtiene es:

- **Acción 1: Dos criados**
Aplicada a la rivera 1:
Se pasará del estado (3,3) posible → al estado (1,3) posible. Y quedará en la rivera 2 el estado (2,0) posible.

Aplicada a la rivera 2:
Se pasará del estado (0,0) posible → al estado (-,0) sin sentido.
- **Acción 2: Dos señores**
Aplicada a la rivera 1:
Se pasará del estado (3,3) posible → al estado (3,1) no aceptable.

Aplicada a la rivera 2:
Se pasará del estado (0,0) posible → al estado (0,-) sin sentido.
- **Acción 3: Un criado y un señor**
Aplicada a la rivera 1:
Se pasará del estado (3,3) posible → al estado (2,2) posible. Y quedará en la rivera 2 el estado (1,1).

Aplicada a la rivera 2:
Se pasará del estado (0,0) posible → al estado (-,-) sin sentido.
- **Acción 4: Un criado**
Aplicada a la rivera 1:
Se pasará del estado (3,3) posible → al estado (2,3) posible. Y quedará en la rivera 2 el estado (1,0) posible.

Aplicada a la rivera 2:
Se pasará del estado (0,0) posible → al estado (0,-) sin sentido.
- **Acción 5: Un señor**
Aplicada a la rivera 1:
Se pasará del estado (3,3) posible → al estado (3,2) no aceptable.

Aplicada a la rivera 2:
Se pasará del estado (0,0) posible → al estado (0,-) sin sentido.

Por ejemplo, si partimos de que en algún momento estamos en la situación (2,3), y aplicamos las acciones en cualquiera de los estados de las dos riveras, lo que se obtiene es:

- **Acción 1: Dos criados**
Aplicada a la rivera 1:
Se pasará del estado (2,3) posible → al estado (0,3) posible. Y quedará en la rivera 2 el estado (3,0) posible.

Aplicada a la rivera 2:
Se pasará del estado (1,0) posible → al estado (-, 0) sin sentido.
- **Acción 2: Dos señores**
Aplicada a la rivera 1:
Se pasará del estado (2,3) posible → al estado (2,1) no aceptable.

Aplicada a la rivera 2:
Se pasará del estado (1,0) posible → al estado (1,-) sin sentido.
- **Acción 3: Un criado y un señor**
Aplicada a la rivera 1:
Se pasará del estado (2,3) posible → al estado (1,2) posible. Y quedará en la rivera 2 el estado (2,1) no aceptable.

Aplicada a la rivera 2:
Se pasará del estado (1,0) posible → al estado (0,-) sin sentido.
- **Acción 4: Un criado**
Aplicada a la rivera 1:
Se pasará del estado (2,3) posible → al estado (1,3) posible. Y quedará en la rivera 2 el estado (2,0) posible.

Aplicada a la rivera 2:
Se pasará del estado (1,0) posible → al estado (0,0) posible. Y quedará en la rivera 1 el estado (3,3) posible.
- **Acción 5: Un señor**
Aplicada a la rivera 1:
Se pasará del estado (2,3) posible → al estado (2,2) posible. Y quedará en la rivera 2 el estado (1,1) posible.

Aplicada a la rivera 2:
Se pasará del estado (1,0) posible → al estado (1,-) sin sentido.

Por ejemplo, si partimos de que en algún momento estamos en la situación (1,3), y aplicamos las acciones en cualquiera de los estados de las dos riveras, lo que se obtiene es:

- **Acción 1: Dos criados**
Aplicada a la rivera 1:
Se pasará del estado (1,3) posible → al estado (-,3) sin sentido.

Aplicada a la rivera 2:
Se pasará del estado (2,0) posible → al estado (0,0) posible. Y quedará en la rivera 1 el estado (3,3) posible.
- **Acción 2: Dos señores**
Aplicada a la rivera 1:
Se pasará del estado (1,3) posible → al estado (1,1) posible. Y quedará en la rivera 2 el estado (2,2) posible.

Aplicada a la rivera 2:
Se pasará del estado (2,0) posible → al estado (2,-) sin sentido.
- **Acción 3: Un criado y un señor**
Aplicada a la rivera 1:
Se pasará del estado (1,3) posible → al estado (0,2) posible. Y quedará en la rivera 2 el estado (3,1) no aceptable.

Aplicada a la rivera 2:
Se pasará del estado (2,0) posible → al estado (1,-) sin sentido.
- **Acción 4: Un criado**
Aplicada a la rivera 1:
Se pasará del estado (1,3) posible → al estado (0,3) posible. Y quedará en la rivera 2 el estado (3,0) posible.

Aplicada a la rivera 2:
Se pasará del estado (2,0) posible → al estado (1,0) posible. Y quedará en la rivera 1 el estado (2,3) posible.
- **Acción 5: Un señor**
Aplicada a la rivera 1:
Se pasará del estado (1,3) posible → al estado (1,2) posible. Y quedará en la rivera 2 el estado (2,1) no aceptable.

Aplicada a la rivera 2:
Se pasará del estado (2,0) posible → al estado (1,-) sin sentido.

Por ejemplo, si partimos de que en algún momento estamos en la situación (0,3), y aplicamos las acciones en cualquiera de los estados de las dos riveras, lo que se obtiene es:

- **Acción 1: Dos criados**
Aplicada a la rivera 1:
Se pasará del estado (0,3) posible → al estado (-,3) sin sentido.

Aplicada a la rivera 2:
Se pasará del estado (3,0) posible → al estado (1,0) posible. Y quedará en la rivera 1 el estado (2,3) posible.
- **Acción 2: Dos señores**
Aplicada a la rivera 1:
Se pasará del estado (0,3) posible → al estado (0,1) posible. Y quedará en la rivera 2 el estado (3,2) no aceptable.

Aplicada a la rivera 2:
Se pasará del estado (3,0) posible → al estado (3,-) sin sentido.
- **Acción 3: Un criado y un señor**
Aplicada a la rivera 1:
Se pasará del estado (0,3) posible → al estado (-,2) sin sentido.

Aplicada a la rivera 2:
Se pasará del estado (3,0) posible → al estado (2,-) sin sentido.
- **Acción 4: Un criado**
Aplicada a la rivera 1:
Se pasará del estado (0,3) posible → al estado (-,3) sin sentido.

Aplicada a la rivera 2:
Se pasará del estado (3,0) posible → al estado (2,0) posible. Y quedará en la rivera 1 el estado (1,3) posible.
- **Acción 5: Un señor**
Aplicada a la rivera 1:
Se pasará del estado (0,3) posible → al estado (0,2) posible. Y quedará en la rivera 2 el estado (3,1) no aceptable.

Aplicada a la rivera 2:
Se pasará del estado (3,0) posible → al estado (3,-) sin sentido.

Por ejemplo, si partimos de que en algún momento estamos en la situación (2,2), y aplicamos las acciones en cualquiera de los estados de las dos riveras, lo que se obtiene es:

- **Acción 1: Dos criados**
Aplicada a la rivera 1:
Se pasará del estado (2,2) posible → al estado (0,2) posible. Y quedará en la rivera 2 el estado (3,1) no aceptable.

Aplicada a la rivera 2:
Se pasará del estado (1,1) posible → al estado (-,1) sin sentido.
- **Acción 2: Dos señores**
Aplicada a la rivera 1:
Se pasará del estado (2,2) posible → al estado (2,0) posible. Y quedará en la rivera 2 el estado (1,3) aceptable.

Aplicada a la rivera 2:
Se pasará del estado (1,1) posible → al estado (1,-) sin sentido.
- **Acción 3: Un criado y un señor**
Aplicada a la rivera 1:
Se pasará del estado (2,2) posible → al estado (1,1) posible. Y quedará en la rivera 2 el estado (2,2) posible.

Aplicada a la rivera 2:
Se pasará del estado (1,1) posible → al estado (0,0) posible. Y quedará en la rivera 1 el estado (3,3) posible.
- **Acción 4: Un criado**
Aplicada a la rivera 1:
Se pasará del estado (2,2) posible → al estado (1,2) posible. Y quedará en la rivera 2 el estado (2,1) no aceptable.

Aplicada a la rivera 2:
Se pasará del estado (1,1) posible → al estado (0,1) posible. Y quedará en la rivera 1 el estado (3,2) no aceptable.
- **Acción 5: Un señor**
Aplicada a la rivera 1:
Se pasará del estado (2,2) posible → al estado (2,1) no aceptable.

Aplicada a la rivera 2:
Se pasará del estado (1,1) posible → al estado (1,0) no aceptable.

Por ejemplo, si partimos de que en algún momento estamos en la situación (1,1), y aplicamos las acciones en cualquiera de los estados de las dos riveras, lo que se obtiene es:

- **Acción 1: Dos criados**
Aplicada a la rivera 1:
Se pasará del estado (1,1) posible → al estado (-,1) no aceptable.

Aplicada a la rivera 2:
Se pasará del estado (2,2) posible → al estado (0,2) posible. Y quedará en la rivera 1 el estado (3,1) no aceptable.
- **Acción 2: Dos señores**
Aplicada a la rivera 1:
Se pasará del estado (1,1) posible → al estado (1,-) sin sentido.

Aplicada a la rivera 2:
Se pasará del estado (2,2) posible → al estado (2,0) posible. Y quedará la rivera 1 en el estado (1,3) posible.
- **Acción 3: Un criado y un señor**
Aplicada a la rivera 1:
Se pasará del estado (1,1) posible → al estado (0,0) posible. Y quedará la rivera 2 en el estado (3,3) posible.

Aplicada a la rivera 2:
Se pasará del estado (2,2) posible → al estado (1,1) posible. Y quedará en la rivera 1 el estado (2,2) posible.
- **Acción 4: Un criado**
Aplicada a la rivera 1:
Se pasará del estado (1,1) posible → al estado (0,1) posible. Y quedará la rivera 2 en el estado (3,2) no aceptable.

Aplicada a la rivera 2:
Se pasará del estado (2,2) posible → al estado (1,2) posible. Y quedará en la rivera 2 el estado (2,1) no aceptable.
- **Acción 5: Un señor**
Aplicada a la rivera 1:
Se pasará del estado (1,1) posible → al estado (1,0) aceptable. Y quedará en la rivera 2 el estado (2,3) posible.

Aplicada a la rivera 2:
Se pasará del estado (2,2) posible → al estado (2,1) no aceptable.

Por ejemplo, si partimos de que en algún momento estamos en la situación (3,0), y aplicamos las acciones en cualquiera de los estados de las dos riveras, lo que se obtiene es:

- **Acción 1: Dos criados**
Aplicada a la rivera 1:
Se pasará del estado (3,0) posible → al estado (1,0) posible. Y quedará en la rivera 2 el estado (2,3) posible.

Aplicada a la rivera 2:
Se pasará del estado (0,3) posible → al estado (-,3) sin sentido.
- **Acción 2: Dos señores**
Aplicada a la rivera 1:
Se pasará del estado (3,0) posible → al estado (3,-) sin sentido.

Aplicada a la rivera 2:
Se pasará del estado (0,3) posible → al estado (0,1) posible. Y quedará la rivera 1 en el estado (3,2) no aceptable.
- **Acción 3: Un criado y un señor**
Aplicada a la rivera 1:
Se pasará del estado (3,0) posible → al estado (2,-) sin sentido.

Aplicada a la rivera 2:
Se pasará del estado (0,3) posible → al estado (-,2) sin sentido.
- **Acción 4: Un criado**
Aplicada a la rivera 1:
Se pasará del estado (3,0) posible → al estado (2,0) posible. Y quedará en la rivera 2 el estado (1,3) posible.

Aplicada a la rivera 2:
Se pasará del estado (0,3) posible → al estado (-,3) sin sentido.
- **Acción 5: Un señor**
Aplicada a la rivera 1:
Se pasará del estado (3,0) posible → al estado (3,-) sin sentido.

Aplicada a la rivera 2:
Se pasará del estado (0,3) posible → al estado (0,2) posible. Y quedará la rivera 1 en el estado (3,1) no aceptable.

Por ejemplo, si partimos de que en algún momento estamos en la situación (2,0), y aplicamos las acciones en cualquiera de los estados de las dos riveras, lo que se obtiene es:

- **Acción 1: Dos criados**
Aplicada a la rivera 1:
Se pasará del estado (2,0) posible → al estado (0,0) posible. Y quedará en la rivera 2 el estado (3,3) posible.

Aplicada a la rivera 2:
Se pasará del estado (1,3) posible → al estado (-,3) sin sentido.
- **Acción 2: Dos señores**
Aplicada a la rivera 1:
Se pasará del estado (2,0) posible → al estado (2,-) sin sentido.

Aplicada a la rivera 2:
Se pasará del estado (1,3) posible → al estado (1,1) posible. Y quedará en la rivera 1 el estado (2,2) posible.
- **Acción 3: Un criado y un señor**
Aplicada a la rivera 1:
Se pasará del estado (2,0) posible → al estado (1,-) sin sentido.

Aplicada a la rivera 2:
Se pasará del estado (1,3) posible → al estado (0,2) posible. Y quedará en la rivera 1 el estado (3,1) no aceptable.
- **Acción 4: Un criado**
Aplicada a la rivera 1:
Se pasará del estado (2,0) posible → al estado (1,0) posible. Y quedará en la rivera 2 el estado (2,3) posible.

Aplicada a la rivera 2:
Se pasará del estado (1,3) posible → al estado (0,3) posible. Y quedará en la rivera 1 el estado (3,0) posible.
- **Acción 5: Un señor**
Aplicada a la rivera 1:
Se pasará del estado (2,0) posible → al estado (2,-) sin sentido.

Aplicada a la rivera 2:
Se pasará del estado (1,3) posible → al estado (1,2) posible. Y quedará en la rivera 1 el estado (2,1) no aceptable.

Por ejemplo, si partimos de que en algún momento estamos en la situación (1,0), y aplicamos las acciones en cualquiera de los estados de las dos riveras, lo que se obtiene es:

- **Acción 1: Dos criados**
Aplicada a la rivera 1:
Se pasará del estado (1,0) posible → al estado (-,0) sin sentido.

Aplicada a la rivera 2:
Se pasará del estado (2,3) posible → al estado (0,3) posible. Y quedará en la rivera 1 el estado (3,0) posible.
- **Acción 2: Dos señores**
Aplicada a la rivera 1:
Se pasará del estado (1,0) posible → al estado (1,-) sin sentido.

Aplicada a la rivera 2:
Se pasará del estado (2,3) posible → al estado (2,1) no posible.
- **Acción 3: Un criado y un señor**
Aplicada a la rivera 1:
Se pasará del estado (1,0) posible → al estado (1,-) sin sentido.

Aplicada a la rivera 2:
Se pasará del estado (2,3) posible → al estado (1,2) posible. Y quedará en la rivera 1 el estado (2,1) no aceptable.
- **Acción 4: Un criado**
Aplicada a la rivera 1:
Se pasará del estado (1,0) posible → al estado (0,0) posible. Y quedará en la rivera 2 el estado (3,3) posible.

Aplicada a la rivera 2:
Se pasará del estado (2,3) posible → al estado (1,3) posible. Y quedará en la rivera 1 el estado (2,0) posible.
- **Acción 5: Un señor**
Aplicada a la rivera 1:
Se pasará del estado (1,0) posible → al estado (1,-) sin sentido.

Aplicada a la rivera 2:
Se pasará del estado (2,3) posible → al estado (2,2) posible. Y quedará en la rivera 1 el estado (1,1) posible.

Por ejemplo, si partimos de que en algún momento estamos en la situación $(0,0)$, y aplicamos las acciones en cualquiera de los estados de las dos riveras, lo que se obtiene es:

- **Acción 1: Dos criados**
Aplicada a la rivera 1:
Se pasará del estado $(0,0)$ posible \rightarrow al estado $(-,0)$ sin sentido.

Aplicada a la rivera 2:
Se pasará del estado $(3,3)$ posible \rightarrow al estado $(1,3)$ posible. Y quedará en la rivera 1 el estado $(2,0)$ posible.
- **Acción 2: Dos señores**
Aplicada a la rivera 1:
Se pasará del estado $(0,0)$ posible \rightarrow al estado $(0,-)$ sin sentido.

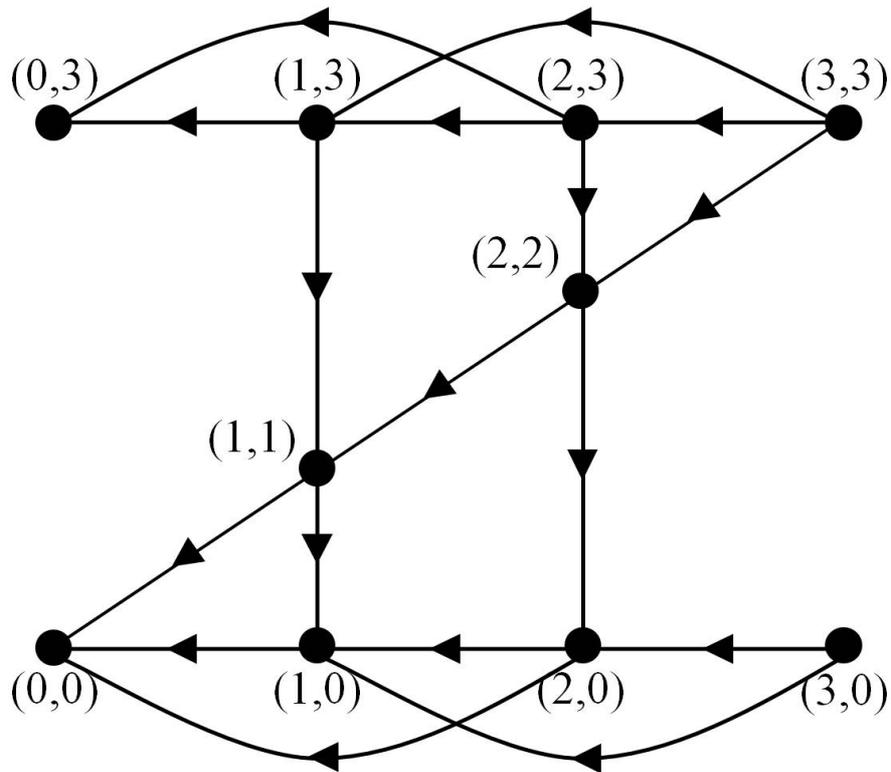
Aplicada a la rivera 2:
Se pasará del estado $(3,3)$ posible \rightarrow al estado $(3,1)$ no posible.
- **Acción 3: Un criado y un señor**
Aplicada a la rivera 1:
Se pasará del estado $(0,0)$ posible \rightarrow al estado $(-,-)$ sin sentido.

Aplicada a la rivera 2:
Se pasará del estado $(3,3)$ posible \rightarrow al estado $(2,2)$ posible. Y quedará en la rivera 1 el estado $(1,1)$ posible.
- **Acción 4: Un criado**
Aplicada a la rivera 1:
Se pasará del estado $(0,0)$ posible \rightarrow al estado $(-,0)$ sin sentido.

Aplicada a la rivera 2:
Se pasará del estado $(3,3)$ posible \rightarrow al estado $(2,3)$ posible. Y quedará en la rivera 1 el estado $(1,0)$ posible.
- **Acción 5: Un señor**
Aplicada a la rivera 1:
Se pasará del estado $(0,0)$ posible \rightarrow al estado $(0,-)$ sin sentido.

Aplicada a la rivera 2:
Se pasará del estado $(3,3)$ posible \rightarrow al estado $(3,2)$ posible. Y quedará en la rivera 1 el estado $(0,1)$ posible.

Si se aplican las acciones a cada uno de los estados de la rivera 1 representados en la matriz con colores en rojo, y se dibujan los resultados válidos de esas acciones, lo que se obtiene es el siguiente grafo dirigido que representa el problema de los tres criados y tres señores.



Ahora para resolver el problema tenemos que partir de la situación inicial representada por el estado $(3,3)$ en la rivera 1 y alcanzar la situación final representada por el estado $(0,0)$ en la misma rivera. Para encontrar la solución, lo que falta es introducir en el grafo que el bote va pasando de una orilla a la otra, obsérvese que en la rivera 1 cuando se va el bote siempre hay decrecimiento de personas, pero cuando vuelve el bote siempre hay crecimiento.

Ahora construyamos visualmente el grafo solución del problema uniendo los vértices (c,s) con una arista continua y dirigida, interpretando que la acción que representa es cruzar el río, de la primera orilla a la segunda. Y se unirán los vértices (c,s) con una arista a trazos y dirigida, interpretando que la acción que representa es cruzar el río, de la segunda orilla a la primera.

Observando el grafo anterior y si se empieza en la rivera 1 del río en el estado $(3,3)$, entonces son viables como estado final del **primer** cruce en la misma rivera el $(1,3)$, el $(2,3)$ y el $(2,2)$, de ellos que cumplan con la disminución del número de personas son los tres.

Fíjense que el desarrollo puede seguirse o con el estado $(1,2)$ o con el estado $(2,3)$ o con el estado $(2,2)$. Seleccionemos uno de ellos y cuando se finalice con este recorrido que nos puede conducir o no a una solución, habría que volver a este punto y seleccionar alguno de los otros para buscar otra posible solución. Una vez finalizado el nuevo recorrido, habría que volver de nuevo a este punto para seleccionar la última de las posibilidades.

Supongamos que se selecciona el estado $(1,3)$, ahora la barca cruza en sentido inverso, entonces estados viables como estado final del **segundo** cruce serían el $(0,3)$, el $(1,1)$, el $(2,3)$ y el $(3,3)$ y como tiene que producirse un crecimiento del número de personas, entonces el único estado accesible sería el $(2,3)$.

Sigamos con el $(2,3)$, ahora la barca va en sentido directo y el número de personas tiene que disminuir, los estados viables como estado final del **tercer** cruce serían el $(0,3)$, el $(1,3)$, el $(2,2)$ y el $(3,3)$. Entonces los estados accesible serían el $(2,2)$ y el $(0,3)$.

Sigamos con el $(2,2)$, ahora la barca va en sentido inverso y el número de personas tiene que crecer, entonces los estados viables como estado final del cuarto cruce serían el $(2,0)$ y el $(1,1)$, pero ninguno es válido. Entonces hay que seguir con el $(0,3)$, en este caso los estados viables como estado final del **cuarto** cruce serían el $(1,3)$ y el $(2,3)$. En este caso el único estado accesible es el $(1,3)$.

Sigamos con el $(1,3)$, ahora la barca va en sentido directo y el número de personas tiene que decrecer, los estados viables como estado final del **quinto** cruce serían $(0,3)$, el $(2,3)$, el $(1,1)$ y el $(3,3)$. Entonces el único estado accesible es el $(1,1)$.

Sigamos con el $(1,1)$ ahora la barca va en sentido inverso y el número de personas tiene que crecer, los estados viables como estado final del **sexto** cruce serían el $(1,3)$, $(1,0)$, $(0,0)$ y el $(2,2)$. Entonces el único estado accesible sería el $(2,2)$.

Sigamos con el $(2,2)$ ahora la barca va en sentido directo y el número de personas tiene que decrecer, los estados viables como estado final del **séptimo** cruce serían $(2,3)$, $(3,3)$, $(1,1)$, $(2,0)$. Entonces el único estado accesible sería el $(2,0)$.

Sigamos con el $(2,0)$, ahora la barca va en sentido inverso y el número de personas tiene que crecer, los estados viables como estado final del **octavo** cruce serían el $(3,0)$, $(2,2)$, $(1,0)$, $(0,0)$. Entonces el único estado accesible sería el $(3,0)$.

Sigamos con el $(3,0)$, ahora la barca va en sentido directo y el número de personas tiene que decrecer, los estados viables como estado final del **noveno** cruce serían el $(2,0)$ y el $(1,0)$. Entonces el único estado accesible sería el $(1,0)$.

Sigamos con el $(1,0)$, ahora la barca va en sentido inverso y el número de personas tiene que crecer, los estados viables como estado final del **décimo** cruce serían el $(2,0)$, el $(0,0)$ y el $(1,1)$. Entonces los únicos estados visitables serían el $(2,0)$ y el $(1,1)$.

Fíjense que el desarrollo puede seguirse o con el estado $(2,0)$ o con el estado $(1,1)$. Seleccionemos uno de ellos y cuando se finalice con este recorrido que nos puede conducir o no a una solución, habría que volver a este punto y seleccionar el otros para buscar otra posible solución.

Sigamos al $(2,0)$ ahora la barca va en sentido directo y el número de personas tiene que decrecer, los estados viables como estado final del **undécimo** cruce serían el $(2,2)$, el $(3,0)$, el $(1,0)$ y el $(0,0)$. Entonces el único estado visitable sería el $(0,0)$.

¡Y hemos alcanzado el objetivo!

La solución encontrada es que se consigue en 11 cruces

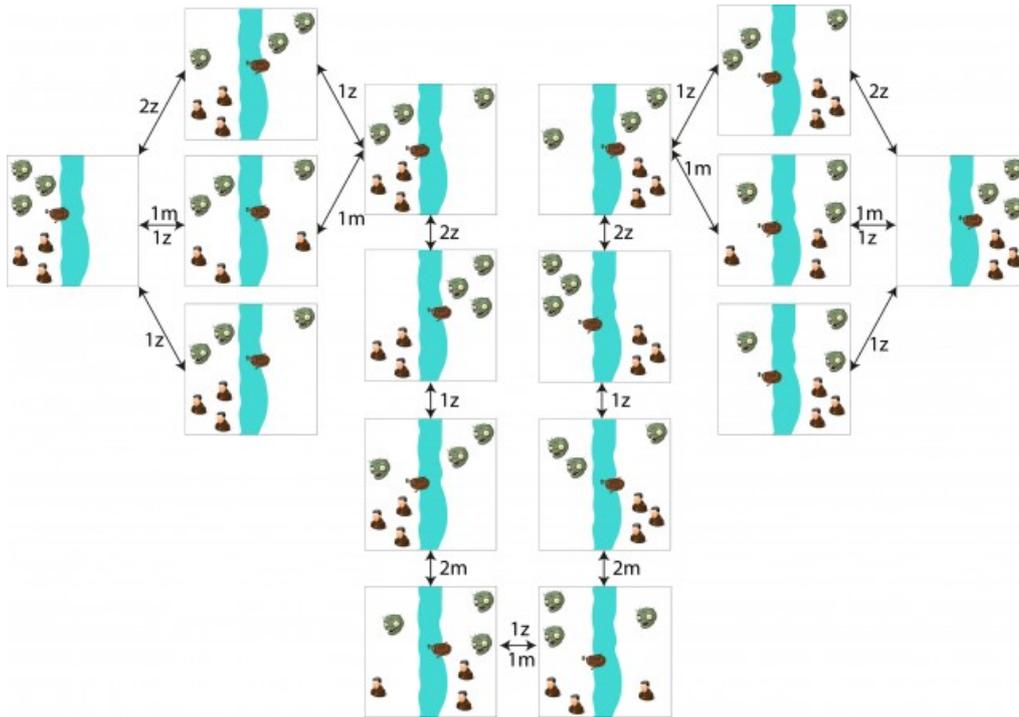
$(3,3)$, $(1,3)$, $(2,3)$, $(0,3)$, $(1,3)$, $(1,1)$, $(2,2)$, $(2,0)$, $(3,0)$, $(1,0)$, $(2,0)$, $(0,0)$

Existen otras tres soluciones,

$(3,3)$, $(2,2)$, $(2,3)$, $(0,3)$, $(1,3)$, $(1,1)$, $(2,2)$, $(2,0)$, $(3,0)$, $(1,0)$, $(2,0)$, $(0,0)$

$(3,3)$, $(1,3)$, $(2,3)$, $(0,3)$, $(1,3)$, $(1,1)$, $(2,2)$, $(2,0)$, $(3,0)$, $(1,0)$, $(1,1)$, $(0,0)$

$(3,3)$, $(2,2)$, $(2,3)$, $(0,3)$, $(1,3)$, $(1,1)$, $(2,2)$, $(2,0)$, $(3,0)$, $(1,0)$, $(1,1)$, $(0,0)$

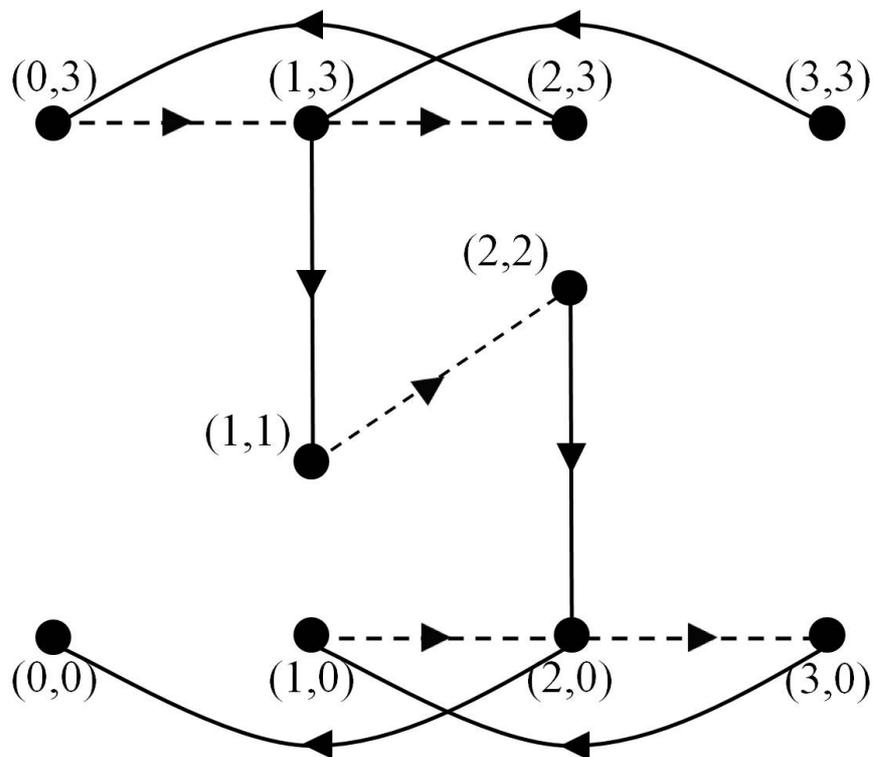


Gráfica de Guillermo Peris Ripollés

<<https://naukas.com/2015/10/05/maridos-celosos-misioneros-zombis/>>

El grafo dirigido de la solución encontrada que se consigue en 11 pasos es

$(3,3)$, $(1,3)$, $(2,3)$, $(0,3)$, $(1,3)$, $(1,1)$, $(2,2)$, $(2,0)$, $(3,0)$, $(1,0)$, $(2,0)$, $(0,0)$



Al estar representada cada solución como un par de números y utilizar un grafo es sencillo resolver el problema mediante un programa informático general que trabaje con cualquier número de personas. Por ejemplo, ¿cuál es el número mínimo de viajes si hay 4 zombis y 4 misioneros? ¿Y si dejamos que en la barca viajen tres individuos? Por curiosidad vea algunos resultados:

- 4 misioneros, 4 zombis, bote con 2 individuos: no hay solución.
- 4 misioneros, 4 zombis, bote con 3 individuos: 9 viajes.
- 5 misioneros, 5 zombis, bote con 3 individuos: 11 viajes.
- 6 misioneros, 6 zombis, bote con 3 individuos: no hay solución.

Por Guillermo Peris Ripollés, el 5 octubre, 2015.

<https://naukas.com/2015/10/05/maridos-celosos-misioneros-zombis/>

Reflexión

Al parecer existe un conjunto de problemas que se caracterizan por estar definidos en un entorno observable, discreto, conocido y determinístico. En estos casos la solución es una secuencia fija de acciones que se obtienen mediante un proceso de búsqueda. Mientras se ejecuta una acción, se supone que el entorno no cambia.

El problema queda completamente definido por los siguientes componentes:

- Definición de qué es un **estado**.
- Definición del **espacio de estados**.
- Definición del **conjunto de acciones** posibles y lo que hacen, a esto se le denomina **modelo de transición**.
- Representación del problema mediante un **grafo** o un (**árbol**).
- Definición del **estado inicial**.
- Definición del **objetivo a alcanzar (o estado solución)**.

La solución del problema es intentar encontrar un **camino** en el espacio de estados que permita alcanzar el objetivo propuesto, a este proceso se le denomina **búsqueda de la solución**. El algoritmo que se diseñe tendrá un coste temporal y de almacenamiento de información, y se considera que la solución obtenida será óptima cuando el coste sea el menor de todos los posibles.

Algunas situaciones que podemos encontrar que dependen del tipo de solución son:

- Que el estado solución no esté en la estructura, por lo tanto, no existe solución.
- Que el estado solución sea único.
- Que el estado solución sea múltiple (debe existir algún criterio para seleccionar uno de ellos).

Con objeto de poder hablar de complejidad de cada algoritmo asociado a un grafo, caractericémoslo del modo siguiente:

- b , factor de ramificación o máximo número de sucesores de cualquier nodo.
- d , la profundidad del nodo objetivo más superficial.
- m , la máxima profundidad del árbol de búsqueda.
- p , es el límite de profundidad.
- C , función coste, es un valor numérico asignado a cada arco.

Desde un punto de vista teórico, la evaluación completa de un algoritmo de búsqueda debería saber responder a las siguientes preguntas:

- ¿Se puede encontrar solución?
- Si la hay, ¿se puede encontrar una solución óptima?

- Si la hay, ¿cuánto tiempo va a requerir? Y ¿cuánta memoria será necesaria?

Existen diferentes estrategias de búsqueda que se suelen agrupar en dos grandes bloques:

- Búsquedas ciegas o no informadas.
- Búsquedas informadas mediante heurísticas.

Algoritmos de búsquedas no informadas o búsquedas a ciegas

Estos algoritmos solo cuentan con la información dada en la definición del problema. El proceso de búsqueda es sistemático pasando por cada nodo del grafo y comprobando si alguno de ellos es la solución del problema. Los algoritmos se diferencian en cómo seleccionan cuál es el siguiente nodo a visitar. Los más referenciados en la literatura son los siguientes:

- *Búsqueda en anchura*, se extraen y se analizan todos los nodos que surgen del nodo raíz denominados *sus sucesores*, a continuación, y para cada sucesor, se extraen y se analizan todos sus nodos sucesores, y se sigue así mientras siga habiendo nodos o se encuentre una solución.

Sus propiedades son (en azul las ventajas, en rojo las desventajas):

- **Completo**: si b finito entonces sí. Si existe solución la encuentra.
- **Óptimo**: sí, si los costes son iguales, ya que encuentra la solución más superficial que es la que menos coste requiere.
- **Complejidad temporal**: exponencial $O(b^{d+1})$.
- **Complejidad espacial**: exponencial $O(b^{d+1})$.

Recomendable para problemas muy pequeños:

- *Búsqueda de coste uniforme*, se extrae y se analiza el nodo con el coste más pequeño que surge del nodo raíz denominado su *sucesor*, a continuación, se extrae y se analiza de todos sus nodos sucesores aquel con el coste acumulado más pequeño, y se sigue así mientras siga habiendo nodos o se encuentre una solución. En este caso si se supone que C^* es el coste de la solución óptima y que cada acción cuesta al menos ϵ .

Sus propiedades son (en azul las ventajas, en rojo las desventajas):

- **Completo**: si b finito, y $\epsilon > 0$ entonces sí. Si existe solución la encuentra.
- **Óptimo**: sí, porque obtiene el camino con el coste menor.
- **Complejidad temporal**: exponencial y superior a $O(b^{\lceil C^*/\epsilon \rceil})$.
- **Complejidad espacial**: exponencial y superior a $O(b^{\lceil C^*/\epsilon \rceil})$.

Recomendable para problemas muy pequeños:

- *Búsqueda primero en profundidad*, se extraen y se analiza el primero de los nodos que surgen del nodo raíz denominado su *sucesor*, a continuación, se extrae y se analiza el primero de los nodos del sucesor, y se sigue así mientras siga habiendo nodos o se encuentre una solución. Cuando ya no quedan más nodos que visitar en dicho camino, regresa hacia atrás, de modo que repite el mismo proceso con cada uno de los hermanos del nodo ya procesado.

Sus propiedades son (en azul las ventajas, en rojo las desventajas):

- **Completo**: no.
- **Óptimo**: no, puede encontrar otras soluciones antes que la óptima.
- **Complejidad temporal**: exponencial $O(b^m)$, m suele ser mayor que d .
- **Complejidad espacial**: lineal $O(b \times m)$.

Recomendable para problemas muy pequeños:

- *Búsqueda en profundidad limitada*. Primero se establece una profundidad máxima (p) a la que se está dispuesto a llegar, posteriormente aplica el algoritmo *primero en profundidad*. Cuando se llega a la profundidad máxima permitida, si no hay solución, se retrocede.

Sus propiedades son (en azul las ventajas, en rojo las desventajas):

- **Completo**: dependiendo de p :
 - Si $p \geq d$ es **completo**.
 - Si $p < d$ no es **completo**, aunque exista solución si está más allá del límite marcado no puede encontrarla.
- **Óptimo**:
 - Si $l < d$ no es **óptimo**.
 - Si $l > d$ la solución no tiene por qué ser la **óptima**.
 - Si $l = d$ encontramos la solución de camino más corto, pero no tiene por qué ser la de menor coste.
- **Complejidad temporal**: exponencial $O(b^p)$.
- **Complejidad espacial**: lineal $O(b \times p)$.

Recomendable para problemas muy pequeños:

- *Búsqueda en profundidad iterativa*. Primero se aplica el algoritmo de *búsqueda en profundidad limitada* con $p=0$, y se analiza lo que ocurre. Si no se encuentra solución, se vuelve a aplicar el algoritmo de *búsqueda en profundidad limitada* con $p=1$, y se analiza lo que ocurre. Si no se encuentra solución, se vuelve a aplicar el algoritmo de *búsqueda en profundidad limitada* con $p=2$, y así sucesivamente hasta encontrar una solución o finalizar cuando se llega a la profundidad máxima que se permita. Si no hay solución, se retrocede.

Sus propiedades son (en azul las ventajas, en rojo las desventajas):

- **Completo**: sí, si b es finita, aunque si existe solución y está más allá del límite marcado no puede encontrarla.
- **Óptimo**: sí, si los costes son iguales, ya que, si encuentra una solución, es la más superficial que es la que menos costes requiere.

- **Complejidad temporal:** exponencial $O(b^d)$.
- **Complejidad espacial:** lineal $O(b \times p)$.

Se considera como un buen algoritmo de búsqueda no informada, sobre todo si el espacio de búsqueda es grande y no se conoce a qué profundidad está la solución.

- *Búsqueda bidireccional.* Requiere que se conozca la solución (el estado final). Se basa en el algoritmo *primero en anchura*. Se realizan dos búsquedas, una desde el estado inicial y otra desde el estado final. Para ello las acciones deben ser reversibles. Lo que se pretende es encontrar un estado intermedio común a las dos búsquedas. En ese momento se ha encontrado un camino que permite ir del estado inicial al estado final.

Sus propiedades son (en azul las ventajas, en rojo las desventajas):

- **Completo:** sí, si hay solución, la encuentra.
- **Óptimo:** no, no se puede garantizar que la solución encontrada sea la mejor, salvo que los costes sean iguales.
- **Complejidad temporal:** exponencial $O(b^{d/2})$.
- **Complejidad espacial:** exponencial $O(b^{d/2})$.

Se considera como un buen algoritmo de búsqueda no informada, si se cumplen los requisitos necesarios.

Para hacerse una idea de las complejidades que pueden surgir en este tipo de algoritmos, veamos la siguiente tabla de valores de b^{d+1} suponiendo que $b=10$ y diferentes valores de profundidad d a la que se ha encontrado la solución. Se ha supuesto que se pueden generar 10 000 nodos por segundo y que un nodo requiere 1000 bytes para almacenarlo.

Profundidad	Nodos	Tiempo	Memoria
2	1100	11 s	1 MB
4	111 100	11 s	106 MB
6	10^7	19 m	10 GB
8	10^9	31 h	1 TB
10	10^{11}	129 días	101 TB
12	10^{13}	35 años	10 PB
14	10^{15}	3523 años	1 EB

Tabla de unidades de medida utilizada.

		Factor binario
Bytes	B	$2^0 = 1$
KiloBytes	Kb	$2^{10} = 1024$
MegaBytes	Mb	$2^{20} = 1\,048\,576$
GigaBytes	Gb	$2^{30} = 1\,073\,741\,824$
TeraBytes	Tb	$2^{40} = 1\,099\,511\,627\,776$
PetaBytes	Pb	$2^{50} = 1\,125\,899\,906\,842\,624$
ExaBytes	Eb	$2^{60} = 1\,152\,921\,504\,606\,846\,976$
ZettaBytes	Zb	$2^{70} = 118\,059\,162\,071\,741\,130\,342\,4$
YottaBytes	Yb	$2^{80} = 1208\,925\,8196\,146\,291\,747\,061\,76$

Algoritmos de búsquedas no informadas con información parcial

Recordemos:

- Un entorno es totalmente observable si se tiene conocimiento total sobre el problema, las opciones de solución que se planteen van a causar siempre resultados conocidos e invariables. Al tomar la decisión solo se debe pensar en la opción que genere mayor beneficio.
- Cuando se actúa en un entorno determinista, se sabe que siempre que se realiza la misma acción, se obtiene el mismo resultado.

Hasta ahora hemos supuesto que el entorno ha sido totalmente observable y determinista y que el agente conoce cuáles son los efectos de cada acción. Por lo tanto, el agente puede calcular exactamente cuál es el resultado de cualquier secuencia de acciones y siempre sabe en qué estado está.

Recordemos:

- En un entorno parcialmente observable, la información con la que se cuenta para solucionar el problema es completa, es decir, se conoce el problema, se conocen las posibles soluciones, pero no se conoce con certeza los resultados que pueden arrojar.

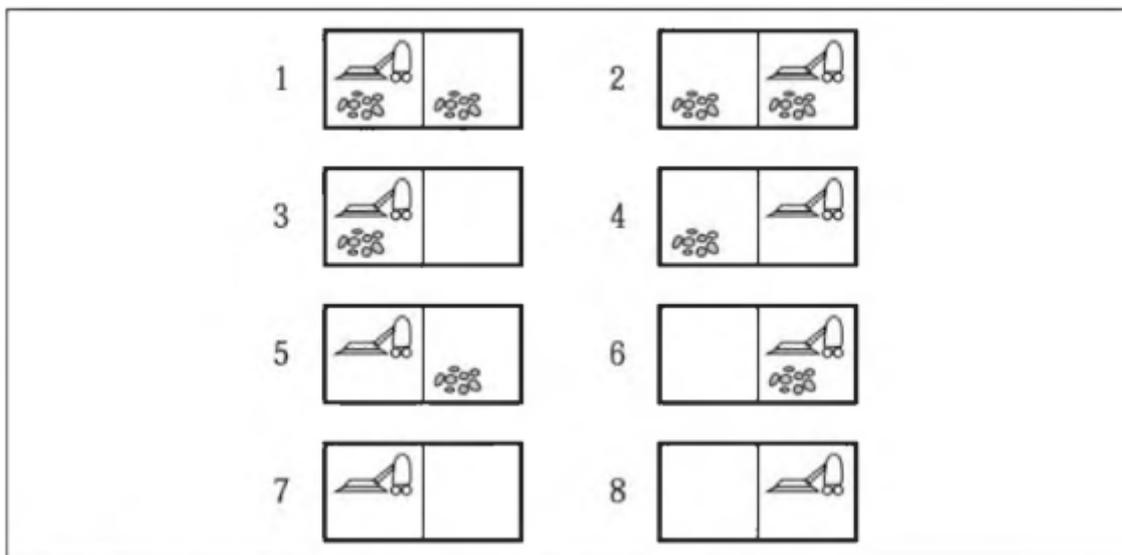
¿Qué pasa cuando el conocimiento de los estados o acciones es incompleto? Entonces hay que distinguir entre tres tipos de incompletitud que conducen a tres tipos de problemas diferentes:

- Entornos parcialmente observables y deterministas o no deterministas respecto a las acciones. Y entornos totalmente observables y no deterministas respecto a las acciones, también denominados problemas conformados.
- Entornos parcialmente observables y no deterministas respecto a las acciones, también denominados *problemas de contingencia*. En estas situaciones se puede obtener más información del entorno de la que inicialmente se dispone, aunque la predicción exacta es imposible. Por esta razón, todas las personas mantienen abiertos los ojos mientras caminan. Estos entornos se tratarán más adelante por ser más complejos que los que estamos presentando.
- Cuando se desconocen los estados y las acciones del entorno, el agente debe actuar para descubrirlos. Los problemas de exploración pueden verse como un caso extremo de problemas de contingencia. También se presentarán más adelante.

En los problemas conformados, no hay nada que nos diga en qué estado nos encontramos. Por lo tanto, no conocemos el estado inicial (supondremos que sí conocemos lo que produce cada acción).

Ejemplo: El problema de la aspiradora:

- La aspiradora puede percibir en qué cuadrante se encuentra y si hay suciedad en él.
- El espacio de estados tiene ocho estados, como se muestra en la figura.

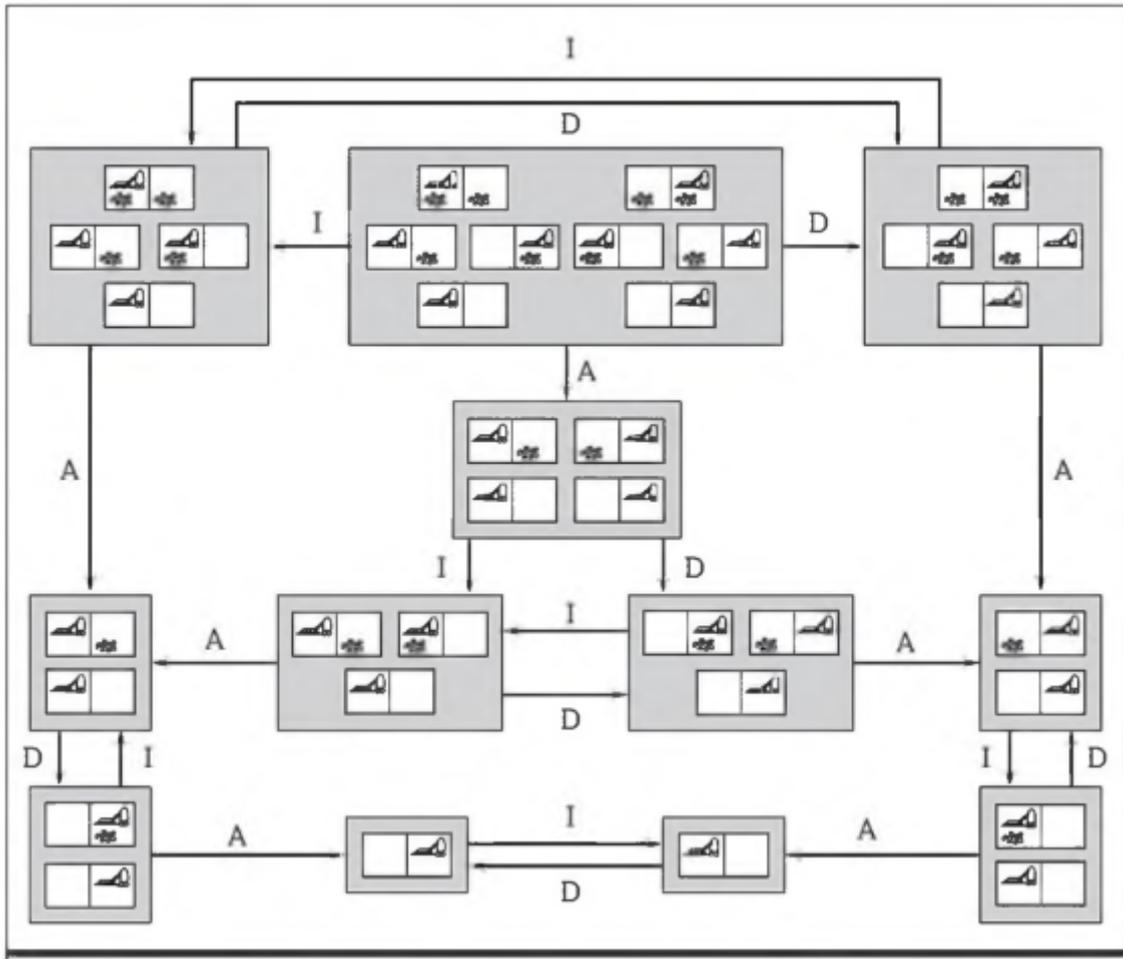


- Hay tres acciones que puede realizar (moverse a izquierda, moverse a derecha y aspirar).
- El objetivo es limpiar toda la suciedad y llegar a los (estados 7 u 8).

Si el entorno es observable, determinista, y completamente conocido, entonces el problema es trivialmente resoluble por cualquiera de los algoritmos que hemos descrito. Por ejemplo, si el estado inicial es 5, entonces la secuencia de acciones (derecha, aspirar) alcanzará un estado objetivo, 8.

Ahora bien, supongamos en lo que sigue que el entorno es parcialmente observable y deterministas respecto a las acciones. Entonces solo sabe que su estado inicial es uno del conjunto {1, 2, 3, 4, 5, 6, 7, 8}. Como conoce lo que hacen sus acciones, puede, por ejemplo, calcular que la acción derecha produce uno de los estados {2, 4, 6, 8}, y la secuencia de acción (derecha, aspirar) siempre terminará en uno de los estados {4, 8}. Finalmente, la secuencia (derecha, aspirar, izquierda, aspirar) garantiza alcanzar el estado objetivo 7 sea cual sea el estado inicio.

La figura muestra el espacio de estados de creencia accesible para el mundo determinista de la aspiradora. Hay solo 12 estados de creencia accesibles, pero el espacio de estados de creencia entero contiene todo conjunto posible de estados físicos, por ejemplo, $2^8 = 256$ estados de creencia.



Resumiendo: cuando el mundo no es completamente observable, el agente debe decidir sobre los conjuntos de estados que podría encontrar, más que por estados simples. Llamamos a cada conjunto de estados un estado de creencia, representando la creencia actual del agente con los estados posibles físicos en los que podría estar.

En un ambiente totalmente observable, cada estado de creencia contiene un estado físico.

Para resolver problemas conformados, buscamos en el espacio de estados de creencia más que en los estados físicos.

- El estado inicial es un estado de creencia, y cada acción aplica un estado de creencia en otro estado de creencia.
- Una acción se aplica a un estado de creencia uniendo los resultados de aplicar la acción a cada estado físico del estado de creencia.
- Un camino une varios estados de creencia, y una solución es ahora un camino que conduce a un estado de creencia, todos de cuyos miembros son estados objetivos.

En general, si el espacio de estados físico tiene S estados, el espacio de estados de creencia tiene 2^S estados de creencia.

Nuestra discusión hasta ahora de problemas conformados ha supuesto acciones deterministas, pero el análisis es esencialmente el mismo si el entorno es no determinista, es decir, si las acciones pueden tener varios resultados posibles. La razón es que el agente no tiene ningún modo de decir qué resultado ocurrió en realidad, así que varios resultados posibles son estados físicos adicionales en el estado de creencia sucesor.

Algoritmos de búsquedas informadas mediante heurísticas y exploración

Newell, Shaw y Simon en 1963 dieron la siguiente definición para el escurridizo concepto de heurística: «Proceso que puede resolver un problema dado, pero que no ofrece ninguna garantía de que lo hará». Uno puede quedarse pensativo ante esa definición, pero se ha demostrado que los métodos heurísticos son preferibles a los métodos no informados en la solución de problemas para los que una búsqueda exhaustiva está fuera de lugar por sus costes prohibitivos.

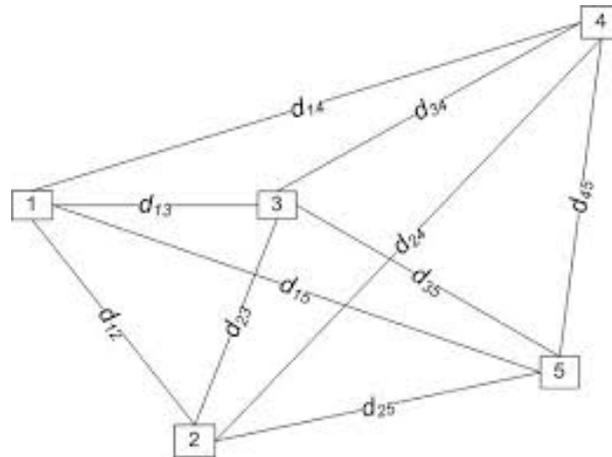
La idea de los métodos de búsquedas informadas es que disponen de alguna información sobre la proximidad de cada estado a un estado objetivo, lo que permite explorar en primer lugar los caminos más prometedores. No garantizan que se encuentre una solución, aunque existan soluciones. Si encuentran una, no asegura que esta tenga las mejores propiedades (camino de longitud mínima o de coste óptimo). Pero en algunas ocasiones, que en general no se podrán determinar *a priori*, encontrarán una solución aceptablemente buena en un tiempo razonable.

Toda la posible información heurística que se pueda utilizar, se suele concentrar en una única función que se denomina *función de evaluación heurística* $h(e)$, evaluable para cada estado del espacio de estados, que devuelve una cierta cantidad numérica que nos orienta de algún modo sobre lo prometedor que es ese estado con respecto al estado objetivo.

Ejemplo: el problema del viajante: dada una lista de N ciudades y las distancias entre cada par de ellas, ¿cuál es la ruta más corta posible que visita cada ciudad exactamente una vez y al finalizar regresa a la ciudad de origen?

- Estado inicial: un viajante se encuentra en una capital de provincia.
- Estado final: quiere volver a la misma capital por la mejor ruta posible (la más corta).
- Información: las capitales de provincia colindantes están unidas por carreteras; se dispone de un mapa con la disposición de las provincias, las distancias reales y sus «coordenadas» en kilómetros respecto a una capital que se considera el centro.
- Una heurística: asignamos a cada estado el valor distancia euclídea aérea (en línea recta) con respecto al estado objetivo.
- Acción: se elige la siguiente ciudad del camino cuando la suma de la distancia a la ciudad actual más la distancia en línea recta al estado objetivo sea la menor.

Planteado como un problema de grafo no dirigido



- Análisis de complejidad del problema:
 - El tamaño del espacio de estados es $N! = N \times (N-1) \times (N-2) \times \dots \times 3 \times 2 \times 1$.
 - Si seleccionamos una ciudad de partida, entonces es $(N-1)!$
 - Como no importa el sentido en que se desplace el viajante, entonces $\frac{(N-1)!}{2}$

Ejemplos con un cálculo de 10^6 rutas por segundo

N	N.º de rutas	Tiempo
10	181 440	0,181 s
30	4×10^{30}	10^{17} años

Nota: se supone que el universo tiene una edad de 13 400 millones de años, del orden de 10^{10} .

En el siguiente enlace puede encontrar la historia y un conjunto de diferentes soluciones obtenidas:

<https://es.wikipedia.org/wiki/Problema_del_viajante>

Algunos algoritmos de los más referenciados en la literatura:

- *Búsqueda primero el mejor*, se basa en el algoritmo *búsqueda de coste uniforme*, pero selecciona el siguiente nodo a expandir mediante una función $f(n)$ que permite calcular un coste estimado. Selecciona el nodo con menor coste. $f(n)$ es una combinación de la función coste $g(n)$ utilizada en la búsqueda de coste uniforme, más la función heurística $h(n)$. Normalmente $f(n)=g(n)+h(n)$ o una combinación lineal de ambos términos.

Las propiedades de la función $h(n)$ es que no debe sobreestimar el coste real y ser consistente (monótona), es decir, no decreciente.

- *Búsqueda voraz, primero el mejor*, se basa en el algoritmo *Búsqueda primero el mejor*, pero haciendo que $f(n) = g(n) * h(n)$

Sus propiedades son (en azul las ventajas, en rojo las desventajas):

- **Completo**: no, ya que puede caer en callejones sin salida y no aportar solución.
- **Óptimo**: no, aunque encuentre solución puede no ser la óptima, ya que no tiene en cuenta los costes de las acciones.
- **Complejidad temporal**: exponencial $O(b^m)$.
- **Complejidad espacial**: exponencial $O(b^m)$.

Como algoritmo, admite costes variables, reduce la complejidad, evita caminos inútiles.

- *Búsqueda primero el mejor A^** , tiene en cuenta el coste del camino recorrido y el coste de la heurística. La función g es una medida del coste para ir desde el estado inicial hasta el nodo actual (suma de los costes). La función h es una estimación del coste adicional necesario para alcanzar un nodo objetivo a partir del nodo actual, es decir, una estimación de lo que me queda por recorrer hasta alcanzar el objetivo. La función combinada f es la que definirá el siguiente nodo a expandir que será el que tenga menor coste estimado por la función $f(n) = g(n) + h(n)$.

Sus propiedades son (en azul las ventajas, en rojo las desventajas):

- **Completo**: sí, si existe solución, la encuentra.
- **Óptimo**: sí. En árboles, si la heurística es admisible y, en grafos, si es consistente. De hecho, ningún algoritmo expande menos nodos que él.
- **Complejidad temporal**: exponencial $O(b^d)$, pero si la heurística es buena puede llegar a polinómico.
- **Complejidad espacial**: exponencial $O(b^m)$.

Como algoritmo, admite costes variables, reduce la complejidad bastante si la heurística es buena. Evita caminos inútiles.

- *Búsqueda primero el mejor IDA**, combina A^* con la *Búsqueda en profundidad iterativa*. En este algoritmo, como en cualquier otro de profundización iterativa, cada iteración es una búsqueda *primero en profundidad*. En este caso la profundidad se basa en la información heurística y terminará no a una determinada profundidad, sino cuando se llegue a un nodo cuyo coste de la función heurística de evaluación $f=g+h$ sea mayor que el actual límite de coste de f . De esta forma en cada iteración se revisan todos los nodos con un coste de f menor o igual que el actual límite de coste.

Sus propiedades son (en azul las ventajas, en rojo las desventajas):

- **Completo**: sí, si existe solución, la encuentra.
- **Óptimo**: sí. En árboles, si la heurística es admisible y, en grafos, si es consistente.
- **Complejidad temporal**: exponencial.
- **Complejidad espacial**: lineal.

Como algoritmo, admite costes variables, reduce la complejidad bastante si la heurística es buena. Evita caminos inútiles.

- Existen otros algoritmos de este mismo tipo de los que se va a dar simplemente el nombre.
 - Búsqueda recursiva primero el mejor.
 - Búsqueda SMA* (Simplified Memory-Bounded A^*).

Algoritmos de búsqueda local

Los algoritmos citados previamente exploran de manera sistemática el espacio de estados buscando un camino que una el estado inicial con el estado objetivo, una vez encontrado esa es una solución del problema.

Existen estrategias de búsqueda en las que el camino seguido no importa y no se requiere mucho almacenamiento en memoria. En estos casos también se explora la idea de que los problemas se pueden resolver buscando en un espacio de estados, pero se realizan búsquedas locales, es decir, se opera utilizando un nodo y se producen movimientos solo alrededor de sus vecinos. No se mantienen los caminos recorridos.

Por ejemplo, veamos el conocido problema de la mochila:

Problema. El problema de la mochila es un problema simple de entender: hay una persona que tiene una mochila que soporta un peso máximo «W» y tiene que elegir entre N productos, de los cuales hay «n diferentes» y de cada ítem se tienen «p_i» elementos disponibles. Si cada uno de los elementos, denotados por la letra «i», con «i» yendo del «1» al «n», tiene un peso «w_i» y aporta un beneficio «b_i», el objetivo de la persona es elegir los elementos que va a meter en la mochila, que le permitan maximizar el beneficio sin excederse de la capacidad permitida.

La solución del problema sería la secuencia [y₁, y₂, y₃, ..., y_n] donde el valor de «y_i» indica cuantas copias se meterán en la mochila del tipo de elemento «i». Observe que estamos trabajando todo el rato con valores enteros. Y la solución también es una secuencia de enteros.

El problema se puede expresar matemáticamente del modo siguiente:

$$\text{Maximizar } [v_1 \times y_1 + v_2 \times y_2 + v_3 \times y_3 \dots + v_{n-1} \times y_{n-1} + v_n \times y_n]$$

$$\text{Verificándose que } [w_1 \times y_1 + w_2 \times y_2 + w_3 \times y_3 \dots + w_{n-1} \times y_{n-1} + w_n \times y_n] \leq \text{que } W$$

$$\text{Estando los valores } y_i \text{ comprendidos entre } 0 \leq y_i \leq q_i$$

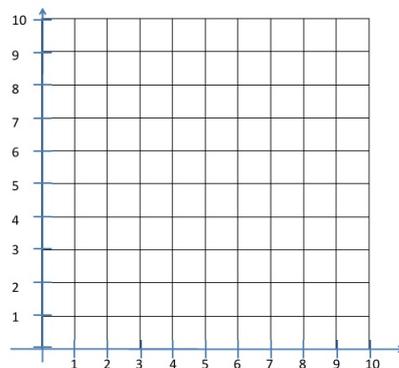
Veamos el tamaño del espacio de estados en un caso simplificado: si contáramos con 4 productos, para saber cuál es la mejor solución deberíamos probar las $2^4 = 16$ posibilidades. El 2 se desprende del hecho de que cada decisión es incluir o no al producto y el 4 de la cantidad de productos. 16 posibilidades es un número manejable, sin embargo, si la cantidad de elementos por ejemplo ascendiera a 20, tendríamos que analizar nada más y nada menos que $2^{20} = 1,048,576$ posibilidades. Y si en vez de una mochila, estuviéramos hablando de un contenedor en el que se tuvieran que considerar 60 elementos entonces $2^{60} = 1,3152,921,504,606,1846,976$, y si fueran 96 elementos, $2^{96} = 39,614,081,257,3132,168,2796,771,975,168$.

Este tipo de problemas consiste en dada una función real, ir eligiendo **sistemáticamente** valores de entrada, tomados de un conjunto permitido, y calculando el valor de la función. De forma general, la solución consiste en encontrar los «mejores valores», es decir, los que maximizan o minimizan dicha función. Este tipo de problemas se pueden realizar trabajando con espacios de estados que toman valores discretos, como en el caso del ejemplo de la mochila o con valores continuos como sería un problema del siguiente tipo.

Problema: Un químico ha sintetizado un nuevo fertilizante «f» hecho a partir de dos materias primas (mp_1) y (mp_2). Al combinar ciertas cantidades de cada materia (cmp_1) y (cmp_2), se obtiene una cierta cantidad de fertilizante (cf), dicha cantidad viene dada por la función conocida $cf(cmp_1, cmp_2)$. Se sabe que cada unidad de materia del tipo mp_1 requiere una inversión conocida dada por imp_1 y la del tipo (mp_2) una inversión de (imp_2). El problema se plantea del modo siguiente, si la compañía tiene una capacidad de inversión mensual de X euros, determinar la cantidad de materias primas necesarias de forma que se maximice la cantidad de fertilizante obtenido.

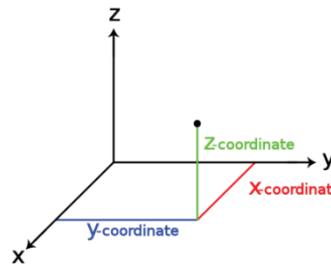
Visualización gráfica de una función asociada a un espacio de estados: Supongamos que la función a maximizar «f», depende solo de estados definidos mediante una pareja de números (x, y).

Entonces el espacio de estados podríamos representarlo mediante un plano coordenado como el de la figura.

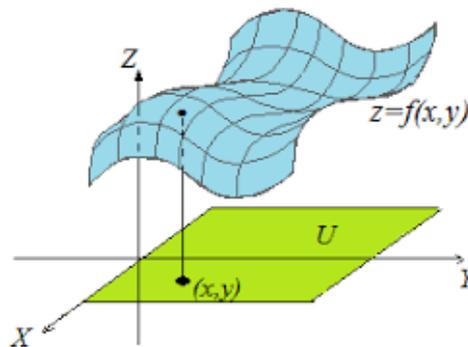


Si los números de la pareja que define el estado son naturales, cada cruce de dos líneas en la figura representaría un estado. En el caso de que la pareja tomara valores continuos, cualquier punto del plano sería un estado.

Ahora, para cada estado (x, y), dibujemos el valor que toma la función « $f(x, y) = z$ » representando todo en 3D, véase la figura

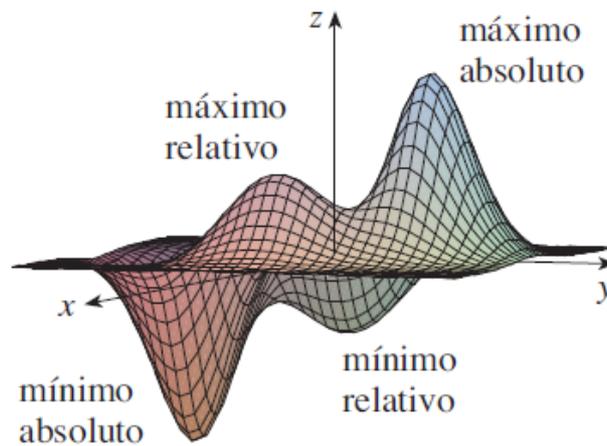


Y si ahora recorremos el espacio de estados y lo visualizamos todo, podríamos ver algo como esto



Los puntos en azul se corresponden con los valores de «f» de los puntos en verde en el caso del espacio de estados continuo, los puntos correspondientes a los cruces de las curvas sobre la superficie, serían los valores de «f» de los puntos correspondientes a un espacio de estados discreto.

Reflexión: Obsérvese que en cualquiera de los dos problemas planteados previamente, espacio de estados discreto o espacio de estados continuo, aparece el concepto de *búsqueda de máximos o mínimos de una función*. Estos son los valores más grandes (máximos) o más pequeños (mínimos), que toma esa función en un punto. Si un extremo se encuentra situado dentro de un intervalo en particular del espacio de estados, el extremo recibe el adjetivo de local, ya que es un extremo que se localiza en una parte del dominio. Si el intervalo en el que se encuentra el valor extremo es todo el espacio de estados, el extremo se denomina *global* o *absoluto*.

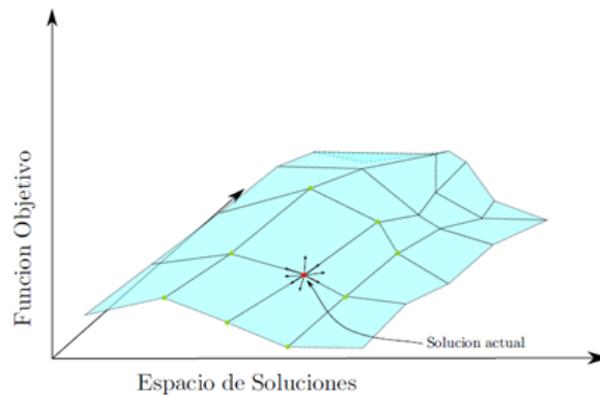


Los algoritmos de los que vamos a hablar inicialmente se centran en espacios de estados discretos y se denominan *locales*, ya que se posicionan en un estado, evalúan la función a maximizar o a minimizar, obtienen un valor y lo comparan con los valores de los estados que lo rodean y van tomando decisiones de cambio de punto.

- *Ascenso de colinas*. Este algoritmo parte desde un punto inicial del espacio de estados a partir del cual se va a intentar explorar cual es el siguiente estado de los que lo rodean que sea más prometedor. En cada paso se selecciona el mejor vecino y se sigue así hasta alcanzar un extremo.

Sus propiedades son (en azul las ventajas, en rojo las desventajas):

- **Completo**: no, encuentra el extremo más cercano al punto de partida, sin saber si ha alcanzado un extremo local o el global. Puede quedar atrapado en mesetas (zonas planas del espacio de estados que al evaluar la función devuelve siempre el mismo valor).
- **Óptimo**: no. Solo encuentra la solución extrema más cercana.
- **Complejidad temporal**: lineal o incluso logarítmica.
- **Complejidad espacial**: constante.
-



- *Ascenso de colinas (variantes)*. Todos los algoritmos parten desde un punto inicial del espacio de estados a partir del cual se va a intentar explorar cuál es el siguiente estado de los que lo rodean que sea más prometedor. En cada paso se selecciona el mejor vecino y se sigue así hasta alcanzar un extremo.

Estocástico. De todos los vecinos mejores que el de partida, se escoge uno de forma aleatoria.

Primera opción. Se generan vecinos de forma aleatoria hasta que se encuentra uno que es mejor que el actual.

Reinicio aleatorio. Se lanza la búsqueda varias veces desde posiciones iniciales generados de forma aleatoria y se elige el mejor.

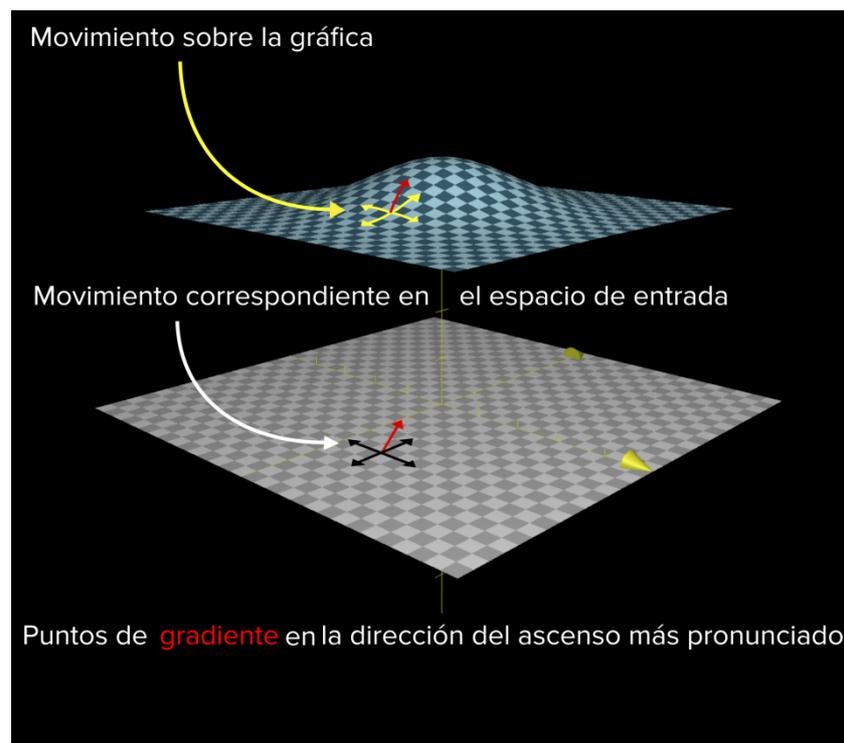
Sus propiedades son análogas al ascenso de colinas.

- *Otros algoritmos*. Se han planteado otros algoritmos inspirados en analogías físicas o biológicas. Se aplican a problemas reales con bastante éxito. A ellos les dedicaremos un capítulo más adelante. Entre los más conocidos se encuentran:
 - *Temple simulado*.
 - *Algoritmos genéticos*.
 - *Optimización mediante enjambres de puntos*.
 - *Optimización por colonias de hormigas*.
 - *Gota de agua inteligente*.
 - *Algoritmo de búsqueda gravitacional*.
 - ...

Los algoritmos de los que hemos hablado inicialmente se han centrado en espacios de estados discretos. Pero en el mundo real muchos problemas son continuos, es decir, dado un estado inicial, este tiene infinitos estados que lo rodean. La mayoría de ellos se caracterizan por ser también algoritmos locales, ya que se posicionan en un estado y analizan su entorno cercano.

Los algoritmos se pueden clasificar como:

- (i) Búsqueda sin el uso de derivadas. Intentan copiar las ideas de los algoritmos discretos, muestreando de alguna manera astuta su entorno y seleccionando un conjunto discreto de estados. Ejemplos típicos de este tipo de métodos son: el método simplex, el algoritmo de Hooke y Jeeves, el método de Rosenbrock y el método de las direcciones conjugadas.
- (ii) Búsqueda usando información de la derivada primera pero en vez de evaluar la función a maximizar o a minimizar, hacen uso del operadores diferencial del tipo **gradiente** que permite encontrar la dirección de la pendiente más escarpada por la que habría que deslizarse para encontrar un máximo o un mínimo. Véase la gráfica



Entre los algoritmos que usan información del gradiente están el método de máximo descenso, el método del gradiente conjugado y los métodos cuasi Newton.

El método del máximo descenso realiza una búsqueda a lo largo de la dirección opuesta al gradiente para minimizar la función.

El método de gradiente conjugado combina la información del último gradiente con la información de gradientes de iteraciones previas.

Los métodos cuasi Newton construyen una aproximación de la curvatura de la función utilizando solo información del gradiente.

- (iii) Búsqueda usando información de la derivada segunda. Dos de los más eficaces son el método de Newton y el de Newton Raphson.

Problemas de satisfacción de restricciones (PSR)

Problema de coloración de mapas:

Es un problema clásico que se puede formular como un PSR. En este problema hay un conjunto de colores y un mapa dividido en regiones. El objetivo es colorear cada región del mapa de manera que regiones adyacentes tengan distintos colores. Una forma de modelar este problema dentro del PSR sería asociando una variable por cada región del mapa (el dominio de cada variable es el conjunto de posibles colores disponibles), y para cada par de regiones contiguas añadir una restricción sobre los valores de las variables correspondientes que impida asignarles el mismo valor (color). Como suele ser habitual, este mapa puede ser representado mediante un grafo donde los nodos representan las diversas regiones del mapa y cada par de regiones adyacentes están unidas por una arista. Veremos que esta representación en forma de grafo, que en este problema es natural, será usada como metodología general para dar una estructura a las restricciones de cualquier problema PSR.

Tres colores son suficientes para mapas simples, pero en algunos casos es necesario un cuarto color adicional. Véase

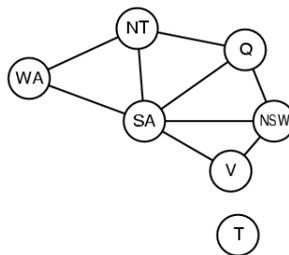
https://es.wikipedia.org/wiki/Teorema_de_los_cuatro_colores

Ejemplo, color usando tres colores el mapa siguiente



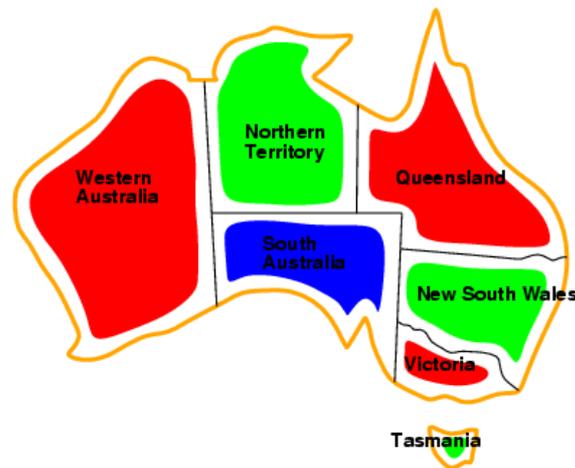
Definición del problema:

- **Variabes** a utilizar:
 - X1 Western Australia (WA)
 - X2 Northern Territory (NT)
 - X3 Queensland (Q)
 - X4 New South Wales (NSW)
 - X5 Victoria (V)
 - X6 South Australia (SA)
 - X7 Tasmania (T)
- **Domínio de cada variable**, conjunto de valores asociados a cada variable:
 - En este caso el dominio es el mismo [a1 (rojo), a2 (verde), a3 (azul)].
- **Restricciones**: Las regiones adyacentes deben tener colores diferentes:
 - Hay nueve restricciones en la adyacencia de colores:
 - $CSA \neq CWA, CSA \neq CNT, CSA \neq CQ, CSA \neq CNSW, CSA \neq V$
 - $CNT \neq CWA, CNT \neq CQ$
 - $CNSW \neq CQ, CNSW \neq CV$
 - Si se empieza por WA y se sigue por NT, entonces estados posibles podrían ser: (WA, NT) [(rojo, verde), (rojo, azul), (verde, rojo), (verde, azul), (azul, rojo), (azul, verde)].
- **Grafo asociado**:
 - Los nodos representan a las variables.
 - Los arcos representan las restricciones, en este caso binarias.



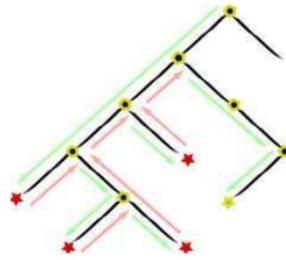
- **Soluciones**: Son asignaciones completas y coherentes:
 - Una solución posible sería: {WA, rojo}, {NT, verde}, {Q, rojo}, {NSW, verde}, {V, rojo}, {SA, azul}, {T, verde}
 - Escrita con notación de variables sería: {X1, rojo}, {X2, verde}, {X3, rojo}, {X4, verde}, {X5, rojo}, {X6, azul}, {X7, verde}
 - Escrita de manera reducida, asociando la posición a la variable: (rojo, verde, rojo, verde, rojo, azul, verde)

Escrita de manera superreducida
(rvrvrav)



- **Resolución:** Una vez formulado el problema hay dos maneras de procesar las restricciones:
 - Técnicas de consistencia: Basadas en la eliminación de valores inconsistentes (no verifican las restricciones impuestas) de los dominios de las variables.
 - Algoritmos de búsqueda: Basados en la exploración sistemática del espacio de soluciones hasta encontrar una solución (o no) que verifica todas las restricciones del problema.
- **Árbol de búsqueda:** Las posibles combinaciones de la asignación de valores a las variables genera un espacio de búsqueda que se puede visualizar como un árbol de búsqueda. De esta forma se va a poder recorrer siguiendo la estrategia que se desee. La forma de construirlo es la siguiente:
 - Los nodos en el primer nivel son los estados parciales a los que se les ha asignado un valor a la variable X_1 , en este nivel el resto de las variables permanecen sin asignarles valor.
 - Los nodos en el segundo nivel son los estados parciales a los que se les ha asignado un valor a la variable X_2 .
 - ...
 - Si n es el número de variables del problema, los nodos en el nivel n son los estados a los que se les ha asignado valores a la variable X_n .
 - De esta manera conforme se van recorriendo todos los nodos desde el nivel 1 hasta el nivel n , si se van obteniendo que los valores de una hoja son consistentes, entonces al final es una solución del problema.
 - La complejidad del árbol es del orden de $(n.º \text{ de valores}^n)$.

Backtracking



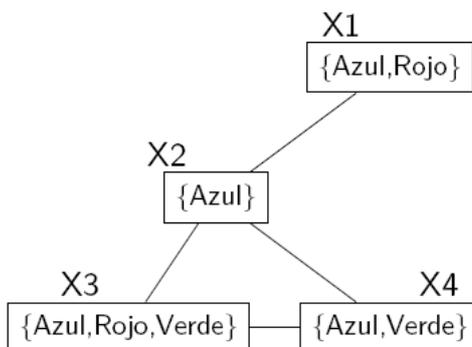
Este algoritmo es mejor que el análisis exhaustivo de todos los nodos del último nivel, es muy simple, pero sigue siendo muy ineficiente, ya que realiza una visión local del problema. Solo comprueba restricciones que están formadas por la variable actual y las pasadas, e ignora la relación entre la variable actual y las futuras.

Para ayudar a combatir este problema se han desarrollado algunos algoritmos de búsqueda más robustos como los algoritmos Look-ahead, estos hacen una comprobación hacia adelante en cada etapa de búsqueda, es decir, llevan a cabo las comprobaciones para obtener las inconsistencias de las variables futuras involucradas además de las variables actual y pasadas.

Antes de seguir vamos a comentar lo que se entiende por arco de consistencia y por propagación de restricciones.

Dadas dos variables (X_i, X_j) unidas por un arco, y para cualquier valor v_k del dominio D_i existe un valor v_l del dominio D_j tal que satisfacen las restricciones. Es decir, se busca que los valores posibles de X_i sean consistentes con la restricción asociada al arco.

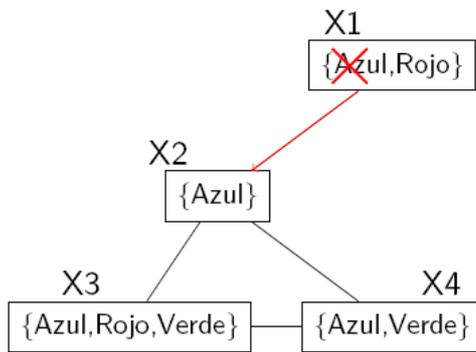
Ejemplo de arco-consistencia (profesor Luigi Ceccaroni):



Lista de arcos inicial:

(X1,X2), (X2,X1), (X2,X3), (X3,X2),
(X2,X4), (X4,X2), (X3,X4), (X4,X3)

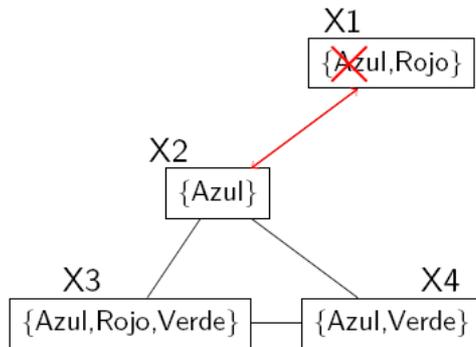
1. $X_1 - X_2 \rightarrow$ Quitar Azul de X_1



Lista de arcos inicial:

- (X_1, X_2), (X_2, X_1), (X_2, X_3), (X_3, X_2),
 (X_2, X_4), (X_4, X_2), (X_3, X_4), (X_4, X_3)

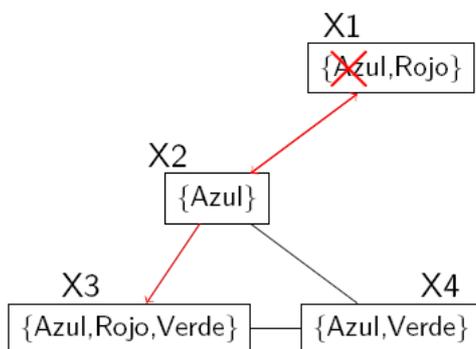
1. $X_1 - X_2 \rightarrow$ Quitar Azul de X_1
2. $X_2 - X_1 \rightarrow$ Todo consistente



Lista de arcos inicial:

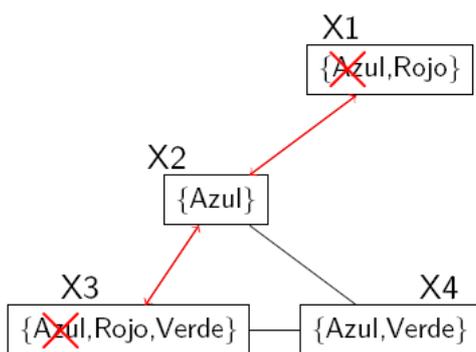
- (X_1, X_2), (X_2, X_1), (X_2, X_3), (X_3, X_2),
 (X_2, X_4), (X_4, X_2), (X_3, X_4), (X_4, X_3)

1. $X_1 - X_2 \rightarrow$ Quitar Azul de X_1
2. $X_2 - X_1 \rightarrow$ Todo consistente
3. $X_2 - X_3 \rightarrow$ Todo consistente



Lista de arcos inicial:

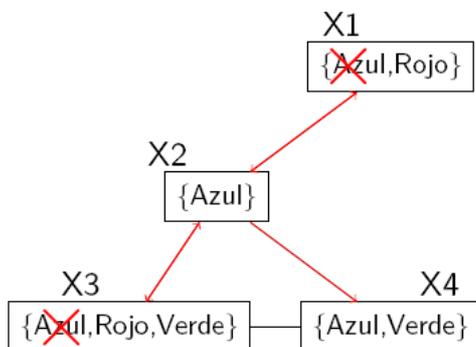
- (X_1, X_2), (X_2, X_1), (X_2, X_3), (X_3, X_2),
 (X_2, X_4), (X_4, X_2), (X_3, X_4), (X_4, X_3)



1. $X_1 - X_2 \rightarrow$ Quitar Azul de X_1
2. $X_2 - X_1 \rightarrow$ Todo consistente
3. $X_2 - X_3 \rightarrow$ Todo consistente
4. $X_3 - X_2 \rightarrow$ Quitar Azul de X_3 , Tendríamos que añadir $X_4 - X_3$ pero ya está

Lista de arcos inicial:

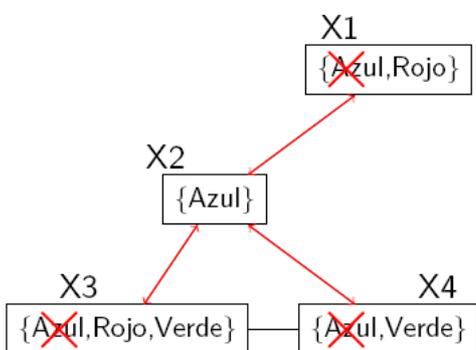
- (X_1, X_2) , (X_2, X_1) , (X_2, X_3) , (X_3, X_2) ,
 (X_2, X_4) , (X_4, X_2) , (X_3, X_4) , (X_4, X_3)



1. $X_1 - X_2 \rightarrow$ Quitar Azul de X_1
2. $X_2 - X_1 \rightarrow$ Todo consistente
3. $X_2 - X_3 \rightarrow$ Todo consistente
4. $X_3 - X_2 \rightarrow$ Quitar Azul de X_3 , Tendríamos que añadir $X_4 - X_3$ pero ya está
5. $X_2 - X_4 \rightarrow$ Todo consistente

Lista de arcos inicial:

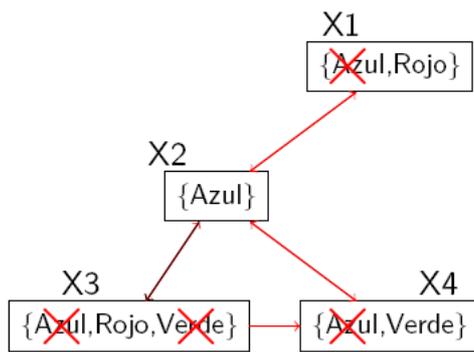
- (X_1, X_2) , (X_2, X_1) , (X_2, X_3) , (X_3, X_2) ,
 (X_2, X_4) , (X_4, X_2) , (X_3, X_4) , (X_4, X_3)



1. $X_1 - X_2 \rightarrow$ Quitar Azul de X_1
2. $X_2 - X_1 \rightarrow$ Todo consistente
3. $X_2 - X_3 \rightarrow$ Todo consistente
4. $X_3 - X_2 \rightarrow$ Quitar Azul de X_3 , Tendríamos que añadir $X_4 - X_3$ pero ya está
5. $X_2 - X_4 \rightarrow$ Todo consistente
6. $X_4 - X_2 \rightarrow$ Quitar Azul de X_4 , Tendríamos que añadir $X_3 - X_4$ pero ya está

Lista de arcos inicial:

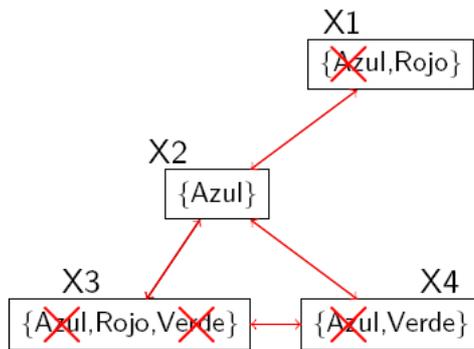
- (X_1, X_2) , (X_2, X_1) , (X_2, X_3) , (X_3, X_2) ,
 (X_2, X_4) , (X_4, X_2) , (X_3, X_4) , (X_4, X_3)



Lista de arcos inicial:

$(X1, X2)$, $(X2, X1)$, $(X2, X3)$, $(X3, X2)$,
 $(X2, X4)$, $(X4, X2)$, $(X3, X4)$, $(X4, X3)$

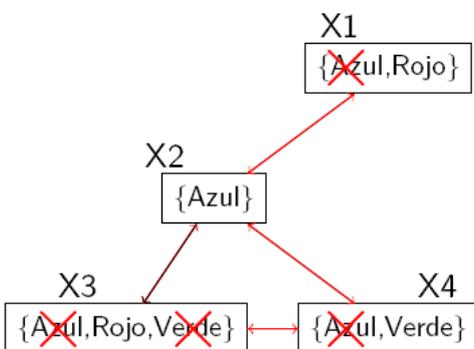
1. $X_1 - X_2 \rightarrow$ Quitar Azul de X_1
2. $X_2 - X_1 \rightarrow$ Todo consistente
3. $X_2 - X_3 \rightarrow$ Todo consistente
4. $X_3 - X_2 \rightarrow$ Quitar Azul de X_3 , Tendríamos que añadir $X_4 - X_3$ pero ya está
5. $X_2 - X_4 \rightarrow$ Todo consistente
6. $X_4 - X_2 \rightarrow$ Quitar Azul de X_4 , Tendríamos que añadir $X_3 - X_4$ pero ya está
7. $X_3 - X_4 \rightarrow$ Quitar Verde de X_3 , Añadimos $X_2 - X_3$



Lista de arcos inicial:

$(X1, X2)$, $(X2, X1)$, $(X2, X3)$, $(X3, X2)$,
 $(X2, X4)$, $(X4, X2)$, $(X3, X4)$, $(X4, X3)$

1. $X_1 - X_2 \rightarrow$ Quitar Azul de X_1
2. $X_2 - X_1 \rightarrow$ Todo consistente
3. $X_2 - X_3 \rightarrow$ Todo consistente
4. $X_3 - X_2 \rightarrow$ Quitar Azul de X_3 , Tendríamos que añadir $X_4 - X_3$ pero ya está
5. $X_2 - X_4 \rightarrow$ Todo consistente
6. $X_4 - X_2 \rightarrow$ Quitar Azul de X_4 , Tendríamos que añadir $X_3 - X_4$ pero ya está
7. $X_3 - X_4 \rightarrow$ Quitar Verde de X_3 , Añadimos $X_2 - X_3$
8. $X_4 - X_3 \rightarrow$ Todo consistente
9. $X_2 - X_3 \rightarrow$ Todo consistente



Lista de arcos inicial:

$(X1, X2)$, $(X2, X1)$, $(X2, X3)$, $(X3, X2)$,
 $(X2, X4)$, $(X4, X2)$, $(X3, X4)$, $(X4, X3)$

1. $X_1 - X_2 \rightarrow$ Quitar Azul de X_1
2. $X_2 - X_1 \rightarrow$ Todo consistente
3. $X_2 - X_3 \rightarrow$ Todo consistente
4. $X_3 - X_2 \rightarrow$ Quitar Azul de X_3 , Tendríamos que añadir $X_4 - X_3$ pero ya está
5. $X_2 - X_4 \rightarrow$ Todo consistente
6. $X_4 - X_2 \rightarrow$ Quitar Azul de X_4 , Tendríamos que añadir $X_3 - X_4$ pero ya está
7. $X_3 - X_4 \rightarrow$ Quitar Verde de X_3 , Añadimos $X_2 - X_3$
8. $X_4 - X_3 \rightarrow$ Todo consistente

Teniendo en cuenta lo anterior, se puede modificar el algoritmo de *Algoritmo de backtracking cronológico*, de manera que las situaciones sin salida se pueden identificar antes y los valores inconsistentes se pueden descubrir y podar para las variables futuras. Uno de los algoritmos de este tipo más comunes es *Forward checking*.

En cada momento el algoritmo se encontrará en un cierto nivel (k), con una solución parcial $(X_1, X_2, \dots, X_k, \dots)$ con $k \leq n$, es decir que los (k) valores cumplen las restricciones aplicables.

- Si puede añadirse un elemento X_{k+1} a la solución parcial se avanza al nivel ($k+1$), es decir, que los ($k+1$) valores siguen cumpliendo con las restricciones aplicables y se consultan las restricciones sobre las variables futuras con arco desde la actual. Se eliminan valores no compatibles de los dominios correspondientes a dichas variables futuras. Equivale a hacer arco-consistente la variable actual con las futuras en cada paso.
- Si no se prueba con otro valor válido para X_k .
- Si no existe ningún valor que sea válido para probar, se retrocede al nivel ($k-1$).
- Se continúa con este proceso hasta que la solución parcial sea una solución del problema o hasta que no queden más posibilidades por probar en el caso de que no se encuentre ninguna solución o se busquen todas las soluciones del problema.

Las soluciones que se pueden encontrar para colorear el mapa son:

- $(WA,r) (NT,v) (SA,a) (Q,r) (NSW,v) (V,r)$ y (T,r)
- $(WA,r) (NT,v) (SA,a) (Q,r) (NSW,v) (V,r)$ y (T,v)
- $(WA,r) (NT,v) (SA,a) (Q,r) (NSW,v) (V,r)$ y (T,a)

- $(WA,r) (NT,a) (SA,v) (Q,r) (NSW,a) (V,r)$ y (T,r)
- $(WA,r) (NT,a) (SA,v) (Q,r) (NSW,a) (V,r)$ y (T,v)
- $(WA,r) (NT,a) (SA,v) (Q,r) (NSW,a) (V,r)$ y (T,a)

- $(WA,v) (NT,r) (SA,a) (Q,v) (NSW,r) (V,v)$ y (T,r)
- $(WA,v) (NT,r) (SA,a) (Q,v) (NSW,r) (V,v)$ y (T,v)
- $(WA,v) (NT,r) (SA,a) (Q,v) (NSW,r) (V,v)$ y (T,a)

- $(WA,v) (NT,v) (SA,r) (Q,v) (NSW,a) (V,v)$ y (T,r)
- $(WA,v) (NT,v) (SA,r) (Q,v) (NSW,a) (V,v)$ y (T,v)
- $(WA,v) (NT,v) (SA,r) (Q,v) (NSW,a) (V,v)$ y (T,a)

- $(WA,a) (NT,v) (SA,v) (Q,a) (NSW,a) (V,a)$ y (T,r)

- (WA,a) (NT,v) (SA,v) (Q,a) (NSW,a) (V,a) y (T,v)
- (WA,a) (NT,v) (SA,v) (Q,a) (NSW,a) (V,a) y (T,a)

- (WA,a) (NT,v) (SA,r) (Q,a) (NSW,a) (V,a) y (T,r)
- (WA,a) (NT,v) (SA,r) (Q,a) (NSW,a) (V,a) y (T,v)
- (WA,a) (NT,v) (SA,r) (Q,a) (NSW,a) (V,a) y (T,a)

En la siguiente tabla se ve el algoritmo aplicando las restricciones

X1 WA	X2 NT	X3 SA	X4 Q	X5 NSW	X6 V	X7 T
1	2	3	4	5	6	7
(R)(V)(A)	RVA	RVA	RVA	RVA	RVA	RVA
R	V A (por 1)	V A (por 1)				
	V (A)	A (por 2)	R V (por 3)	R V (por 3)	R V (por 3)	
			R (por 2)	V (por 4)	R (por 5)	RVA
	A	V	RA	RA	RA	
			R	A	R	RVA
X1 WA	X2 NT	X3 SA	X4 Q	X5 NSW	X6 V	X7 T
1	2	3	4	5	6	7
(R)(V)(A)	RVA	RVA	RVA	RVA	RVA	RVA
V	RA (por 1)	RA (por 1)				
	R (A)	A (por 2)	V A (por 3)	R V (por 3)	R V (por 3)	
			V (por 2)	R (por 4)	V (por 5)	RVA
	A	R (por 2)	V A (por 3)	V A (por 3)	V A (por 3)	
			V	A (por 4)	V (por 5)	RVA
X1 WA	X2 NT	X3 SA	X4 Q	X5 NSW	X6 V	X7 T
1	2	3	4	5	6	7
(R)(V)(A)	RVA	RVA	RVA	RVA	RVA	RVA
A	R V (por 1)	R V (por 1)				
	R (V)	V (por 2)	V A (por 3)	R A (por 3)	R A (por 3)	
			A (por 2)	R (por 4)	A (por 5)	RVA
	V	R (por 2)	V A (por 3)	V A (por 3)	V A (por 3)	
			A (por 2)	V	A (por 5)	RVA

Una vez visto cómo encontrar soluciones al problema propuesto, veamos qué es un problema de satisfacción de restricciones (PSR) y realicemos una descripción más formalista.

Un problema de satisfacción de restricciones (PSR) es un problema en el que la asignación de valor a una variable, restringe los posibles valores asignables a otras variables.

Tipos de restricciones

- Restricciones unarias.
- Restricciones binarias.
- ...

Existen muchos problemas relevantes que se pueden expresar como PSR como, por ejemplo, el diseño de una cadena de montaje, la asignación de recursos, planificación..., representación del conocimiento, razonamiento automático...

Los tipos de problemas se pueden clasificar en tres grandes categorías:

- PSR con variables discretas y dominios finitos. Si el tamaño máximo del dominio de cualquier variable es d , y si existen n variables, entonces la complejidad es del orden de $O(d^n)$, es decir, exponencial en el número de variables.
- PSR con variables discretas y dominios infinitos. En este caso la enumeración de combinaciones de valores no es factible. Por lo que se hace necesario construir un lenguaje para representar las restricciones. Existen soluciones especiales para las restricciones lineales sobre variables enteras. Y no existen algoritmos para resolver restricciones no lineales sobre variables enteras.
- PSR con dominios continuos. En este caso la mejor categoría conocida de dominios continuos para problemas PSR son los problemas de programación lineal. En la optimización no lineal con restricciones, los diferentes métodos pertenecen a las clases: (i) métodos de penalización exterior, (ii) métodos de penalización interior (barrera), (iii) métodos de proyección de gradiente, (iv) método de gradiente reducido generalizado, (v) programación lineal sucesiva, (vi) programación cuadrática sucesiva.

- En el caso de un PSR con variables discretas y dominios finitos, su especificación está formada por tres tipos de componentes:
 - X el conjunto de variables $\{X_1, X_2, \dots, X_n\}$.
 - D es el conjunto de los dominios de cada variable $\{D_1, D_2, \dots, D_n\}$.
 - D_i es el conjunto de valores admisibles $\{v_1, v_2, \dots, v_k\}$ para la variable X_i .
 - C es el conjunto de restricciones que especifica las combinaciones de valores permitidos.

- Para resolver un problema PSR de estas características, es necesario definir el espacio de estados y la noción de una solución:
 - Cada estado en un PSR se define por una asignación de valores a alguna o todas las variables. $\{\dots, X_i = v_i, X_j = v_j, \dots\}$
 - Una asignación que no viola ninguna restricción se denomina *asignación consistente* o *legal*.
 - Una asignación completa es aquella en que cada variable tiene una asignación.
 - Una solución de un PSR es una asignación consistente y completa.
 - Una asignación parcial es aquella que asigna valores solo a algunas variables.

- El árbol de búsqueda de estados se construye del modo siguiente:
 - Raíz, es el estado con asignación vacía.
 - Cada nivel del árbol va asociado a una variable. El número de niveles es igual que el número de variables.
 - Un cierto nivel (k), si se recorre desde el nivel 1, permite obtener todos los posibles estados parciales del tipo $(X_1, X_2, \dots, X_k, \dots)$ con $k \leq n$.
 - Los sucesores de un nodo son todos los valores de la variable asociada a ese nivel.
 - Las soluciones completas aparecen solo a profundidad n .

- Es bueno visualizar las restricciones de un PSR como un grafo. Los nodos del grafo corresponden a variables del problema y los arcos corresponden a restricciones. La estructura del grafo de restricciones suele usarse para simplificar el proceso de búsqueda.

Técnicas para variables discretas y dominios finitos

- Existen diferentes métodos de resolución:

Como las soluciones aparecen solo a profundidad n , los algoritmos de búsqueda, primero en profundidad, son populares:

- *Método 1.* Partiendo del árbol de búsquedas generado, se va recorriendo en profundidad para ir obteniendo todo el espacio de estados completos y cada uno obtenido se va comprobando si es solución. Es muy poco eficiente, genera muchas asignaciones parciales que violan las restricciones. La idea principal de los procedimientos de resolución es eliminar grandes porciones del espacio de búsqueda de golpe identificando parejas variable / valor que violen las restricciones.
- *Backtracking (BT).* La estrategia es:
 - Se construye una solución parcial, es decir, se van realizando asignaciones parciales que satisfacen las restricciones.
 - Se extiende la solución parcial, incluyendo una variable cada vez hasta la solución total.
 - Si no se puede extender se realiza el proceso de vuelta hacia atrás (*Backtracking*), esto producirá una poda de todo el subárbol que cuelga de él. Existe el BT cronológico, en el que se elimina la última decisión. El BT no cronológico, en el que se elimina una decisión anterior.
- *Forward checking.* La estrategia es:
 - En cada momento el algoritmo se encontrará en un cierto nivel (k), con una solución parcial ($X_1, X_2, \dots, X_k, \dots$) con $k \leq n$, es decir, que los (k) valores cumplen las restricciones aplicables.
 - Si puede añadirse un elemento X_{k+1} a la solución parcial se avanza al nivel ($k+1$), es decir, que los ($k+1$) valores siguen cumpliendo con las restricciones aplicables y se consultan las restricciones sobre las variables futuras con arco desde la actual. Se eliminan valores no compatibles de los dominios correspondientes a dichas variables futuras. Esto equivale a hacer arco-consistente la variable actual con las futuras en cada paso.
 - Si no se prueba con otro valor válido para X_k .
 - Si no existe ningún valor que sea válido para probar, se retrocede al nivel ($k-1$).
 - Se continúa con este proceso hasta que la solución parcial sea una solución del problema o hasta que no queden más posibilidades por probar en el caso de que no se encuentre ninguna solución o se busquen todas las soluciones del problema.

- *Existen muchos más métodos que intentan optimizar las ideas previas:*
 - Realy Full Lookahead.
 - Maintaining Arc Consistency.
 - Heurísticas.

Para saber más:

- <https://es.wikipedia.org/wiki/Teor%C3%ADa_de_grafos#Aplicaciones>

Algoritmos basados en el uso de la lógica (modelo simbólico)

El conocimiento, su representación, y los procesos de razonamiento que permiten que el conocimiento evolucione, juegan un papel importante cuando se trata de resolver problemas en entornos observables.

En esta parte del libro se aplicarán los principios generales de la lógica. Con ella podremos representar el conocimiento que se tenga e inferir otro conocimiento a partir de él. Este paradigma es flexible y potente. Se caracteriza por que este tipo de conocimiento siempre es categórico, es decir, cada proposición acerca del mundo es verdadera o falsa, si bien el agente puede ser agnóstico acerca de algunas proposiciones.

La lógica es un ejemplo didáctico muy interesante a la hora de representar y utilizar el conocimiento, pero tiene serias limitaciones. Las personas cuando trabajan en entornos parcialmente observables manejan conocimiento que es incierto y la lógica no puede representar bien esa incertidumbre, por lo que más adelante tendremos que cambiar de paradigma.

El componente principal de estos sistemas es su **base de conocimiento** en el que se recoge todo el conocimiento que el sistema tiene de partida **conocimiento de antecedentes** y el que va adquiriendo conforme evoluciona. Dicho conocimiento se representa mediante lo que posteriormente definiremos como **proposiciones**. A una base de conocimiento se le pregunta y ella responde en función del conocimiento que tiene hasta el momento de la pregunta. La respuesta es una acción que se debe ejecutar. Las capacidades de representación, razonamiento y aprendizaje se apoyan en la teoría y en la tecnología de la lógica. En cada caso en el que el sistema saca una conclusión, a partir de la información que tiene disponible, se garantiza que dicha conclusión es correcta si la información disponible también lo es.

Las **proposiciones** expresan aserciones acerca del mundo (entendido como un potencial entorno real), en este contexto, a las proposiciones se les va a poder asignar uno y solo uno de dos valores, **verdadero** o **falso**, de acuerdo con la sintaxis del lenguaje de representación que se utilice, que es quien especifica la forma de una proposición bien formada. A los dos posibles valores se les denomina **valores de verdad**.

Una lógica también debe definir la semántica del lenguaje, se encarga de definir el valor de verdad de cada proposición respecto a un mundo posible del que se está hablando. Cuando queremos ser más precisos, en vez de mundo posible se hablará del término modelo. Un **modelo** es la abstracción matemática de un mundo posible en la que se desenvuelve la lógica, es decir, es una configuración concreta de valores de verdad que pueden tomar las premisas.

Lógica

El razonamiento lógico es un conjunto de proposiciones que mantienen entre sí relaciones lógicas de tal forma que partiendo de algunas proposiciones dadas a las que denominamos *premisas* podemos llegar deductivamente a un juicio que no teníamos y que denominamos *conclusión*. Si las premisas fueran verdaderas, la conclusión también lo sería. Al proceso seguido se le denomina *inferencia*. La lógica formal es la ciencia de los principios de la validez formal de la inferencia.

Decir que la lógica es una ciencia formal, es decir que trabaja con objetos puramente abstractos, que estudia los principios de la demostración y la inferencia válida, las falacias, las paradojas y la noción de verdad.

- Llamemos **proposición** a un enunciado al que se le puede asignar uno y solo uno de los valores verdadero o falso dependiendo del mundo en el que estemos inmersos.
- Un **argumento** es la expresión de un razonamiento, mediante el cual se intenta probar, refutar o justificar una proposición o tesis, partiendo de un conjunto de premisas.
- La **demostración** o bien prueba, es un argumento deductivo construido para asegurar la verdad de una proposición.
- La **inferencia** es el proceso por el cual se derivan conclusiones a partir de premisas. En este contexto, una premisa es cada una de las proposiciones anteriores a la conclusión del argumento. **Cuando una inferencia es aceptable, lo es por su estructura lógica y no por el contenido específico del argumento.** Por eso se pueden construir sistemas formales que capturan los factores relevantes de las deducciones. Tradicionalmente se distinguen tres clases de inferencias:
 - **Deducciones.** Un razonamiento deductivo es un argumento donde la conclusión se infiere necesariamente de las premisas.
 - **Inducciones.** Un razonamiento inductivo es una forma de razonamiento en que la verdad de las premisas apoya la conclusión, pero no la garantizan.
 - **Abducciones.** Es un tipo de razonamiento que a partir de la descripción de un hecho o fenómeno ofrece o llega a una hipótesis, la cual explica las posibles razones o motivos del hecho mediante las premisas obtenidas.

- Recordemos que se denomina **modelo** a una configuración concreta de valores de verdad que pueden tomar todas las premisas de entre todas las posibles.
- **Consecuencia.** Si se puede derivar una nueva proposición correcta a partir de otras (premisas), entonces la nueva proposición es consecuencia lógica de las primeras. Cada modelo que haga ciertas las premisas debe hacer cierta al **consecuente**. Entonces se dice que hay **consistencia**. Si existe alguna situación que haga ciertas a las premisas, pero haga falsa la consecuencia, entonces se dice que hay **inconsistencia**. Un algoritmo que solo deriva proposiciones que son consecuencias, se dice que es correcto, ya que preserva la verdad.

Dos proposiciones son **equivalentes** si cada una es consecuencia de la otra. O lo que es equivalente, si tienen los mismos valores de verdad en el mismo conjunto de modelos. Su notación es $\alpha \vdash \beta$.

- La **verdad** es la coincidencia entre una afirmación y los hechos, o la realidad a la que dicha afirmación se refiere o la fidelidad a una idea.

Cuando en un modelo se comprueba que una proposición es verdad, se dice que el modelo **satisface** a la proposición o que la proposición es satisfecha por el modelo.

Cuando en un modelo una proposición es verdadera siempre, en cualquier situación, se denomina **tautología** ($p \vee \neg p$). Su negación se denomina absurdo ($p \wedge \neg p$).

Cuando en un modelo una proposición no es verdadera en ninguna situación, se denomina **contradicción** o *no satisfacible*.

Cuando se tiene una proposición que es verdadera o falsa dependiendo del modelo en el que se enuncie, se denomina **contingente** ($p \vee q$).

- La **validez** es una propiedad que tienen los argumentos cuando las premisas implican la conclusión. Si la conclusión es una consecuencia lógica de las premisas, se dice que el argumento es deductivamente válido. Para que un argumento sea deductivamente válido, no es necesario que las premisas o la conclusión sean verdaderas, solo se requiere que la conclusión sea una

consecuencia lógica de las premisas. La lógica formal exige únicamente una relación condicional entre las premisas y las conclusiones:

- Si las premisas son verdaderas, entonces la conclusión también lo es (caracterización semántica de la noción de consecuencia lógica).
- Que la conclusión sea deducible de las premisas conforme a las reglas de un sistema lógico (caracterización sintáctica de la noción de consecuencia lógica).
- Si un argumento, además de ser válido, tiene premisas verdaderas, entonces se dice que es sólido.

Si una base de conocimiento (conjunto de premisas) es verdadera en el mundo real, entonces cualquier proposición que se derive de la base mediante un procedimiento de inferencia sólido también será verdadera en el mundo real.

- La validez y la satisfactibilidad están íntimamente relacionadas:
 - Una proposición α es válida si y solo si $\neg\alpha$ es no satisfacible.
 - En contraposición, α es satisfacible si y solo si $\neg\alpha$ no es válida.
- Una falacia es un argumento que parece válido, pero no lo es. Que un argumento sea falaz no implica que sus premisas o su conclusión sean falsas ni que sean verdaderas. Lo que hace falaz a un argumento es la invalidez del argumento en sí.
- Una paradoja es una idea extraña opuesta a lo que se considera verdadero a la opinión general. También se considera paradoja a una proposición en apariencia falsa o que infringe el sentido común, pero no conlleva una contradicción lógica.

La lógica computacional es la aplicación de la lógica matemática a las ciencias de la computación. Ambas estudian el problema de la inferencia mediante sistemas formales como la lógica proposicional, la lógica de primer orden, la lógica modal...

- Un sistema formal o sistema lógico es un sistema abstracto compuesto por un lenguaje formal, axiomas, reglas de inferencia y a veces una semántica formal, que se utiliza para deducir o demostrar y dar una definición rigurosa del concepto de *demostración*. Las cualidades fundamentales de un sistema lógico son:

- La **consistencia** es la propiedad que tienen los sistemas formales cuando no es posible deducir una contradicción dentro del sistema. Es decir, no es posible deducir una fórmula y su negación.
- La **completitud** es la propiedad que tienen los sistemas formales cuando todas las fórmulas lógicamente válidas del sistema pertenecen al sistema lógico. O lo que es lo mismo, si un algoritmo puede derivar cualquier consecuencia posible.

Lógica proposicional

Recordemos que una **proposición** es un enunciado al que se le puede asignar uno y solo uno de los valores verdadero o falso dependiendo del mundo (modelo) en el que estemos inmersos.

Ejemplos de proposiciones: Mañana es miércoles, Hoy está soleado, No estoy corriendo.

Ejemplos de no proposiciones: ¡Hola!

Existen proposiciones simples como las anteriores, pero también proposiciones compuestas construidas mediante **conectores** como «y», «o», «si entonces»...

Ejemplos de proposiciones compuestas: Mañana es miércoles o mañana es jueves, Si llueve entonces cogeré el paraguas.

Considérese el siguiente **argumento**:

- Mañana es miércoles o mañana es jueves.
- Mañana no es jueves.
- Por lo tanto, mañana es miércoles.

Este argumento es válido.

Considérese otro argumento:

- Hoy está soleado o está nublado.
- Hoy no está nublado.
- Por lo tanto, hoy está soleado.

De nuevo, este argumento es válido.

Considérese otro argumento:

- Estoy caminando o estoy corriendo.
- No estoy corriendo.
- Por lo tanto, estoy caminando.

Este argumento es válido.

La validez de los argumentos anteriores no depende del significado de las expresiones que hemos utilizado, sino de que las proposiciones tengan un valor de verdad y de la estructura del argumento. En estos ejemplos dicha estructura depende del significado de las expresiones «o» y «no».

Considérese este otro argumento:

- Xnhgrj ndgfro **o** xnhgrj hgrfseto.
- **No** xnhgrj hgrfseto.
- **Por lo tanto**, xnhgrj ndgfro.

Sin ninguna duda, este argumento también es válido.

Con objeto de escribir menos y representar exclusivamente la estructura del argumento, se va a sustituir a las expresiones por letras como p, q, r, s, etc.

Por ejemplo, todos los argumentos anteriores podrían representarse por:

- p **o** q.
- **No** q.
- **Por lo tanto**, p.

A continuación se muestra una tabla que muestra todas las **conectivas lógicas** que se utilizan y los símbolos que los representan en lenguaje formal de la lógica proposicional.

<i>Conectiva</i>	<i>Lenguaje natural</i>	<i>Ejemplos</i>	<i>Símbolo</i>
Negación	no	No está lloviendo	\neg
Conjunción	y	Está lloviendo y está nublado	\wedge
Disyunción	o	Está lloviendo o está soleado	\vee
Condición material	si... entonces	Si está soleado entonces es de día	\rightarrow
Bicondicional	si y solo si	Está nublado si y solo si hay nubes visibles	\leftrightarrow
Disyunción opuesta	ni... ni	Ni está soleado ni está nublado	\downarrow
Disyunción exclusiva	o bien... o bien	O bien está soleado o bien está nublado	\oplus

El valor de verdad (verdadero o falso) de una proposición compuesta puede determinarse perfectamente si se conoce el valor (verdadero o falso) de las proposiciones simples que la componen y de la conectividad que las une. Teniendo en cuenta el valor que la conectividad devuelve frente a todas las combinaciones posibles de valores de verdad que puede recibir. Estas están definidas mediante lo que se conoce como *tablas de verdad*.

p	q	$\neg p$	$p \wedge q$	$p \vee q$	$p \rightarrow q$	$p \leftrightarrow q$	$p \downarrow q$	$p \oplus q$
V	V	F	V	V	V	V	F	F
F	V	V	F	V	V	F	F	V
V	F		F	V	F	F	F	V
F	F		F	F	V	V	V	F

Por ejemplo, determinar el valor de verdad de la siguiente proposición compuesta $[(p \wedge (\neg q)) \vee r] \rightarrow ((\neg p) \vee q)$.

p	q	r	$\neg p$	$\neg q$	$p \wedge (\neg q)$	$[(p \wedge (\neg q)) \vee r]$	$(\neg p) \vee q$	$[(p \wedge (\neg q)) \vee r] \rightarrow ((\neg p) \vee q)$
V	V	V	F	F	F	V	V	V
V	V	F	F	F	F	F	V	V
V	F	V	F	V	V	V	F	F
V	F	F	F	V	V	V	F	F
F	V	V	V	F	F	V	V	V
F	V	F	V	F	F	F	V	V
F	F	V	V	V	F	V	V	V
F	F	F	V	V	F	F	V	V

El orden seguido es el siguiente:

- En la proposición aparecen tres proposiciones p, q y r, por lo tanto, hay 2^3 posibilidades de los valores de verdad, todos los que aparecen en negro. Cada configuración concreta de valores para p, q y r, se denomina modelo. En este ejemplo hay 8 modelos.
- Dados esos valores de verdad se pueden calcular los valores en azul.
- Posteriormente los morados.
- Después los verdes.
- Por fin los naranjas.

Cada configuración posible de las proposiciones simples, se denomina *modelo*. Por lo tanto, lo que se acaba de calcular es la validez de la proposición compuesta $[(p \wedge (\neg q)) \vee r] \rightarrow ((\neg p) \vee q)$, para cada modelo posible para p, q y r.

Métodos de inferencia: Son los procedimientos que existen para partiendo de un conjunto de proposiciones llamadas premisas se alcance otra proposición llamada conclusión. Se trata de transmitir la verdad de las premisas a la conclusión.

Existen diferentes métodos de inferencia que se denominan:

- Tablas de verdad.
- Deducción mediante reglas de inferencia.
- Resolución.
- Encadenamiento hacia adelante.
- Encadenamiento hacia atrás.
- Tableaux.

Tablas de verdad. Se construye una base de conocimiento. Recordemos que una **base de conocimiento (BC)** recoge todo el conocimiento que el sistema tiene de partida **conocimiento de antecedentes**. Sea BC una base de conocimiento y α una nueva proposición o pregunta, el objetivo consiste en saber si se puede derivar α de BC.

En el caso de la inferencia mediante tablas de verdad el procedimiento que se sigue sería:

- Crear la tabla de verdad con todos los valores posibles de las proposiciones simples y compuestas que aparecen en la BC.
- Se obtienen todos los modelos que hacen cierta a la BC.
- Se mira si la pregunta α es cierta para todos ellos. Si la respuesta es que sí, entonces se dice que α se puede derivar.

Las propiedades de este método son (en azul las ventajas, en rojo las desventajas):

- **Complejidad temporal:** exponencial $O(2^n)$.
- **Complejidad espacial:** exponencial $O(2^n)$.

Es un algoritmo de fuerza bruta.

Deducción mediante reglas de inferencia. No siempre se puede inferir inmediatamente la conclusión de un conjunto de premisas sino que, por el contrario, es necesario llegar a la conclusión a través de pasos sucesivos, cada paso en función de alguna regla. Este proceso recibe el nombre de *deducción* o *demostración natural*. Este método fue presentado en 1934 por Gentzen. El concepto de *deducción* puede seguir diferentes estrategias. La deducción directa y la indirecta (reducción al absurdo). La deducción natural busca capturar la manera en que las personas razonan naturalmente al construir demostraciones matemáticas, sirve para intentar demostrar que un razonamiento es correcto, no sirve para demostrar la invalidez de una suposición.

En un sistema de deducción natural se distinguen dos clases de reglas:

- Las reglas de inferencia.
- Las reglas de construcción de una deducción.

Las reglas mínimas de inferencia que permiten llegar a la conclusión son:

- **Introducción de la conjunción.** Esta regla dice que si en una línea tenemos escrita una cosa cierta (A), y en otra línea tenemos otra (B), también cierta, podemos dejar escrito en una sola línea que las dos cosas son ciertas ($A \wedge B$).
- **Eliminación de la conjunción.** Se pueden separar en varias líneas los elementos conjuntados que aparecen a los lados del operador \wedge . De ($A \wedge B$) se puede inferir A y se puede inferir B .
- **Introducción de la implicación.** Si hemos supuesto algo (llamémosle A), y hemos descubierto mediante reglas que suponer A hace cierto a B (lo que sea), entonces tenemos una cosa clara: no podemos asegurar que B siempre se cumple, pero sí que $A \rightarrow B$.
- **Eliminación de la implicación.** Si cuando pasa A también pasa B ($A \rightarrow B$), y también nos dicen que ahora pasa A , entonces podemos asegurar B . (Regla llamada Modus ponens).
- **Introducción de la disyunción.** Si sabemos que se cumple A podemos deducir que se cumple ($A \vee B$) y ($B \vee A$).
- **Eliminación de la disyunción.** De ($A \vee B$) y ($A \rightarrow C$) y ($B \rightarrow C$), se infiere C .
- **Introducción de la negación.** Si al suponer ($A \rightarrow B$) y ($A \rightarrow \neg B$), se infiere $\neg A$.

- **Eliminación de la negación.** De $\neg A$, se infiere $(A \rightarrow B)$.
- **Eliminación de la doble negación.** De $\neg\neg A$, se infiere A .

Solo con esas reglas bastaría, pero a partir de ellas se suelen construir otras a partir de ellas que facilitan el proceso de demostración. La tabla de reglas de inferencia ampliada suele ser muy útil.

Nombre	Consecuente	Descripción
Modus ponens	$((p \rightarrow q) \wedge p) \vdash q$	Si p entonces q ; p ; por lo tanto q
Modus tollens	$((p \rightarrow q) \wedge \neg q) \vdash \neg p$	Si p entonces q ; no q ; por lo tanto no p
Silogismo hipotético	$((p \rightarrow q) \wedge (q \rightarrow r)) \vdash (p \rightarrow r)$	Si p entonces q ; si q entonces r ; por lo tanto, si p entonces r
Silogismo disyuntivo	$((p \vee q) \wedge \neg p) \vdash q$	Si p o q ; $\neg p$; por lo tanto, q
Dilema constructivo	$((p \rightarrow q) \wedge (r \rightarrow s) \wedge (p \vee r)) \vdash (q \vee s)$	Si p entonces q ; y si r entonces s ; pero p o r ; por lo tanto q o s
Dilema destructivo	$((p \rightarrow q) \wedge (r \rightarrow s) \wedge (\neg q \vee \neg s)) \vdash (\neg p \vee \neg r)$	Si p entonces q ; y si r entonces s ; pero no q o no s ; por lo tanto no p o no r
Dilema bidireccional	$((p \rightarrow q) \wedge (r \rightarrow s) \wedge (p \vee \neg s)) \vdash (q \vee \neg r)$	Si p entonces q ; y si r entonces s ; pero p o no s ; por lo tanto q o no r
Simplificación	$(p \wedge q) \vdash p$	p y q son verdaderos; por lo tanto p es verdadero
Conjunción	$p, q \vdash (p \wedge q)$	p y q son verdaderos separadamente; entonces son verdaderos conjuntamente.
Adición	$p \vdash (p \vee q)$	p es verdadero; por lo tanto la disyunción $(p \vee q)$ es verdadera
Composición	$((p \rightarrow q) \wedge (p \rightarrow r)) \vdash (p \rightarrow (q \wedge r))$	Si p entonces q ; y si p entonces r ; por lo tanto si p es verdadero entonces q y r son verdaderos
Ley de De Morgan(1)	$\neg(p \vee q) \vdash (\neg p \wedge \neg q)$	La negación de $(p \vee q)$ es equivalente a $(\neg p \wedge \neg q)$
Ley de De Morgan(2)	$\neg(p \wedge q) \vdash (\neg p \vee \neg q)$	La negación de $(p \wedge q)$ es equivalente a $(\neg p \vee \neg q)$
Conmutación (1)	$(p \vee q) \vdash (q \vee p)$	$(p \vee q)$ es equivalente a $(q \vee p)$
Conmutación (2)	$(p \wedge q) \vdash (q \wedge p)$	$(p \wedge q)$ es equivalente a $(q \wedge p)$
Conmutación (3)	$(p \leftrightarrow q) \vdash (q \leftrightarrow p)$	$(p$ es equivalente a $q)$ es equivalente a $(q$ es equivalente a $p)$
Asociación (1)	$(p \vee (q \vee r)) \vdash ((p \vee q) \vee r)$	p o $(q$ o $r)$ es equivalente a $(p$ o $q)$ o r
Asociación (2)	$(p \wedge (q \wedge r)) \vdash ((p \wedge q) \wedge r)$	p y $(q$ y $r)$ es equivalente a $(p$ y $q)$ y r
Distribución (1)	$(p \wedge (q \vee r)) \vdash ((p \wedge q) \vee (p \wedge r))$	p y $(q$ o $r)$ es equivalente a $(p$ y $q)$ o $(p$ y $r)$
Distribución (2)	$(p \vee (q \wedge r)) \vdash ((p \vee q) \wedge (p \vee r))$	p o $(q$ y $r)$ es equivalente a $(p$ o $q)$ y $(p$ o $r)$
Doble negación	$p \vdash \neg\neg p$	p es equivalente a la negación de no p
Transposición	$(p \rightarrow q) \vdash (\neg q \rightarrow \neg p)$	Si p entonces q es equivalente a si no q entonces no p
Implicación material	$(p \rightarrow q) \vdash (\neg p \vee q)$	Si p entonces q es equivalente a no p o q

Equivalencia material(1)	$(p \leftrightarrow q) \vdash ((p \rightarrow q) \wedge (q \rightarrow p))$	$(p$ si y solo si $q)$ es equivalente a $($ si p es verdadero entonces q es verdadero) y $($ si q es verdadero entonces p es verdadero)
Equivalencia material (2)	$(p \leftrightarrow q) \vdash ((p \wedge q) \vee (\neg p \wedge \neg q))$	$(p$ si $q)$ es equivalente a cualquiera de los dos $(p$ y q son verdaderos) o $($ tanto p como q son falsos)
Equivalencia material (3)	$(p \leftrightarrow q) \vdash ((p \vee \neg q) \wedge (\neg p \vee q))$	$(p$ si $q)$ es equivalente a: tanto $(p$ como no q son verdaderos) y $($ no p o q es verdadero)
Exportación ³	$((p \wedge q) \rightarrow r) \vdash (p \rightarrow (q \rightarrow r))$	desde $($ si p y q son verdaderos, entonces r es verdadero) se puede probar que $($ si q es verdadero entonces r es verdadero, si p es verdadero)
Importación	$(p \rightarrow (q \rightarrow r)) \vdash ((p \wedge q) \rightarrow r)$	p implica que q implica r es equivalente a que p y q implican r
Tautología (1)	$p \vdash (p \vee p)$	p es verdadero es equivalente a p es verdadero o p es verdadero
Tautología (2)	$p \vdash (p \wedge p)$	p es verdadero es equivalente a p es verdadero y p es verdadero
Principio del tercero excluido	$\vdash (p \vee \neg p)$	p o no p es verdadero
Principio de no contradicción	$\vdash \neg(p \wedge \neg p)$	p y no p es falso

El símbolo \vdash indica «por lo tanto».

Las **reglas de construcción** de una deducción son:

- Un supuesto inicial o premisa inicial, fórmulas hipotéticamente dadas desde el principio de la derivación, o
- Un supuesto provisional o subsidiario, que debe estar cancelado antes de la conclusión, o
- Una fórmula derivada lógicamente de las anteriores por inferencia inmediata, que denominaremos *consecuencias lógicas inmediatas*.
- La última línea de la derivación es la conclusión.

Uno puede llegar a pensar que el proceso seguido para partiendo de unas premisas se llega a una conclusión es un proceso único, pero la respuesta es que no, en problemas complejos, existen numerosas formas correctas de hacerlo. Y el proceso que se sigue no es innato, hay que aprender a hacerlo ejercitándose bastante. Pero lo bueno es que es aprendible.

La pregunta que queda por hacer en este contexto es: ¿Hay programas que hagan deducción natural para problemas reales? La verdad es que no conozco ninguno y creo que no hay salvo, programas didácticos.

Ejemplo: Estudiar si es correcto el siguiente razonamiento. Para ello se utilizará el método directo que consiste en los pasos siguientes: Es necesario identificar las afirmaciones, expresarlas de forma simbólica, identificar las premisas y la conclusión y escribirlas como proposiciones compuestas definidas en términos de las afirmaciones. Finalmente se deben organizar en una tabla de deducción y utilizar las reglas de inferencia y las equivalencias lógicas para demostrar que la conclusión es una consecuencia lógica de las premisas.

La ballena es un mamífero, entonces toma oxígeno del aire. Si toma oxígeno del aire, entonces no necesita branquias. La ballena es un mamífero y vive en el océano. Por lo tanto, no necesita branquias.

La conclusión que se desea deducir es la proposición «no necesita branquias».

Simbolización de las proposiciones:

- p «la ballena es un mamífero».
- q «toma oxígeno del aire».
- r «necesita branquias».
- s «habita en el océano».

Simbolización de las premisas:

1. $p \rightarrow q$
2. $q \rightarrow \neg r$
3. $p \wedge s$

Conclusión a la que hay que llegar es $\neg r$.

Deducción proposicional:

1. $p \rightarrow q$ es una premisa.
2. $q \rightarrow \neg r$ es una premisa.
3. $p \wedge s$ es una premisa.
4. p cierta por la regla de simplificación aplicada al paso 3 [la regla 8 de la tabla anterior].
5. q cierta por la regla *modus ponens* aplicada a los pasos 1 y 4 [la regla 8 de la tabla anterior].
6. $\neg r$ cierta por la regla *modus ponens* aplicada a los pasos 2 y 5 [la regla 1 de la tabla anterior].

Puesto que la línea 6 representa la conclusión deseada, la deducción es completa.

Ejemplo: Estudiar si es correcto el siguiente razonamiento. Para ello se utilizará el método indirecto o de *reductio ad absurdum* que consiste en los pasos siguientes: Se inicia asumiendo la negación de la conclusión que se desea probar. Si a partir de la suposición y de las premisas se llega a una contradicción, entonces se concluye que la suposición es falsa. La conclusión se considera verdadera por ser la negación de la suposición. Las proposiciones intermedias generadas durante la deducción indirecta no se pueden usar fuera del alcance de esta última porque dependen de una suposición.

Si no hay probabilidad de lluvia y la máquina de cortar funciona, el jardinero cortará el césped. Si la temperatura es superior a los 25 °C, no hay probabilidades de lluvia. Hoy la temperatura es de 30 °C, y la máquina funciona, por lo tanto el jardinero cortará el césped.

La conclusión que se desea deducir es la proposición «el jardinero cortará el césped».

Simbolización de las proposiciones:

- p «hay probabilidad de lluvia».
- q «la máquina de cortar funciona».
- r «el jardinero cortará el césped».
- s «la temperatura es superior a los 25 °C».

Las premisas y la conclusión en formato proposicional son:

1. $(\neg p \wedge q) \rightarrow r$. Si no hay probabilidad de lluvia y la máquina de cortar funciona, el jardinero cortará el césped.
2. $s \rightarrow \neg p$. Si la temperatura es superior a 25 °C, no hay probabilidades de lluvia.
3. $s \wedge p$. Hoy la temperatura es de 30 °C, y la máquina funciona.

Conclusión: r.

Deducción proposicional:

1. $(\neg p \wedge q) \rightarrow r$ es una premisa.
2. $s \rightarrow \neg p$ es una premisa.
3. $s \wedge q$ es una premisa.
4. s cierta por la regla de simplificación aplicada al paso 3
[la regla 8 de la tabla anterior].
5. $\neg p$ cierta por la regla *modus ponens* aplicada a los pasos 2, 4
[la regla 1 de la tabla anterior].
6. $q \wedge s$ por la regla conmutativa aplicada al paso 3
[la regla 15 de la tabla anterior].
7. q por la regla de simplificación aplicada al paso 6
[la regla 8 de la tabla anterior].
8. $\neg r$ Suposición para realizar la demostración indirecta
9. $\neg(\neg p \wedge r)$ por la regla *modus tollens* aplicada a los pasos 1, 8
[la regla 2 de la tabla anterior].
10. $\neg\neg p \vee \neg q$ por la regla de Morgan para la conjunción aplicada a 9
[la regla 12 de la tabla anterior].
11. $\neg q \vee \neg\neg p$ por la regla de la conmutatividad aplicada al paso 10
[la regla 14 de la tabla anterior].
12. $\neg\neg p$ por la regla del silogismo disyuntivo aplicada a los pasos 7,
11
[la regla 4 de la tabla anterior].
13. p por la regla de la doble negación a 12
[la regla 21 de la tabla anterior].
14. $p \wedge \neg p$ por la regla de la conjunción a 5, 13
[la regla 9 de la tabla anterior] ¡Absurdo!
15. r Deducción indirecta (reducción al absurdo) teniendo en
cuenta los pasos 8, 9, 10, 11, 12, 13, 14

Resolución. En este momento, se va a introducir una regla de inferencia sencilla, denominada *resolución*, que va a permitir construir un algoritmo de inferencia completo cuando se empareja con un algoritmo de búsqueda completo. Se utiliza para construir demostradores automatizados de teoremas. Los demostradores contruidos son completos (actuando por contradicción) y correctos.

Este procedimiento solo funciona cuando se escriben las proposiciones en términos de premisas que constan solo de disyunciones y literales (una proposición es literal o atómica cuando es indivisible y solo puede ser positiva o negativa). Puesto que toda proposición lógica se puede escribir en términos de disyunciones, conjunciones y negaciones, lo anterior no supone una limitación del método más allá de transformar proposiciones lógicas. A las proposiciones escritas en esta forma se las denomina *cláusulas*.

Toda sentencia en lógica proposicional es equivalente lógicamente a una conjunción de disyunciones de literales. Una sentencia representada mediante una conjunción de disyunciones de literales se dice que está en forma normal conjuntiva (FNC).

Por ejemplo: $(\neg p \vee q) \wedge (\neg q \vee p)$.

Otro ejemplo: La FNC de $\neg(p \wedge (q \rightarrow r))$ es $(\neg p \vee q) \wedge (\neg p \vee \neg r)$.

Algoritmo de cálculo de forma normal conjuntiva para una fórmula F. Se utilizará el símbolo \equiv para indicar equivalencia:

- Eliminamos los bicondicionales usando la equivalencia
 1. $A \leftrightarrow B \equiv (A \rightarrow B) \wedge (B \rightarrow A)$
- Eliminamos los condicionales usando la equivalencia
 2. $A \rightarrow B \equiv \neg A \vee B$
- Interiorizar las negaciones usando las equivalencias
 3. $\neg(A \wedge B) \equiv \neg A \vee \neg B$
 4. $\neg(A \vee B) \equiv \neg A \wedge \neg B$
 5. $\neg\neg A \equiv A$
- Interiorizar las disyunciones usando las equivalencias
 6. $A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C)$
 7. $(A \wedge B) \vee C \equiv (A \vee C) \wedge (B \vee C)$

Ejemplo de cálculo de la FNC de $\neg(p \wedge (q \rightarrow r))$:

$$\begin{aligned} & \neg(p \wedge (q \rightarrow r)) \\ \equiv & \neg(p \wedge (\neg q \vee r)) && [\text{por (2)}] \\ \equiv & \neg p \vee \neg(\neg q \vee r) && [\text{por (3)}] \\ \equiv & \neg p \vee (\neg\neg q \wedge \neg r) && [\text{por (4)}] \\ \equiv & \neg p \vee (q \wedge \neg r) && [\text{por (5)}] \\ \equiv & (\neg p \vee q) \wedge (\neg p \vee \neg r) && [\text{por (6)}] \end{aligned}$$

Teniendo en cuenta que la forma sintáctica general de una fórmula FNC siempre tiene la forma

$$(L_{1,1} \vee L_{1,2} \vee L_{1,3} \vee L_{1,4} \vee \dots L_{1,n-1} \vee L_{1,n}) \wedge \dots \wedge (L_{m,1} \vee L_{m,2} \vee L_{m,3} \vee L_{m,4} \vee \dots L_{m,n-1} \vee L_{m,s})$$

Podemos escribir de forma más compacta ahorrándonos símbolos utilizando la siguiente notación denominada cláusulas:

- La cláusula de $(L_{1,1} \vee L_{1,2} \vee L_{1,3} \vee L_{1,4} \vee \dots L_{1,n-1} \vee L_{1,n})$ es

$$\{L_{1,1}, L_{1,2}, L_{1,3}, L_{1,4}, \dots L_{1,n-1}, L_{1,n}\}$$

- La cláusula de $(L_{m,1} \vee L_{m,2} \vee L_{m,3} \vee L_{m,4} \vee \dots L_{m,n-1} \vee L_{m,s})$ es

$$\{L_{m,1}, L_{m,2}, L_{m,3}, L_{m,4}, \dots L_{m,n-1}, L_{m,s}\}$$

- La cláusula de

$$(L_{1,1} \vee L_{1,2} \vee L_{1,3} \vee L_{1,4} \vee \dots L_{1,n-1} \vee L_{1,n}) \wedge \dots \wedge (L_{m,1} \vee L_{m,2} \vee L_{m,3} \vee L_{m,4} \vee \dots L_{m,n-1} \vee L_{m,s})$$

sería

$$\{\{L_{1,1}, L_{1,2}, L_{1,3}, L_{1,4}, \dots L_{1,n-1}, L_{1,n}\}, \dots, \{L_{m,1}, L_{m,2}, L_{m,3}, L_{m,4}, \dots L_{m,n-1}, L_{m,s}\}\}$$

Ejemplo: Acabamos de ver que la FNC de $\neg(p \wedge (q \rightarrow r))$ es $(\neg p \vee q) \wedge (\neg p \vee \neg r)$ a la que le corresponde la cláusula $\{\{\neg p, q\}, \{\neg p, \neg r\}\}$.

De fórmulas a cláusulas:

- Una forma clausal de una fórmula F expresada en la forma FNC es un conjunto de cláusulas equivalente a F .
- Una forma clausal de un conjunto de fórmulas $\{F_1, F_2, \dots, F_n\}$ es un conjunto de cláusulas $\{S_1, S_2, \dots, S_n\}$, que verifican que $S_1 \cup S_2 \cup \dots \cup S_n$ es una forma clausal de $\{F_1, F_2, \dots, F_n\}$

Los procedimientos de inferencia basados en la resolución trabajan utilizando el principio de prueba mediante contradicción, es decir, para demostrar que a partir de una base de conocimiento (BC), α se deduce de ella mediante un algoritmo «i», lo que se representa $BC \vdash_i \alpha$, lo que hay que demostrar es que $(BC \wedge \neg\alpha)$ es insatisfacible.

Las reglas de resolución son simples:

- Si en una cláusula aparece un literal y en otra aparece la negación del mismo literal, se pueden unir ambas cláusulas eliminando dicho literal.

$$\{p_1, p_2, \dots, r, \dots, p_n\}$$

$$\{q_1, q_2, \dots, \neg r, \dots, q_n\}$$

Resultado

$$\{p_1, p_2, \dots, p_n, q_1, q_2, \dots, q_n\}$$

- Si en una cláusula aparece un literal y su negación, se puede eliminar dicha cláusula, ya que equivale a verdadera.

$$\{p_1, p_2, \dots, r, \neg r, \dots, p_n\} \equiv \text{Verdadero}$$

Ejemplo de refutación por resolución: Refutar $[\{p, q\}, \{\neg p, q\}, \{p, \neg q\}, \{\neg p, \neg q\}]$

1. $\{p, q\}$ Por hipótesis.
2. $\{\neg p, q\}$ Por hipótesis.
3. $\{p, \neg q\}$ Por hipótesis.
4. $\{\neg p, \neg q\}$ Por hipótesis.
5. $\{q\}$ Resolvente de 1 y 2.
6. $\{\neg q\}$ Resolvente de 3 y 4.
7. absurdo Resolvente de 5 y 6.

Como hemos dicho, para demostrar que a partir de una base de conocimiento (BC), que α se deduce de ella mediante un algoritmo «i», lo que hay que demostrar es que $(BC \wedge \neg\alpha)$ es insatisfacible. Por lo tanto, el procedimiento de trabajo consiste en:

- Construir la base de conocimiento ampliada $(BC \wedge \neg\alpha)$.
- Expresarla en formato FNC.
- Aplicar las reglas de resolución indicadas.
- El proceso continúa hasta que sucede una de estas dos cosas:
 - No hay nuevas cláusulas que se puedan añadir, en cuyo caso no es factible la deducción $BC \vdash_i \alpha$.
 - Se deriva la cláusula vacía en cuyo caso $BC \vdash_i \alpha$.

5. $\text{tiene_pelos} \wedge \text{tiene_pezuñas} \wedge \text{tiene_rayas_negras}$

Forma cláusula { tiene_pelos } { tiene_pezuñas } { $\text{tiene_rayas_negras}$ }

Argumentación y resolución:

- | | |
|--|------------------------|
| 1. { \neg tiene pelos, es mamífero} | Hipótesis. |
| 2. { \neg da leche, es mamífero} | Hipótesis. |
| 3. { \neg es mamífero, \neg tiene pezuñas, es ungulado} | Hipótesis. |
| 4. { \neg es mamífero, \neg rumia, es ungulado} | Hipótesis. |
| 5. { \neg es ungulado, \neg tiene cuello largo, es jirafa} | Hipótesis. |
| 6. { \neg es ungulado, \neg tiene rayas negras, es cebra} | Hipótesis. |
| 7. {tiene pelos} | Hipótesis. |
| 8. {tiene pezuñas} | Hipótesis. |
| 9. {tiene rayas negras} | Hipótesis. |
| 10. {-es cebra} | Hipótesis. |
| 11. {es mamífero} | Resolvente de 1 y 7. |
| 12. { \neg tiene pezuñas, es ungulado} | Resolvente de 11 y 3. |
| 13. {es ungulado} | Resolvente de 12 y 8. |
| 14. { \neg tiene rayas negras, es cebra} | Resolvente de 13 y 6. |
| 15. {es cebra} | Resolvente de 14 y 9. |
| 16. Cláusula vacía | Resolvente de 15 y 10. |

⊢ Resolución es_cebra

La pregunta que queda por hacer en este contexto es: ¿Hay programas que implementen el método de la resolución para problemas reales? La respuesta es afirmativa, ya que si se observa el procedimiento que se sigue, no es más que un mero procesado de símbolos fácilmente automatizable y las reglas de inferencia son simples y actúan sobre símbolos.

Cláusulas de Horn

En muchos casos prácticos no se necesita todo el poder del método anterior (Resolución). Las bases de conocimiento en el mundo real a menudo contienen solo cláusulas, de un tipo restringido, denominadas *cláusulas de Horn*.

Una cláusula de Horn es una disyunción de literales, es decir, está formada por proposiciones atómicas, de las cuales como mucho una es positiva.

Ejemplo de cláusula de Horn: $\neg p \vee \neg q \vee \neg r \vee \dots \vee \neg t \vee u$

Este tipo de fórmula también puede escribirse cómo

$$(p \wedge q \wedge r \wedge \dots \wedge t) \rightarrow u$$

Denominando:

Cuerpo a $(p \wedge q \wedge r \wedge \dots \wedge t)$

Cabeza a u

Un hecho q puede considerarse una regla con cuerpo vacío $\rightarrow u$

La inferencia en las BC con cláusulas de Horn se pueden realizar mediante los algoritmos denominados, «de encadenamiento hacia adelante» y «de encadenamiento hacia atrás», en ellos los pasos de inferencia son bastante evidentes a la hora de seguirlos. Además, son lineales en tiempo.

El encadenamiento hacia adelante permite un tipo de razonamiento que parte de los datos conocidos para derivar conclusiones. Mientras que el encadenamiento hacia atrás trabaja a partir de la conclusión.

Encadenamiento hacia adelante. Se van agregando conclusiones a la base de conocimiento si todos los literales de una premisa se cumplen. Se continúa hasta que se obtiene la conclusión a comprobar o si no se pueden hacer más inferencias.

Las reglas de inferencia utilizadas son:

- Llamemos $I \wedge a$: A cierta y B cierta $\vdash A \wedge B$.
- *Modus ponens* (MP). $A \rightarrow B, A$ cierta $\vdash B$ cierta.

Ejemplo: Deducir a partir de la $BC \vdash Q$

- BC

$$BC = \{ P \rightarrow Q, L \wedge M \rightarrow P, B \wedge L \rightarrow M, A \wedge P \rightarrow L, A \wedge B \rightarrow L, A, B \}$$

- | | |
|--------------------------------|------------------------------|
| 1. A | Hipótesis. |
| 2. B | Hipótesis. |
| 3. $A \wedge B$ | $I \wedge$ aplicada a 1 y 2. |
| 4. $A \wedge B \rightarrow L$ | Hipótesis. |
| 5. L | MP aplicada a 3 y 4. |
| 6. $B \wedge L$ | $I \wedge$ aplicada a 2 y 5. |
| 7. $B \wedge L \rightarrow M$ | Hipótesis. |
| 8. M | MP aplicada a 6 y 7. |
| 9. $L \wedge M$ | $I \wedge$ aplicada a 5 y 8. |
| 10. $L \wedge M \rightarrow P$ | Hipótesis. |
| 11. P | MP aplicada a 9 y 10. |
| 12. $P \rightarrow Q$ | Hipótesis. |
| 13. Q | MP aplicada a 11 y 12. |

- Luego $BC \vdash Q$

La pregunta que queda por hacer en este contexto es: ¿Hay programas que implementen el método cláusulas de Horn y encadenamiento hacia adelante? La respuesta es afirmativa, ya que si se observa el procedimiento que se sigue, no es más que un mero procesado de símbolos fácilmente automatizable y las reglas de inferencia son simples y actúan sobre símbolos. De hecho hay hasta un lenguaje de programación OPS-5, que implementa esta metodología, véase

<<https://en.wikipedia.org/wiki/OPS5>>

Encadenamiento hacia atrás:

- Se parte del literal que hay que demostrar.
- Se busca alguna cláusula donde dicho literal sea la conclusión.
- Se mira si sus premisas son verdad.
- Si son verdaderas, se ha finalizado.
- Si no lo son, se repite la operación previa para cada una de esas premisas.
- Se procede de forma recursiva hasta que todas las premisas se satisfagan.

Ejemplo: Deducir a partir de la BC $\vdash r$

Supongamos que un paciente va al doctor, el doctor después de escuchar el problema del paciente cree que tiene una infección de garganta. Ahora bien, veremos como un sistema experto basado en reglas de encadenamiento hacia atrás puede solucionar este problema.

- p hay señales de infección de garganta.
- q hay evidencia de presencia de estreptococo.
- r el paciente tiene una infección de garganta.
- t el teñido del organismo es oscuro.
- u la morfología del organismo es coco.
- v el crecimiento del organismo es en cadena.

BC = { $p \wedge q \rightarrow r$, $t \wedge u \wedge v \rightarrow q$, p, t, u, v } Demostrar r

1. p Hecho.
2. t Hecho.
3. u Hecho.
4. v Hecho.
5. $p \wedge q \rightarrow r$.
6. $t \wedge u \wedge v \rightarrow q$.

- Para demostrar r, por 5 hay que demostrar que es verdad p y q.
- Para demostrar p y q, hay que demostrar por un lado p y por otro q.
- A partir de 1, p es válida.
- Para demostrar que q, por 6 hay que demostrar que es verdad t y u y v.
- Para demostrar t y u y v, hay que demostrar independientemente que son ciertos t, u y v.
- Por 2, t es cierto.
- Por 3, u es cierto.
- Por 4, v es cierto.

Por lo tanto, el paciente tiene una infección de garganta.

Otro ejemplo, supongamos que una nueva mascota, *Gustavo*, se entrega en una caja opaca junto con dos hechos sobre *Gustavo*: croa y come moscas.

El objetivo es decidir si *Gustavo* es de color verde, según una base de conocimiento que contiene las seis reglas siguientes:

1. p Hecho.
2. q Hecho.
3. $p \wedge q \rightarrow r$.
4. $s \rightarrow t$.
5. $r \rightarrow u$.
6. $t \rightarrow v$.

siendo:

- p «croat».
- q «come moscas».
- s «canta».
- u «verde».
- v «amarillo».
- t «es un canario».
- r «es una rana»

Con el razonamiento hacia atrás, veamos si un motor de inferencia puede determinar si *Gustavo* es verde. Para empezar, la consulta se formula como una afirmación objetivo que es ser probada: «*Gustavo* es verde».

Empezamos en 5, para llegar a *Gustavo* es verde (u), es necesario probar que *Gustavo* es una rana (r).

Seguimos por 3, para llegar a *Gustavo* es una rana (r), entonces es necesario probar que «croat» (p) y que «come moscas» (q).

Para probar las dos a la vez, es necesario probar cada una por su lado, pero son exactamente la 1 y la 2.

Por lo tanto, *Gustavo* es verde . Obsérvese que no se han empleado todas la BC.

La pregunta que queda por hacer en este contexto es: ¿Hay programas que implementen el método cláusulas de Horn y encadenamiento hacia atrás? La respuesta es afirmativa, ya que si se observa el procedimiento que se sigue, no es más que un mero procesado de símbolos fácilmente automatizable y las reglas de inferencia son simples y actúan sobre símbolos. De hecho hay hasta un lenguaje de

programación PROLOG, que implementa esta metodología, véase <<https://es.wikipedia.org/wiki/Prolog>>

Existen dos familias de algoritmos eficientes para la inferencia en lógica proposicional. Estos algoritmos se usan para la comprobación de la satisfacibilidad (SAT). A modo de información, se denominan DPLL y WalkSAT.

Para saber más véanse:

- DPLL. El algoritmo Davis-Putnam-Logemann-Loveland (DPLL) es un algoritmo completo basado en la vuelta atrás que sirve para decidir la satisfactibilidad de las fórmulas de lógica proposicional en una forma normal conjuntiva. Véase <https://es.wikipedia.org/wiki/Algoritmo_DPLL>
- WalkSAT y GSAT. Algoritmos de búsqueda local que permiten resolver problemas de satisfacibilidad Booleana convertidos a una forma normal conjuntiva.
Véase <<https://en.wikipedia.org/wiki/WalkSAT>>

Otros tipos de lógica

Primer criterio: La lógica formal moderna se puede ver como lógicas anidadas, en donde sobre la filosofía de una se monta otra que contiene más recursos expresivos, que necesita de nuevos elementos, y se eliminan restricciones del uso de estos recursos. Esto debe interpretarse de una forma monótona ascendente, es decir, que todo lo que vale en un nivel inferior vale para el nivel superior, pero no al revés.

Si describimos desde la más interior a la más exterior podríamos ver:



Segundo criterio: Además existen otras formas de clasificación



Véase <<https://webs.ucm.es/info/pslogica/cdn.pdf>>

Veamos a continuación una sucinta caracterización de cada una de ellas.

Primer criterio

Lógica de proposiciones o de enunciados: Es el cálculo básico, basado en proposiciones (hechos) en las que cada símbolo representa una proposición atómica, Las proposiciones complejas se construyen exclusivamente a partir de las atómicas utilizando las conectivas lógicas (\neg , \wedge , \vee , \rightarrow , \leftrightarrow). La relación lógica que estudiar es la que se establece en una relación condicional entre las premisas y la conclusión. Es decir, si ocurren las premisas, entonces ocurre la conclusión. La lógica proposicional no escala para entornos no acotados en tamaño porque carece de la potencia expresiva para tratar de manera concisa con el tiempo, el espacio y los patrones universales de relación entre objetos.

Lógica de predicados o cuantificacional: Pese a las virtudes que ofrece un sistema de representación y razonamiento formal como la lógica proposicional (LP), esta puede ser no apropiada para expresar ciertos tipos de conocimiento.

Por ejemplo, una sentencia tan sencilla como «Algunas máquinas son inteligentes» solo dispone de una posible representación en LP, como variable proposicional, ya que es una afirmación atómica. Y si representamos esta sentencia por una variable proposicional «p», el sistema no será capaz de diferenciar su contenido de una frase similar pero que da una información bien distinta: «Las máquinas son inteligentes». La primera afirmación no se refiere específicamente a ningún conjunto de máquinas, pero tampoco informa de una propiedad general de las mismas, solo indica que existe un conjunto de máquinas que son inteligentes.

Observemos el siguiente razonamiento que sabemos que es cierto:

- Todos los atenienses son griegos «p».
- Los griegos son europeos «q».
- Los atenienses son europeos «r».

Como este argumento no contiene ninguna de las conectivas «no», «y», «o», etc., según la lógica proposicional, su formalización será la siguiente:

- p.
- q.
- Por lo tanto, r.

La LP es incapaz de deducir nada.

Por ejemplo, considérese el siguiente argumento:

- Todos los hombres son mortales.
- Sócrates es un hombre.
- Por lo tanto, Sócrates es mortal.

Como este argumento no contiene ninguna de las conectivas «no», «y», «o», etc., según la lógica proposicional, su formalización será la siguiente:

1. p.
2. q.
3. Por lo tanto, r.

La LP es incapaz de deducir nada.

Parte del problema que muestra la LP es que no tiene capacidad para diferenciar propiedades en familias de elementos de un dominio. O bien habla de objetos particulares que tienen una descripción propia (un nombre que los identifica), o habla de propiedades del dominio completo como una unidad. No tiene capacidad de ajuste fino.

La lógica de predicados se apoya en el cálculo básico dado por la lógica de proposiciones y a ello se le añade el análisis de las oraciones dividiéndolas en sus componentes, sujeto y predicado.

Ejemplos de proposiciones en lógica de predicados:

- Juan bebe sujeto: Juan predicado: bebe
- Cinco es mayor que dos sujetos: cinco y dos predicado: es mayor que
- El Ebro es caudaloso sujeto: Ebro predicado: es caudaloso

La sintaxis utilizada es:

- Letras minúsculas para los sujetos.
- Letras mayúsculas para los predicados.

Ejemplos de representación sintáctica:

- «i» representa a «Juan» y F representa a «bebe»:
 - Juan bebe = Fi.
- «i» representa a «cinco», j representa a «dos», y F representa a «es mayor que»:
 - Cinco es mayor que dos = F i j.

Esas expresiones sintácticas se denominan *esquemas cuantificacionales atómicos*.

Estos esquemas se pueden combinar utilizando las conectivas lógicas ya conocidas (\neg , \wedge , \vee , \rightarrow , \leftrightarrow).

Ejemplo:

- Si Ernesto (i) canta (F), entonces Ernesto (i) es feliz (G):
 - $F_i \rightarrow G_i$.
- Si Ernesto (i) canta (F), entonces Luis (j) canta (F):
 - $F_i \rightarrow F_j$.
- Tomás (i) canta (F) o Ricardo (j) baila (G):
 - $F_i \vee G_j$.
- Pablo (i) baila (F) si y solo si Juan (j) toca (G):
 - $F_i \leftrightarrow G_j$.

También se pueden escribir esquemas variables.

Ejemplo:

- x es mayor que cinco (H):
 - Hx .
 - Si hacemos $x = 7$, entonces $H7$ es cierta.
 - Si hacemos $x = 3$, entonces $H3$ es falsa.
- « $x < u$ » y « $z > y$ », si $<$ lo representamos por F y $>$ lo representamos por G:
 - $Fxu \wedge Gzy$.

La lógica proposicional añade a las conectivas lógicas ya conocidas (\neg , \wedge , \vee , \rightarrow , \leftrightarrow) dos cuantificadores: « \forall » denominado *cuantificador universal* y « \exists » denominado *cuantificador existencial*.

Ejemplos:

- Todos los asistentes al congreso llevan corbata:
 - Si M es el conjunto de todos los asistentes.
 - Si V es el conjunto de todos los asistentes que llevan corbata.
 - Fx « x asiste al congreso».
 - Gx « x lleva corbata».
 - Entonces $(\forall x) [Fx \rightarrow Gx]$ es verdad si y solo si $V = M$.
- Ninguno quiere ser el primero:
 - Fx «no quiere ser el primero».
 - Entonces $(\forall x) \neg(Fx)$.

- Ningún francés es americano:
 - Fx «ser francés».
 - Gx «ser americano».
 - Entonces $(\forall x) [Fx \rightarrow \neg(Gx)]$.

- Sea M el conjunto de los números $\{1, 2, 3, 4, 5, 6\}$:
 - Fx « x es un número impar».
 - Entonces podemos afirmar que $(\exists x) (Fx)$, que se lee existe un valor de x en M que verifica que es un número impar.

- Todos los estudiantes de Informática son listos:
 - Sujeto x = Estudiante.
 - Predicado:
 - $I(x)$ = «Estudia Informática».
 - $L(x)$ = «Son listos».
 - Cuantificador \forall .
 - Términos $I(X) \rightarrow L(x)$.
 - Entonces $\forall x I(X) \rightarrow L(x)$.

- Vease el siguiente argumento:
 - Todos los hombres son mortales.
 - Sócrates es un hombre.
 - Por lo tanto, Sócrates es mortal.

Demostración:

1. $(\forall x) [Hx \rightarrow Mx]$ H «ser un hombre», M «ser mortal», x Sócrates.
2. Hs $x = s =$ Sócrates.
3. Conclusión Ms Sócrates es mortal.

La lógica de predicados utiliza reglas de inferencia análogas a las de proposiciones, ampliadas con reglas de inferencia útiles para el uso de los cuantificadores. Véase

https://es.wikipedia.org/wiki/Reglas_de_inferencia#:~:text=Las%20reglas%20significativas%20de%20inferencia,para%20liderar%20con%20cuantificadores%20l%C3%B3gicos.>

Como acabamos de ver la lógica de predicados puede cuantificar sobre individuos, es decir, podemos tratar con todos o con algunos de los elementos que pueden ser sujetos de una oración. Es una lógica de clases donde la relación lógica que se estudia es la pertenencia a un conjunto o la posesión de propiedades por los distintos individuos de los que se habla. El análisis lógico del lenguaje que lleva a cabo la lógica

de predicados tiene en cuenta elementos de la proposición que la lógica de enunciados no tiene, por ejemplo, «para todos», «existe un», «ningún»...

Lógica de primer orden: La unión del cálculo proposicional y del cálculo de predicados es lo que se denomina *lógica de 1.º orden*.

La lógica de proposiciones se apoya en hechos, en las que cada símbolo representa una proposición atómica, y utiliza las conectivas lógicas (\neg , \wedge , \vee , \rightarrow , \leftrightarrow).

La lógica de predicados se apoya en el cálculo básico dado por la lógica de proposiciones y a ello se le añade el análisis de las oraciones dividiéndolas en sus componentes, sujeto y predicado. Utiliza las conectivas lógicas (\neg , \wedge , \vee , \rightarrow , \leftrightarrow) y los cuantificadores (\forall y \exists).

La lógica de primer orden añade a la lógica de predicados la posibilidad de establecer relaciones entre las variables.

Ejemplos:

- El rey Felipe VI es el padre de la princesa Leonor:
 - La relación «ser padre de» la podemos expresar del modo siguiente:
 - x «el rey» a Felipe VI.
 - y «la princesa» b Leonor.
 - Si Fxy indica que « x » es padre de « y ».
 - La proposición sería Fab .
- La princesa Leonor tiene un padre $\exists x Fxb$.
- El rey Felipe VI tiene una hija $\exists y Fay$.
- $\exists x Fbx$ nos habla de un hijo de la princesa.
- $\exists x Fxa$ nos habla del padre del rey.
- $\forall x \exists y Fyx$ todos tienen un padre.
- $\forall x \exists y Fxy$ todos son padre de alguien.

La lógica de primer orden trabaja con hechos, objetos (gente, números, colores...) y relaciones unarias (rojo, redondo, primo...), n-arias (hermano de, más grande que, parte de, tiene color...). Permite hacer cuantificación sobre los objetos de un dominio, por ejemplo, puede expresar cosas como: «Todas las máquinas son inteligentes», «Algunas máquinas funcionan correctamente», etc. Permite representar propiedades de los objetos particulares del dominio por medio de predicados y funciones. Permite trabajar con subconjuntos de objetos que pueden

venir caracterizados por propiedades que se describen por medio de propiedades y funciones.

Esta lógica tiene la restricción de que solo se pueden utilizar los cuantificadores con elementos individuales. Es decir, no se puede hablar de todas o de algunas de las clases de algún tipo.

Lógica de órdenes superiores (2.º, 3.º, 4.º...) orden: Estas lógicas se construyen sobre la lógica de primer orden, según admitamos cuantificar sobre propiedades o predicados, o predicados de predicados, iremos subiendo de orden. Existe, por lo tanto, una infinita jerarquía de niveles de predicación:

- Nivel 1. Predicados de individuos.
- Nivel 2. Predicados de predicados de individuos.
- Nivel 3. Predicados de predicados de predicados de individuos.
- ...

También es evidente que podemos cuantificar sobre propiedades y no solo sobre individuos.

Por ejemplo:

- Un padre y un hijo tienen ciertos rasgos en común:
 - Fxy indica «x es padre de y».
 - G actuaría como una variable predicativa.
- Entonces $\forall x \forall y (Fxy \rightarrow \exists G [Gx \wedge Gy])$.

El paso de la lógica de primer orden a la lógica de orden superior es mucho más drástico de lo que pueda imaginarse inicialmente.

Segundo criterio

Un segundo criterio de clasificación se fija en el número de valores de verdad que se acepten.

Lógica clásica. Las fórmulas lógicas solo pueden ser verdaderas o falsas y no puede ocurrir que lo sean a la vez.

Lógica no clásica. Las fórmulas pueden contemplar más valores de verdad que verdadero o falso u otros mecanismos expresivos. Entre ellas podemos destacar:

- La lógica trivalente. Que contempla [lo verdadero, lo falso y lo que no es ni verdadero ni falso], este último por desconocido o incierto.
<https://es.wikipedia.org/wiki/L%C3%B3gica_trivalente#:~:text=Se%20llama%20l%C3%B3gica%20ternaria%20o,y%20alg%C3%BAAn%20otro%20valor%20indeterminado.er>
- Las lógicas polivalentes. Son lógicas probabilistas en la que los valores de verdad se corresponden con el intervalo [0, 1].
Véase
<https://es.wikipedia.org/wiki/L%C3%B3gica_plurivalente#:~:text=Una%20l%C3%B3gica%20plurivalente%20o%20l%C3%B3gica,los%20tradicionales%20verdadero%20y%20falso>
- La lógica modal. Que incorpora como operadores los modificadores [lo necesario, lo posible].
Véase <https://es.wikipedia.org/wiki/L%C3%B3gica_modal>
- La lógica temporal. Que incorpora parámetros temporales. Para muchas aserciones su verdad depende del momento en que se produce.
Véase <https://es.wikipedia.org/wiki/L%C3%B3gica_temporal>
- La lógica epistémica. Es una lógica intencional que pretende formalizar enunciados de creencia, opinión, etc.
Véase
<https://es.wikipedia.org/wiki/L%C3%B3gica_epist%C3%A9mica#:~:text=La%20l%C3%B3gica%20epist%C3%A9mica%20es%20un,del%20razonamiento%20sobre%20el%20conocimiento>

- La lógica no monotónica. Que pretende formalizar situaciones reales en las que decidimos sin una total información y que posteriormente admite, conforme se prueben o refuten creencias, revisar el sistema total de creencia.

Véase <https://es.wikipedia.org/wiki/L%C3%B3gica_no_monot%C3%B3nica>

- ...

Lógica difusa o Lógica Fuzzi

Véanse:

<<http://www.cs.us.es/~fsancho/?e=97>>

<https://www.esi.uclm.es/www/cglez/downloads/docencia/2011_Softcomputing/LogicaDifusa.pdf>



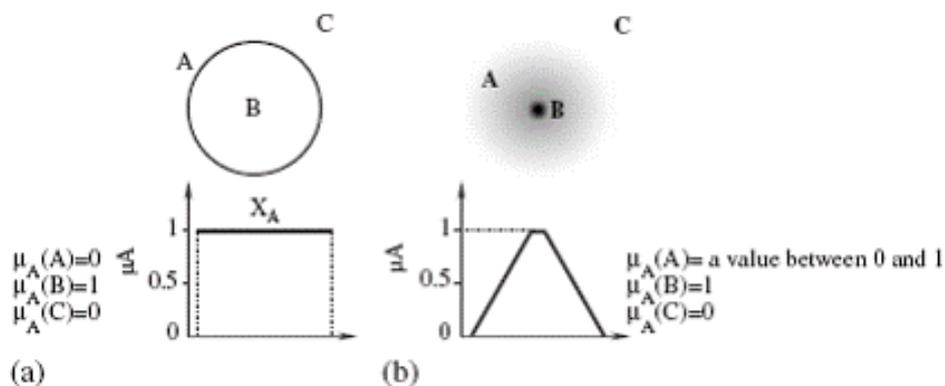
La forma en que la gente piensa es, inherentemente, difusa debido a que cuando expresamos lo que percibimos sobre el mundo este no siempre se puede definir en términos de sentencias verdaderas o falsas, sino más bien mediante probabilidades.

Lo cual va a dejar fuera a lo que se denominan «cisnes negros» que son los sucesos altamente improbables que cuando ocurren impactan.

Consideremos como ejemplo el conjunto de vasos del mundo, que pueden estar vacíos o llenos de agua. Ahora tomemos un vaso vacío y comencemos a echar agua poco a poco, ¿en qué momento decidimos que el vaso pasa de estar vacío a estar lleno? Evidentemente, hay dos situaciones extremas que reconocemos sin ninguna duda, la primera cuando el vaso está completamente vacío, sin una sola gota de agua en su interior, y la segunda cuando está completamente lleno. Las definiciones de **vaso completamente vacío** y **vaso completamente lleno** son demasiado estrictas como para que resulten interesantes en un razonamiento en el que se consideran operaciones de llenado y vaciado de vasos, y entre los términos de *lleno* y *vacío* hay un área que no está claramente definida de pertenencia a ninguno de esos extremos.

En el lenguaje natural que usamos en el mundo real hemos cubierto esta imprecisión por medio de una jerarquía de términos intermedios junto con modificadores que permiten cubrir un espectro más grande de áreas usando un número limitado de ellos, y podemos hablar de lleno, medio lleno, completamente lleno, casi lleno, etc.

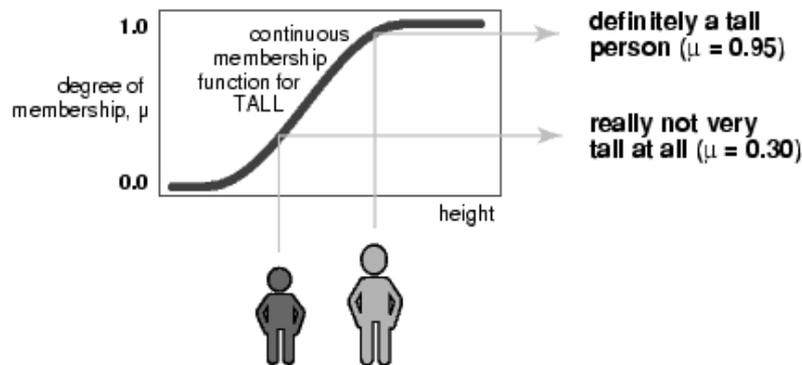
Matemáticamente, los conceptos de *sí / no*, *verdadero / falso* están representados por medio del concepto clásico de *conjunto*, pero necesitamos extenderlo para poder representar este tipo de información más difusa.



Un conjunto difuso permite a sus elementos tener un grado de pertenencia. Si el valor 1 se asigna a los elementos que están completamente en el conjunto, y 0 a los que están completamente fuera, entonces los objetos que están parcialmente en el conjunto tendrán un valor de pertenencia estrictamente entre 0 y 1. Por tanto, si un vaso completamente lleno tiene un grado de pertenencia a **los vasos llenos** de valor 1, y un vaso completamente vacío un grado de pertenencia a **los vasos llenos** de valor 0, entonces al añadir una gota a este último, su grado de pertenencia a **los vasos llenos** sería ligeramente superior a 0.

Definición matemática de conjuntos difusos

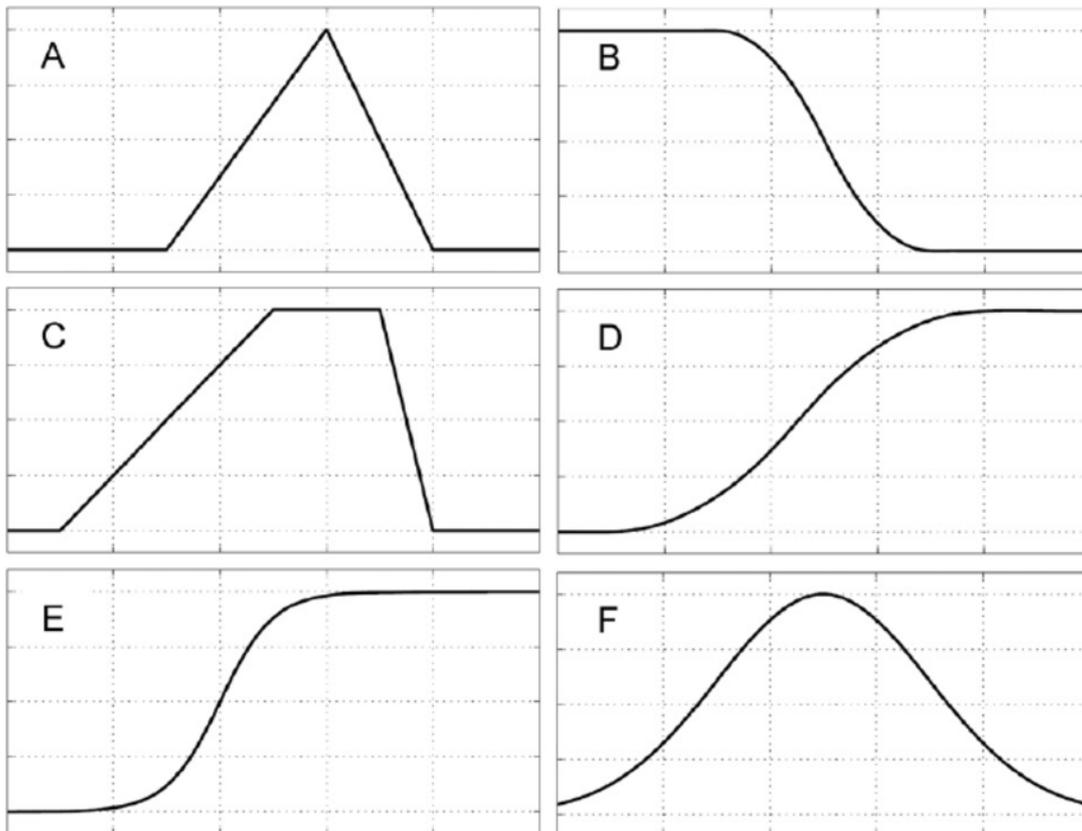
Analicemos el conjunto de los altos. El conjunto se define por medio de una función continua que puede tomar valores intermedios entre los extremos 0 y 1.



La función de pertenencia μ define el conjunto difuso para los posibles valores de altura (mostrados en el eje horizontal) y proporciona el grado de pertenencia de la altura al conjunto difuso (mostrado en el eje vertical con valores entre 0 y 1). De esta forma, el grado de pertenencia de la primera persona es 0,3 y, por tanto, no es muy alto, mientras que el segundo tiene un grado de pertenencia de 0,95 y, definitivamente, es alto.

Los **conjuntos difusos** fueron propuestos inicialmente por Lofti A. Zadeh en su artículo de 1965 titulado «Fuzzy Sets». Este artículo establece los fundamentos de la lógica difusa que se deduce de la definición de conjunto difuso y sus propiedades.

En él no se indicaba cómo han de ser las funciones de pertenencia μ_A asociada a un conjunto difuso A , ya que eso dependerá de las características propias del conjunto real que se quiere representar, pero suelen usarse algunas funciones clásicas comunes como las que se muestran a continuación:



La función $\mu_A(x) = X \rightarrow [0,1]$ entendiéndose que:

- $\mu_A(x) = 1$, si x está totalmente en A .
- $\mu_A(x) = 0$, si x no está en A .
- $0 < \mu_A(x) < 1$, si x está parcialmente en A .

El valor asociado representa el grado de pertenencia de un elemento « x » a un conjunto « A ».

Operaciones lógicas difusas

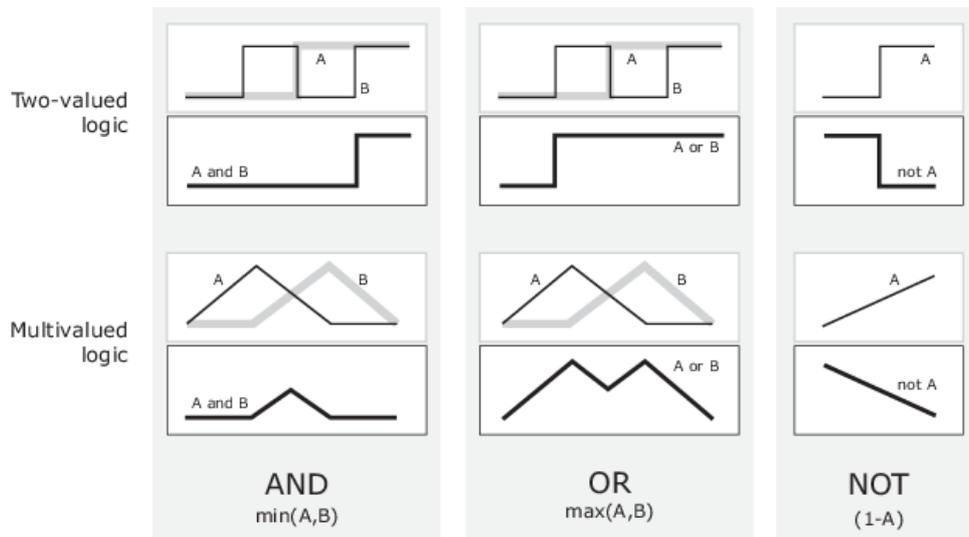
La lógica tradicional bivaluada usa los operadores booleanos [\wedge (AND), \vee (OR), y \neg (NOT)] para llevar a cabo las operaciones de conjuntos de intersección, unión y complementario. Estos operadores funcionan bien con conjuntos nítidos, clásicos, y se pueden definir sencillamente a partir de las tablas de verdad asociadas a cada operador.

Estas tablas de verdad funcionan bien para la lógica bivaluada, pero debido a que los conjuntos difusos no tienen por qué tomar una cantidad finita de valores, no es fácil extender las tablas para su uso en este caso. Estos operadores necesitan ser redefinidos como funciones para todos los posibles valores difusos de los grados de pertenencia, es decir, para todo el intervalo $[0,1]$, y no solo para los valores extremos.

Una posible definición (hay otras) vendría dada por:

- $x \wedge y = \text{mín.}(x,y)$.
- $x \vee y = \text{máx.}(x,y)$.
- $\neg x = 1-x$.

Estas definiciones se pueden usar tanto para aplicar los operadores clásicos como para obtener combinaciones difusas. Las siguientes gráficas representan el resultado obtenido con las definiciones anteriores sobre todos los posibles valores de entrada:

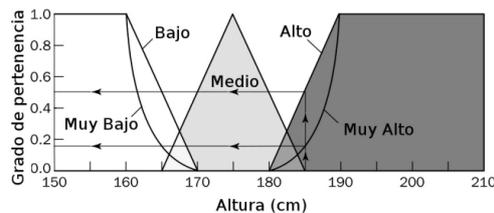


Para representar el conocimiento en razonamiento borroso, se utilizan variables lingüísticas extraídas del lenguaje natural:

- Términos primarios: «bajo», «alto»...
- Modificadores: «Muy», «más», «menos», «cerca de»...
- Conectores lógicos: Normalmente NOT, AND y OR.

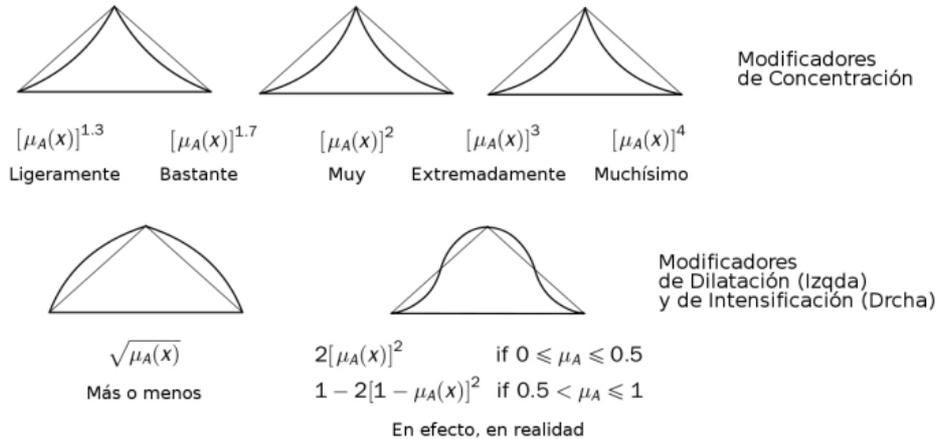
Modificadores lingüísticos

La figura muestra un ejemplo de modificadores (en este caso «muy»)



En la práctica existen tres tipos de modificadores:

- De concentración.
- De dilatación.
- De intensificación.

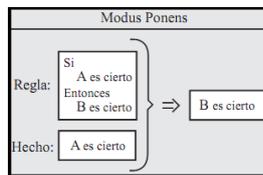


Ejemplo:

Si Pedro tiene un valor de pertenencia $\mu_A(\text{Pedro}) = 0,86$ al conjunto difuso de los altos (A), entonces tendría un valor de $\sqrt{\mu_A(\text{Pedro})} = \sqrt{0,86} = 0,92$ al conjunto difuso de los más o menos altos.

Reglas difusas de inferencia

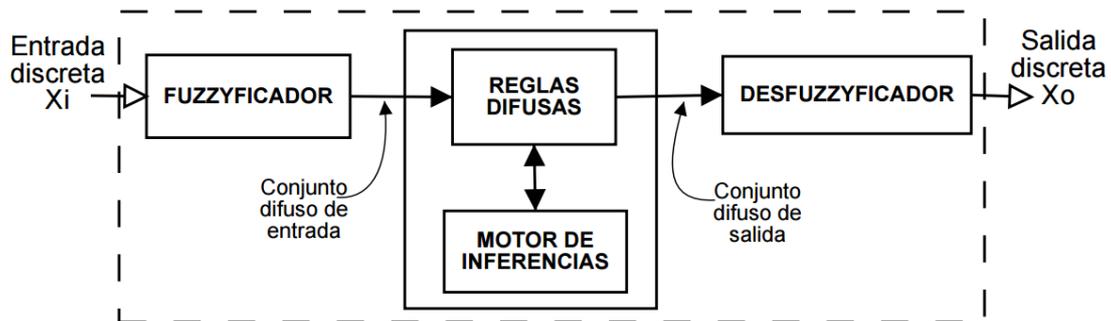
La mayoría de las decisiones que la gente toma son decisiones lógicas, miran la situación, la valoran, y toman una decisión basándose en ella. La forma generalizada de una decisión de este tipo se llama **modus ponens generalizado**, que tiene la forma:



Esta forma de razonamiento lógico es bastante estricta. La lógica difusa pierde esta forma estricta diciendo que B se dará con más opciones si la veracidad de A es más alta. Donde ahora A y B toman valores difusos. El razonamiento anterior requiere que se defina un conjunto de reglas que lo lleven a cabo. Estas reglas son reglas lingüísticas que relacionan diferentes conjuntos y valores difusos. La forma general de estas reglas es: «si x está en A, entonces y está en B», donde x e y son valores difusos en los conjuntos difusos A y B, respectivamente (y que vendrán definidos por medio de sus funciones de pertenencia).

Las reglas lingüísticas se usan para relacionar las entradas con las salidas, y vamos a ver a continuación cómo se puede hacer una valoración correcta de las mismas. Ha habido varias propuestas para determinar la ejecución de estas reglas, las más importantes han sido los métodos de inferencia y agregación de Mamdani, Larsen, Takagi-Sugeno-Kang y Tsukamoto.

El método de inferencia de Mamdani puede que sea el método más utilizado. Este proceso se realiza en cuatro pasos, y puede resumirse de la siguiente forma:

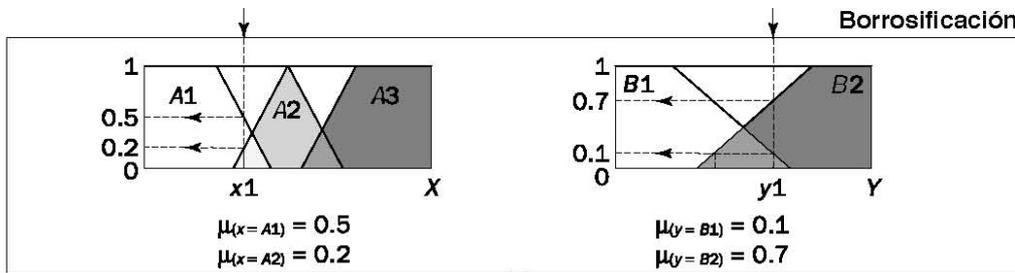


- Se parte de una entrada discreta X_{input} .
- Esta pasa al fuzzyficador que convierte las entradas del sistema, que son valores numéricos nítidos en conjuntos borrosos.
- Ese conjunto difuso de entrada pasa a la base de conocimiento que está formado por las reglas difusas: Ese almacén de reglas SI-ENTONCES se ha obtenido a partir de los expertos.
- Entonces actúa el motor de inferencia que simula el razonamiento humano haciendo inferencia sobre las entradas recibidas y las reglas SI-ENTONCES almacenadas, generando el conjunto difuso de salida.
- Dicha salida pasa al **desfuzzyficador**, convirtiendo el conjunto borroso obtenido en un valor numérico nítido (X_{output}) que puede ser reutilizado.

Ejemplo:

- Términos primarios:
 - $x \rightarrow$ (financiación-del-proyecto).
 - $y \rightarrow$ (plantilla-del-proyecto).
 - $z \rightarrow$ (riesgo).

- Modificadores:
 - Dominio X → A1 (inadecuado), A2 (marginal), A3 (adecuado).
 - Dominio Y → B1 (pequeña), B2 (grande).



- Dominio Z → C1 (bajo), C2 (normal), C3 (alto).
- Reglas:
 - R1: IF x es A3 OR y es B1 THEN z es C1.
 - R2: IF x es A2 AND y es B2 THEN z es C2.
 - R3: IF x es A1 THEN z es C3.

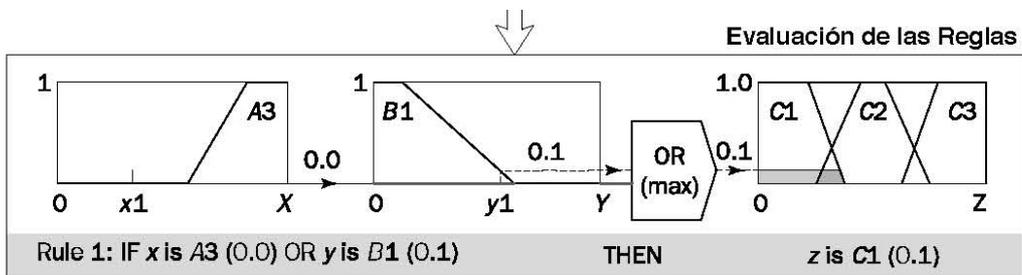
Veamos el proceso completo

- Fuzzificación, valores dados por un experto:
 - A «x1» se le asigna el valor 35 %, por lo tanto, $\mu_{A1} = 0,5$ y $\mu_{A2} = 0,2$.
 - A «y1» se le asigna el valor 60 %, por lo tanto, $\mu_{B1} = 0,1$ y $\mu_{B2} = 0,7$.
- Evaluación de reglas.

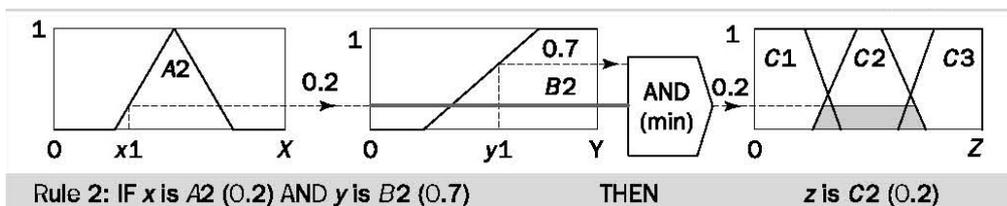
Recordemos que:

- $x \wedge y = \text{mín.}(x,y)$.
- $x \vee y = \text{máx.}(x,y)$.
- $\neg x = 1-x$.

R1: IF x es A3 OR y es B1 THEN z es C1



R2: IF x es A2 AND y es B2 THEN z es C2



R3: IF x es A1 THEN z es C3



- Se agregan las salidas, unificando todas las reglas, es decir, combinando las funciones de pertenencia de todos los consecuentes, previamente recortados (en este caso), también se habrían podido escalar.

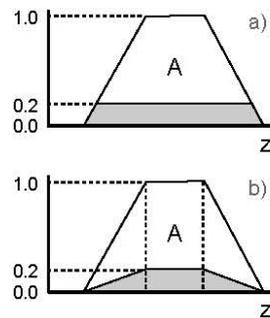
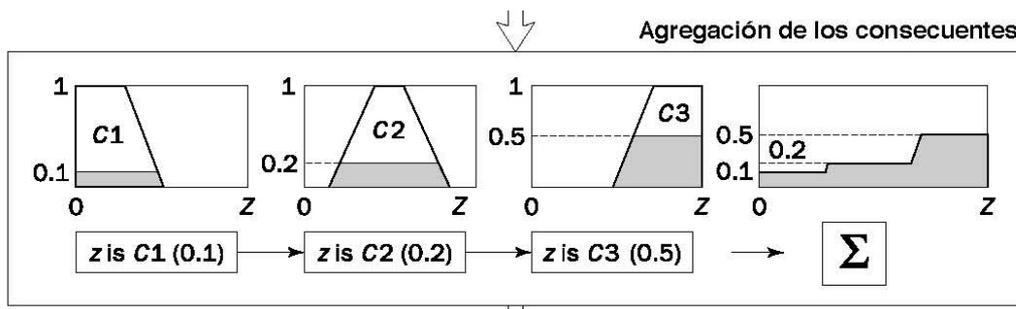


Figura 3.2: Conjunto recortado (a) y escalado (b).



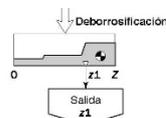
- Se defuzzifica. Existen varios métodos para realizar esa tarea, pero probablemente el más ampliamente usado es el centroide, que calcula el punto donde una línea vertical divide el conjunto en dos áreas iguales.

$$\text{Centroide} = \frac{\sum_{z=a}^b (\mu_A(z) z)}{\sum_{z=a}^b (\mu_A(z))}$$

Suponiendo que el eje de las z recorre el intervalo [0-100]:

- Que C1 cubre los valores del eje z en el intervalo [0-40).
- Que C2 cubre los valores del eje z en el intervalo [40-80).
- Que C3 cubre los valores del eje z en el intervalo [80-100].

$$z1 = \frac{(0 + 10 + 20 + 30) \times 0.1 + (40 + 50 + 60 + 70) \times 0.2 + (80 + 90 + 100) \times 0.5}{0.1 + 0.1 + 0.1 + 0.1 + 0.2 + 0.2 + 0.2 + 0.2 + 0.5 + 0.5 + 0.5} = \frac{6 + 44 + 135}{2.7} = \frac{185}{2.7} = 68,52$$



El proceso completo se puede ver en el gráfico siguiente:

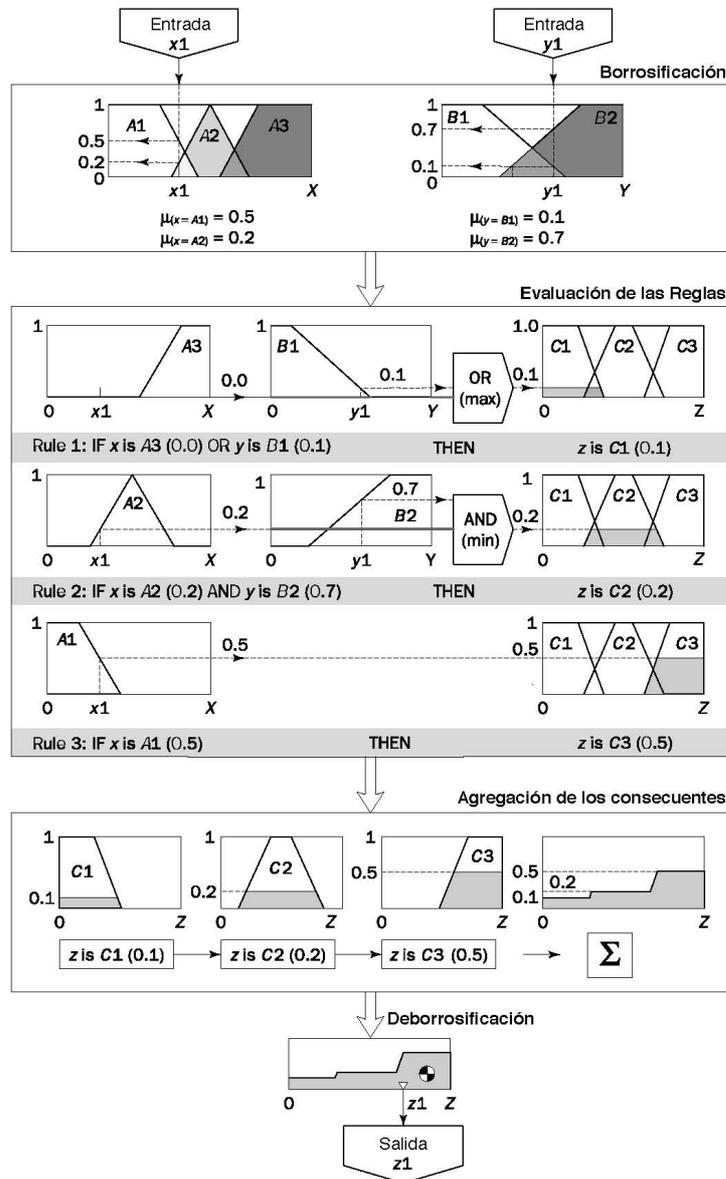


Figura 3.3: Estructura básica de inferencia de Mamdani.

Ventajas

Los conceptos matemáticos dentro del razonamiento borroso son muy simples y se pueden implementar con relativa facilidad.

Da buenos resultados en procesos no lineales y de difícil modelado.

La lógica borrosa es flexible: es fácil transformar un sistema borroso añadiendo o eliminando reglas sin tener que empezar desde cero.

La lógica borrosa admite datos imprecisos (pero cuidado, no estudia la incertidumbre): maneja elementos de un conjunto borroso, es decir, valores de una función de pertenencia. Por ejemplo, en lugar de manejar el dato «Mide 180 cm», maneja «Es alto con una precisión de 0,8».

Desventajas

La ley del tercio exclusivo ($p \vee \neg p$) deja de ser una tautología, por lo que no se pueden realizar inferencias por contradicción.

La lógica borrosa no resuelve problemas nuevos, sino que utiliza nuevos métodos para resolver los problemas de siempre.

La lógica borrosa admite datos imprecisos (pero cuidado, no estudia la incertidumbre): maneja elementos de un conjunto borroso, es decir, valores de una función de pertenencia. Por ejemplo, en lugar de manejar el dato «Mide 180 cm», maneja «Es alto con una precisión de 0,8».

La lógica borrosa se construye sobre la experiencia de los expertos: confía en la experiencia de quien ya conoce el sistema.

Ante un problema que es resoluble mediante un modelo matemático, la lógica difusa no tiene por qué generar mejores resultados.

No hay actualmente un análisis matemático riguroso que garantice que el uso de un sistema experto difuso, para controlar un sistema, dé cómo resultado un sistema estable.

A veces no es fácil interpretar los resultados difusos.

Existen varias definiciones de operadores y reglas de inferencia difusa.

Para saber más: Véase

https://es.wikipedia.org/wiki/L%C3%B3gica_difusa

Planificación

La capacidad de planificación (capacidad para anticipar el resultado de las acciones y encadenarlas) es claramente necesaria para que un agente alcance una inteligencia de propósito general.

Actuamos sin planificar:

- Cuando el propósito es inmediato.
- Cuando realizamos tareas bien aprendidas.
- Cuando nuestro curso de acción puede adaptarse libremente sin demasiadas consecuencias.

Planificamos:

- Cuando nos encontramos ante situaciones nuevas.
- Cuando las tareas son complejas.
- Cuando existen riesgos (costes) que asumir.
- Cuando colaboramos con otros.

Llamamos planificación al proceso automático de búsqueda y articulación de una secuencia de acciones que permitan alcanzar un objetivo. A eso se le denomina un *plan*.

La planificación es un área de gran interés de la IA. Una razón de esto es que combina dos áreas importantes, la de búsquedas y la de lógica. La planificación se puede ver inicialmente como un problema de búsqueda en un espacio de estados, o como uno que proporcione la existencia de una solución. Pero como ya sabemos, los algoritmos de búsqueda en ocasiones presentan problemas de eficiencia cuando hay un alto factor de ramificación. Muchos algoritmos de planificación suelen reducir la complejidad del problema descomponiéndolo en subobjetivos y normalmente operan con representaciones proposicionales explícitas de primer orden de estado y de acciones.

Uno de los principales problemas con los que se enfrenta la planificación es el de la explosión combinatoria del espacio de estados. Si tenemos un número p de primitivas en un dominio, tendremos 2^p posibles estados. Para dominios complejos p puede ser muy alto. Además, si el dominio tiene propiedades (localización, color) y relaciones (en, sobre, entre), con d objetos en un dominio que mantenga relaciones cada tres, tendremos 2^{d^3} estados. Es evidente entonces que el problema de planificación es una tarea sin solución. Lo que no quiere decir que el problema no sea afrontable con la estrategia divide y vencerás, es decir, podemos ir construyendo un plan para alcanzar (un objetivo final, unos subobjetivos o unos sub-subobjetivos)... para que no se tengan en cuenta todos los detalles a la vez. La planificación a un nivel

adecuado de abstracción puede llevar a «podar» el espacio de búsqueda, para que algunos detalles no tengan que examinarse para nada. De cualquier modo vamos a hacer una primera aproximación al problema.

Ejemplos de problemas de planificación: planificación de movimientos de robots móviles, diseño de agentes para juegos o programas de entrenamiento, consultar servicios web, manejo de crisis, campañas militares, decisiones logísticas y estratégicas, misiones aeroespaciales, control de aeronaves y vehículos autónomos, gestión de flujos de trabajo...

La planificación clásica considera entornos que son:

- Completamente observables. Se conoce perfectamente el estado del entorno y también el efecto de sus acciones en el entorno.
- Deterministas. Se pueden predecir y predefinir los efectos de todas las acciones.
- Finitos. Existe un conjunto finito de acciones y de estados.
- Estáticos. El entorno solo cambia cuando el agente planificador actúa sobre él.
- Discretos. El entorno se puede describir de forma discreta:
 - Tiempo (medido en ciclos de ejecución).
 - Acciones (vistas como unidades que requieren un ciclo de ejecución).
 - Objetos.
 - Efectos (observables al final de cada ciclo de ejecución).

En la actualidad, existen planificadores capaces de adaptarse a entornos que sean observables en parte (modelo del mundo incompleto y/o incorrecto) y probabilístico. Y son capaces de entablar continuamente la formulación, la ejecución, la adaptación y el abandono de objetivos, según cambie el medio.

Un planificador consiste normalmente en:

- Una descripción del mundo.
- Una descripción del estado inicial del mundo.
- Una descripción del objetivo que queremos alcanzar.
- Un conjunto de posibles acciones. Generalmente, cada acción especifica precondiciones que se deben cumplir como requisito para tal acción, así como poscondiciones, que constituyen el efecto sobre el estado actual del mundo.

Los sistemas de IA, por lo general, usan técnicas de «encadenamiento hacia adelante» y «encadenamiento hacia atrás» que explican cómo encontró la solución el programa, lo que ayuda al usuario a juzgar si la acción / dictamen del programa es la apropiada.

Si se plantea el planificador como un algoritmo de búsqueda hacia adelante en un grafo (Depth First Search, DFS), entonces el proceso de búsqueda sería:

- Estado \leftarrow estado inicial inicial.
- Plan \leftarrow $\langle \rangle$.
- Mientras el estado no satisface el objetivo:
 - Aplicables \leftarrow acciones válidas para estado.
 - Si aplicables está vacío, devolver fallo.
 - Acción \leftarrow seleccionar una acción de aplicables.
 - Estado \leftarrow resultado (estado, acción).
 - Plan \leftarrow plan \cdot \langle acción \rangle .
- Devolver plan.

La búsqueda hacia adelante es correcta, si devuelve un plan, este es válido. Además es completa, ya que si existe un plan, la búsqueda lo termina encontrando.

La búsqueda hacia adelante puede no ser la mejor opción si existe un número elevado de acciones aplicables en cada estado. O el factor de ramificación es demasiado grande. O el número de pasos lo hace inviable. Una alternativa es plantear la búsqueda hacia atrás en el grafo (Depth First Search, DFS):

- Subobjetivo \leftarrow objetivo final.
- Plan \leftarrow $\langle \rangle$.
- Mientras el estado inicial no satisface el subobjetivo:
 - Aplicables \leftarrow acciones que nos llevan a subobjetivo.
 - Si aplicables está vacío, devolver fallo.
 - Acción \leftarrow seleccionar una acción de aplicables.
 - Subobjetivo \leftarrow resultado-1 (subobjetivo, acción).
 - Plan \leftarrow \langle acción $\rangle \cdot$ plan.
- Devolver plan.

La búsqueda hacia atrás es correcta y completa. Su problema vuelve a ser que los espacios de búsqueda suelen ser demasiado grandes.

Una búsqueda genérica no parece la estrategia más adecuada para resolver un problema de planificación, ya que:

- No se tienen en cuenta las diferencias entre el estado inicial y el objetivo para elegir las acciones adecuadas.
- Los planes podrían empezar con acciones obvias y después refinarse (con acciones que quizá haya que realizar antes de las obvias).
- Muchos objetivos son compuestos y se podrían obtener de forma independiente (o, al menos, de forma relativamente independiente).

Para seguir avanzando en el problema de la planificación, surgieron los lenguajes de especificación. La adopción de un formalismo común para la descripción de los dominios de planificación, permite describir los dominios de planificación a la vez que facilita la investigación y la comparación directa de los sistemas y los enfoques de resolución. Para lograr una planificación eficiente es importante contar con buenos lenguajes de modelización.

El primer avance se produjo a principios de los setenta, con la aparición de un planificador lineal, denominado STRIPS (Stanford Research Institute Problem Solver) que se caracterizó por:

- Uso de predicados (lógica de primer orden).
- Solo literales positivos.
- Solo conjunciones de literales simples en el objetivo.
- Solo se especifica aquello que cambia.
- Hipótesis de mundo cerrado.
- Una acción tiene tres elementos clave:
 - Las precondiciones son proposiciones que deben cumplirse para poder aplicar la acción.
 - Consecuencias de la aplicación de la acción:
 - Los predicados que añadir, proposiciones que pasan a ser ciertas.
 - Los predicados que borrar, proposiciones que pasan a ser falsas.
- Para que una acción pueda aplicarse en un estado, este ha de satisfacer sus precondiciones.
- Cuando se realiza la acción, este estado se transforma en un nuevo estado añadiendo los predicados que están en la lista «añadir» y borrando los que aparecen en la lista «borrar».
- Una acción se incluye en el plan cuando contiene en la lista «añadir» un objetivo del problema, y sus precondiciones se vuelven subobjetivos a cumplir.
- Este esquema de sustitución de unos objetivos por otros, idealmente más sencillos, es lo que se conoce con el nombre de *subloading*.
- Todas estas nociones se formalizan lógicamente mediante el denominado *situation calculus*.
- Si tenemos «n» subobjetivos que conseguir, puede que tengamos que probar n! formas distintas de ordenarlos hasta obtener una solución.
- Para reducir el espacio de búsqueda, se tratará de ir logrando los objetivos en cierto orden.
- La anomalía de Sussman mostró que los planificadores lineales pueden no alcanzar nunca los objetivos. Debido a que puede que deshagamos un objetivo ya conseguido.

Algoritmo básico de STRIPS:

- STRIPS (estado, objetivo):
 - Plan $\leftarrow \langle \rangle$.
 - Mientras el estado no satisface el objetivo.
 - Aplicables \leftarrow acciones relevantes (estado, objetivo).
 - Si aplicables está vacío, devolver fallo.
 - Acción \leftarrow seleccionar una acción de aplicables.
 - Subplan \leftarrow STRIPS (estado, precondiciones de acción).
 - Si subplan falla, devolver fallo.
 - Estado \leftarrow resultado (estado, subplan \cdot (acción)).
 - Plan \leftarrow plan \cdot subplan \cdot (acción).
- Devolver plan.

El lenguaje STRIPS surgió a principios de 1970 como un sistema de planificación para el proyecto del robot Shakey. Véase

<https://es.wikipedia.org/wiki/STRIPS>

Una de las principales ampliaciones que se ha llevado a cabo es el lenguaje ADL (Action Description Language) de 1989. Véase

https://en.wikipedia.org/wiki/Architecture_description_language

Otra ampliación famosa es PDDL (Planning Domain Definition Language) de 1998. Véase https://en.wikipedia.org/wiki/Planning_Domain_Definition_Language

Una de las tendencias que marcó la evolución fue la utilización de algoritmos para grafos. Graphplan (1997) fue el principal exponente y superó el rendimiento de los planificadores existentes. El algoritmo toma como entrada un problema de planificación expresado en STRIPS y produce, si es posible, un grafo de planificación en el que se realiza la búsqueda para encontrar una secuencia de operaciones que llegue a la solución en el espacio de estados.

Durante los años setenta y ochenta, el esfuerzo se centró en problemas tipo *puzzle*, como el mundo de bloques o las torres de Hanói, que presentaban fuertes interacciones entre objetivos y acciones. El objetivo era encontrar el plan óptimo y, aunque se realizó un progreso considerable en las técnicas de búsqueda, ningún planificador fue capaz de resolver problemas con una mínima aplicación práctica.

La aparición de Graphplan a principios de los noventa generó un gran interés pues constituía una aproximación totalmente nueva. Graphplan realiza un análisis de alcanzabilidad del espacio de estados del problema. La representación compacta de la información obtenida permitió superar ampliamente el rendimiento del resto de aproximaciones de la época.

Tras el éxito de Graphplan, el interés en la resolución de problemas de planificación resurgió. Aparecieron entonces nuevas aproximaciones, como la planificación heurística o las técnicas de transformación del problema de planificación, que pronto superaron el rendimiento de Graphplan. Estos nuevos avances también permitieron la reaparición de antiguas técnicas, como la planificación de orden parcial, que prácticamente se habían olvidado por cuestiones de eficiencia. Actualmente, una de las últimas tendencias es la combinación de distintas técnicas para sacar partido de las ventajas que ofrece cada una de ellas.

Cuando se considera la aplicación de técnicas de planificación en entornos reales, es imprescindible tener en cuenta los aspectos relacionados con la ejecución de los planes. La planificación clásica ignora estos aspectos, realizando diversas simplificaciones poco realistas, como que el entorno es estático, determinista y completamente observable. Para intentar superar estas simplificaciones, se han desarrollado numerosos trabajos de investigación que se engloban generalmente bajo el nombre de *planificación práctica*.

Las investigaciones sobre los aspectos prácticos de la planificación se han llevado a cabo en dos vertientes muy relacionadas: la mejora de la expresividad de los lenguajes de especificación y el desarrollo de nuevas técnicas de planificación.

Respecto a las mejoras en la expresividad, los nuevos lenguajes propuestos, como SADL o PPDDL permiten modelar la incertidumbre en los dominios, acciones de sensorización, y otras características de gran utilidad en entornos reales.

El lenguaje de representación básico del planificador clásico STRIPS es el siguiente:

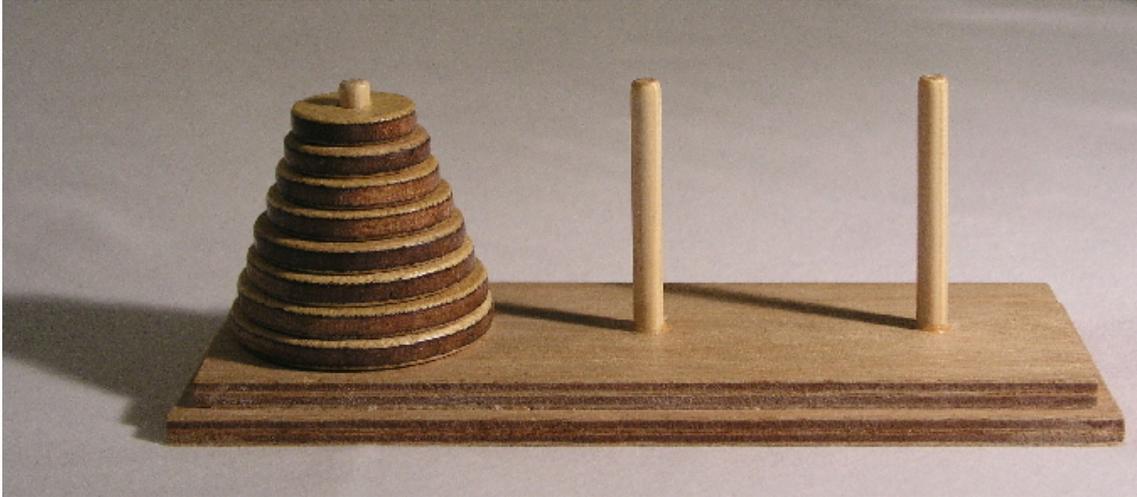
- Representación de estados (descripción de la situación del mundo):
 - Un estado se representa como una secuencia consistente en una conjunción de literales positivos conectados. Es un predicado lógico.
- Representación de un objetivo como un estado parcialmente especificado representado como una secuencia de literales positivos y simples.
- Representación de acciones. Cada acción se especifica en términos de precondiciones que se han de cumplir antes de que se pueda ejecutar y de poscondiciones que indican los efectos que se van a producir después de su actuación. La acción es aplicable en cualquier estado que satisfaga la precondición. El resultado de ejecutar la acción en un estado s es otro estado s' . El operador asociado a la acción añade y/o elimina predicados al estado actual.
 - Cada acción puede depender de una lista de parámetros de los que dependa.

- Cada precondition será la unión de literales positivos sin dependencia funcional que especifica lo que debe ser verdadero antes de la ejecución de la acción. Todas las variables en la precondition serán parámetros de la acción.
- El efecto será la unión de literales sin dependencia funcional describiendo cómo cambia el estado del mundo cuando la acción se ejecuta. Todas las variables han de aparecer también en la lista de parámetros de la acción.
- Búsqueda en el espacio de estados:
 - Idea. Aplicar los métodos de búsqueda en el espacio de estados. Los operadores son las acciones posibles. La solución, la secuencia de acciones desde el estado inicial.
 - Búsqueda progresiva (hacia adelante) no suele ser adecuada debido normalmente al alto factor de ramificación. Existe dificultad comprobada para definir heurísticas útiles.
 - Búsqueda regresiva (hacia atrás). Se comienza en un objetivo y para cada uno de los estados se generan los posibles predecesores. Finaliza cuando se alcanza un objetivo que sea cierto en el estado inicial. Este procedimiento suele mejorar la eficiencia.

- Ejemplo aplicado al problema de las torres de Hanói.

<https://es.wikipedia.org/wiki/Torres_de_Hanói>

<<http://estadisticando.blogspot.com/2015/06/las-torres-de-hanoi.html>>



Édouard Lucas (Amiens, 4 de abril de 1842 - París, 3 de octubre de 1891), fue un matemático francés reconocido por sus trabajos sobre la serie de Fibonacci, la prueba de Lucas o el test de primalidad, en 1883 publicó el juego de «La Torre de Hanói» («La Tour de Hanoi») bajo el pseudónimo de Profesor N. Claus de Siam.

El juego de las torres de Hanói es un dispositivo que consta de tres varillas verticales A, B y C, y un número variable de discos. Los n discos son todos de diferente tamaño y, en la posición de partida del juego, todos los discos están colocados en la varilla A ordenados de mayor a menor tamaño, esto es, el mayor en el lugar más bajo y el menor arriba. El juego consiste en pasar todos los discos a la tercera varilla C colocándolos de mayor a menor. Conforme aumenta el número de discos la dificultad del juego también, así como el tiempo de resolución sin haber cometido ningún error.

En las instrucciones que acompañaban al juego original, Édouard Lucas incluía una breve referencia a una leyenda relacionada con los brahmanes de Benarés (India) y sus templos: «En el gran templo de Benarés, debajo de la cúpula que marca el centro del mundo, yace una base de bronce, en donde se encuentran acomodadas tres agujas de diamante, cada una del grueso del cuerpo de una abeja y de una altura de 50 cm aproximadamente. En una de estas agujas, Dios, en el momento de la Creación, colocó sesenta y cuatro discos de oro, el mayor sobre la base de bronce y el resto de menor tamaño conforme se va ascendiendo. Día y noche, incesantemente, los sacerdotes del templo se turnan en el trabajo de mover los discos de una aguja a otra de acuerdo con las leyes impuestas e inmutables de Brahma, que requieren que siempre haya algún sacerdote trabajando, que no muevan más de un disco a la vez y que deben colocar cada disco en alguna de las agujas de modo que no cubra a un disco de radio

menor. Cuando los sesenta y cuatro discos hayan sido transferidos de la aguja en la que Dios los colocó en el momento de la Creación a otra aguja, el templo y los brahmanes se convertirán en polvo y, junto con ellos, el mundo desaparecerá». De ahí que a las torres de Hanói también se le conoce como «Las torres de Brahma» o «El problema del fin del mundo». La razón es que el número de movimientos necesarios para mover correctamente una torre de 64 discos es $2^{64} - 1$ que es igual a $18_3446\ 744_2073\ 709_1551\dots615$. A una velocidad de un movimiento por segundo, iese serían $584\ 942_1417\ 355$ años!

Solución:

Como ya hemos dicho este juego consiste en un número de discos perforados de radio creciente que se apilan insertándose en uno de los tres postes fijados a un tablero. El objetivo del juego es, si se parte de la configuración que se ve en la imagen, trasladar la pila al otro poste de más a la izquierda siguiendo ciertas reglas:

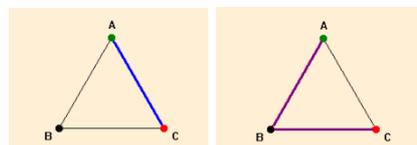
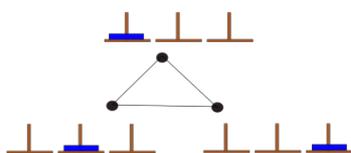
- Se desplaza un disco cada vez.
- Solo se pueden desplazar los discos de arriba.
- No se puede colocar un disco más grande encima de un disco más pequeño. La fórmula para transferir «n» discos desde un puesto a otro es $(2^n - 1)$ por el camino más corto.

La construcción del grafo asociado al espacio de estados sería en el caso de $n = 3$ piezas cada una de un tamaño diferente y $m = 3$ postes de tamaño n^m es decir 3^3 . Si denominamos a cada poste «A» el de partida, «B» el del medio y «C» el de llegada, el espacio de estados sería:

Estado
AAA
AAB
AAC
ABA
ABB
ABC
ACA
ACB
ACC
BAA
BAB
BAC
BBA
BBB
BBC
BCA
BCB
BCC

CAA
CAB
CAC
CBA
CBB
CBC
CCA
CCB
CCC

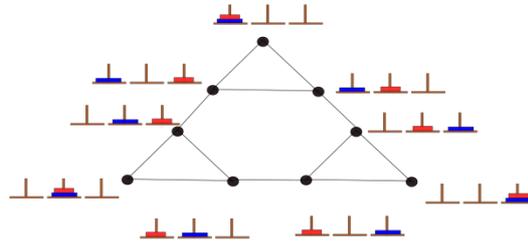
Veamos el caso de 1 disco: Se requiere un movimiento 2^1-1 . Denominemos a los postes «A» el de partida, «B» el central y «C» el de la izquierda.



En la figura se puede ver la solución.

- Camino corto:
 - Se inserta el disco en el poste «C».
- Camino largo:
 - Se inserta el disco en el poste «B» y posteriormente en el «C».

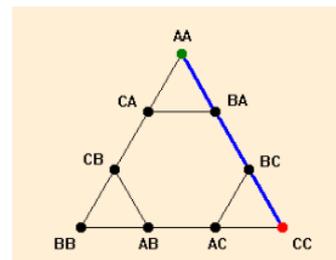
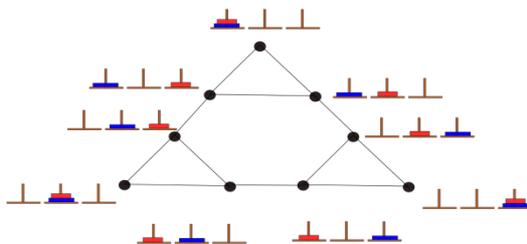
Veamos el caso de 2 discos: Se requieren tres movimientos ($2^2 - 1$) por el camino más corto. Denominemos a los postes «A» el de partida, «B» el central y «C» el de la izquierda. Y a los discos (al azul lo denominaremos 2, y al rojo 1).



En la figura se puede ver la solución. Recorramos el grafo.

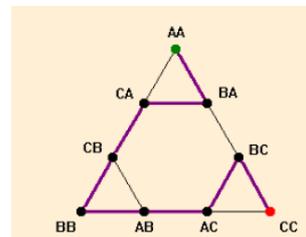
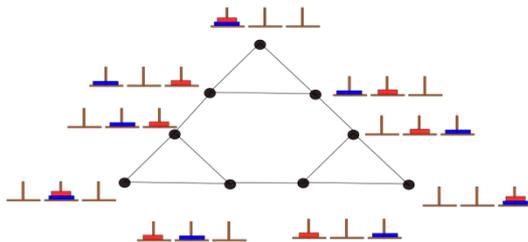
Camino corto:

- Estado de partida:
 - Los dos discos en el poste «A» por lo tanto (AA).
 - El orden de arriba hacia abajo [1, 2].
- 1.^{er} movimiento:
 - El disco 1 en «B» y el disco 2 en «A», por lo tanto (BA).
- 2.^o movimiento:
 - El disco 1 en «B» y el disco 2 en «C», por lo tanto (BC).
- 3.^{er} movimiento
 - El disco 1 en «C» y el disco 2 en «C», por lo tanto (CC).
 - El orden de arriba a bajo [1, 2] **SOLUCIÓN en tres pasos.**

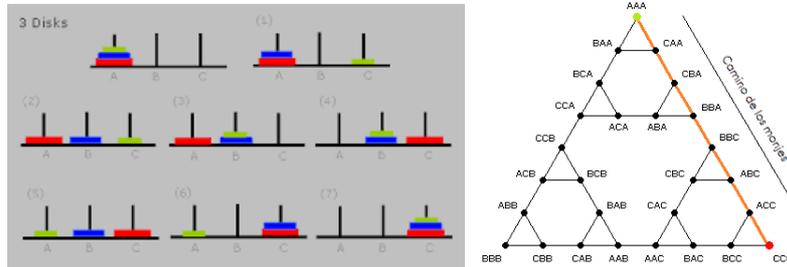


Camino largo:

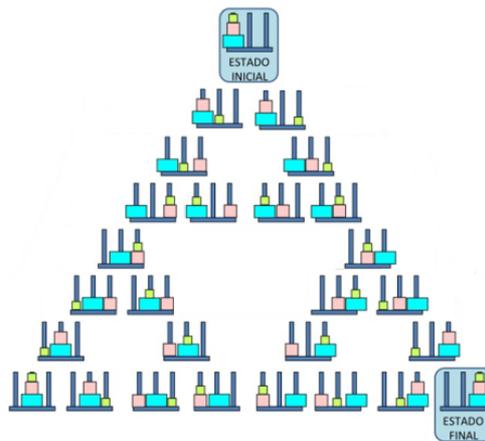
- Estado de partida:
 - Los dos discos en el poste «A» por lo tanto (AA).
 - El orden de arriba hacia abajo [1, 2].
- 1.^{er} movimiento:
 - El disco 1 en «B» y el disco 2 en «A», por lo tanto (BA).
- 2.^o movimiento:
 - El disco 1 en «C» y el disco 2 en «A», por lo tanto (CA).
- 3.^{er} movimiento:
 - El disco 1 en «C» y el disco 2 en «B», por lo tanto (CB).
- 4.^o movimiento:
 - El disco 1 en «B» y el disco 2 en «B», por lo tanto (BB).
 - El orden de arriba hacia abajo [1, 2].
- 5.^o movimiento:
 - El disco 1 en «A» y el disco 2 en «B», por lo tanto (AB).
- 6.^o movimiento:
 - El disco 1 en «A» y el disco 2 en «C», por lo tanto (AC).
- 7.^o movimiento:
 - El disco 1 en «C» y el disco 2 en «C», por lo tanto (CC).
 - El orden de arriba a bajo [1, 2] **SOLUCIÓN en siete pasos.**



Veamos el caso de 3 discos: Se requieren tres movimientos ($2^3 - 1$) por el camino más corto. Denominemos a los postes «A» el de partida, «B» el central y «C» el de la izquierda. Y a los discos (al azul lo denominaremos 3, al amarillo 2 y al rojo 1)



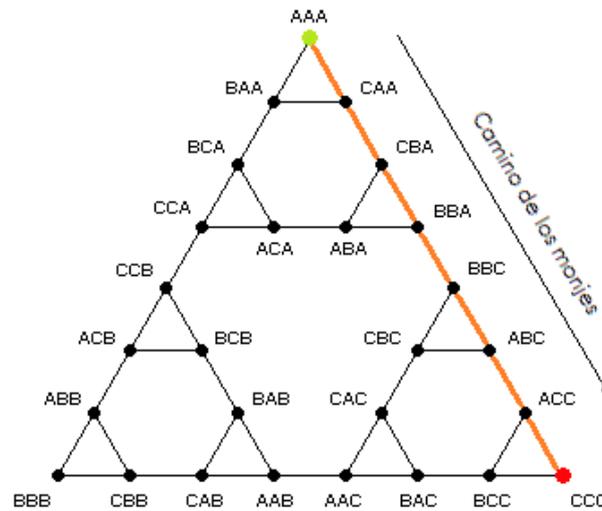
Construcción del grafo



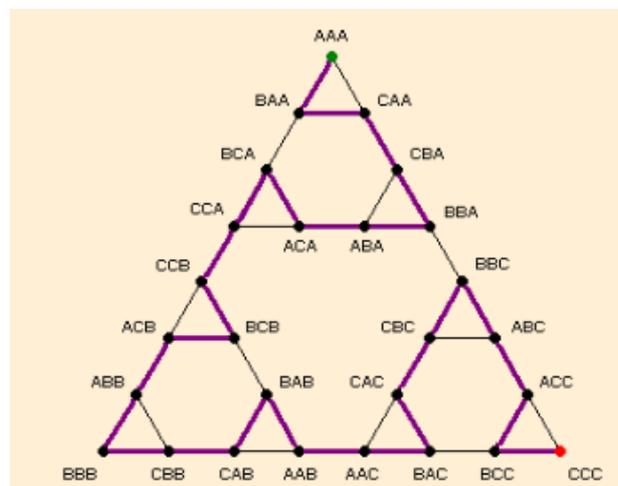
Caso corto:

- Estado de partida:
 - Los dos discos en el poste «A», por lo tanto (aaa).
 - El orden de arriba hacia abajo [1, 2, 3].
- 1.^{er} movimiento:
 - El disco 1 en «C», el disco 2 en «A», el disco 3 en «A», por lo tanto (caa).
 - El orden de arriba hacia abajo en «A» [2, 3].
- 2.^o movimiento:
 - El disco 1 en «C», el disco 2 en «B», el disco 3 en «A», por lo tanto (cba).
- 3.^{er} movimiento:
 - El disco 1 en «B», el disco 2 en «B», el disco 3 en «A», por lo tanto (bba).
 - El orden de arriba hacia abajo en «B» [1, 2].
- 4.^o movimiento:
 - El disco 1 en «B», el disco 2 en «B», el disco 3 en «C», por lo tanto (bbc).E
 - El orden de arriba hacia abajo en «B» [1, 2].

- 5.º movimiento:
 - El disco 1 en «A», el disco 2 en «B», el disco 3 en «C», por lo tanto (abc).
- 6.º movimiento:
 - El disco 1 en «A», el disco 2 en «C», el disco 3 en «C», por lo tanto (acc).
 - El orden de arriba hacia abajo en «C» [1, 2].
- 7.º movimiento:
 - El disco 1 en «C», el disco 2 en «C», el disco 3 en «C», por lo tanto (ccc).
 - El orden de arriba a bajo [1, 2, 3]. **SOLUCIÓN en siete pasos.**

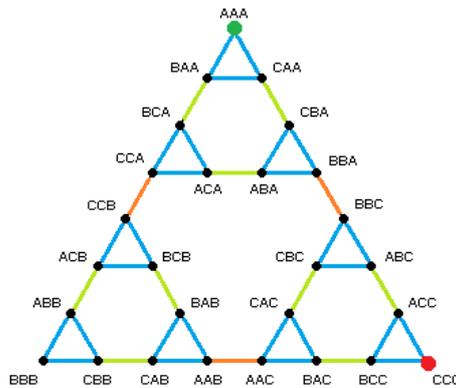


Caso largo:

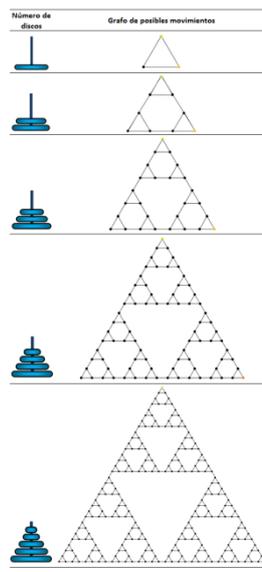


Comentarios curiosos sobre los grafos asociados al problema de las torres de Hanói

Con respecto a los grafos de las torres de Hanói, existe una conexión con el triángulo de Sierpinski. Mediante la teoría de grafos, el profesor Ian Stewart de The University of Warwick (Inglaterra), indica que cada serie tiene una única configuración. El grafo sigue una configuración similar a un triángulo equilátero cuyos vértices contienen varios triángulos colocados en su interior. Cada lado de un **triángulo pequeño** representa un movimiento del disco pequeño, cada lado de un **triángulo mediano** que no pertenezca a un triángulo pequeño representa un movimiento del disco segundo y cada lado del **triángulo grande** que no pertenezca a un triángulo menor representa los movimientos del disco grande. Este gráfico representa todos los posibles movimientos para la solución y cuando se incrementa el número de discos, el gráfico se convierte en un fractal. Cada una de sus partes reproduce el modelo del grafo completo. Por ejemplo, para 3 discos se tiene el grafo:

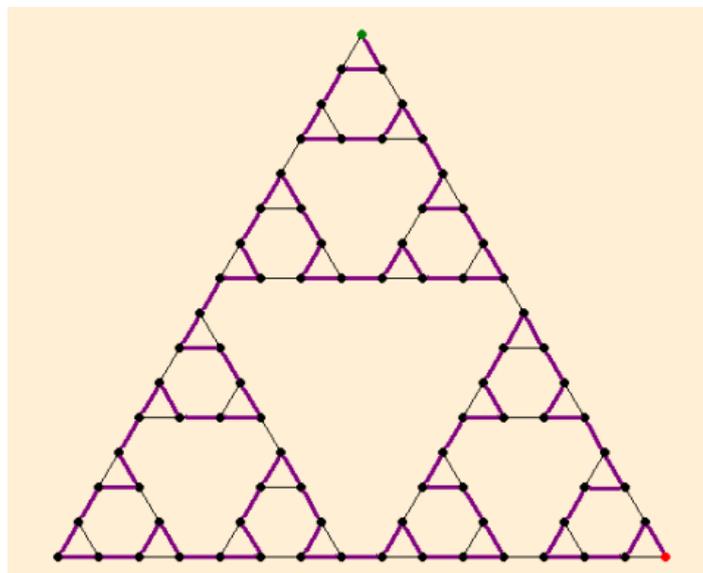


Grafo y triángulo de Sierpinski:



Veamos el caso de 4 discos: Se requiere quince movimientos ($2^4 - 1$) por el camino más corto.

Caso largo:



Veamos cómo se plantearía la solución utilizando STRIPS

Véase

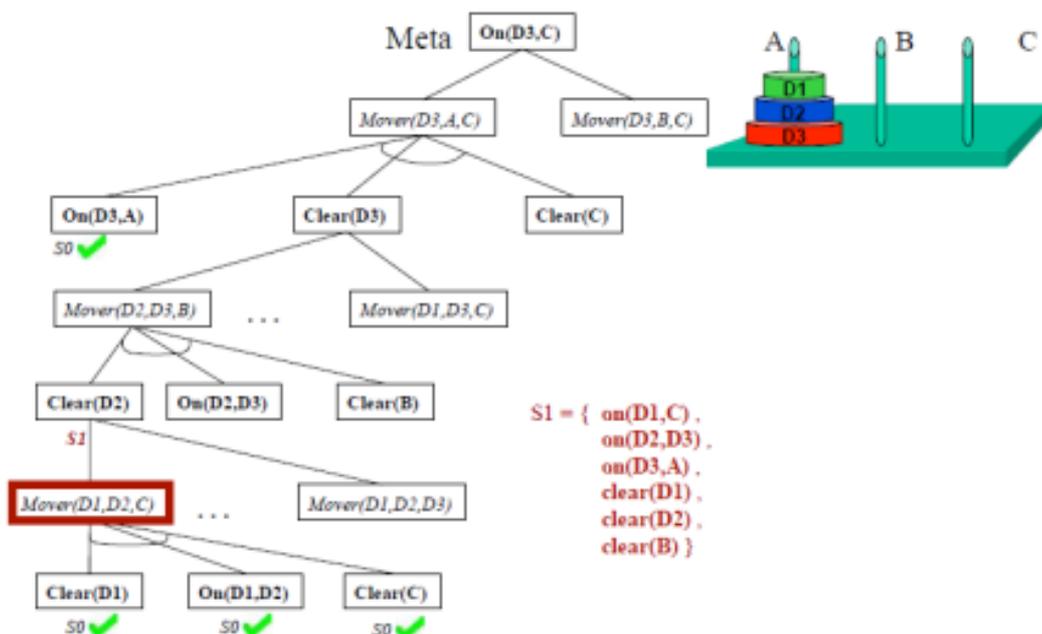
<http://www.udb.edu.sv/udb/archivo/guia/informatica-ingenieria/sistemas-expertos-e-inteligencia-artificial/2016/i/guia-11.pdf>

- Se tienen tres discos de tamaños diferentes D1 el pequeño, D2 el mediano, D3 el grande, y tres postes «A» el de partida, «B» el del medio y «C» el de llegada
- Elementos: $I = \{D1, D2, D3, A, B, C\}$
- Conjunto de propiedades:
 - $P = \{on(x,y): x,y \in I \wedge menor^*(x, y)\} \cup \{clear(x): x \in I\}$.
 - $clear(D_i)$ «no hay nada encima del disco i ».
 - $clear(x)$ «no hay ningún disco en la varilla x ».
 - $menor^*(x, y)$.
- Ejemplos:
 - $on(D1, D2), on(D1, D3), on(D2, D3)$.
 - $on(D1, A), on(D1, B), on(D1, C)$.
 - $on(D2, A), on(D2, B), on(D2, C)$.
 - $on(D3, A), on(D3, B), on(D3, C)$.
 - $clear(D1), clear(D2), clear(D3)$: «no hay nada encima del disco».
 - $clear(A), clear(B), clear(C)$: «no hay ningún disco en la varilla».
 - ...
- Conjunto de operadores:
 - $O = \{mover(x, y, z): x \in \{D1, D2, D3\} \wedge y, z \in I \wedge menor^*(x, y) \wedge menor^*(x, z) \wedge y \neq z\}$.
- Estructura de un operador:
 - Precondiciones (PC): Lista de elementos que tienen que ser ciertos en el estado para que la acción pueda ser aplicada.
 - Lista borrar (B): Lista de elementos que dejan de ser ciertos una vez aplicada la acción.
 - Lista añadir (A): Lista de elementos que pasan a ser ciertos una vez aplicada la acción.
- Plantilla para los efectos de los operadores:
 - $mover(\langle x \rangle, \langle y \rangle, \langle z \rangle)$:
 - PC: $\{clear(\langle x \rangle), clear(\langle y \rangle), on(\langle x \rangle, \langle y \rangle)\}$.
 - B: $\{on(\langle x \rangle, \langle y \rangle), clear(\langle z \rangle)\}$.
 - A: $\{on(\langle x \rangle, \langle z \rangle), clear(\langle y \rangle)\}$.

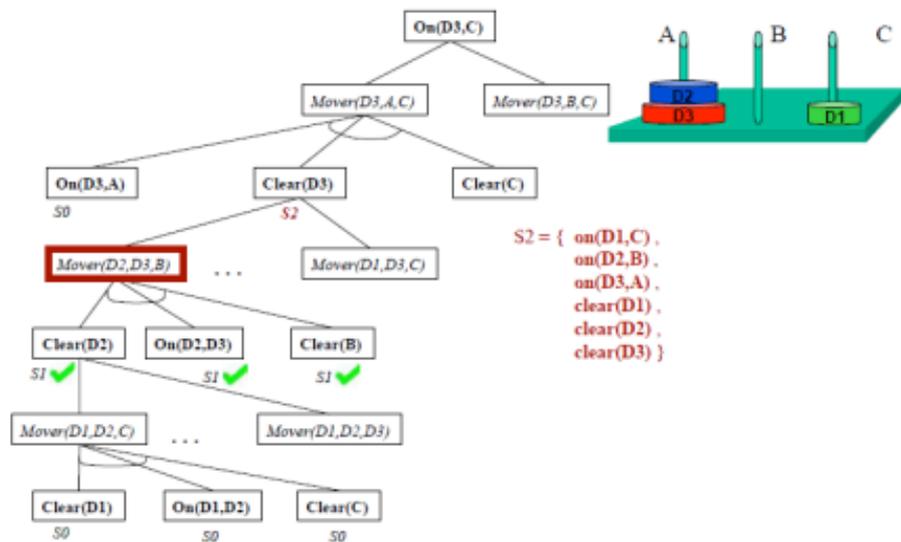
- Ejemplos:
 - Mover (D1, D2, D3):
 - PC: {clear (D1), clear (D3), on (D1, D2)}.
 - B: {on (D1, D2), clear (D3)}.
 - A: {on(D1, D3), clear (D2)}.
- Predicados auxiliares:
 - Menor* es el cierre transitivo de menor.
- Ejemplos:
 - menor (D1, D2), menor(D2, D3).
 - menor(D3, A), menor (D3, B), menor (D3, C).
 - ...

Propuesta de solución:

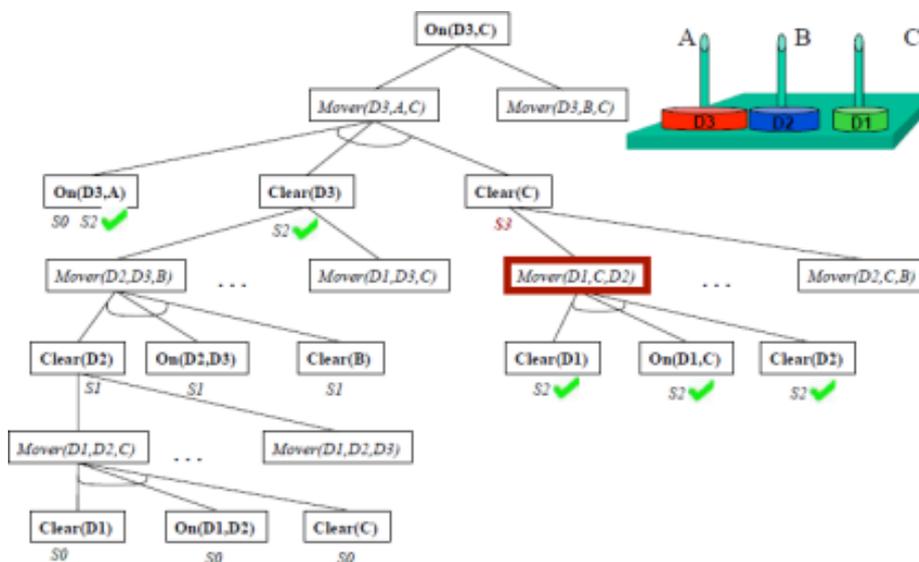
- Situación inicial:
 - $S_0 = \{on (D1, D2), on (D2, D3), on (D3, A), clear (D1), clear (B), clear(C)\}$.



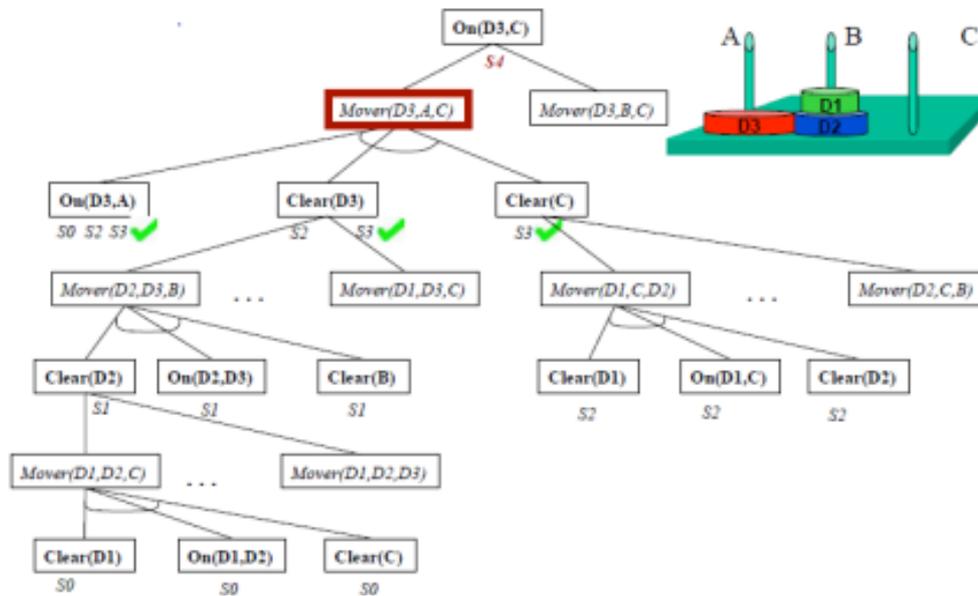
- Primer movimiento:
 - $S1 = \{on(D1, C), on(D2, D3), on(D3, A), clear(D1), clear(D2), clear(B)\}$.



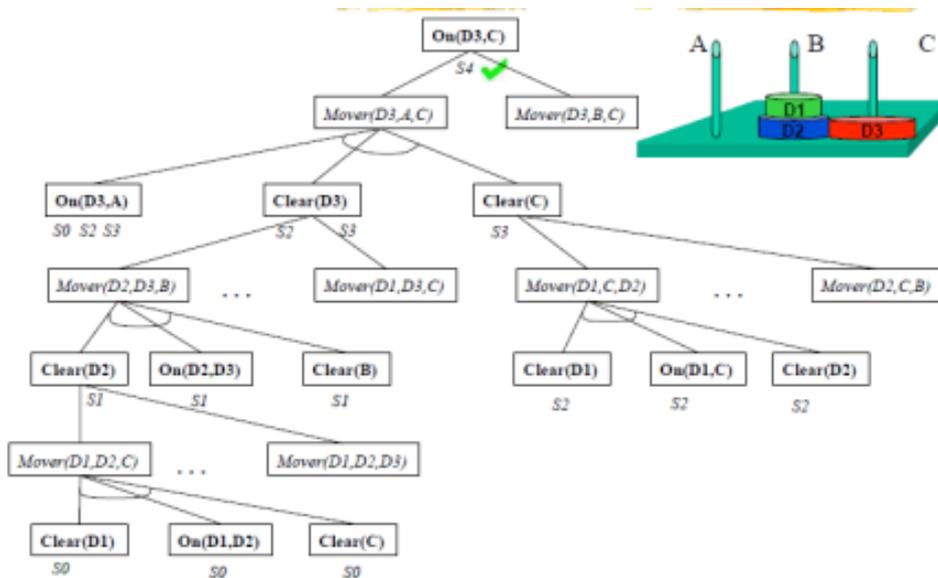
- Segundo movimiento:
 - $S2 = \{on(D1, C), on(D2, B), on(D3, A), clear(D1), clear(D2), clear(D3)\}$.



- Tercer movimiento:
 - $S3 = \{on(D1, D2), on(D2, B), on(D3, A), clear(D1), clear(D3), clear(C)\}$.



- Cuarto movimiento:
 - $S4 = \{on(D1, D2), on(D2, B), on(D3, C), clear(D1), clear(D3), clear(A)\}$.



• ...

Dado que es un problema de juguete, veamos también la solución algorítmica.

Pensemos en este problema desde abajo hacia arriba. Supongamos que usted tiene una torre de cinco discos, originalmente en un poste.

- Si usted ya sabe cómo mover una torre de cuatro discos al poste dos, entonces podría mover fácilmente el disco inferior al poste tres, y luego mover la torre de cuatro discos desde el poste dos al poste tres.

Pero ¿si usted no sabe cómo mover una torre de altura cuatro?

- Supongamos que usted sabe cómo mover una torre de altura tres al poste tres; entonces sería fácil mover el cuarto disco al poste dos y mover los tres discos del poste tres encima de aquel.

Pero ¿si usted no sabe cómo mover una torre de tres discos?

- ¿Qué tal si usted mueve una torre de dos discos al poste dos y luego mueve el tercer disco al poste tres, y luego mueve la torre de altura dos encima de dicho disco?

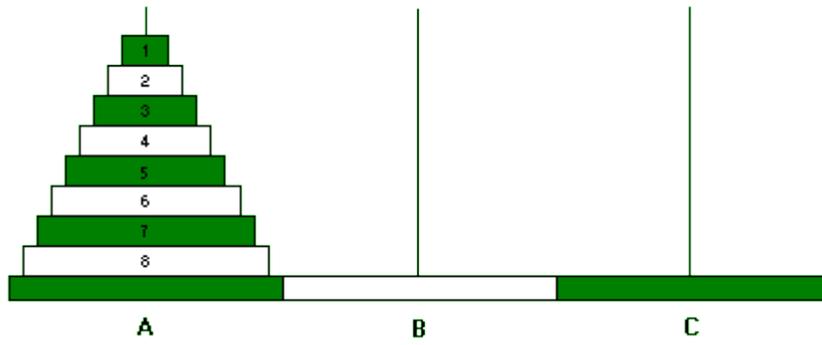
Pero ¿si todavía no sabe cómo hacer esto?

- Seguramente estará de acuerdo en que mover un solo disco al poste tres es bastante fácil, de hecho es trivial. Esto suena como un caso base.

Este planteamiento se denomina *planteamiento recursivo*, que funciona muy bien con los ordenadores, pero son difíciles de aplicar para un ser humano, ya que es muy fácil perderse.

Existe una alternativa a los algoritmos recursivos que son los algoritmos iterativos, en los que no hay llamadas a sí mismos, sino uno o varios bucles. Por simplicidad supongamos que los discos están pintados de la forma que se indica en la figura.

- Lleva el disco 1 sobre un disco (o trozo de base) que no sea de su mismo color.
- Si todos los discos están en C, fin.
- Si no, mueve un disco distinto de 1 y vuelve al paso 1.
- El disco más pequeño no es el único que no se coloca nunca sobre otro de distinto color, en ningún momento hay dos discos del mismo color juntos.



Véase <<http://www.rodoval.com/heureka/hanoi/>>

Algoritmos basados en el uso de la probabilidad (modelo simbólico)

Aplicar las técnicas mostradas hasta ahora a ciertos problemas del mundo real a veces no es posible por diferentes motivos como:

- La especificación completa del problema requeriría un esfuerzo que lo haría bastante inútil de aplicar.
- No se tiene toda la información del dominio de actuación, por diferentes razones como ruido o quizá la información recibida no es muy exacta o no es adecuada para el tipo de situación...
- Incluso conociendo las reglas, su aplicación produciría resultados inciertos.

Qué pasa cuando el entorno es parcialmente observable, o no determinístico, o no se conoce al completo. En este contexto se denomina *estado de creencia de un sistema* a una representación de las probabilidades de todos los posibles estados actuales del mundo en el que está inmerso el sistema. Empecemos con la teoría de la probabilidad.

Vivimos en entornos en constante cambio y necesitamos soluciones creativas para ajustarnos y adaptarnos a la incertidumbre. ¿Cómo se desenvuelve nuestro cerebro por los equívocos de lo incierto? El teorema de la inferencia bayesiana (que luego veremos) ideado por el teólogo y matemático inglés Thomas Bayes, y publicado póstumamente en 1763, que se ha convertido en uno de los principios fundamentales de la ciencia cognitiva moderna, lo explica. «Cualquier sistema biológico que resista la tendencia al desorden se adherirá a él». Al parecer, nuestro cerebro es un órgano estadístico de inferencia que opera bajo el principio de probabilidad bayesiano. «El cerebro, al tratar de anticipar lo que la próxima ola de sensaciones le comunicará, constantemente hace inferencias y actualiza sus creencias en función de lo que le transmiten los sentidos e intenta minimizar las señales de error de predicción y previene la sorpresa. Literalmente, es un órgano fantástico en el sentido de que genera hipótesis y fantasías que le son apropiadas para tratar de explicar los innumerables patrones y el flujo de información sensorial que está recibiendo».

Probabilidad

Como se acaba de decir, existen casos en los que se puede hacer uso de la probabilidad, ya sea objetiva o subjetiva, para estimar el posible resultado. Entendiendo por:

- La probabilidad objetiva es la posibilidad de que ocurra un resultado basándose en hechos concretos. Por ejemplo, tirar una moneda.
- La probabilidad subjetiva determina el resultado basándose en opiniones y juicios personales. No son afirmaciones sobre el mundo, sino sobre nuestro conocimiento. Por ello, las probabilidades pueden ir cambiando conforme vayan apareciendo evidencias nuevas. Por ejemplo, «muy difícil», «poco peligroso».

Subyacente a la teoría de la probabilidad está el azar o aleatoriedad, entendido como una casualidad que está presente en diversos fenómenos que no parecen ser predictibles en todos sus detalles. La razón de esa impredecibilidad depende del ámbito del conocimiento en el que nos movamos. El resultado de todo suceso aleatorio no puede determinarse en ningún caso antes de que este se produzca. Todo fenómeno aleatorio lleva asociado un espacio muestral, que no es más que el conjunto de todos los resultados posibles de dicho fenómeno.

Ejemplo: El lanzamiento de una moneda o el de los dados son eventos aleatorios.

- El espacio muestral en el lanzamiento de una moneda es $\{1,2\}$.
- El espacio muestral en el lanzamiento de un dado es $\{1,2,3,4,5,6\}$.

El elemento básico en la teoría de la probabilidad es la «variable aleatoria». Esta puede concebirse como un valor numérico perteneciente a un conjunto de valores discretos, o perteneciente a un conjunto de valores continuos, que está afectado por el azar. Es decir, dada una variable aleatoria no es posible conocer con certeza el valor que tomará esta al ser medida, aunque sí se conoce que cada medida posible tiene asociada una probabilidad concreta. Quizá pueda interpretarse en algunas situaciones como ignorancia acerca de las causas de un suceso. Y en otras ocasiones tendría que ver con el hecho de que no siempre podemos predecir el mundo, ya que en esas ocasiones el mundo realmente tiene un funcionamiento indeterminado o aleatorio.

La probabilidad asociada a un suceso o evento aleatorio es una medida del grado de certidumbre de que dicho suceso pueda ocurrir. Se suele expresar como un número comprendido en el intervalo $[0, 1]$, entendiendo que un suceso imposible tiene probabilidad cero y un suceso seguro tiene probabilidad uno.

Pero ¿qué es realmente la probabilidad? ¿Qué quiere decir? Nadie jamás ha observado una posibilidad, solo podemos observar los hechos. Cuando se tira un dado, solo se puede observar un resultado. Las otras cinco posibilidades solo han existido en la imaginación y en los cálculos. ¿Por qué entonces decimos que existían? ¿Por qué decimos que es posible que hubiera salido un resultado diferente al que ocurrió verdaderamente?

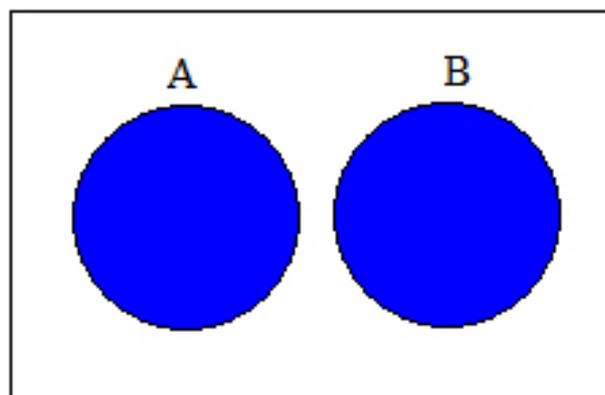
Si se denota como $P(E)$ la probabilidad de que ocurra un evento E , la forma de calcularla viene definida por: $P(E) = \frac{\text{casos favorables}}{\text{casos posibles}}$ [fórmula 1]

Cuando se producen dos o más eventos, los dos pueden ser independientes o dependientes entre sí. Se considera que un suceso B es independiente de un suceso A si la probabilidad de B no cambia cuando se cuenta con la información de que ha ocurrido A . Si los dos sucesos son dependientes se escribe $P(B|A)$. A la $P(B|A)$ se la denomina probabilidad condicional y se lee «la probabilidad de que ocurra B dado A » o $P(A|B)$ que se lee «la probabilidad de que ocurra A dado B ». Ambas son diferentes.

Cuando se producen dos o más eventos, las probabilidades pueden calcularse mediante las siguientes reglas:

- Regla de la adición:
 - La probabilidad de que ocurra uno de dos eventos que son excluyentes, es decir, que no pueden ocurrir a la vez es

$$P(A \vee B) = P(A) + P(B) \text{ [fórmula 2]}$$



Piense que el área de cada círculo es la probabilidad de cada suceso.

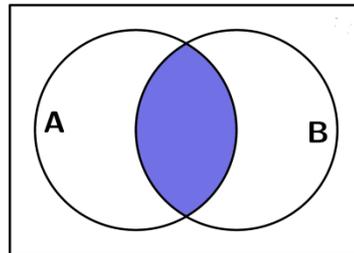
- Por ejemplo lanzar una moneda y sacar cara y sacar cruz
 $P(\text{cara} \vee \text{cruz}) = P(\text{cara}) + P(\text{cruz}) = \frac{1}{2} + \frac{1}{2} = 1$

- La probabilidad de que ocurra uno de dos eventos que no son mutuamente excluyentes es

$$P(A \vee B) = P(A) + P(B) - P(A \wedge B) \text{ [fórmula 3]}$$

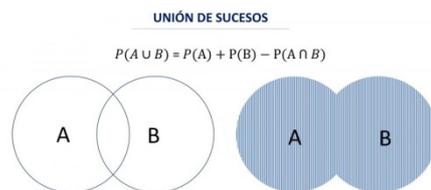
Que también se puede escribir como

$$P(A \wedge B) = P(A) + P(B) - P(A \vee B)$$



La zona común a las dos áreas es lo que se conoce como $A \wedge B$.

Por lo tanto:



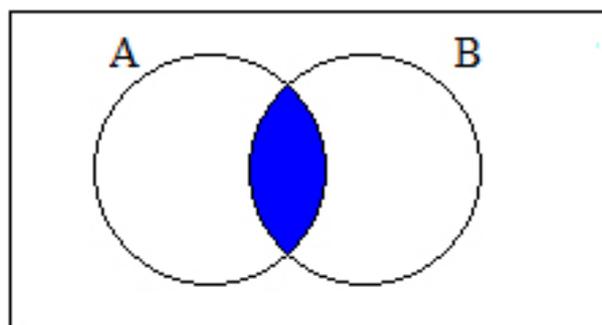
ECONOMIPEDIA.COM

- Por ejemplo, sacar de una baraja de cuarenta cartas y cuatro palos (oros, copas, bastos y espadas) un 5 o una carta de espadas

$$P(\text{un } 5 \vee \text{espada}) = P(\text{un } 5) + P(\text{espada}) - P(\text{cinco} \wedge \text{espada}) = \frac{1}{4} + \frac{10}{40} + \frac{1}{10} = \frac{3}{5}$$

- Regla de la multiplicación:
 - La probabilidad de que ocurran a la vez dos eventos que son independientes viene dada por

$$P(A \wedge B) = P(A) \times P(B) \text{ [fórmula 4]}$$



- Por ejemplo, una caja contiene 4 bolas rojas, 3 verdes y 2 azules. Cuál es la probabilidad de que al sacar una bola esta sea azul y la siguiente verde. Comentario. Cada vez que se saca una bola, esta se devuelve a la caja.

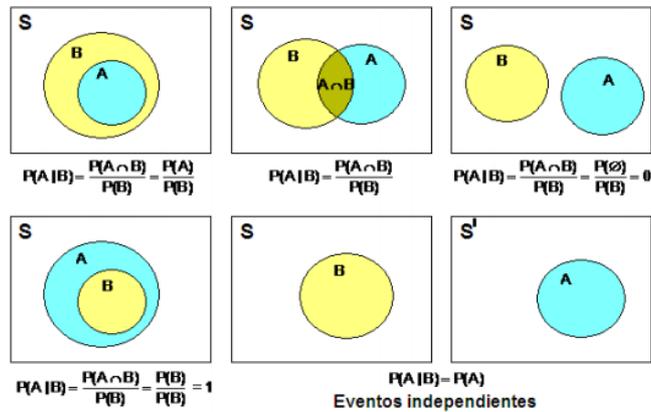
$$P(\text{roja} \wedge \text{verde}) = P(\text{roja}) \times P(\text{verde}) = \frac{4}{9} + \frac{3}{9} = \frac{7}{9}$$

- La probabilidad de que ocurran dos eventos siendo uno dependiente del otro es

$$P(A \wedge B) = P(B) \times P(A|B) \text{ [fórmula 5]}$$

indicando con $(A|B)$, que ocurre A una vez que ha ocurrido B .

INTERPRETACIÓN GRÁFICA



- Por ejemplo, una caja contiene 4 bolas rojas, 3 verdes y 2 azules. Cuál es la probabilidad de que al sacar una bola esta sea roja y la siguiente verde. Comentario. Cada vez que se saca una bola, esta no se devuelve a la caja.

$$P(\text{verde} \wedge \text{roja}) = P(\text{roja}) \times P(\text{verde}|\text{roja}) = \frac{4}{9} + \frac{3}{8} = \frac{59}{72}$$

- Reglas de Laplace:
 - La probabilidad de ocurrencia de un suceso imposible es 0.
 - La probabilidad de ocurrencia de un suceso seguro es 1.

Inferencia probabilística o cálculo de probabilidades posteriores dadas nuevas evidencias.

A la $P(A|B)$ se la denomina *probabilidad condicional* y se lee «la probabilidad de que ocurra A dado B ». Para su cálculo no tiene por qué haber una relación causal o temporal entre A y B . A puede preceder en el tiempo a B , sucederlo o pueden ocurrir simultáneamente. A puede causar B , B puede causar A , o pueden no tener relación

causal. Las relaciones causales o temporales son nociones que no pertenecen al ámbito de la probabilidad. Pueden desempeñar un papel o no.

Ejemplo: Lanzamiento de una moneda para luego lanzar un dado. ¿Cuál es la probabilidad de que en el dado salga un 6 dado que ya ha salido una cara en la moneda? Esa probabilidad se denotaría $P(6|cara)$.

La probabilidad condicional de que ocurra A dado B se calcula utilizando la fórmula 5, del modo siguiente

$$P(A|B) = \frac{P(A \wedge B)}{P(B)} \text{ [fórmula 5]}$$

Ejemplo:

- Probabilidad de que salga un 6 en un dado $P(6) = \frac{1}{6}$
- Probabilidad de que haya salido una cara en un moneda $P(cara) = \frac{1}{2}$
- Entonces:
 - La probabilidad de que salga un 6 en un dado habiendo salido una cara

$$P(6|cara) = \frac{\frac{1}{6} \times \frac{1}{2}}{\frac{1}{2}} = \frac{1}{6}$$
 - La probabilidad de que salga una cara habiendo salido un 6 en un dado

$$P(cara|6) = \frac{\frac{1}{2} \times \frac{1}{6}}{\frac{1}{6}} = \frac{1}{2}$$

Teorema de Bayes

En teoría de la probabilidad, es una proposición planteada por el matemático inglés Thomas Bayes (1702-1761) que expresa la probabilidad condicional de un evento aleatorio A dado B en términos de la distribución de probabilidad condicional del evento B dado A.

Si (1) $P(A|B) = \frac{P(A \wedge B)}{P(B)}$ y si (2) $P(B|A) = \frac{P(B \wedge A)}{P(A)}$ entonces [fórmula 5]

De (1) $P(A \wedge B) = P(A|B) \times P(B)$

De (2) $P(B \wedge A) = P(B|A) \times P(A)$

Como $P(A \wedge B) = P(B \wedge A)$ entonces $P(A|B) \times P(B) = P(B|A) \times P(A)$

Teorema de Bayes $P(A|B) = \frac{P(B|A) \times P(A)}{P(B)}$ [fórmula 6]

En su forma más genérica, sea $\{A_1, A_2, A_3, \dots, A_i, \dots, A_n\}$ un conjunto de sucesos mutuamente excluyentes y exhaustivos, y tales que la probabilidad de cada uno de ellos es distinta de cero (0). Sea B un suceso cualquiera del que se conocen las probabilidades condicionales $P(B|A_i)$. Entonces:

1. La probabilidad $P(B|A_i)$ viene dada por la expresión:

$$P(A_i|B) = \frac{P(B|A_i) \times P(A_i)}{P(B)} \text{ (teorema de Bayes 1763) [fórmula 7]}$$

Denominándose:

- $P(A_i)$ probabilidades *a priori*
- $P(B|A_i)$ probabilidad de B en la hipótesis A_i
- $P(A_i|B)$ Probabilidades *a posteriori*

Ejemplo

Supóngase un colegio mixto donde el 60 % de los estudiantes son chicos y el 40 % son chicas. Las chicas llevan pantalón o falda en probabilidades iguales; los chicos siempre llevan pantalones. Un observador ve desde lejos a un estudiante aleatorio; lo único que puede distinguir el observador es que el o la estudiante lleva pantalones. ¿Cuál es la probabilidad de que sea una chica? La respuesta correcta se puede hallar usando el teorema de Bayes.

El evento A es que el estudiante observado sea una chica, mientras que el evento B es que el estudiante observado lleva pantalones. Para hallar $P(A|B)$, primero necesitamos saber:

- $P(A)$, o la probabilidad de que el estudiante sea una chica a pesar de cualquier otra información. Ya que el observador ve un estudiante aleatorio, y dado que todos los estudiantes tienen la misma probabilidad de ser observados, y el porcentaje de chicas entre los estudiantes es el 40 %, esta probabilidad es 0,4.
- $P(A')$, o la probabilidad de que el estudiante observado sea un chico a pesar de cualquier otra información (A' es el complementario del evento A). Esto es el 60 %, o 0,6.
- $P(B|A)$, o la probabilidad de que el estudiante lleve pantalones siendo una chica. Como tienen la misma probabilidad de llevar falda o pantalones, esto es 0,5.
- $P(B|A')$, o la probabilidad de que el estudiante lleve pantalones siendo un chico. Esto es 1.
- $P(B)$, o la probabilidad de que un estudiante (aleatorio) lleve pantalones. Dado que $P(B) = P(B|A)P(A) + P(B|A')P(A')$, esto es $0,5 \times 0,4 + 1 \times 0,6 = 0,8$.

Dados todos estos datos, la probabilidad de que el observador haya visto a una chica habiendo observado que lleva pantalones puede ser calculada sustituyendo estos valores en la fórmula:

$$P(A|B) = P(B|A)P(A)/p(B) = 0,25$$

2. Teorema de la probabilidad total. Se tiene que

Sea $\{A_1, A_2, A_3, \dots, A_i, \dots, A_n\}$ una partición sobre el espacio muestral y sea B un suceso cualquiera del que se conocen las probabilidades condicionales $P(B|A_i)$, entonces la probabilidad del suceso B viene dada por la expresión.

El suceso B se puede escribir como $B = (B \wedge A_1) \vee (B \wedge A_2) \vee \dots \vee (B \wedge A_n)$, siendo cada $(B \wedge A_i)$ disjuntos dos a dos, ya que lo son los A_i , por lo tanto la $P(B) = (B \wedge A_1) + (B \wedge A_2) + \dots + (B \wedge A_n)$. Y como sabemos que $P(B \wedge A_i) = P(B|A_i)P(A_i)$

Entonces

$$P(B) = \sum_{i=1}^n P(B|A_i) \times P(A_i) \text{ [fórmula 8]}$$

Ejemplo: Una fábrica de coches fabrica tres tipos de coches C1, C2 y C3, en una proporción respectivamente de (4/10), (5/10) y (1/10). Además, la probabilidad de que un coche de tipo C1 se avería durante el primer año es de 0,07, para el coche C2 es 0,04 y para el coche 3 es 0,09. ¿Cuál es la probabilidad de que ocurra el suceso B definido como que un coche producido en esa fábrica tenga una avería antes de un año?

$$P(B) = \sum_{i=1}^n P(B|C_i) \times P(C_i)$$

$$P(B) = 0,07 \times (4/10) + 0,04 \times (5/10) + 0,09 \times (1/10) = 0,057$$

Teniendo en cuenta los dos teoremas anteriores se deduce de forma directa que

$$P(A_j|B) = \frac{P(B|A_j) \times P(A_j)}{P(B)} = \frac{P(B|A_j) \times P(A_j)}{\sum_{i=1}^n P(B|A_i) \times P(A_i)} \text{ teorema de Bayes [fórmula 9]}$$

Fórmula que se interpreta del modo siguiente. En muchas ocasiones se dispone de una descomposición del espacio muestral en un sistema completo de sucesos $\{A_1, A_2, \dots, A_n\}$ mutuamente excluyentes, cuyas probabilidades $P(A_i)$ se conocen, en principio, para todos los A_i (probabilidades *a priori*). Supongamos que los A_i no son directamente observables y que nos interesa calcular la probabilidad de que haya ocurrido concretamente el suceso A_j . Si es posible realizar un experimento que produzca un resultado B, cuyas probabilidades condicionales $P(B|A_i)$ también se conocen para todos los A_i , entonces el teorema de Bayes permite usar la información aportada por B para calcular la probabilidad de que haya ocurrido A_j , esto es, la probabilidad $P(A_j|B)$ (probabilidad *a posteriori*).

Ejemplo: Consideremos una prueba diagnóstica para la detección de determinado tipo de cáncer. Para que esta sea aceptable ha de tener alta sensibilidad y alta especificidad. Estos conceptos se definen del siguiente modo (se entiende que dar positivo en la prueba significa que esta detecta la presencia del cáncer):

- Sensibilidad = $P(\text{Que la prueba dé positivo} \mid \text{El individuo está enfermo})$.
- Especificidad = $P(\text{Que la prueba dé negativo} \mid \text{El individuo está sano})$.

Como las pruebas diagnósticas no son exactas, puede ocurrir que den positivo en individuos sanos (falsos positivos) y negativo en individuos enfermos (falsos negativos), entonces se definen también:

- Coeficiente falso_positivo = $1 - \text{especificidad}$.
- Coeficiente falso_negativo = $1 - \text{sensibilidad}$.

Si se denomina:

- C «padece cáncer».
- T «da positivo en la prueba diagnóstica».

Se puede expresar y le damos valores para un ejemplo:

- Coeficiente falso_positivo = $P(T \mid \neg C)$ = 0,01.
- Especificidad = $P(\neg T \mid \neg C)$ = 0,99.
- Coeficiente falso_negativo = $P(\neg T \mid C)$ = 0,001.
- Sensibilidad = $P(T \mid C)$ = 0,999.

Supongamos que se desea saber si una persona elegida arbitrariamente de una población padece cáncer. Si $p_c = (1/1000)$ es la proporción de enfermos de cáncer en esa población, *a priori* la probabilidad de que esa persona tenga cáncer puede tomarse como $P(C) = p_c = (1/1000)$ y de que esté sana es $P(\neg C) = p_s = (999/1000)$.

Para mejorar su información, el médico le realiza una prueba diagnóstica cuya sensibilidad y especificidad se conocen. Supongamos que el resultado de la prueba es positivo. El médico y el presunto enfermo pueden ahora preguntarse:

- ¿Cuál es la probabilidad de que el paciente esté realmente enfermo sabiendo que ha dado positivo en la prueba?

Apliquemos el teorema de Bayes [fórmula 8]

$$P(C \mid T) = \frac{P(T \mid C) \times P(C)}{P(T)} = \frac{P(T \mid C) \times P(C)}{P(T \mid C) \times P(C) + P(T \mid \neg C) \times P(\neg C)}$$

$$P(C|T) = \frac{0,999 \times \frac{1}{1000}}{0,999 \times \frac{1}{1000} + 0,01 \times \frac{999}{1000}} = 0,091$$

- ¿Cuál es la probabilidad de que el paciente esté realmente sano sabiendo que ha dado negativo en la prueba?

Apliquemos el teorema de Bayes [fórmula 8]

$$P(\neg C|\neg T) = \frac{P(\neg T|\neg C) \times P(\neg C)}{P(\neg T)} = \frac{P(\neg T|\neg C) \times P(\neg C)}{P(\neg T|C) \times P(C) + P(\neg T|\neg C) \times P(\neg C)}$$

$$P(\neg C|\neg T) = \frac{0,99 \times \frac{999}{1000}}{0,001 \times \frac{1}{1000} + 0,99 \times \frac{999}{1000}} = 0,999$$

¿Cómo podríamos interpretar $P(A|B \wedge C)$?

Esa fórmula se interpreta del modo siguiente. En muchas ocasiones se dispone de una descomposición del espacio muestral en un sistema completo de sucesos $\{A_1, A_2, \dots, A_n\}$ mutuamente excluyentes, cuyas probabilidades $P(A_i)$ se conocen, en principio, para todos los A_i (probabilidades *a priori*). Supongamos que los A_i no son directamente observables y que nos interesa calcular la probabilidad de que haya ocurrido concretamente el suceso A_j . Si es posible realizar dos experimentos que produzcan un resultado B y otro resultado C , de tal manera ambos sean independientes entre sí, y sus probabilidades condicionales $P(B|A_i)$ $P(C|A_i)$ también se conocen para todos los A_i , entonces el teorema de Bayes permite usar la información aportada por B y por C para calcular la probabilidad de que haya ocurrido A_j , esto es, la probabilidad $P(A_j|B)$ y la probabilidad $P(A_j|C)$ (probabilidades *a posteriori*).

La forma de calcularla es rápida, consideremos D a $(B \wedge C)$ y calculemos $P(A|D)$ según el teorema de Valles

$$P(A_j|D) = \frac{P(D|A_j) \times P(A_j)}{P(D)} = \frac{P(B \wedge C|A_j) \times P(A_j)}{P(B \wedge C)} = \frac{P(B|A_j) \times P(C|A_j) \times P(A_j)}{P(B) \times P(C)} = \alpha (P(B|A_j) \times P(C|A_j) \times P(A_j)) \quad \text{[fórmula 10]}$$

Esa expresión se denomina *teorema de Bayes simple*, ya que se ha supuesto que B y C no son independientes condicionalmente.

Es decir $P(B \wedge C|A_j) = P(B|A_j) \times P(C|A_j)$

Siendo

$$\alpha = \frac{1}{P(B) \times P(C)}$$

Generalizando

$$P(A_j|B_1, B_2, B_3, \dots, B_n) = \alpha P(A_j) \prod_1^n P(B|A_j) \quad \text{[fórmula 10]}$$

Siendo

$$\alpha = \frac{1}{P(B_1) \times P(B_2) \times P(B_3) \times \dots \times P(B_n)}$$

Veamos ahora lo que se conoce como regla de la cadena, que permite calcular la probabilidad siguiente $P(X_1 \wedge X_2 \wedge X_3 \wedge \dots \wedge X_n)$, que por eficiencia la vamos a escribir a partir de ahora como $P(X_1, X_2, X_3, \dots, X_n)$, teniendo en cuenta la fórmula 5
Para eventos dependientes

$$P(A \wedge B) = P(B) \times P(A|B) \text{ [fórmula 5]}$$

$$\begin{aligned} P(X_1, X_2, X_3, \dots, X_n) &= P(X_n | X_{n-1}, X_{n-2}, \dots, X_1) P(X_{n-1}, X_{n-2}, \dots, X_1) = \\ &P(X_n | X_{n-1}, X_{n-2}, \dots, X_1) P(X_{n-1} | X_{n-2}, X_{n-3}, \dots, X_1) P(X_{n-2}, X_{n-3}, \dots, X_1) = \\ &P(X_n | X_{n-1}, X_{n-2}, \dots, X_1) P(X_{n-1} | X_{n-2}, X_{n-3}, \dots, X_1) \dots P(X_2 | X_1) P(X_1) \end{aligned}$$

Por lo tanto, la expresión de la regla de la cadena es

$$P(X_1, X_2, X_3, \dots, X_n) = \prod_{i=1}^n P(X_i | X_1, \dots, X_{i-1}) \text{ [fórmula 11]}$$

Razonamiento probabilístico

La probabilidad *a priori* o incondicional asociada a una proposición «a» es el grado de creencia que se le otorga en ausencia de cualquier información; y se escribe como $P(a)$. Ese valor solo se puede utilizar cuando no hay otra información. Cuando queremos hacer referencia a todas las probabilidades de todos los valores posibles de una variable aleatoria, se utilizará la expresión $P(A)$, que denota un vector de valores que se corresponden a las probabilidades de cada estado individual.

Ejemplo: Si $A = \{\text{rojo, verde, azul, amarillo}\}$ y $P(\text{rojo}) = 0,54$, $P(\text{verde}) = 0,23$, $P(\text{azul}) = 0,13$ y $P(\text{amarillo}) = 0,1$, entonces se puede escribir lo siguiente:

$$P(A) = (0,54, 0,23, 0,13, 0,1)$$

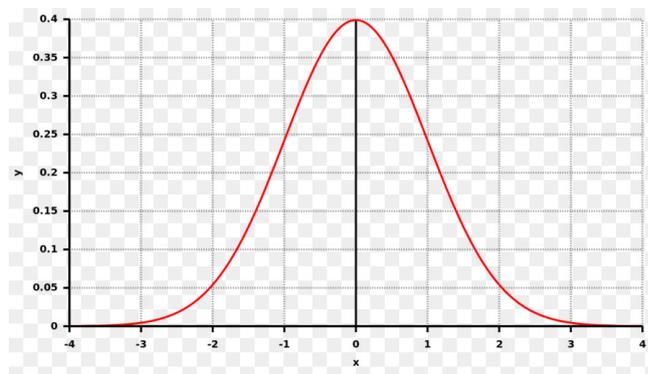
Esa expresión define una distribución de probabilidad *a priori* para la variable aleatoria A .

Cuando tenemos varias variables aleatorias A, B, \dots, Z , se utiliza la distribución de probabilidad conjunta completa.

Ejemplo: Dadas las variables $A = \{a_1, a_2, a_3\}$, $B = \{b_1, b_2, b_3, b_4\}$ y $C = \{c_1, c_2\}$, entonces la distribución de probabilidad conjunta sería una tabla de $3 \times 4 \times 2$, formado por tuplas del tipo $P(a_i, b_j, c_k)$

Para variables aleatorias continuas, no es posible escribir la distribución completa como una tabla, ya que hay infinitos valores. Las distribuciones de probabilidad para variables continuas se llaman *funciones de densidad de probabilidad*.

Ejemplo: Una función densidad de probabilidad podría ser la de la figura. Observe que para cualquier valor del intervalo continuo $[-4, 4]$ representado en el eje X , la función devuelve una probabilidad representada en el eje Y .



Una vez que se obtiene alguna evidencia referente a las variables aleatorias desconocidas que constituyen el dominio completo, las probabilidades *a priori* ya no son aplicables. En vez de ellas, ahora se pueden calcular las probabilidades *a posteriori* o condicionales. La notación que se va a usar es $P(a|b)$, es decir, la probabilidad de a , supuesto que todo lo que conozco es b . Ya sabemos que se puede calcular aplicando el teorema de Bayes. Se puede utilizar la notación $P(\mathbf{X}|\mathbf{Y})$ interpretada como $P(X=x_i|Y=y_j)$ para cada posible i, j .

Por ejemplo: Si $\mathbf{A} = \{a_1, a_2, a_3\}$, $\mathbf{B} = \{b_1, b_2, b_3, b_4\}$, $P(\mathbf{A}|\mathbf{B})$ sería

$P(a_i b_j)$	a_1	a_2	a_3
b_1	$P(a_1 b_1)$	$P(a_2 b_1)$	$P(a_3 b_1)$
b_2	$P(a_1 b_2)$	$P(a_2 b_2)$	$P(a_3 b_2)$
b_3	$P(a_1 b_3)$	$P(a_2 b_3)$	$P(a_3 b_3)$
b_4	$P(a_1 b_4)$	$P(a_2 b_4)$	$P(a_3 b_4)$

La tarea básica de cualquier sistema de inferencia probabilística es calcular la distribución de probabilidad *a posteriori* para un conjunto de variables pregunta, dado algún evento observado, es decir, para alguna asignación de valores a un conjunto de variables evidencia.

Notación:

- X denota la variable pregunta.
- \mathbf{E} denota el conjunto de variables evidencia $E_1, E_2, E_3, \dots, E_i, \dots, E_m$
- \mathbf{e} es un evento observado particular, $E_1 = e_1, E_2 = e_2, \dots, E_i = e_i, \dots, E_m = e_m$
- \mathbf{Y} denota las variables no evidencia u ocultas $Y_1, Y_2, Y_3, \dots, Y_i, \dots, Y_L$

Por lo tanto, el conjunto completo de variables es $\mathbf{X} = \{X\} \cup \mathbf{E} \cup \mathbf{Y}$

Una pregunta típica consiste en calcular la probabilidad *a posteriori* $P(X|\mathbf{e})$ suponiendo que la variable pregunta no está entre las variables evidencia. Por simplicidad se supone que la pregunta queda representada mediante una sola variable.

Modelo: Es lo que el agente sabe sobre las relaciones entre las variables observadas y las ocultas. Es lo que va a permitir hacer inferencias. Para ello se va a utilizar la distribución de probabilidad conjunta completa.

Ejemplo: Un fabricante de caramelos fabrica cinco tipos de bolsas (h), las hace con diferente probabilidad y cada una con una proporción diferente entre limón y naranja. El número de caramelos por bolsa es muy grande.

	Probabilidad	Naranja	Limón
h_1	0,1	100 %	0 %
h_2	0,2	75 %	25 %
h_3	0,4	50 %	50 %
h_4	0,2	25 %	75 %
h_5	0,1	0 %	100 %

Tomamos una bolsa (que es del tipo 5 pero NO lo sabemos) y vamos abriendo algunos caramelos, anotando su sabor y posteriormente se devuelven a la bolsa. ¿Podemos predecir el sabor del siguiente caramelo?

- Conocemos las composiciones de cada bolsa.
- Los datos son los distintos sabores de los caramelos abiertos
 $\mathbf{D} = \{d_1, d_2, d_3, \dots, d_n\}$, sabiendo que $d_i =$ caramelo de limón, $\forall i$.
- Las probabilidades *a priori* son:
 - Las probabilidades de fabricación de cada bolsa, $p(h)$.
 - Las probabilidades de la forma $\mathbf{P}(\mathbf{D}|h_i) = \prod_1^n P(d_j|h_i)$ [véase fórmula 10].
- Y queremos las respuestas a las preguntas:
 - $P(h_i|\mathbf{D}) = \alpha P(h_i) \prod_1^n P(d_j|h_i)$.
- Y también queremos la respuesta a la pregunta:
 - $P(d_{n+1}|\mathbf{D}) = \sum_{i=1}^5 P(d_{n+1}|\mathbf{D}, h_i) \times P(h_i|\mathbf{D}) = \sum_{i=1}^5 P(d_{n+1}|h_i) \times P(h_i|\mathbf{D})$.

$P(\mathbf{D} h_i)$	d_1 limón	d_1, d_2 limón	d_1, d_2, d_3 limón	...	d_1, d_2, \dots, d_{10} limón
h_1	$P(d_1 h_1)$	$P(d_1 h_1)P(d_2 h_1)$	$P(d_1 h_1)P(d_2 h_1)P(d_3 h_1)$...	
h_2	$P(d_1 h_2)$	$P(d_1 h_2)P(d_2 h_2)$	$P(d_1 h_2)P(d_2 h_2)P(d_3 h_2)$...	
h_3	$P(d_1 h_3)$	$P(d_1 h_3)P(d_2 h_3)$	$P(d_1 h_3)P(d_2 h_3)P(d_3 h_3)$...	
h_4	$P(d_1 h_4)$	$P(d_1 h_4)P(d_2 h_4)$	$P(d_1 h_4)P(d_2 h_4)P(d_3 h_4)$...	
h_5	$P(d_1 h_5)$	$P(d_1 h_5)P(d_2 h_5)$	$P(d_1 h_5)P(d_2 h_5)P(d_3 h_5)$...	

Sabiendo que al ser todos los caramelos que se extraen y se devuelven son de limón:

- $P(d_i|h_1) = 0$ para todo i.
- $P(d_i|h_2) = 0,25$ para todo i.
- $P(d_i|h_3) = 0,50$ para todo i.
- $P(d_i|h_4) = 0,75$ para todo i.
- $P(d_i|h_5) = 1$ para todo i.

$P(D h_i)$	d_1 limón	d_1, d_2 limón	d_1, d_2, d_3 limón	...	d_1, d_2, \dots, d_{10} limón
h_1	0	0	0	...	0
h_2	0,25	$0,25 \times 0,25$	$0,25 \times 0,25 \times 0,25$...	$0,25^{10}$
h_3	0,50	$0,50 \times 0,50$	$0,50 \times 0,50 \times 0,50$...	$0,50^{10}$
h_4	0,75	$0,75 \times 0,75$	$0,75 \times 0,75 \times 0,75$...	$0,75^{10}$
h_5	1	1	1	...	1

- La respuesta a la pregunta $P(h_i|d) = \alpha P(h_i) \prod_1^n P(d_j|h_i)$

$P(h_i D)$	d_1 limón	d_1, d_2 limón	d_1, d_2, d_3 limón	...	d_1, d_2, \dots, d_{10} limón
h_1	0	0	0	...	0
h_2	$\alpha_{2,1} \times 0,2 \times 0,25$	$\alpha_{2,2} \times 0,2 \times 0,25 \times 0,25$	$\alpha_{2,3} \times 0,2 \times 0,25 \times 0,25 \times 0,25$...	$\alpha_{2,10} \times 0,2 \times 0,25^{10}$
h_3	$\alpha_{3,1} \times 0,4 \times 0,50$	$\alpha_{3,2} \times 0,4 \times 0,50 \times 0,50$	$\alpha_{3,3} \times 0,4 \times 0,50 \times 0,50 \times 0,50$...	$\alpha_{3,10} \times 0,4 \times 0,50^{10}$
h_4	$\alpha_{4,1} \times 0,2 \times 0,75$	$\alpha_{4,2} \times 0,2 \times 0,75 \times 0,75$	$\alpha_{4,3} \times 0,2 \times 0,75 \times 0,75 \times 0,75$...	$\alpha_{4,10} \times 0,2 \times 0,75^{10}$
h_5	$\alpha_{5,1} \times 0,1 \times 1$	$\alpha_{5,2} \times 0,1 \times 1$	$\alpha_{5,3} \times 0,1 \times 1$...	$\alpha_{5,10} \times 0,1 \times 1$

Siendo

$$\alpha_{i,j} = \frac{1}{\prod_1^j P(d_i)}$$

donde

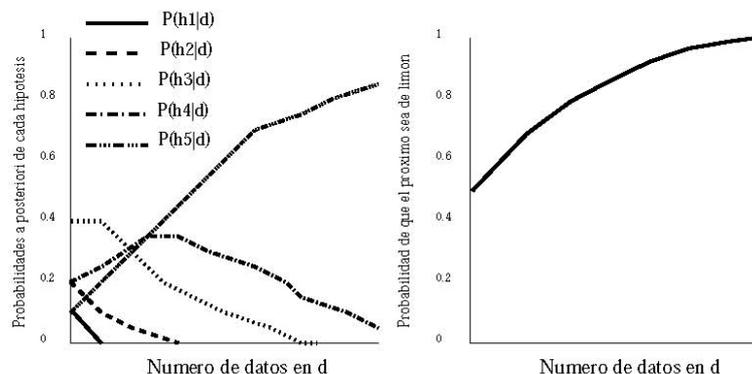
$$P(d_i) = \sum_{i=1}^n P(d_i|h_i) \times P(h_i)$$

$$P(\text{limón}) = 0 \times 0,1 + 0,25 \times 0,2 + 0,5 \times 0,4 + 0,75 \times 0,2 + 1 \times 0,1 = 0,5$$

- La respuesta a la pregunta X = predecir el sabor del siguiente caramelo

$$P(d_{n+1}|D) = \sum_{i=1}^5 P(d_{n+1}|D, h_i) \times P(h_i|D) = \sum_{i=1}^5 P(d_{n+1}|h_i) \times P(h_i|D)$$

- Aplicando las fórmulas anteriores se obtienen los siguientes resultados, como reflejan las gráficas:

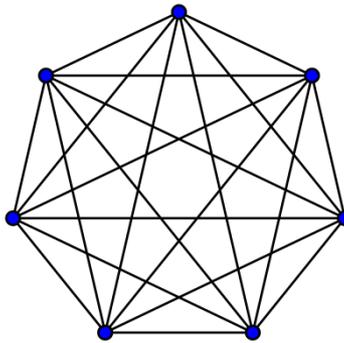


La gráfica de la izquierda representa las probabilidades *a posteriori* $P(h_i|d_1, d_2, \dots, d_n)$. La gráfica de la derecha es la predicción bayesiana $P(d_{n+1}|d_1, d_2, \dots, d_n)$. El número de observaciones va de 1 a 10.

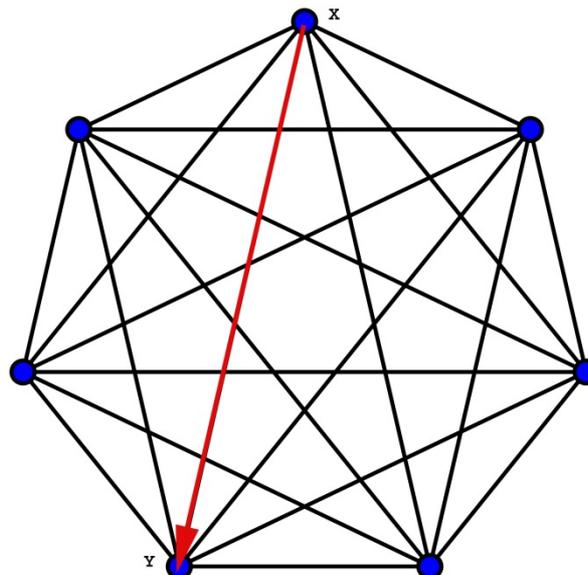
Redes bayesianas

En teoría de la probabilidad, no hay ninguna forma *a priori* de conocer qué variables influyen sobre otras. En general, las distribuciones de probabilidad conjuntas se deben conocer para poder realizar una inferencia correcta. Pero a la hora de construir modelos para el mundo real, si se tuvieran en cuenta todas las posibilidades, esas distribuciones serían muy grandes y en los casos reales no se podrían almacenar en un computador. Veamos un ejemplo.

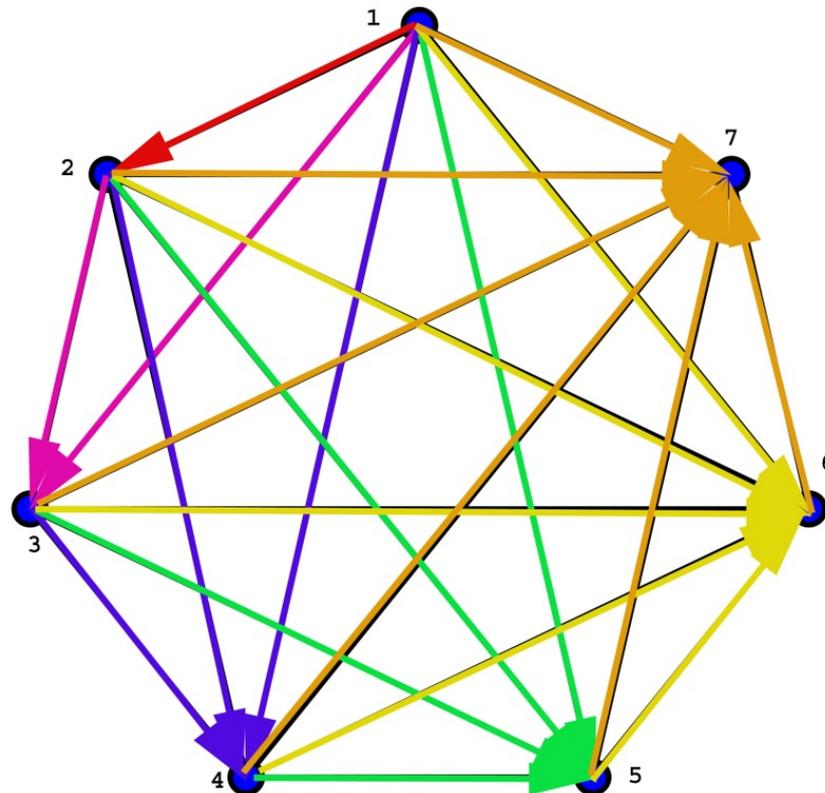
Supongamos que tenemos un dominio modelado por el conjunto de variables aleatorias $\{X_1, X_2, X_3, \dots, X_n\}$, si tenemos en cuenta todas las posibilidades de relaciones y lo reflejásemos visualmente obtendríamos algo de este estilo, todas con todas. Supongamos $n=7$ y en esta gráfica cada enlace es bidireccional.



Ahora bien, si hay un enlace que tiene que representar que X es el padre de Y, es decir, que X tiene una influencia directa sobre Y, o lo que es análogo, que existe una dependencia condicional, entonces eso lo vamos a representar con enlaces dirigidos que conectan esa pareja de nodos. Por ejemplo



Si se tuvieran en cuenta la probabilidad conjunta expresada visualmente por todos los factores posibles de las probabilidades condicionadas, deberíamos empezar por ordenar las variables independientes según su posible influencia; consideremos por ejemplo {1,2,3,4,5,6,7} empezando por la parte superior y contando hacia la izquierda, y tendríamos para esa configuración de datos un gráfico como el siguiente.



Que se interpreta que el nodo 1 es el padre de todos, el nodo 2 es el padre de 3,4,5,6,7, el nodo 3 es el padre de 4,5,6,7,... hasta llegar a que el 7 es hijo de todos.

Pero si ahora rotan los números, resultaría que el 7 pasaría a ser el padre de todos...

Y cualquier ordenación de los 7 valores, daría lugar a una nueva estructura de dependencias condicionales.

Si tienen curiosidad, el gráfico correspondiente al ejemplo {1,2,3,4,5,6,7} se puede representar matemáticamente mediante lo que ya conocemos como regla de la cadena [fórmula 11].

$$P(X_1, X_2, X_3, \dots, X_n) = \prod_{i=1}^n P(X_i | X_1, \dots, X_{i-1})$$

Por lo tanto, si tenemos n variables, iresultan n! reformulaciones distintas.

Por ejemplo, supongamos que tenemos las variables 1, 2, 3, 4 y 5, tendríamos 120 posibilidades.

$$P(1,2,3,4,5) = P(1)P(2|1)P(3|1,2)P(4|1,2,3)P(5|1,2,3,4)$$
$$\dots$$
$$P(3,2,5,1,4) = P(3) P(2|3) P(5|3,2) P(1|3,2,5) P(4|3,2,5,1)$$
$$\dots$$

Pero si tuviéramos 15 variables aleatorias tendríamos la necesidad de calcular 123076741368000 de posibilidades para cubrir todas las posibilidades. Evidentemente, esto no es viable ni mediante gráfico, ni mediante fórmula.

Ahora bien, como acabamos de comprobar, trabajar con razonamiento probabilístico es muy interesante pero no es trivial cuando el conjunto de variables y las relaciones de independencia condicional son grandes. Para ello se han inventado las redes bayesianas (RB).

Una red bayesiana (RB) permite modelar la incertidumbre de forma probabilística incluso cuando el número de variables es elevado. Proporciona una forma compacta y modular de representar la distribución conjunta de varias variables aleatorias. Estas redes constan de dos partes: La cualitativa que describe las relaciones entre las distintas variables, y la cuantitativa que describe la fuerza de dichas relaciones mediante probabilidades condicionadas.

Uno de los trabajos iniciales cuando se quiere construir un modelo bayesiano es que el constructor tiene que utilizar el sentido común y el conocimiento del mundo real para hacer decrecer la complejidad del modelo, dejando únicamente las variables que describen el modelo y definiendo las asociaciones entre las variables que producen cambios en el sistema y sobre qué variables influyen.

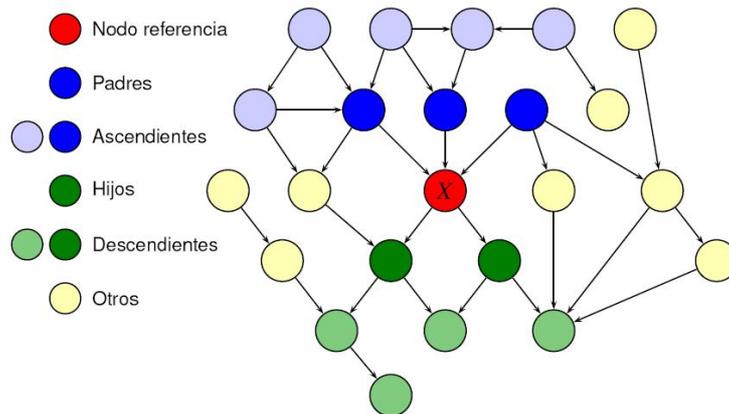
Las RB son más compactas que las distribuciones conjuntas completas, por el simple hecho de que no todas las variables están relacionadas con todas. Cada variable interactúa directamente con un grupo reducido de otras variables. Es decir, el dominio está estructurado localmente. Pero, a pesar de todo, la construcción de una RB estructurada localmente no es un problema trivial.

Teniendo en cuenta lo dicho, el procedimiento para construir una RB es el siguiente:

- Elegir un grupo de variables $\{X_1, X_2, X_3, \dots, X_n\}$ que describen un dominio. Cada variable describe un conjunto de estados que representan todas las situaciones posibles y diferentes de la variable. Este conjunto se define como «mutuamente exclusivo y exhaustivo».
- Use el conocimiento causal para definir las conectividades en el grafo.

- Utilizar el conocimiento *a priori* para especificar las distribuciones condicionales.

Las probabilidades originales del modelo se conocen como *probabilidades a priori*. Y las probabilidades calculadas después de utilizar las evidencias se conocen como *probabilidades a posteriori*.



Se entiende que para cada variable X se considera que es independiente de sus «no descendientes» dados sus padres.

Recordemos que los datos asociados a una RB son:

- Un conjunto de variables aleatorias que forman los nodos de la red. Las variables pueden ser discretas o continuas $\{X_1, X_2, X_3, \dots, X_n\}$.
- Entre ellas están las variables de interés, supongamos que son $\mathbf{X}_i = \{X_1, X_2, X_3, \dots, X_i\}$, por sencillez en los ejemplos, se va a considerar solo una de ellas $\{X\}$.
- Supongamos que las variables observadas son $\mathbf{X}_o = \{X_{i+1}, \dots, X_j\}$ y sus evidencias son $\mathbf{X}_o = \mathbf{e}_o = \{X_{i+1} = e_{i+1}, \dots, X_j = e_j\}$.
- El resto de las variables son las variables ocultas $\mathbf{Y} = \{X_{j+1}, \dots, X_n\}$.
- De esta manera $\mathbf{X} = \{X\} \cup \mathbf{X}_o \cup \mathbf{Y}$.

Hay que tener en cuenta que cada vez que se tiene una nueva evidencia, las variables de interés, las variables observadas y las ocultas pueden cambiar de categoría. Es decir, la configuración de tipos de las variables aleatorias puede cambiar.

Desde un punto de vista gráfico, una RB es un grafo dirigido acíclico en el que cada nodo contiene información probabilística, que se representa mediante:

- Un conjunto de enlaces dirigidos (arcos) que conectan parejas de nodos. Si hay un enlace que va del nodo X al Y , se dice que X es el padre de Y , es decir,

que X tiene una influencia directa sobre Y , o lo que es análogo existe una dependencia condicional).

- Cada nodo X_i tiene asociada una distribución de probabilidad condicional:

$$P(X_i | X_1 \dots X_{i-1}) = P(X_i | \text{Padres}(X_i)) \text{ [ver fórmula 10]}$$

Que mide el efecto de los padres de ese nodo.

- Eso quiere decir que una red bayesiana es una representación correcta de un dominio solo si cada nodo es condicionalmente independiente de sus predecesores en orden, dados sus padres. Para lograr esto, los padres de una variable X_i deben ser aquellos entre las variables X_1, \dots, X_{i-1} .
- Suponiendo que cada nodo de la red tenga como máximo k padres ($k \ll n$), un nodo necesitará 2^k para representar la influencia de sus padres, por lo tanto, el espacio necesario será del orden $O(n 2^k)$.

Por ejemplo:

- Con 10 variables y suponiendo 3 padres como máximo, tenemos 80 frente a 1024.
- Con 100 variables y suponiendo 5 padres como máximo, tenemos 3200 frente a aproximadamente 10^{30} .
- El problema a resolver es $P(X_i | X_0 = e)$, teniendo en cuenta la fórmula 5:

$$P(X_i | X_0 = e) = \frac{P(X_i, X_0 = e)}{P(X_0 = e)} = \frac{P(X_1, X_2, \dots, X_i, X_{i+1} = e_{i+1}, X_{i+2} = e_{i+2}, \dots, X_j = e_j)}{P(X_{i+1} = e_{i+1}, X_{i+2} = e_{i+2}, \dots, X_j = e_j)}$$

- Teniendo en cuenta la fórmula 8 y que solo se considera una variable de interés X

$$P(X_i | X_0 = e) = \alpha P(X, e) = \alpha \sum_y P(X, e, y)$$

Siendo el sumatorio en y todas las combinaciones de valores posibles de las variables ocultas Y .

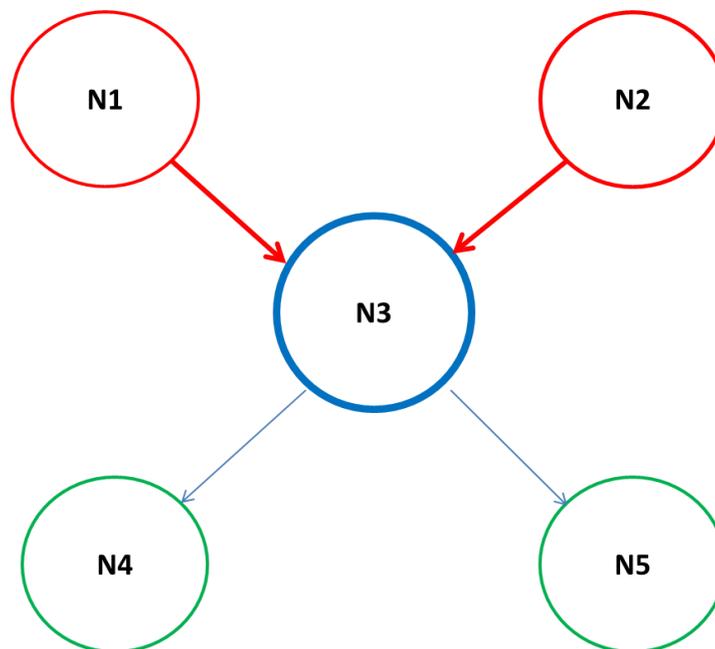
Ejemplo de red bayesiana y su funcionamiento

Supongamos la siguiente situación:

- Existe algo que vamos a denotar por Z.
- N3 es capaz de detectar que N1 o N2 se activa.
- N4 es capaz de detectar que N3 se activa y en ese caso interactúa con Z.
- N5 es capaz de detectar que N3 se activa y en ese caso interactúa con Z.
- N4 y N5 no son capaces de detectar ni N1, ni N2.
- N4 y N5 no interactúan entre sí.
- No hay más información relevante conocida o que se tenga que tener en cuenta. En este caso Z solo sirve para describir la acción de N4 o N5.

Pero no todo funciona perfectamente, por lo que son necesarias que surjan las probabilidades que están definidas *a priori*, sus frecuencias de comportamiento se han podido observar a lo largo del tiempo y, por lo tanto, en este momento son conocidas.

En este caso, podemos suponer la siguiente RB en la que tenemos un nodo N3 que tiene dos padres N1 y N2, y dos hijos N4 y N5



Con las siguientes tablas de probabilidades:

La probabilidad de ocurrencia de N1, $P(N1) = 0,001$ $P(-N1) = 0,999$.

La probabilidad de ocurrencia de N2, $P(N2) = 0,002$ $P(-N2) = 0,998$.

La probabilidad de detección de N3 viene dada por la siguiente tabla

N1	N2	P(N3 N1, N2)	P(-N3 N1, N2)
Se activa (N1)	Se activa (N2)	P(N3 N1, N2) 0,95	P(-N3 N1, N2) 0,05
Se activa (N1)	No se activa (-N2)	P(N3 N1, -N2) 0,94	P(-N3 N1, -N2) 0,06
No se activa (-N1)	Se activa (N2)	P(N3 -N1, N2) 0,29	P(-N3 -N1, N2) 0,71
No se activa (-N1)	No se activa (-N2)	P(N3 -N1, -N2) 0,001	P(-N3 -N1, -N2) 0,999

La probabilidad de actuación de N4 cuando N3 se activa, viene dada por la siguiente tabla

N3	P(N4 N3)	P(-N4 N3)
Se activa (N3) y N4 reacciona	P(N4 N3) 0,90	P(-N4 N3) 0,10
No se activa (-N3) y N4 reacciona	P(N4 -N3) 0,05	P(-N4 -N3) 0,95

La probabilidad de actuación de N5 cuando N3 se activa, viene dada por la siguiente tabla

N3	P(N5 N3)	P(-N5 N3)
Se activa (N3) y N5 reacciona	P(N5 N3) 0,70	P(-N5 N3) 0,30
No se activa (-N3) y N5 reacciona	P(N5 -N3) 0,01	P(-N5 -N3) 0,99

Observe que una tabla para una variable booleana con k padres booleanos tiene 2^k valores, y 2^k valores complementarios.

Veamos ahora algunos cálculos de probabilidades

Caso 1: Calcular la probabilidad de que N3 se active, que no hayan ocurrido ni N1, ni N2, y N4 y N5 hayan actuado llamando a X.

Teniendo en cuenta la distribución conjunta completa usada para responder cualquier pregunta sobre el dominio [fórmula 11] tendríamos

$$P(X_1, X_2, X_3, \dots, X_n) = \prod_{i=1}^n P(X_i | X_1, \dots, X_{i-1}) \text{ [fórmula 11]}$$

Y teniendo en cuenta la topología de la RB entonces

$$P(N4, N5, N3, -N1, -N2) = P(N4|N3) P(N5|N3) P(N3|-N1 \wedge -N2) P(-N1) P(-N2) = 0,90 \times 0,70 \times 0,001 \times 0,999 \times 0,998$$

Por lo tanto, $P(N4, N5, N3, -N1, -N2) = 0,00062$

Caso 2: Calcular la probabilidad de que se haya activado N1 o no se haya activado N1, si N4 y N5 han actuado llamando a X. En este caso las variables ocultas son N2 y N3. En realidad, son dos preguntas que llevaremos en paralelo.

Las preguntas son $\langle P(N1|N4 \text{ cierto y } N5 \text{ cierto}), P(\neg N1|N4 \text{ cierto y } N5 \text{ cierto}) \rangle$

- $$P(N1|N4 \text{ cierto y } N5 \text{ cierto}) = \alpha \sum_{N2} \sum_{N3} P(N1, N2, N3, N4, N5)$$

$$= \alpha \sum_{N2} \sum_{N3} P(N1)P(N2)P(N3|N1, N2)P(N4|N3)P(N5|N3) =$$

$$\alpha [P(N1)P(N2)P(N3|N1, N2)P(N4|N3)P(N5|N3) +$$

$$P(N1)P(N2)P(\neg N3|N1, N2)P(N4|\neg N3)P(N5|\neg N3) +$$

$$P(N1)P(\neg N2)P(N3|N1, \neg N2)P(N4|N3)P(N5|N3) +$$

$$P(N1)P(\neg N2)P(\neg N3|N1, \neg N2)P(N4|\neg N3)P(N5|\neg N3)] =$$

$$\alpha [(0,001 \times 0,002 \times 0,95 \times 0,90 \times 0,70) +$$

$$(0,001 \times 0,002 \times 0,05 \times 0,05 \times 0,01) +$$

$$(0,001 \times 0,998 \times 0,94 \times 0,90 \times 0,70) +$$

$$(0,001 \times 0,998 \times 0,06 \times 0,05 \times 0,01)] =$$

$$\alpha [0,000001197 + 0,0000000001 + 0,0005910156 + 0,0002994] =$$

$$\alpha \times 0,00059224254$$

- $$P(\neg N1|N4 \text{ cierto y } N5 \text{ cierto}) = \alpha \sum_{N2} \sum_{N3} P(\neg N1, N2, N3, N4, N5)$$

$$= \alpha \sum_{N2} \sum_{N3} P(\neg N1)P(N2)P(N3|\neg N1, N2)P(N4|N3)P(N5|N3) =$$

$$\alpha [P(\neg N1)P(N2)P(N3|\neg N1, N2)P(N4|N3)P(N5|N3) +$$

$$P(\neg N1)P(N2)P(\neg N3|\neg N1, N2)P(N4|\neg N3)P(N5|\neg N3) +$$

$$P(\neg N1)P(\neg N2)P(N3|\neg N1, \neg N2)P(N4|N3)P(N5|N3) +$$

$$P(\neg N1)P(\neg N2)P(\neg N3|\neg N1, \neg N2)P(N4|\neg N3)P(N5|\neg N3)] =$$

$$\alpha [(0,999 \times 0,002 \times 0,29 \times 0,90 \times 0,70) +$$

$$(0,999 \times 0,002 \times 0,71 \times 0,05 \times 0,01) +$$

$$(0,999 \times 0,998 \times 0,001 \times 0,90 \times 0,70) +$$

$$(0,999 \times 0,998 \times 0,999 \times 0,05 \times 0,01)] =$$

$$\alpha [0,0003650346 + 0,00000070929 + 0,00062811126 + 0,0004980025] =$$

$$\alpha \times 0,00149185765$$

- Resultado

$$\langle P(N1|N4 \text{ cierto y } N5 \text{ cierto}), P(\neg N1|N4 \text{ cierto y } N5 \text{ cierto}) \rangle =$$

$$\alpha \langle 0,00059224254, 0,00149185765 \rangle$$

- Falta calcular el valor de α pero interpretado como un factor de normalización que es un factor común que hace que las probabilidades sumen uno.

Entonces, en vez de calcularlo directamente podemos realizar el cálculo alternativo

$$\alpha [0,00059224254 + 0,00149185765] = 1$$

$$\alpha = \frac{1}{0,00208410019} = 479,823381236$$

Con lo cual la respuesta a las dos preguntas es

$$\begin{aligned} &\langle P(N1|N4 \text{ cierto y } N5 \text{ cierto}), P(\neg N1|N4 \text{ cierto y } N5 \text{ cierto}) \rangle = \\ &\quad \alpha \langle 0,00059224254, 0,00149185765 \rangle = \\ &\quad 479,823381236 \langle 0,00059224254, 0,00149185765 \rangle = \\ &\quad \langle 0,284171818, 0,715828182 \rangle = \\ &\quad \langle 0,284, 0,716 \rangle \end{aligned}$$

$$\langle P(N1|N4 \text{ cierto y } N5 \text{ cierto}), P(\neg N1|N4 \text{ cierto y } N5 \text{ cierto}) \rangle = \langle 0,284, 0,716 \rangle$$

El proceso que se ha seguido se denomina *inferencia exacta por enumeración*. Dicho proceso puede mejorarse eliminando cálculos repetidos, tal y como se puede ver en el ejemplo anterior.

La idea sencilla es hacer los cálculos una sola vez y salvar el resultado para usarlo más tarde. Una de las implementaciones se denomina *eliminación de variables*. Véase

https://es.wikipedia.org/wiki/Algoritmo_de_eliminaci%C3%B3n_de_variables

Ese algoritmo es sencillo y eficiente para responder a preguntas particulares, pero si se desea calcular probabilidades posteriores para todas las variables de una red, es poco eficiente. Para ello se utiliza el algoritmo denominado *árboles de uniones*. Véase Stuart Russell & Peter Norvig

Dada la intratabilidad de la inferencia exacta en grandes redes es imprescindible tratar con métodos de inferencia aproximados. Es decir, aplicar técnicas de muestreo para el cálculo de probabilidades *a posteriori*. Véase

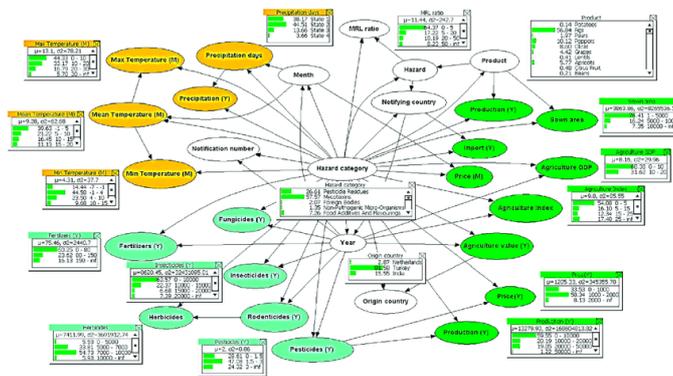
https://es.wikipedia.org/wiki/Red_bayesiana

Ejemplos de ellos son los algoritmos de muestreo:

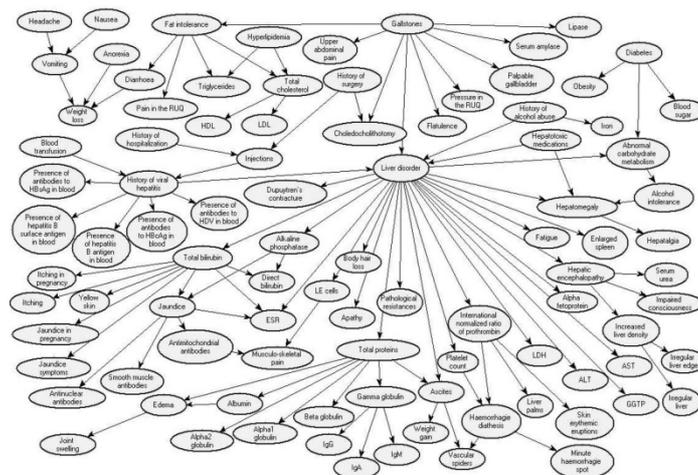
- Directo.
- Rechazo.
- Ponderación de verosimilitud.
- MonteCarlo para cadenas de Markov.

Otras opciones son las que se basan en combinar representaciones lógicas de primer orden con la teoría de la probabilidad para hacer sistemas de razonamiento en entornos con incertidumbre. Los modelos probabilistas relacionales trabajan con esas ideas que pueden acabar expresándose con una red bayesiana equivalente.

Reflexión: Es evidente que tanto la teoría de la probabilidad es una herramienta matemática de primer orden para entornos que son parcialmente observables, o no determinísticos, o que no se conocen al completo. Y que dada la complejidad de los problemas del mundo real las redes bayesianas pueden acudir en nuestra ayuda a la hora de modelarlos. Estos modelos pueden tener diversas aplicaciones, para clasificación, predicción, diagnóstico, etc. Además, pueden dar información interesante en cuanto a cómo se relacionan las variables del dominio, las cuales pueden ser interpretadas en ocasiones como relaciones de causa-efecto. Inicialmente, estos modelos eran construidos a mano, basados en un conocimiento experto. Pero ahora observen solo la topología de las siguientes redes bayesianas extraídas de Internet.



Bayesian network model to predict hazard category in fruits and vegetables: (i) RASFF factors (white nodes), (ii) economic factors (green nodes); (iii) agricultural factors (pistachio nodes); (iv) and climatic factors (orange nodes).



The example below is taken from the publication «A Bayesian Network Model for Diagnosis of Liver Disorders» – Agnieszka Onisko, M.S., Marek J. Druzdzel, Ph.D., and Hanna Wasyluk, M.D., Ph.D.- September 1999.

A la vista está que surge la necesidad y, por lo tanto, la explicación de que en los últimos años se hayan desarrollado diversas técnicas para facilitar la automatización de este tipo de redes bayesianas (grafos). A modo de ejemplo algunas herramientas que se pueden encontrar en la red son las siguientes:

- Norsys:
 - Programa Netica.
 - <http://www.norsys.com/netica.html>
- Microsoft:
 - MSBNd.
 - <https://www.microsoft.com/en-us/research/publication/msbnd-a-component-centric-toolkit-for-modeling-and-inference-with-bayesian-networks/>

Y las herramientas más sofisticadas existentes en la actualidad, son capaces de aprender a partir de datos, tanto la estructura como los parámetros asociados al modelo y también es posible combinar el conocimiento del experto a la hora de construir un modelo.

El conjunto de aplicaciones de este tipo de redes bayesianas es inmenso, lo que demuestra su versatilidad. Algunas aplicaciones de las redes bayesianas pueden encontrarse en algunas empresas de referencia:

- Microsoft:
 - Answer Wizard (Office)-
 - Diagnóstico de problemas de usuario (Aladdin)-
 - Home Health en la red de Microsoft (MSN)-
- Intel:
 - Diagnóstico de fallos de procesadores.
- HP:
 - Diagnóstico de problemas de impresora.
- Nokia:
 - Diagnóstico de redes celulares.
- Nasa:
 - Sistema de ayuda a la decisión en misiones espaciales.
- ...

O en cierto tipo de problemas, como los que aparecen descritos en

<https://data-flair.training/blogs/bayesian-network-applications/>

- Redes reguladoras de genes.
- Medicina para el diagnóstico, tratamiento o prevención.
- Bio-monitorización, para cuantificar la concentración de productos químicos.

- Clasificación de documentos, problema típico de biblioteconomía, informática y ciencias de la información.
- Recuperación de información de las bases de datos para no producir sobrecarga de información.
- Búsqueda semántica. Intentando comprender la intención del buscador y el significado contextual de los términos, para mejorar la precisión de la búsqueda.
- Procesamiento de imagen.
- Filtros antispam para detectar correos electrónicos no solicitados y no deseados.
- Códigos de corrección de errores en transmisión de señales.
- Inferencia de diferentes tipos de redes biológicas.
- ...

Razonamiento probabilista en el tiempo

En los apartados anteriores, se han desarrollado técnicas de razonamiento probabilista en el contexto de mundos estáticos, en los que cada variable aleatoria tiene un único valor fijo. Ahora se va a considerar un problema ligeramente diferente en el que se necesita tener en cuenta los aspectos dinámicos del problema.

El proceso de cambio se va a ver como una serie de fotos tomadas en ciertos instantes, cada una de las cuales describe el estado del mundo en un momento particular. El intervalo entre los cortes temporales dependerá del problema. Mientras no se diga lo contrario, se supondrán procesos discretos.

Supondremos, por simplicidad, la siguiente notación:

- X_t describirá el conjunto de variables del estado observables del instante t .
- E_t describirá el conjunto de variables evidencia observable, la observación de estas variables en el instante de tiempo t se denotará mediante e_t .
- Etiquetaremos los intervalos temporales con enteros, empezando en $t=1$.
- Las variables de estado en su evolución vendrán descritas por X_1, X_2, X_3, \dots
- Las variables de evidencia en su evolución vendrán descritas por e_1, e_2, e_3, \dots
- La notación $Z_{a:b}$ indicará el correspondiente conjunto de variables desde el instante «a» hasta el instante «b».
- El sistema evoluciona de unos estados a otros con transiciones probabilísticas.

Ejemplo: Se supone que el tiempo de cada día puede describirse mediante la variable aleatoria X_t , siendo «t» la variable temporal, que puede tener los estados o variables evidencia {calor, frío, lluvia}.

El siguiente paso es determinar las dependencias entre las variables. Podríamos seguir el procedimiento seguido previamente, poner las variables en algún orden y preguntar cuestiones sobre la independencia condicional de los predecesores, dado algún conjunto de padres. Una elección obvia es ordenar las variables en su orden temporal natural, ya que la causa usualmente precede al efecto y se prefiere añadir las variables en el orden causal. Sin embargo, chocaríamos rápidamente con un obstáculo: el conjunto de variables es no acotado, ya que incluye las variables de estado y de evidencia para cada instante temporal, lo que induciría a tener que especificar un número no acotado de tablas de probabilidad condicional, y además se podría llegar a un número no acotado de padres.

El primer problema se resuelve si se supone que los cambios de estado en el tiempo están causados por un proceso estacionario, es decir, las leyes de cambio permanecen las mismas en el tiempo. Por ello, las distribuciones condicionadas de probabilidades se definen una sola vez para un intervalo temporal dado.

El segundo problema, el de la manipulación de un número de padres potencialmente infinito, se resuelve mediante la hipótesis de Markov, que dice que el estado actual depende de un conjunto finito de estados precedentes. El más simple es el proceso de Markov de primer orden, en el que el estado actual depende solo del estado previo y de ningún otro. Lo que podemos expresar como

$$P(X_{t+1} = e_{t+1} \mid X_{1:t} = e_{1:t}) = P(X_{t+1} = e_{t+1} \mid X_t = e_t)$$

	Estado observable	Estado oculto (observaciones)
No hay acciones (sólo transiciones estocásticas)	Cadenas de Markov (Markov Chains)	Modelos Ocultos de Markov (HMMs)
Hay acciones (que producen transiciones estocásticas)	Procesos de Decisión de Markov (MDPs)	Procesos de Decisión de Markov Parcialmente Observables (POMDPs)

Vista como red bayesiana, una cadena de Markov tiene la siguiente estructura

$$\dots \rightarrow X_{t-2} \rightarrow X_{t-1} \rightarrow X_t \rightarrow X_{t+1} \rightarrow X_{t+2} \rightarrow \dots$$

- Cada nodo tiene asociada una tabla de probabilidad correspondiente a $P(X_t | X_{t-1})$.
- La misma tabla en todos los nodos.
- Excepto en el instante inicial X_1 , cuya tabla es $P(X_1)$.
- La tabla de probabilidad $P(X_t | X_{t-1})$, viene dada por una matriz de probabilidades de transición $A = (a_{ij})$:
 - Donde cada $a_{ij} = P(X_t = e_j | X_{t-1} = e_i)$.

La estructura de la red implica que, dada la inmediatamente anterior, cada variable es independiente de todas las anteriores.

[A]

	Calor en t	Frío en t	Lluvia en t
Calor en (t-1)	0,5	0,2	0,3
Frío en (t-1)	0,2	0,5	0,3
Lluvia en (t-1)	0,3	0,1	0,6

{P}

Calor en t=1	0,5
Frío en t=1	0,3
Lluvia en t=1	0,2

Con las cadenas de Markov se suelen hacer dos tipos de preguntas:

- ¿Cuál es la probabilidad de que ocurra una determinada secuencia de estados $E_t = e_1, e_2, \dots, e_t$?, es decir, $P(X_1=e_1, X_2=e_2, \dots, X_t= E_t)$.
- ¿Cuál es la probabilidad de que en el instante «t» se llegue al estado «e», es decir, $P(X_t=e)$.

Primera pregunta. $P(X_1=e_1, X_2=e_2, \dots, X_t=e_t)$

- Aplicando la regla de la cadena [fórmula 11]

$$P(X_1, X_2, X_3, \dots, X_n) = P(X_n | X_{n-1}, X_{n-2}, \dots, X_1) P(X_{n-1} | X_{n-2}, \dots, X_1) = \\ P(X_n | X_{n-1}, X_{n-2}, \dots, X_1) P(X_{n-1} | X_{n-2}, X_{n-3}, \dots, X_1) P(X_{n-2} | X_{n-3}, \dots, X_1) = \\ P(X_n | X_{n-1}, X_{n-2}, \dots, X_1) P(X_{n-1} | X_{n-2}, X_{n-3}, \dots, X_1) \dots P(X_2 | X_1) P(X_1)$$

- Por lo tanto, la expresión de la regla de la cadena es

$$P(X_1, X_2, X_3, \dots, X_n) = \prod_{i=1}^n P(X_i | X_1, \dots, X_{i-1}) \text{ [fórmula 11]}$$

- Por la propiedad de Markov

$$P(X_1, X_2, X_3, \dots, X_n) = P(X_n | X_{n-1}) P(X_{n-1} | X_{n-2}) \dots P(X_2 | X_1) P(X_1)$$

Luego $P(X_1, X_2, X_3, \dots, X_n) = \pi_{e_1} \times a_{e_1 e_2} \times a_{e_2 e_3} \times \dots \times a_{e_{t-1} e_t}$, es decir, la probabilidad de una secuencia es el producto de las correspondientes probabilidades de transitar de cada estado al siguiente.

En el ejemplo anterior:

$$P(\text{calor}, \text{calor}, \text{lluvia}, \text{frío}) = 0,5 \times 0,5 \times 0,3 \times 0,1 = 0,0075$$

Segunda pregunta, $P(X_t=e)$

- *Primera idea:* $P(X_t=e) = \sum_E P(E)$, donde tenemos un sumando por cada secuencia $E = e_1, e_2, \dots, e_t$ que acaba en el estado $e_t = e$. Total n^{t-1}
- En el ejemplo anterior, tenemos tres estados {calor, frío, lluvia} si queremos calcular la $P(X_3=\text{calor})$, es decir, que al cabo de tres avances acabemos en calor lo que tenemos que hacer es ver todas las posibilidades existentes que acaben en calor.

calor	calor	calor
calor	frío	calor
calor	lluvia	calor
frío	calor	calor
frío	frío	calor
frío	lluvia	calor
lluvia	calor	calor
lluvia	frío	calor
lluvia	lluvia	calor

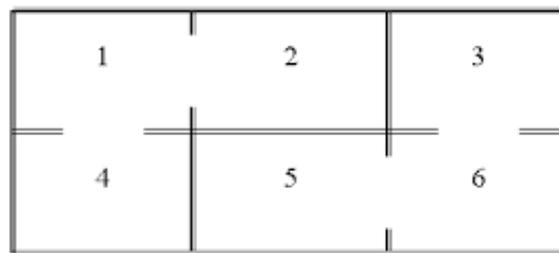
- Proceso ineficiente

- *Segunda idea:* $P(X_t=e)$: Calcular recursivamente los valores $p_t(e_i)$ para todos los estados $1 \leq i \leq n$

- $p_1(\text{calor}) = 0.5$
- $p_1(\text{frío}) = 0.3$
- $p_1(\text{lluvia}) = 0.2$
- $p_2(\text{calor}) = 0.5 \cdot p_1(\text{calor}) + 0.2 \cdot p_1(\text{frío}) + 0.3 \cdot p_1(\text{lluvia}) = 0.37$
- $p_2(\text{frío}) = 0.2 \cdot p_1(\text{calor}) + 0.5 \cdot p_1(\text{frío}) + 0.1 \cdot p_1(\text{lluvia}) = 0.27$
- $p_2(\text{lluvia}) = 0.3 \cdot p_1(\text{calor}) + 0.3 \cdot p_1(\text{frío}) + 0.6 \cdot p_1(\text{lluvia}) = 0.36$
- $p_3(\text{calor}) = 0.5 \cdot p_2(\text{calor}) + 0.2 \cdot p_2(\text{frío}) + 0.3 \cdot p_2(\text{lluvia}) = 0.347$
- ...

Ejemplo: Considérese el siguiente laberinto

<https://www.ugr.es/~bioestad/_private/cpfund10.pdf>



Supóngase que se introduce un ratón de forma aleatoria en una de las celdas de dicho laberinto. El ratón se va a trasladar de forma aleatoria de cada celda a una de las contiguas.

Notación:

- X_n , es la celda ocupada en el instante $t=n$.
- Los posibles estados son 1, 2, 3, 4, 5 y 6.
- Las probabilidades iniciales de estar en cada estado vienen dadas por el vector $P_1 = (p_1, p_2, p_3, p_4, p_5, p_6)$.
- Las probabilidades de transición teóricas serían $p_{ij} = P(X_{n+1} = j | X_n = i)$.

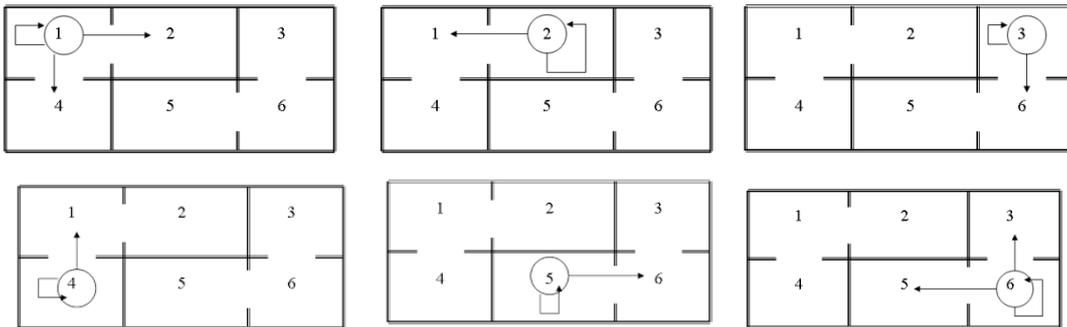
Que se interpreta como la probabilidad de pasar de la casilla «i» en el instante «n» a la casilla «j» en el instante «n+1». Lo que define la matriz de transición T para cualquier instante

$$T = \begin{pmatrix} p_{11} & p_{12} & p_{13} & p_{14} & p_{15} & p_{16} \\ p_{21} & p_{22} & p_{23} & p_{24} & p_{25} & p_{26} \\ p_{31} & p_{32} & p_{33} & p_{34} & p_{35} & p_{36} \\ p_{41} & p_{42} & p_{43} & p_{44} & p_{45} & p_{46} \\ p_{51} & p_{52} & p_{53} & p_{54} & p_{55} & p_{56} \\ p_{61} & p_{62} & p_{63} & p_{64} & p_{65} & p_{66} \end{pmatrix}$$

Verificándose que

$$\left(\begin{array}{c} \sum_{j=1}^6 p_{ij} = 1 \\ p_{ij} = P(X_t = e_j | X_{t-1} = e_i), 1 \leq i, j \leq 6 \\ p_{ij} \geq 0 \end{array} \right)$$

- Observemos ahora las posibles transiciones entre estados, que además son estacionarias en el tiempo, es decir, que no hay otras y siempre son las mismas.



Todas aquellas que son inaccesibles desde un estado, tendrán valores nulos

$$T = \begin{pmatrix} p_{11} & p_{12} & 0_{13} & p_{14} & 0_{15} & 0_{16} \\ p_{21} & p_{22} & 0_{23} & 0_{24} & 0_{25} & 0_{26} \\ 0_{31} & 0_{32} & p_{33} & 0_{34} & 0_{35} & p_{36} \\ p_{41} & 0_{42} & 0_{43} & p_{44} & 0_{45} & 0_{46} \\ 0_{51} & 0_{52} & 0_{53} & 0_{54} & p_{55} & p_{56} \\ 0_{61} & 0_{62} & p_{63} & 0_{64} & p_{65} & p_{66} \end{pmatrix}$$

Ahora pasemos del instante «0» al instante «1», es decir, $P_1 = P_0 \times T$

$$(p_1, p_2, p_3, p_4, p_5, p_6)_0 \times \begin{pmatrix} p_{11} & p_{12} & 0_{13} & p_{14} & 0_{15} & 0_{16} \\ p_{21} & p_{22} & 0_{23} & 0_{24} & 0_{25} & 0_{26} \\ 0_{31} & 0_{32} & p_{33} & 0_{34} & 0_{35} & p_{36} \\ p_{41} & 0_{42} & 0_{43} & p_{44} & 0_{45} & 0_{46} \\ 0_{51} & 0_{52} & 0_{53} & 0_{54} & p_{55} & p_{56} \\ 0_{61} & 0_{62} & p_{63} & 0_{64} & p_{65} & p_{66} \end{pmatrix}$$

- Entonces la probabilidad de estar en la celda 1 en el instante 1 vendría dada por la multiplicación del vector P_0 por la primera columna de la matriz T

$$p_1 \times p_{11} + p_2 \times p_{21} + p_3 \times p_{31} + p_4 \times p_{41} + p_5 \times p_{51} + p_6 \times p_{61}$$

que en este caso equivale a

$$p_1 \times p_{11} + p_2 \times p_{21} + p_3 \times 0_{31} + p_4 \times p_{41} + p_5 \times 0_{51} + p_6 \times 0_{61}$$

...

En general la probabilidad de estar en la celda «i» en el instante 1 vendría dada por la multiplicación de vector P_0 por la columna «i» de la matriz T

$$p_1 \times p_{1i} + p_2 \times p_{2i} + p_3 \times p_{3i} + p_4 \times p_{4i} + p_5 \times p_{5i} + p_6 \times p_{6i}$$

- Supongamos un caso concreto que venga definido por:

La probabilidad inicial de que el ratón esté en una casilla se considera que es, $P_1 = (\frac{2}{10}, \frac{2}{10}, \frac{1}{10}, \frac{5}{100}, \frac{5}{100}, \frac{4}{10})$.

Una vez dentro del laberinto el ratón, las acciones del ratón son: se queda donde está o se va a la celda contigua, ambas con la misma probabilidad.

Por lo tanto, las probabilidades p_{ij} de la matriz T vienen definidas de la misma forma

$$T = \begin{pmatrix} p_{11} & p_{12} & 0_{13} & p_{14} & 0_{15} & 0_{16} \\ p_{21} & p_{22} & 0_{23} & 0_{24} & 0_{25} & 0_{26} \\ 0_{31} & 0_{32} & p_{33} & 0_{34} & 0_{35} & p_{36} \\ p_{41} & 0_{42} & 0_{43} & p_{44} & 0_{45} & 0_{46} \\ 0_{51} & 0_{52} & 0_{53} & 0_{54} & p_{55} & p_{56} \\ 0_{61} & 0_{62} & p_{63} & 0_{64} & p_{65} & p_{66} \end{pmatrix} = \begin{pmatrix} 1/3 & 1/3 & 0_{13} & 1/3 & 0_{15} & 0_{16} \\ 1/2 & 1/2 & 0_{23} & 0_{24} & 0_{25} & 0_{26} \\ 0_{31} & 0_{32} & 1/2 & 0_{34} & 0_{35} & 1/2 \\ 1/2 & 0_{42} & 0_{43} & 1/2 & 0_{45} & 0_{46} \\ 0_{51} & 0_{52} & 0_{53} & 0_{54} & 1/2 & 1/2 \\ 0_{61} & 0_{62} & 1/3 & 0_{64} & 1/3 & 1/3 \end{pmatrix}$$

Ahora pasemos del instante «1» al instante «2», es decir, $P_2 = P_1 \times T$

$$\left(\frac{2}{10}, \frac{2}{10}, \frac{1}{10}, \frac{5}{100}, \frac{5}{100}, \frac{4}{10}\right) \times \begin{pmatrix} 1/3 & 1/3 & 0_{13} & 1/3 & 0_{15} & 0_{16} \\ 1/2 & 1/2 & 0_{23} & 0_{24} & 0_{25} & 0_{26} \\ 0_{31} & 0_{32} & 1/2 & 0_{34} & 0_{35} & 1/2 \\ 1/2 & 0_{42} & 0_{43} & 1/2 & 0_{45} & 0_{46} \\ 0_{51} & 0_{52} & 0_{53} & 0_{54} & 1/2 & 1/2 \\ 0_{61} & 0_{62} & 1/3 & 0_{64} & 1/3 & 1/3 \end{pmatrix} =$$

$$(0,191666, 0,1666, 0,18333, 0,091666, 0,158333, 0,208333)$$

Que se interpreta como que

- La probabilidad de estar en la celda 1 en el instante 2 es de 0,191666.
 - La probabilidad de estar en la celda 2 en el instante 2 es de 0,1666.
 - La probabilidad de estar en la celda 3 en el instante 2 es de 0,18333.
 - La probabilidad de estar en la celda 4 en el instante 2 es de 0,091666.
 - La probabilidad de estar en la celda 5 en el instante 2 es de 0,158333.
 - La probabilidad de estar en la celda 6 en el instante 2 es de 0,208333.
- Supongamos otro caso concreto que venga definido por:

La probabilidad inicial de que el ratón esté en una casilla se considera que es equiprobable. Por lo tanto, $P_1 = (\frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6})$.

Una vez dentro del laberinto el ratón, las acciones del ratón son: se queda donde está o se va a la celda contigua, con una matriz T definida del modo siguiente

$$T = \begin{pmatrix} p_{11} & p_{12} & 0_{13} & p_{14} & 0_{15} & 0_{16} \\ p_{21} & p_{22} & 0_{23} & 0_{24} & 0_{25} & 0_{26} \\ 0_{31} & 0_{32} & p_{33} & 0_{34} & 0_{35} & p_{36} \\ p_{41} & 0_{42} & 0_{43} & p_{44} & 0_{45} & 0_{46} \\ 0_{51} & 0_{52} & 0_{53} & 0_{54} & p_{55} & p_{56} \\ 0_{61} & 0_{62} & p_{63} & 0_{64} & p_{65} & p_{66} \end{pmatrix} = \begin{pmatrix} \frac{1}{7} & \frac{2}{7} & 0_{13} & \frac{4}{7} & 0_{15} & 0_{16} \\ \frac{1}{3} & \frac{2}{3} & 0_{23} & 0_{24} & 0_{25} & 0_{26} \\ 0_{31} & 0_{32} & \frac{3}{9} & 0_{34} & 0_{35} & \frac{6}{9} \\ \frac{1}{5} & 0_{42} & 0_{43} & \frac{4}{5} & 0_{45} & 0_{46} \\ 0_{51} & 0_{52} & 0_{53} & 0_{54} & \frac{5}{11} & \frac{6}{11} \\ 0_{61} & 0_{62} & \frac{3}{14} & 0_{64} & \frac{5}{14} & \frac{6}{14} \end{pmatrix}$$

Ahora pasemos del instante «1» al instante «2», es decir, $P_2 = P_1 \times T$

$$\left(\frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6} \right) \times \begin{pmatrix} \frac{1}{7} & \frac{2}{7} & 0_{13} & \frac{4}{7} & 0_{15} & 0_{16} \\ \frac{1}{3} & \frac{2}{3} & 0_{23} & 0_{24} & 0_{25} & 0_{26} \\ 0_{31} & 0_{32} & \frac{3}{9} & 0_{34} & 0_{35} & \frac{6}{9} \\ \frac{1}{5} & 0_{42} & 0_{43} & \frac{4}{5} & 0_{45} & 0_{46} \\ 0_{51} & 0_{52} & 0_{53} & 0_{54} & \frac{5}{11} & \frac{6}{11} \\ 0_{61} & 0_{62} & \frac{3}{14} & 0_{64} & \frac{5}{14} & \frac{6}{14} \end{pmatrix} =$$

$$\left(\frac{71}{630}, \frac{10}{63}, \frac{23}{252}, \frac{8}{35}, \frac{125}{924}, \frac{379}{1386} \right)$$

Que se interpreta como que

- La probabilidad de estar en la celda 1 en el instante 2 es de $\frac{71}{630}$.
- La probabilidad de estar en la celda 2 en el instante 2 es de $\frac{10}{63}$.
- La probabilidad de estar en la celda 3 en el instante 2 es de $\frac{23}{252}$.
- La probabilidad de estar en la celda 4 en el instante 2 es de $\frac{8}{35}$.
- La probabilidad de estar en la celda 5 en el instante 2 es de $\frac{125}{924}$.
- La probabilidad de estar en la celda 6 en el instante 2 es de $\frac{379}{1386}$.

Visto el ejemplo anterior, es más sencillo interpretar la ecuación

$$P(X_1, X_2, \dots, X_t, E_1, E_2, \dots, E_t) = P(X_1) \prod_{i=1}^t P(X_i | X_{i-1}) P(E_i | X_i)$$

Veamos el siguiente modelo de cadena de Markov diseñado para predecir el tiempo de mañana usando únicamente información sobre los días anteriores, partiendo de que tenemos un modelo con tres estados que representan los posibles estados {Soleado, Lluvioso y Nublado} con las siguientes probabilidades de transición.

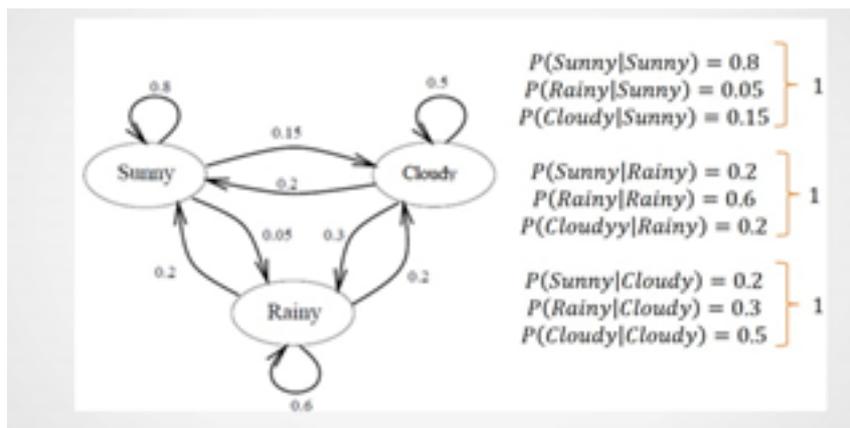
[T]	Soleado	Lluvioso	Nublado
Soleado	0,8	0,05	0,15
Lluvioso	0,2	0,6	0,2
Nublado	0,2	0,3	0,5

Por ejemplo, una secuencia de estados podría ser Soleado, Lluvioso, Nublado, Nublado, Soleado... de forma que un día puede estar en cualquiera de los tres estados.

Para calcular la probabilidad del tiempo podremos usar la propiedad de Markov, que nos dice que la distribución de probabilidad del valor futuro depende únicamente de su valor presente, y no de la historia de esa variable, es decir

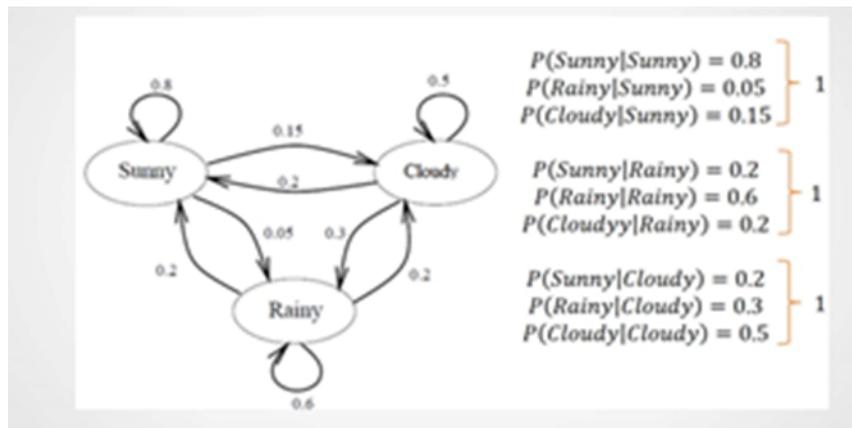
$$P(X_1, X_2, \dots, X_t) = \prod_{i=1}^t P(X_i | X_{i-1})$$

Obsérvese la siguiente representación gráfica, habitual en los diagramas asociados a las cadenas de Markov. Compare el grafo correspondiente con la tabla previa y verá que da la misma información.



Pasemos ahora a intentar responder algunas preguntas.

- **Pregunta 1.** Si sabemos que hoy el día es soleado, ¿cuál es la probabilidad de que mañana sea soleado y que al día siguiente sea lluvioso?



X en el instante 1 = X_1 = soleado.
 X en el instante 2 = X_2 = soleado.
 X en el instante 3 = X_3 = lluvioso.

Pregunta:

$$p(\text{soleado, lluvioso}|\text{soleado}) = p(\text{soleado}|\text{soleado}) \times p(\text{lluvioso}|\text{soleado, soleado})$$

$$p(X_2, X_3|X_1) = p(X_2|X_1) \times p(X_3|X_1, X_2)$$

Respuesta teniendo en cuenta la propiedad de Markov que nos dice que la distribución de probabilidad del valor futuro depende únicamente de su valor presente y no de la historia de la variable, es decir, $p(X_3|X_1, X_2) = p(X_3|X_2)$:

$$p(X_2, X_3|X_1) = p(X_2|X_1) \times p(X_3|X_1, X_2) = p(X_2|X_1) \times p(X_3|X_2)$$

que se interpreta como

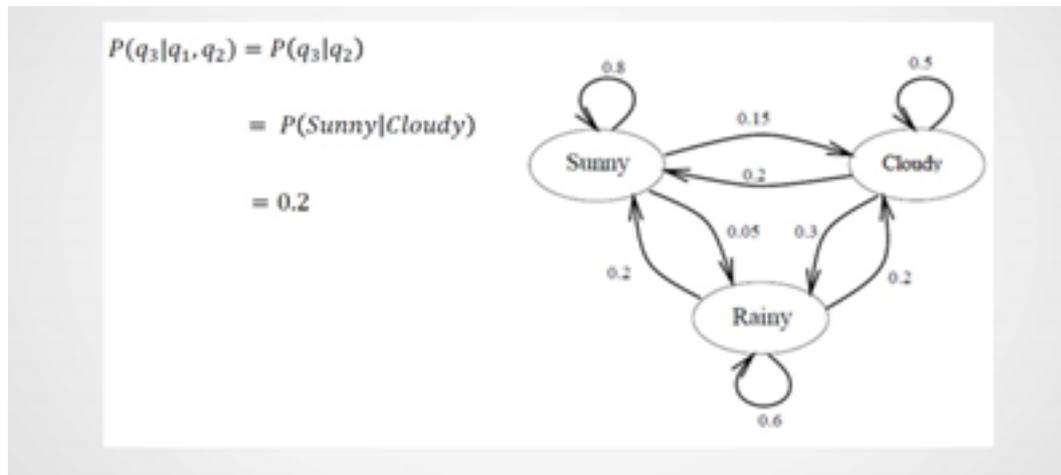
$$p(\text{soleado, lluvioso}|\text{soleado}) = p(\text{soleado}|\text{soleado}) \times p(\text{lluvioso}|\text{soleado, soleado}) =$$

$$p(\text{soleado}|\text{soleado}) \times p(\text{lluvioso}|\text{soleado})$$

viendo las tablas:

$$p(X_2, X_3|X_1) = 0,8 \times 0,05 = 0,04$$

- Pregunta 2. Si sabemos que ayer fue un día lluvioso y hoy está nublado, ¿cuál es la probabilidad de que mañana sea soleado?



X en el instante 1 = X_1 = lluvioso.
 X en el instante 2 = X_2 = nublado.
 X en el instante 3 = X_3 = soleado.

Pregunta:

$$p(\text{soleado}|\text{lluvioso, nublado}) = p(\text{soleado}|\text{nublado})$$

$$p(X_3|X_1, X_2) = p(X_3|X_2)$$

Respuesta: Teniendo en cuenta la propiedad de Markov que nos dice que la distribución de probabilidad del valor futuro depende únicamente de su valor presente y no de la historia de la variable, es decir, $p(X_3|X_1, X_2) = p(X_3|X_2)$:

$$p(X_3|X_1, X_2) = p(X_3|X_2) = 0,2$$

Partiendo de una variable de estado X , con unas variables evidencia e . Si tenemos una matriz de transición T , partiendo de una probabilidad inicial P_1 , se va a verificar que

$$P(X_t | X_{1:t-1}) = P(X_t | X_{t-1}) = p_{t-1} T = p_{t-2} T \times T = p_{t-3} T \times T \times T = p_1 T \times T \times T \times \dots \times T = p_1 \prod_{i=0}^{t-1} T^i$$

En el caso de que para ir de $t = 1$, a $t=T$, haya «n pasos» entonces

$$P(X_t | X_{1:t-1}) = p_1 T^n$$

La pregunta inmediata viene cuando el exponente «n» de la potencia T^n es muy grande, y además el número de evidencias de la variable de estado es elevado.

Ejemplo de probabilidad de transición en n pasos

Ejemplo: Probabilidad de transición en n pasos

- Los consumidores de cerveza de una localidad consumen las marcas A, B y C. En marzo del 2006, se hizo una encuesta en la que se entrevistó a los 8450 consumidores acerca de cuál sería la marca de cerveza que consumirían el próximo mes y los resultados fueron:

	Marca A	Marca B	Marca C	Totales:
Marca A	507	845	338	1690
Marca B	676	2028	676	3380
Marca C	845	845	1690	3380
Totales:	2028	3718	2704	8450

Pregunta: Para el mes de julio, ¿cuál será el porcentaje de participación de mercado para cada una de las 3 marcas?

Ejemplo: Probabilidad de transición en n pasos

- Para encontrar la matriz de transición, dividimos cada uno de los resultados de la encuesta entre el total de consumidores de la marca.

$$P = \begin{bmatrix} \frac{507}{1690} & \frac{845}{1690} & \frac{338}{1690} \\ \frac{676}{3380} & \frac{2028}{3380} & \frac{676}{3380} \\ \frac{845}{3380} & \frac{845}{3380} & \frac{1690}{3380} \end{bmatrix} = \begin{bmatrix} 0.3 & 0.5 & 0.2 \\ 0.2 & 0.6 & 0.2 \\ 0.25 & 0.25 & 0.5 \end{bmatrix}$$

Ejemplo: Probabilidad de transición en n pasos

- Para calcular la probabilidad al cabo de 4 etapas (de marzo a julio), elevamos la matriz de transición a la cuarta potencia:

$$P^2 = \begin{bmatrix} 0.3 & 0.5 & 0.2 \\ 0.2 & 0.6 & 0.2 \\ 0.25 & 0.25 & 0.5 \end{bmatrix} \begin{bmatrix} 0.3 & 0.5 & 0.2 \\ 0.2 & 0.6 & 0.2 \\ 0.25 & 0.25 & 0.5 \end{bmatrix} = \begin{bmatrix} 0.24 & 0.5 & 0.26 \\ 0.23 & 0.51 & 0.26 \\ 0.25 & 0.4 & 0.35 \end{bmatrix}$$

$$P^4 = P^2 \cdot P^2 = \begin{bmatrix} 0.24 & 0.5 & 0.26 \\ 0.23 & 0.51 & 0.26 \\ 0.25 & 0.4 & 0.35 \end{bmatrix} \begin{bmatrix} 0.24 & 0.5 & 0.26 \\ 0.23 & 0.51 & 0.26 \\ 0.25 & 0.4 & 0.35 \end{bmatrix} = \begin{bmatrix} 0.2376 & 0.479 & 0.2834 \\ 0.2375 & 0.4791 & 0.2834 \\ 0.2395 & 0.469 & 0.2915 \end{bmatrix}$$

Ejemplo: Probabilidad de transición en n pasos

- Nos preguntan el porcentaje de participación en el mercado de cada marca. Primero calculamos el vector de probabilidad inicial, formado por el porcentaje de personas que toma las marcas A, B y C (la suma de cada columna) en marzo:

$$\Pi(0) = \left[\frac{2028}{8450} \quad \frac{3718}{8450} \quad \frac{2704}{8450} \right] = [0.24 \quad 0.464 \quad 0.296]$$

Ejemplo: Probabilidad de transición en n pasos

- Para encontrar la probabilidad en julio, multiplicamos el vector de estado inicial por

$$\Pi(4) = [0.24 \quad 0.464 \quad 0.296] \begin{bmatrix} 0.2376 & 0.479 & 0.2834 \\ 0.2375 & 0.4791 & 0.2834 \\ 0.2395 & 0.469 & 0.2915 \end{bmatrix}$$

$$\Pi(4) = [0.237 \quad 0.476 \quad 0.285]$$

- Por lo tanto, la distribución del mercado en julio será para la marca A=23.7%, para la marca B=47.6% y para la marca C=28.5% de participación de mercado.

Condición de Equilibrio

- Para una Cadena de Markov, el límite, cuando el número de etapa n tiende al infinito, existe y es independiente de i
- Representado matemáticamente:

$$\lim_{n \rightarrow \infty} P_{ij}^{(n)} = \pi_j$$

- Donde se satisface que los valores del vector de equilibrio son positivos y la suma de las probabilidades del vector son iguales a 1.

Condición de Equilibrio

- El estado de equilibrio para una cadena con tres estados y con probabilidades en estado estable o de equilibrio, tendríamos que:

$$[\pi_1 \quad \pi_2 \quad \pi_3] \cdot \begin{bmatrix} P_{11} & P_{12} & P_{13} \\ P_{21} & P_{22} & P_{23} \\ P_{31} & P_{32} & P_{33} \end{bmatrix} = [\pi_1 \quad \pi_2 \quad \pi_3]$$

Condición de Equilibrio

- Resolviendo simultáneamente este sistema para podemos encontrar la distribución de probabilidades en estado estable.

$$\pi_1 = \pi_1 p_{11} + \pi_2 p_{12} + \pi_3 p_{13}$$

$$\pi_2 = \pi_1 p_{21} + \pi_2 p_{22} + \pi_3 p_{23}$$

$$\pi_3 = \pi_1 p_{31} + \pi_2 p_{32} + \pi_3 p_{33}$$

y:

$$\pi_1 + \pi_2 + \pi_3 = 1$$

Ejemplo para Condición de Equilibrio

- Tenemos la siguiente matriz de transición:

$$P = \begin{bmatrix} 0 & 0.5 & 0.5 \\ 0.75 & 0 & 0.25 \\ 1 & 0 & 0 \end{bmatrix}$$

Para encontrar el estado estable, planteamos su condición:

$$[\pi_1 \quad \pi_2 \quad \pi_3] \cdot \begin{bmatrix} 0 & 0.5 & 0.5 \\ 0.75 & 0 & 0.25 \\ 1 & 0 & 0 \end{bmatrix} = [\pi_1 \quad \pi_2 \quad \pi_3]$$

Ejemplo de Condición de Equilibrio

- Multiplicamos la matriz de estado estable por la matriz de transición y nos quedan las siguientes ecuaciones:

$$\pi_1(0) + \pi_2(0.75) + \pi_3(1) = \pi_1$$

$$\pi_1(0.5) + \pi_2(0) + \pi_3(0) = \pi_2$$

$$\pi_1(0.5) + \pi_2(0.25) + \pi_3(0) = \pi_3$$

- A las cuales agregamos la restricción de la suma de probabilidades:

$$\pi_1 + \pi_2 + \pi_3 = 1$$

Ejemplo de Condición de Equilibrio

- Resolviendo el sistema de ecuaciones, obtenemos la distribución de probabilidad en el estado de equilibrio:

$$4\pi_1 - 3\pi_2 - 4\pi_3 = 0$$

$$\pi_1 - 2\pi_2 = 0$$

$$2\pi_1 + \pi_2 - 4\pi_3 = 0$$

y:

$$\pi_1 + \pi_2 + \pi_3 = 1$$



$$\pi_1 = 0.4705$$

$$\pi_2 = 0.2352$$

$$\pi_3 = 0.2941$$

Lo cual significa que a largo plazo existe

- un 47.05% de probabilidad que se presente el estado 1,
- un 23.52% de probabilidad que se presente el estado 2 y
- un 29.41% de probabilidad que se presente el estado 3.

Modelos ocultos de Markov. Véase

<https://moodle2018-19.ua.es/moodle/pluginfile.php/7304/mod_page/content/21/clase11.pdf>

Un modelo oculto de Markov es un modelo estocástico donde los estados del modelo están ocultos. Cada estado puede emitir una salida observable. El objetivo es determinar los parámetros desconocidos u ocultos a partir de los parámetros observables. Obsérvese la siguiente imagen:

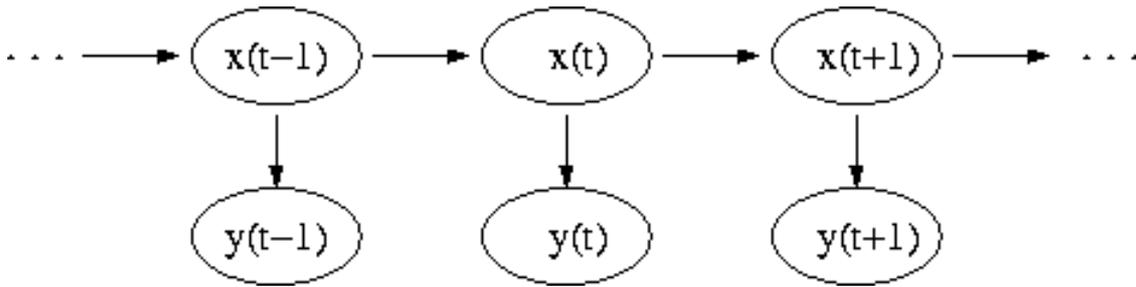
La variable aleatoria $x(t)$ es el valor de la variable oculta en el instante «t». Dicho valor solo depende de la variable oculta $x(t-1)$ por la propiedad de Markov. Es decir, $P(X_{t+1} = e_{t+1} \mid X_{1:t} = e_{1:t}) = P(X_{t+1} = e_{t+1} \mid X_t = e_t)$.

La variable aleatoria $y(t)$ es el valor de la variable observada en el mismo instante «t», que se consideran independientes entre sí. Es decir, $P(e_t \mid X_{1:t} = e_{1:t}) = P(e_t \mid e_t = X_t)$. Las flechas indican dependencias condicionales. Dicho valor solo depende de la variable oculta $x(t)$.

En un modelo oculto de Markov el estado es parcialmente observable (no se conoce, pero existen observaciones y no existen acciones (no hay toma de decisiones)).

	Estado observable	Estado oculto (observaciones)
No hay acciones (sólo transiciones estocásticas)	Cadenas de Markov (Markov Chains)	Modelos Ocultos de Markov (HMMs)
Hay acciones (que producen transiciones estocásticas)	Procesos de Decisión de Markov (MDPs)	Procesos de Decisión de Markov Parcialmente Observables (POMDPs)

Vista como red bayesiana, un modelo oculto de Markov tiene la siguiente estructura:



- Cada nodo X_t tiene asociada una tabla de probabilidad correspondiente a $P(X_t|X_{t-1})$. La misma tabla en todos los nodos.
- Excepto en el instante inicial X_1 , cuya tabla es $P(X_1)$.
- Cada nodo E_t tiene asociada una tabla de probabilidad correspondiente a $P(E_t|X_t)$. La misma tabla en todos los nodos.
- En el caso de la cadena de Markov de los estados (modelo de transición):
 - Para cada $t > 1$, una variable X_t con posibles estados (e_1, e_2, \dots, e_n).
 - La tabla de probabilidad $P(X_t|X_{t-1})$, dada por una matriz $[A] = (a_{ij})$ donde $a_{ij} = P(X_t = e_j | X_{t-1} = e_i)$ que da la probabilidad de pasar de un estado al siguiente.
 - Vector $\pi = (\pi_i)$, donde $\pi_i = P(X_i = e_i)$ (probabilidades *a priori* para cada estado)
- En el caso de la cadena de Markov de las percepciones (modelo sensor):
 - Para cada $t > 1$, una variable E_t con posibles estados (v_1, v_2, \dots, v_m).
 - La tabla de probabilidad $P(E_t|X_{t-1})$, dada por una matriz $[B] = b_i(v_j)$ donde $b_i(v_j) = P(E_t = v_j | X_t = e_i)$ que da la probabilidad de observar v_i cuando el estado es e_i .

La probabilidad de observar la secuencia $X = [x(0), x(1), x(2), \dots, x(L-1)]$ de longitud «L» está dada por

$$P(X) = \sum_X P(Y|X)P(X)$$

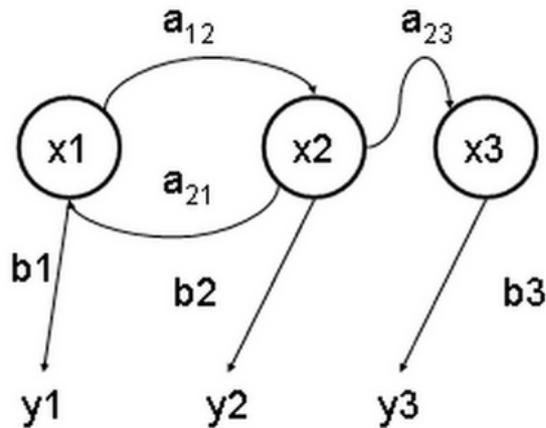
La probabilidad de observar la secuencia $Y = [y(0), y(1), y(2), \dots, y(L-1)]$ de longitud «L» está dada por

$$P(Y) = \sum_X P(Y|X)P(X)$$

Donde el sumatorio se extiende sobre todas las secuencias de nodos ocultos $X=[x(0), x(1), \dots, x(L-1)]$.

Es evidente que el cálculo por la fuerza bruta de $P(Y)$ es impracticable para la mayoría de los problemas reales, dado que el número de secuencias de nodos ocultos será extremadamente alto en tal caso.

El ejemplo de transición de estados en un modelo oculto de Markov es el siguiente:



Siendo:

- x los estados ocultos.
- y las salidas observables.
- a las probabilidades de transición.
- b las probabilidades de salida.

La notación habitual es:

- $Q =$ conjunto de estados $\{1, 2, 3, \dots, N\}$.
- $V =$ conjunto de posibles valores $\{v_1, v_2, \dots, v_m\}$ observables de cada estado.
- $\pi = \{\pi_i\}$, donde π_i es la probabilidad de que el primer estado sea Q_i .
- La matriz de probabilidades $[A] = \{a_{ij}\}$ de transiciones entre estados, es decir, $a_{ij} = P(q_t = j | q_{t-1} = i)$ es la probabilidad de estar en el estado «j» en el instante «t», si en el instante «t-1» se estaba en el estado «i».
- La matriz de probabilidades $[B] = \{b_j(v_k)\}$ de las observaciones, es decir, $b_j(v_k) = P(o_t = v_k | q_t = j)$ es la probabilidad de observar v_k cuando se está en el estado «j» en el instante «t».
- La secuencia de observables se denota como un conjunto $O = \{o_1, o_2, \dots, o_T\}$.

Ejemplo notacional: véase

<https://es.wikipedia.org/wiki/Modelo_oculto_de_M%C3%A1rkov>

Imagínese que tiene un amigo que vive lejos y con quien habla a diario por teléfono acerca de lo que hizo durante el día. A su amigo le interesan tres actividades: caminar por la plaza, salir de compras y limpiar su departamento. Lo que su amigo hace depende exclusivamente del estado del tiempo en ese día. Usted no tiene información clara acerca del estado del tiempo donde su amigo vive, pero conoce tendencias generales. Basándose en lo que su amigo le dice que hizo en el día, usted

intenta adivinar el estado del tiempo. Supóngase que el estado del tiempo se comporta como una cadena de Markov discreta.

- Q = conjunto de estados ocultos {lluvioso, soleado} $N = 2$.
- V = conjunto de posibles valores observables {caminar, comprar, limpiar} observables de cada estado, $m = 3$.
- $\pi = \{\pi_i\}$, supongamos {lluvioso = 0,571, soleado = 0,429}.
- La matriz de probabilidades de transición $[A] = \{a_{ij}\}$.

	Lluvioso en (t)	Soleado en (t)
Lluvioso en (t-1)	0,7	0,3
Soleado en (t-1)	0,4	0,6

- La matriz de probabilidades $[B] = \{b_j(v_k)\}$ de las observaciones

	Caminar en (t)	Comprar en (t)	Limpiar en (t)
Lluvioso en (t)	0,1	0,4	0,5
Soleado en (t)	0,6	0,3	0,1

- La secuencia de observables se denota como un conjunto $O = \{o_1, o_2, \dots, o_T\}$

$$P(X_{t+1} | X_{1:t}) = P(X_{t+1} = e_{t+1} | X_t = e_t) \text{ (probabilidad de transición)}$$

Por lo tanto, en un proceso de primer orden, las leyes que describen cómo evoluciona el estado en el tiempo están contenidas íntegramente dentro de la distribución condicional $P(X_{t+1} | X_t)$ que va a recibir el nombre de *modelo de transición*.

Además de restringir a los padres de las variables de estado X_{t+1} , también se restringen los padres de las variables evidencia E_{t+1} . Es típico suponer que en el instante t solo dependen del estado actual $P(E_{t+1} | X_{1:t}, E_{1:t}) = P(E_t | X_t)$ que va a recibir el nombre de *modelo sensor* o *modelo de observación*.

A todo lo anterior hay que añadir que se debe partir de unas probabilidades *a priori* $P(X_1)$ de los estados en el instante 1.

$$P(X_{t+1} | e_{1:t+1}) = P(X_{t+1} | e_{1:t}, e_{t+1}) = \alpha P(e_{t+1} | X_{t+1}, e_{1:t}) P(X_{t+1} | e_{1:t})$$

Por Markov

$$P(X_{t+1} | e_{1:t+1}) = \alpha P(e_{t+1} | X_{t+1}) P(X_{t+1} | e_{1:t})$$

Y teniendo en cuenta que

$$P(X_{t+1} | e_{1:t}) = \sum_{X_t} P(X_{t+1}, X_t | e_{1:t}) = \text{por la regla de bayes}$$

$$\sum_{X_t} P(X_{t+1} | X_t, e_{1:t}) * P(X_t | e_{1:t})$$

Entonces por Markov $P(X_{t+1} | X_t, e_{1:t}) = P(X_{t+1} | X_t)$

$$P(X_{t+1} | e_{1:t+1}) = \alpha P(e_{t+1} | X_{t+1}) P(X_{t+1} | e_{1:t}) =$$

$$\alpha P(e_{t+1} | X_{t+1}) \sum_{X_t} P(X_{t+1} | X_t) * P(X_t | e_{1:t})$$

A los términos:

- $P(X_{t+1} | X_t)$ se les denomina *modelo de transición*.
- $P(e_{t+1} | X_{t+1})$ se les denomina *modelo sensorial*.

Obsérvese que se ha obtenido una ecuación recursiva

$$P(X_{t+1} | e_{1:t+1}) = \alpha P(e_{t+1} | X_{t+1}) \sum_{X_t} P(X_{t+1} | X_t) * P(X_t | e_{1:t})$$

Existen tres tipos de problemas típicos asociados a los modelos ocultos de Markov.

- Dados los parámetros del modelo, compútese la probabilidad de una secuencia de salida en particular. Este problema se resuelve con el algoritmo de avance-retroceso. Véase
<https://es.wikipedia.org/wiki/Algoritmo_de_avance-retroceso>
- Dados los parámetros del modelo, encuéntrese la secuencia más probable de estados ocultos que puedan haber generado una secuencia de salida dada. Este problema se resuelve con el algoritmo de Viterbi. Véase
<https://es.wikipedia.org/wiki/Algoritmo_de_Viterbi>
- Dada una secuencia de salida o un conjunto de tales secuencias, encuéntrese el conjunto de estados de transición y probabilidades de salida más probables. Este problema se resuelve con el algoritmo de Baum-Welch.
<https://es.wikipedia.org/wiki/Algoritmo_de_Baum-Welch>

Pregunta: Calcular la probabilidad $P(q_1, q_2, \dots, q_t | o_1, o_2, \dots, o_t)$ que nos relacione las secuencias desconocidas con las observadas.

Teniendo en cuenta Valles, se puede calcular:

- La probabilidad para un q_i particular

$$P(q_i | o_i) = \frac{P(o_i | q_i)P(q_i)}{P(o_i)}$$

- La probabilidad para una secuencia de días de tamaño t

$$P(q_1, q_2, \dots, q_t, o_1, o_2, \dots, o_t) = \frac{P(o_1, \dots, o_t | q_1, \dots, q_t)P(q_1, \dots, q_t)}{P(o_1, \dots, o_t)}$$

Y sabemos que:

$$P(o_1, \dots, o_t | q_1, \dots, q_t) = \prod_{i=1}^t P(o_i | q_i)$$

$$P(q_1, \dots, q_t) = \prod_{i=1}^t P(q_i | q_{i-1})$$

Por lo tanto, el modelo oculto de Markov viene definido por

$$P(q_1, q_2, \dots, q_t, o_1, o_2, \dots, o_t) \approx \prod_{i=1}^t P(o_i | q_i) \prod_{i=1}^t P(q_i | q_{i-1})$$

Siendo

- $P(q_i | q_{i-1})$ las probabilidades de transición entre estados, expresadas matricialmente, representada por T

$$T = \begin{pmatrix} p_{11} & p_{12} & \dots & p_{1j} & \dots & p_{1n} \\ p_{21} & p_{22} & \dots & \dots & \dots & p_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ p_{i1} & \dots & \dots & p_{ij} & \dots & p_{in} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ p_{n1} & p_{n2} & \dots & p_{nj} & \dots & p_{nn} \end{pmatrix}$$

Verificándose que

$$\left(\begin{array}{c} \sum_{j=1}^n p_{ij} = 1 \\ p_{ij} = P(q_t = e_j | q_{t-1} = e_i), 1 \leq i, j \leq n \\ p_{ij} \geq 0 \end{array} \right)$$

- $P(o_i | x_i)$ las probabilidades de emisión-observación. Un estado genera una observación de salida, se expresa matricialmente.

Sean:

- $\{v_1, v_2, \dots, v_w\}$ el conjunto de posibles observaciones para cada estado.
- $b_i(v_k) = p(o_t = v_k | x_t = e_i), 1 \leq k \leq w$.

$$B = \begin{pmatrix} b_1(v_1) & \dots & b_i(v_1) & \dots & \dots & b_1(v_w) \\ b_2(v_1) & \dots & \dots & \dots & \dots & b_2(v_w) \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & b_i(v_j) & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ b_n(v_w) & \dots & b_n(v_w) & \dots & \dots & b_n(v_w) \end{pmatrix}$$

- $\pi(q_i = e_i)$ las probabilidades iniciales de los estados $\sum_{i=1}^n \pi_i = 1$.

Ejemplo: Imaginemos que estamos encerrados en una habitación durante días, no hay ventanas y uno se pregunta sobre el tiempo que hace fuera. La única evidencia que se tiene sobre el tiempo es que la persona que entra y sale para traer comida lleve o no un paraguas.

- ¿Qué está oculto?
 - Respuesta: El tiempo que puede estar soleado, lluvioso o nublado. Esos son los posibles estados ocultos.
- ¿Qué puedes observar?
 - Respuesta: El paraguas, que puede llevarse, o puede no llevarse.

Supongamos que pasan «t» días. Por lo tanto, se tiene una secuencia de observaciones $\{o_1, o_2, o_3, \dots, o_t\}$, donde cada o_i puede ser {paraguas, no paraguas}.

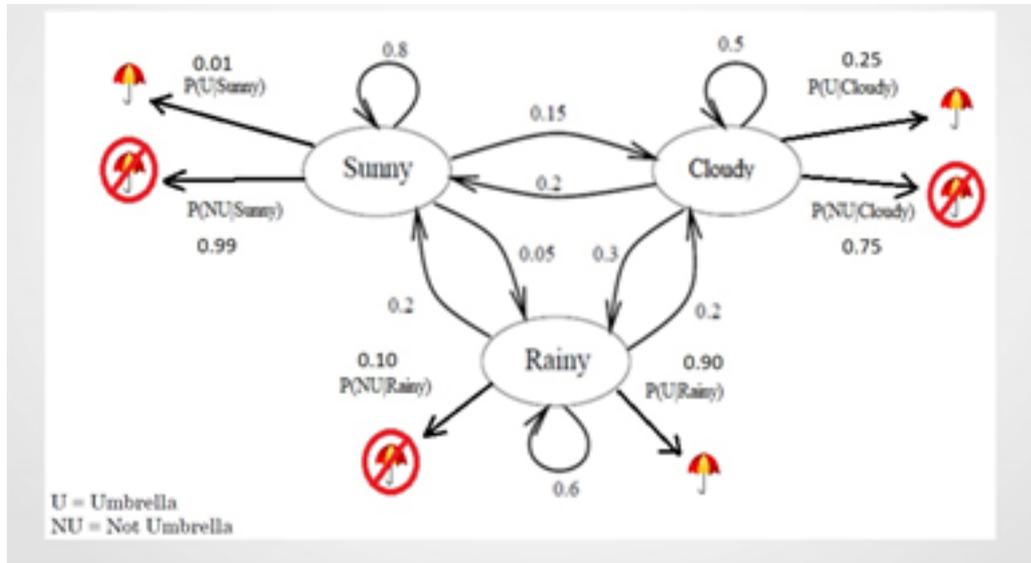
Cada observación viene de un estado desconocido. Por tanto, tenemos también una secuencia desconocida $\{q_1, q_2, \dots, q_t\}$, donde Q_i puede ser {soleado, lluvioso, nublado}

[T]	Soleado en (t)	Lluvioso en (t)	Nublado en (t)
Soleado en (t-1)	0,8	0,05	0,15
Lluvioso en (t-1)	0,2	0,6	0,2
Nublado en (t-1)	0,2	0,3	0,5

[B]	Paraguas en t	No paraguas en t
Soleado en t	0,01	0,99
Lluvioso en t	0,90	0,10
Nublado en t	0,25	0,75

P_1	Probabilidad inicial en t=1
Soleado	0
Lluvioso	0
Nublado	1

Obsérvese la siguiente representación gráfica, habitual en los diagramas asociados a los modelos ocultos de Markov. Compare los grafos correspondientes con las tablas previas y verá que dan la misma información.



- Pregunta: ¿Cuál es la secuencia óptima de estados que explica? La siguiente secuencia de observaciones
(e_1 =nublado, e_2 =nublado, e_3 =soleado, e_4 =nublado, e_5 =soleado)

Existe un algoritmo denominado de *Viterbi* que permite hallar la secuencia más probable de estados ocultos, el denominado *camino de Viterbi*, que produce una secuencia observada de sucesos.

El procedimiento que sigue es definir una variable $\delta_t(i)$ como la probabilidad del mejor camino hasta el estado «i» habiendo ocurrido las «t» primeras observaciones. Esa función se debe calcular para todos los estados e instantes de tiempo.

$$\delta_t(i) = \max_{q_1, q_2, \dots, q_{t-1}} P(q_1, q_2, \dots, q_t = i, o_1, o_2, \dots, o_t | \mu)$$

$$\delta_{t+1}(j) = [\max_{1 \leq i \leq 3} (\delta_t(i) * T_{ij})] * b_j(o_{t+1})$$

Puesto que el objetivo es obtener las secuencias de estados más probables, será necesario almacenar el argumento que hace máxima la ecuación en cada instante de tiempo «t» para cada estado «j», por lo que se utilizará la variable $\varphi_t(j)$

Cálculo

- Observaciones
(o_1 =no_paraguas, o_2 =no_paraguas, o_3 =paraguas, o_4 =no_paraguas, o_5 =paraguas)

- Cálculo para o_1 = no_paraguas

- $\delta_1(\text{soleado}) = p_1 \times b_1(o_1) = 0 \times 0,99 = 0.$
- $\delta_1(\text{lluvioso}) = p_1 \times b_2(o_1) = 0 \times 0,10 = 0.$
- $\delta_1(\text{nublado}) = p_1 \times b_3(o_1) = 1 \times 0,75 = 0,75.$

- Cálculo para o_2 = no_paraguas

- $\delta_2(\text{soleado}) = [\max_{1 \leq i \leq 3} (\delta_1(i) * T_{i1})] * b_1(o_2) =$
 $\max(0*0,8, 0*0,2, 0,75*0,2) * 0,99 = 0,1485$
- $\varphi_2(\text{soleado}) = \operatorname{argmax}_{1 \leq i \leq 3} (\delta_1(i) * T_{i1}) = 3$
- $\delta_2(\text{lluvioso}) = [\max_{1 \leq i \leq 3} (\delta_1(i) * T_{i2})] * b_2(o_2) =$
 $\max(0*0,05, 0*0,6, 0,75*0,3) * 0,1 = 0,0225$
- $\varphi_2(\text{lluvioso}) = \operatorname{argmax}_{1 \leq i \leq 3} (\delta_1(i) * T_{i2}) = 3$
- $\delta_2(\text{nublado}) = [\max_{1 \leq i \leq 3} (\delta_1(i) * T_{i3})] * b_2(o_2) =$
 $\max(0*0,15, 0*0,2, 0,75*0,5) * 0,75 = 0,28125$
- $\varphi_2(\text{nublado}) = \operatorname{argmax}_{1 \leq i \leq 3} (\delta_1(i) * T_{i3}) = 3$

- ...

- Cálculo para o_5 = paraguas

- ...

- Resumiendo, los resultados

$$\delta_t(i)$$

	No_paraguas	No_paraguas	Paraguas	No_paraguas	Paraguas
	t = 1	t = 2	t = 3	t = 4	t = 5
Soleado	0	0,1485	0,001188	0,015035625	0,000120285
Lluvioso	0	0,0225	0,0759375	0,00455625	0,00355957
Nublado	0,75	0,28125	0,03515625	0,013183594	0,001647949

$$\varphi_t(j)$$

	No_paraguas	No_paraguas	Paraguas	No_paraguas	Paraguas
	t = 1	t = 2	t = 3	t = 4	t = 5
Soleado		3	1	2	1
Lluvioso		3	3	2	3
Nublado		3	3	3	3

- $q_5^* = \operatorname{argmax}_{1 \leq i \leq 3} (\delta_5(i)) = 2 = \text{Lluvioso}$
- **Reconstrucción de la secuencia más probable**
 - $q_4^* = \varphi_5(q_5^*) = 3 = \text{Nublado}$.
 - $q_3^* = \varphi_4(q_4^*) = 3 = \text{Nublado}$.
 - $q_2^* = \varphi_3(q_3^*) = 3 = \text{Nublado}$.
 - $q_1^* = \varphi_2(q_2^*) = 3 = \text{Nublado}$.
- **Por lo tanto, la secuencia más probable es:**
 - Nublado, Nublado, Nublado, Nublado, Lluvioso.

Para saber más

- **Filtros de Kalman**

Hasta el momento se ha supuesto que los sistemas analizados eran discretos, pero qué pasa cuando el sistema es continuo lineal o no lineal.

Para ello se desarrolló lo que se conoce como *filtro de Kalman*, desarrollado por Rudolph E. Kalman en 1960, que permite identificar el estado oculto (no medible) de un sistema dinámico lineal. El filtro de Kalman proporciona el estimador con menor error cuadrático medio, combinando la evolución de la variable con observaciones recogidas. Es, además, un proceso recursivo, por lo que no es necesario almacenar la información del estado en todos los instantes. Véase

<https://es.wikipedia.org/wiki/Filtro_de_Kalman#:~:text=El%20filtro%20de%20Kalman%20es,sometido%20a%20ruido%20blanco%20aditivo>

- El filtro de Kalman puede ser extendido para sistemas no lineales. Este filtro se denomina *filtro de Kalman-Schmidt*

<https://es.wikipedia.org/wiki/Filtro_de_Kalman_Extendido>

- **Redes bayesianas dinámicas**

Una red bayesiana dinámica es un tipo particular de red bayesiana que permite modelar estructuras de datos secuenciales o datos temporales. Son una generalización de los modelos ocultos de Markov y de los filtros de Kalman. Véase

<[https://en.wikipedia.org/wiki/Dynamic_Bayesian_network#:~:text=A%20Dynamic%20Bayesian%20Network%20\(DBN,other%20over%20adjacent%20time%20steps](https://en.wikipedia.org/wiki/Dynamic_Bayesian_network#:~:text=A%20Dynamic%20Bayesian%20Network%20(DBN,other%20over%20adjacent%20time%20steps)>

Toma de decisiones. Teoría de la utilidad

Cuando no existe incertidumbre, la toma de decisiones recibe el nombre de *planificación*, pero qué pasa cuando el entorno es parcialmente observable, o no determinístico, o no se conoce al completo. En este tipo de toma de decisiones, las posibles opciones de solución tienen cierta probabilidad conocida de generar un resultado y su utilidad implícita es la que orientará la preferencia. De esta manera, las preferencias expresadas como utilidades se combinan con probabilidades y generan lo que se conoce como *teoría de la decisión*. En este contexto se denomina *estado de creencia de un sistema* a una representación de las probabilidades de todos los posibles estados actuales del mundo en el que está inmerso el sistema.

La idea fundamental de la teoría de la decisión es que un sistema toma decisiones adecuadas si y solo si elige la acción que le produce la utilidad esperada más alta, en promedio, sobre todas las posibles consecuencias de la acción. Lo que se conoce como el *principio de la máxima utilidad esperada*. En este contexto no tiene sentido entrar en consideraciones de las teorías éticas más ambiciosas que pretenden formular criterios generales y uniformes para justificar todo tipo de reglas morales a la hora de tomar decisiones en situaciones de cualquier índole.

Qué es una función de utilidad $U(s)$, siendo $U()$ la función y «s» el estado en el que se encuentra el sistema. Es una forma de asignar un número que nos indica lo deseable que es un estado. Esta medida de la utilidad, combinada con la probabilidad de ocurrencia de las acciones, da como resultado la utilidad esperada para cada acción.

Sea

- E , el factor que representa la «evidencia» disponible por el sistema.
- A , una acción no determinista.
- $\text{Resultado}_i(A)$, el resultado «i» de la actuación de A , «i» recorre el rango de posibles resultados.
- $P(\text{Resultado}_i(A) | \text{Realizar}(A), E)$, representa la probabilidad de que dadas las evidencias E y la ejecución de la acción A , se obtenga el resultado «i» asociado a la acción A .
- $U(\text{Resultado}_i(A))$, Valor de la función utilidad asociada al $\text{Resultado}_i(A)$.
- Al producto $P(\text{Resultado}_i(A) | \text{Realizar}(A), E) \times U(\text{Resultado}_i(A))$, se denomina *utilidad esperada de la acción A*, dada la evidencia E obteniendo el Resultado_i .
- Si se sabe que «i» recorre el rango de posibles resultados, entonces se define la utilidad esperada de la acción A , dada la evidencia E , y se denota como $UE(A|E)$ a la expresión suma en «i».

$$UE(A|E) = \sum_i P(\text{Resultado}_i(A) | \text{Realizar}(A), E) * U(\text{Resultado}_i(A))$$

El principio de la máxima utilidad esperada (MUE) establece que, entre varias acciones posibles, se debe elegir aquella acción que maximice la utilidad esperada del sistema.

$$MUE(A_j|E) = \sum_i P(\text{Resultado}_i(A_j) | \text{Realizar}(A_j), E) * U(\text{Resultado}_i(A_j))$$

La acción seleccionada es la A_k que maximiza $MUE(MUE(A_j|E)) \forall j$.

Esto que es fácil de asimilar, normalmente es muy difícil de calcular, basta con fijarse en los elementos que interviene en la definición. Si no se conocen bien las probabilidades, el cálculo es imposible o muy difícil si hay que calcularlas previamente. Tampoco se dice nada de cómo definir la función de utilidad. Pero sí que es verdad que proporciona un marco conceptual formal a tener en cuenta.

Ejemplo: La paradoja de San Petersburgo

- Juegan dos personas Y y Z.
- Se comienza con un bote de 2 €.
- Se lanza una moneda al aire.
 - Si sale cruz, Y dobla la cantidad que hay en el bote.
 - Si sale cara, Z se lleva el bote.

Explicación

- Se parte de un bote con 2 €.
- Si la primera tirada sale cara, Z gana 2 €.
- Si la primera tirada sale cruz, Y pone 2 €, por lo tanto, en el bote hay 4 €.
 - Si la segunda tirada sale cara, Z gana 4 €.
- Si la primera tirada sale cruz, Y pone 2 €, por lo tanto, en el bote hay 4 €.
 - Si la segunda tirada sale cruz, Y pone 4 €, por lo tanto, en el bote hay 8 €.
 - Si la tercera tirada sale cara, Z gana 8 €.
-

Si después de un conjunto de $n-1$ tiradas siempre sale cruz, y en la tirada n sale cara, Z gana 2^n €.

Obviamente a Z le conviene que salga cara lo más tarde posible. En cualquier caso, Z gana siempre algo de dinero, por lo que es justo que Y cobre alguna cantidad o cuota para permitirle participar en el juego.

La pregunta que se hizo Bernoulli (1725), y que en cierto modo sigue sin resolverse es: ¿Cuál es la cuota de entrada que se debería cobrar para que el juego fuera justo?

Hagamos el análisis, suponga que vamos a elegir U como la ganancia, entonces

$$UE(A|E) = \sum_i P(\text{Resultado}_i(A) | \text{Realizar}(A), E) * U(\text{Resultado}_i(A))$$

- La probabilidad de que salga la primera cara la primera vez es $\frac{1}{2}$ y se gana 2 €.
- La probabilidad de que salga la primera cara a la segunda vez es $\frac{1}{2} \times \frac{1}{2}$ y se gana 4 €.
- ...
- La probabilidad de que salga la primera cara en la n-ésima tirada es $1/2^n$ y se gana 2^n €

$$UE(X) = \sum_{n \geq 1} \frac{1}{2^n} * 2^n = + \infty$$

Por lo tanto, el jugador debería pagar una cantidad infinita para que el juego fuera justo.

En otras palabras, si le ofrecen entrar en el juego con una cuota de 1 millón de euros, debería aceptar, porque la ganancia media en el juego, que es infinita, superará esa y cualquier otra cantidad. Pero el sentido común nos dice que el valor medio de la ganancia no determina la cuota de entrada aceptable.

¿Cómo se determina dicha cuota?

La cuestión radica en la definición de la función de utilidad. Una propuesta interesante podría venir del valor esperado no del dinero, sino de la utilidad del dinero. En un estudio pionero sobre las funciones de utilidad actuales, Grayson (1960) encontró que la utilidad del dinero era casi exactamente proporcional al logaritmo de la utilidad del dinero; idea ya sugerida por Bernuilli en 1738.

Por lo tanto, si $U(x) = \ln(x)$, siendo x el premio del juego, entonces

$$UE(X) = \sum_{n \geq 1} \frac{1}{2^n} * \ln(2^n) \approx 1,38627982$$

La tabla de resultados para los primeros veinte intentos es la siguiente:

Intentos	Probabilidad	X = Premio	U (X)	E [U (X)]	Util.Marg.
1	0.5000000	2	0.69314718	0.3465736	
2	0.2500000	4	1.38629436	0.3465736	0.3465736
3	0.1250000	8	2.07944154	0.2599302	0.1732868
4	0.0625000	16	2.77258872	0.1732868	0.0866434
5	0.0312500	32	3.4657359	0.1083042	0.0433217
6	0.0156250	64	4.15888308	0.0649825	0.0216608
7	0.0078125	128	4.85203026	0.0379065	0.0108304
8	0.0039063	256	5.54517744	0.0216608	0.0054152
9	0.0019531	512	6.23832463	0.0121842	0.0027076
10	0.0009766	1024	6.93147181	0.0067690	0.0013538
11	0.0004883	2048	7.62461899	0.0037230	0.0006769
12	0.0002441	4096	8.31776617	0.0020307	0.0003385
13	0.0001221	8192	9.01091335	0.0011000	0.0001692
14	0.0000610	16384	9.70406053	0.0005923	0.0000846
15	0.0000305	32768	10.3972077	0.0003173	0.0000423
16	0.0000153	65536	11.0903549	0.0001692	0.0000212
17	0.0000076	131072	11.7835021	0.0000899	0.0000106
18	0.0000038	262144	12.4766493	0.0000476	0.0000053
19	0.0000019	524288	13.1697964	0.0000251	0.0000026
20	0.0000010	1048576	13.8629436	0.0000132	0.0000013

Total = 1.38627982

Como se puede observar en la tabla anterior en el intento número 20 la suma del valor esperado de la utilidad del juego ha llegado a 1,38627982, muy cerca ya de su límite. Esto se ve observando que cada nuevo intento agrega un valor esperado de utilidad insignificante.

Por lo tanto, el valor del juego, medido en unidades de «utilidad» es 1,38627982 aproximadamente. Si se mide en euros, el juego tiene un valor igual al antilogaritmo de 1,3862, que aproximadamente es 4 €.

El valor del juego es 4 €

Este ejemplo pone en evidencia el valor de la teoría de toma de decisiones y la importancia de definir la función de utilidad.

Modelos para soporte de toma de decisiones. Árboles de decisión

Un árbol de decisión es una representación gráfica de las alternativas disponibles para un sistema y los aspectos que son inciertos. Son una extensión natural de las redes bayesianas que contienen nodos de diferentes tipos.

Un árbol de decisión tiene tres tipos de nodos:

- Nodos de decisión (cuadrados) que son las alternativas que tiene el sistema.
- Nodos aleatorios (círculos) que son los posibles resultados de un evento incierto. Cada alternativa tiene asociada una determinada probabilidad.
- Nodos terminales (triángulos, hojas del árbol) tienen un costo o utilidad.

El funcionamiento es simple. Al encontrar un nodo de decisión, se debe seleccionar una de las alternativas. Al encontrarse con un nodo aleatorio, pierde el control, la trayectoria está determinada por las probabilidades.

Evaluación. A partir de los nodos terminales:

- Para los nodos aleatorios, E, se calcula la utilidad esperada en función de los costos de cada alternativa y sus probabilidades asociadas.
- Para los nodos de decisión, D, se selecciona la alternativa de mayor utilidad (menor costo) esperada.

Teoría de la decisión en una etapa

Se definen del modo siguiente.

- $E = \{E_1, \dots, E_m\}$, el conjunto de los estados.
- $P = \{p_1, \dots, p_m\}$, sus probabilidades asociadas.
- $A = \{A_1, \dots, A_n\}$, el conjunto de las posibles decisiones.
- X_{ij} ; la consecuencia de tomar la decisión A_i si ocurre el estado E_j (ganancia o coste).

Decisión en una etapa

- Tabla de decisión

	E_1	E_2	...	E_m	Estados
	p_1	p_2	...	p_m	Probabilidades
A_1	X_{11}	X_{12}	...	X_{1m}	Matriz de pagos
A_2	X_{21}	X_{22}	...	X_{2m}	
...	
A_n	X_{n1}	X_{n2}	...	X_{nm}	

Decisiones

Ejemplo 1:

Un comerciante vende un artículo cuya demanda mensual (E) puede ser

- 1 unidad, con probabilidad 0,1.
- 2 unidades, con probabilidad 0,3.
- 3 unidades, con probabilidad 0,4.
- 4 unidades, con probabilidad 0,2.

El precio de venta del artículo es de 6500 €, y el precio de compra 5000€.

El comerciante debe decidir cuántas unidades de dicho artículo debe comprar (A), teniendo en cuenta que cada unidad no vendida al finalizar el mes debe devolverla a un precio de 4000 €

- Si compra 1 unidad gasta 5000 €
 - Si vende 1, gana $6500€ - 5000 € = +1500 €$
 - En cualquier otra situación, como solo tiene una, la ganancia no varía.
- Si compra 2 unidades gasta 10 000 €
 - Si vende 1 y devuelve la otra gana
 - $6500 € - 10 000 € + 4000 € = +0,500 €$
 - Si vende 2, gana $13 000€ - 10 000 € = +3000 €$
 - En cualquier otra situación, como solo tiene dos, la ganancia no varía.
- Si compra 3 unidades gasta 15 000 €
 - Si vende 1 y devuelve las otras dos, gana
 - $6500 € - 15 000 € + 8000 € = -0,500 €$
 - Si vende 2, gana $13 000 € - 15 000 € + 4000 € = +2000 €$
 - Si vende 3, gana $19 500 € - 15 000 € = +4500 €$
 - En cualquier otra situación, como solo tiene tres, la ganancia no varía.
- Si compra 4 unidades gasta 20 000 €
 - Si vende 1 y devuelve las otras tres, gana
 - $6500 € - 20 000 € + 12 000 € = -1500 €$
 - Si vende 2, gana $13 000 € - 20 000 € + 8000 € = +1000 €$
 - Si vende 3, gana $19.500 € - 20.000 € + 4000 € = +3500 €$
 - Si vende 4, gana $26 000 € - 20 000 € = +6000 €$

Decisión en una etapa

	E ₁ =1	E ₂ =2	E ₃ =3	E ₄ =4
	p ₁ =0.1	p ₂ =0.3	p ₃ =0.4	p ₄ =0.2
A ₁ =1	1500	1500	1500	1500
A ₂ =2	500	3000	3000	3000
A ₃ =3	-500	2000	4500	4500
A ₄ =4	-1500	1000	3500	6000

Criterios factibles para tomar decisiones:

- Información perfecta

- Criterio de Laplace (valor esperado), decisión comprar 3 unidades.

$$UE(A|E) = \sum_i P(\text{Resultado}_i(A) | \text{Realizar}(A), E) * U(\text{Resultado}_i(A))$$

- Si A = compra 1 unidad, la ganancia esperada es

$$UE(A|E) = 0,1 \times 1500 + 0,3 \times 1500 + 0,4 \times 1500 + 0,2 \times 1500 = 1500$$

- Si A = compra 2 unidades, la ganancia esperada es

$$UE(A|E) = 0,1 \times 500 + 0,3 \times 3000 + 0,4 \times 3000 + 0,2 \times 3000 = 2750$$

- Si A = compra 3 unidades, la ganancia esperada es

$$UE(A|E) = 0,1 \times (-500) + 0,3 \times 2000 + 0,4 \times 4500 + 0,2 \times 4500 = 3250$$

- Si A = compra 4 unidades, la ganancia esperada es

$$UE(A|E) = 0,1 \times (-1500) + 0,3 \times 1000 + 0,4 \times 3500 + 0,2 \times 6000 = 2750$$

- Criterio de lo más probable

- El estado más probable es E₃, ya que su p(E₃) = 0,4, la mayor ganancia es 4500 €, por lo tanto, la decisión es comprar 3 unidades.

- Criterio del estado medio

- El estado medio es = 1 x 0,1 + 2 x 0,3 + 3 x 0,4 + 4 x 0,2 = 2,7, que no corresponde a ningún estado. Se redondea al valor entero más cercano, por lo tanto, E₃, la mayor ganancia es 4500 €, por lo tanto, la decisión es comprar 3 unidades.

- Bajo incertidumbre.
 - Criterio de Wald (maximin). Suponer para cada posible decisión que ocurrirá el peor estado, y quedarse con la decisión de mayor ganancia mínima. Conservador, pesimista. Decisión comprar 1 unidad.
 - Si $A = 1$, la ganancia mínima es +1500 €.
 - Si $A = 2$, la ganancia mínima es +0,500 €.
 - Si $A = 3$, la ganancia mínima es -0,500 €.
 - Si $A = 4$, la ganancia mínima es -1500 €.
 - Criterio maximax. Suponer para cada posible decisión que ocurrirá el mejor estado, y quedarse con la decisión de mayor ganancia máxima. Arriesgado, optimista. Decisión comprar 4 unidades.
 - Si $A = 1$, la ganancia máxima es +1500 €.
 - Si $A = 2$, la ganancia máxima es +3000 €.
 - Si $A = 3$, la ganancia máxima es +4500 €.
 - Si $A = 4$, la ganancia máxima es +6000 €.
 - Criterio de Hurwicz. Siendo $0 \leq \alpha \leq 1$, un índice de optimismo, valorar cada decisión mediante la expresión

$$\alpha \times \text{máx.} + (1-\alpha) \times \text{mín.}$$
 Adoptar la decisión mejor valorada. Combina las actitudes optimista y pesimista. Si $\alpha = 0$ se obtiene maximin. Si $\alpha = 1$ se obtiene maximax. Decisión comprar 1 unidad.

Tomando $\alpha = 0,3$ entonces

 - Si $A = 1$, $0,3 \times 1500 + 0,7 \times 1500 = 1500$.
 - Si $A = 2$, $0,3 \times 500 + 0,7 \times 3000 = 1250$.
 - Si $A = 3$, $0,3 \times (-500) + 0,7 \times 4500 = 1000$.
 - Si $A = 4$, $0,3 \times (-1.500) + 0,7 \times 6000 = 0,750$.
 - Criterio de Savage. Construir la matriz de costes de oportunidad sustituyendo cada elemento de la matriz original por la diferencia entre el mayor valor del estado correspondiente y aquel elemento. Sobre esa matriz aplicar el criterio minimax (el mínimo de los máximos) u otro criterio. Decisión comprar 3 unidades.

	E_1 (1500)	E_2 (3000)	E_3 (4500)	E_4 (6000)
A_1	1500-1500=0	3000-1500=1500	4500-1500=3000	6000-1500=4500
A_2	1500-500=1000	3000-3000=0	4500-3000=1500	6000-3000=3000
A_3	1500+500=2000	3000-2000=1000	4500-4500=0	6000-4500=1500
A_4	1500+1500=3000	3000-1000=2000	4500-3500=1000	6000-6000=0

Teoría de la decisión en multi-etapa

Son procesos secuenciales de decisión-azar, cuyo objetivo es determinar la secuencia de decisiones que proporcione el mejor resultado de acuerdo con algún criterio establecido, como el de máxima utilidad esperada. La estructura gráfica que se suele utilizar es de un árbol.

Dicho árbol se construye:

- De izquierda a derecha (raíz a hojas).
- Hay unos vértices de decisión. De ellos parten arcos en trazo continuo asociados a decisiones.
- Hay unos vértices de azar «simbolizados por círculos». De ellos parten arcos en trazo discontinuo asociados a estados, valorados mediante su probabilidad.
- Vértice inicial o raíz. Siempre es un vértice de decisión.
- Vértices terminales u hojas «simbolizados por triángulos». Llevan asociado un beneficio o coste.

Valoración

- De derecha a izquierda (de hojas a raíz).
- Cada vértice de azar se valora según el criterio del valor esperado (o algún otro criterio), teniendo en cuenta los valores de los vértices finales de los arcos que parten de él.
- Cada vértice de decisión se valora con el valor máximo (mínimo en caso de costes) de los valores de los vértices finales de los arcos que parten de él. Las decisiones no seleccionadas se rechazan.

Ejemplo 1. Un vendedor ambulante se plantea en enero ir a una feria en septiembre o no hacerlo. Si va tiene que pedir un permiso que cuesta 40 000 €. Un mes antes de la feria sabrá si va a hacer mal tiempo con probabilidad 0,3, en cuyo caso prefiere no ir, o buen tiempo, en cuyo caso puede hacer dos tipos de pedido:

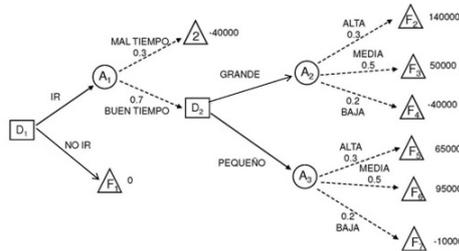
- Grande: 900 unidades a 100 € la unidad, que puede vender a 300 € la unidad.
- Pequeño: 600 unidades a 125 € la unidad, que puede vender a 350 € la unidad.

La demanda puede ser de 900, 600 o 300 unidades, con probabilidades respectivas de 0,3, 0,5 o 0,2.

Si la demanda es mayor que la oferta, ha de rebajar en 50€ el precio de venta de cada unidad.

La representación gráfica es

Procesos de decisión multi-etapa



Siendo las cuentas

- 900 unidades a 100 € de compra y 300 € de venta son:
 - Si la demanda es de 900 y tiene 900, entonces.
 - 40 000 € de gasto de licencia.
 - 90 000 € de gasto de compra.
 - 270 000 € de venta.
 - 140 000 € de beneficio.
 -
 - Si la demanda es de 600 y tiene 900, entonces.
 - 40 000 € de gasto de licencia.
 - 90 000 € de gasto de compra.
 - 180 000 € de venta.
 - 50 000 € de beneficio.
 -
 - Si la demanda es de 300 y tiene 900, entonces.
 - 40 000 € de gasto de licencia.
 - 90 000 € de gasto de compra.
 - 90 000 € de venta.
 - -40 000 € de beneficio.

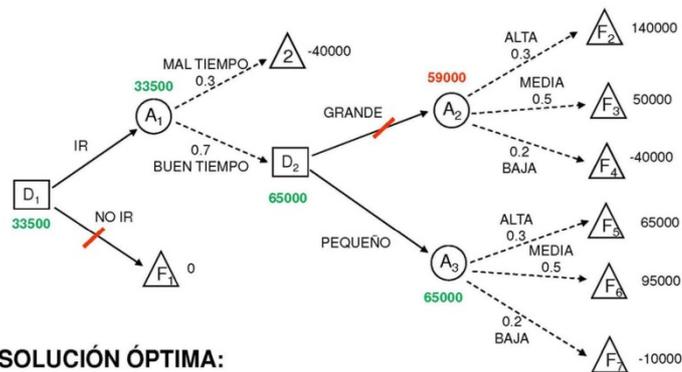
- 600 unidades a 125 € de compra y 350 € de venta son:
 - Si la demanda es de 900 y tiene 600, entonces.
 - 40 000 € de gasto de licencia.
 - 75 000 € de gasto de compra.
 - 180 000 € de venta.
 - 65 000 € de beneficio.
 -
 - Si la demanda es de 600 y tiene 600, entonces.
 - 40 000 € de gasto de licencia.

- 75 000 € de gasto de compra.
- 210 000 € de venta.
- 95 000 € de beneficio.
-
- Si la demanda es de 300 y tiene 600, entonces.
- 40 000 € de gasto de licencia.
- 75 000 € de gasto de compra.
- 105 000€ de venta.
- -10 000 € de beneficio.

Calculemos ahora la MUE

- $MUE(A|E) = \sum_i P(\text{Resultado}_i(A) | \text{Realizar}(A), E) * U(\text{Resultado}_i(A))$

Procesos de decisión multi-etapa



SOLUCIÓN ÓPTIMA:

Pedir el permiso, y si hace buen tiempo, hacer un pedido pequeño
Ganancia esperada: 33500 €

Ejemplo 2. Un vendedor ambulante se plantea en enero ir a una feria en septiembre o no hacerlo. Si va tiene que pedir un permiso que cuesta 40 000 €. Un mes antes de la feria sabrá si va a hacer mal tiempo con probabilidad 0,3, en cuyo caso prefiere no ir, o buen tiempo, en cuyo caso puede hacer dos tipos de pedido:

- Grande: 900 unidades a 100 € la unidad, que puede vender a 300 € la unidad.
- Pequeño: 600 unidades a 125 € la unidad, que puede vender a 350 € la unidad.

La demanda puede ser de 900, 600 o 300 unidades, con probabilidades respectivas de 0,3, 0,5 o 0,2.

Si la demanda es mayor que la oferta, ha de rebajar en 50 € el precio de venta de cada unidad.

Pero además por 10 000 € puede consultar a un meteorólogo sobre el tiempo que va a hacer en septiembre, antes de tomar cualquier decisión. Dicho meteorólogo tiene el siguiente comportamiento:

- Cuando hace buen tiempo, acierta 9 de cada 10 veces. $P(pb|B) = 0,9$.
- Cuando hace mal tiempo, acierta 3 de cada 10 veces. $P(pm|M) = 0,3$.

Con esos nuevos datos, el comerciante debe decidir previamente si consultar al meteorólogo.

Supongamos que hacemos la consulta, entonces a los gastos anteriores hay que añadirles 10 000 € más, por lo tanto, las nuevas cuentas:

- 900 unidades a 100 € de compra y 300 € de venta son:
 - Si la demanda es de 900 y tiene 900, entonces.
 - 10 000 € de la consulta.
 - 40 000 € de gasto de licencia.
 - 90 000 € de gasto de compra.
 - 270 000 € de venta.
 - 130 000 € de beneficio.
 -
 - Si la demanda es de 600 y tiene 900, entonces.
 - 10 000 € de la consulta.
 - 40 000 € de gasto de licencia.
 - 90 000 € de gasto de compra.
 - 180 000 € de venta.
 - 40 000 € de beneficio.
 -
 - Si la demanda es de 300 y tiene 900, entonces.
 - 10 000 € de la consulta.
 - 40 000 € de gasto de licencia.
 - 90 000 € de gasto de compra.

- 90 000 € de venta.
- -50 000 € de beneficio.
- 600 unidades a 125 € de compra y 350 € de venta son:
 - Si la demanda es de 900 y tiene 600, entonces.
 - 10 000 € de la consulta.
 - 40 000 € de gasto de licencia.
 - 75 000 € de gasto de compra.
 - 180 000 € de venta.
 - 55 000 € de beneficio.
 -
 - Si la demanda es de 600 y tiene 600, entonces.
 - 10 000 € de la consulta.
 - 40 000 € de gasto de licencia.
 - 75 000 € de gasto de compra.
 - 210 000 € de venta.
 - 85 000 € de beneficio.
 -
 - Si la demanda es de 300 y tiene 600, entonces.
 - 10 000 € de la consulta.
 - 40 000 € de gasto de licencia.
 - 75 000 € de gasto de compra.
 - 105 000 € de venta.
 - -20 000 € de beneficio.

Las probabilidades *a priori* eran:

- De buen tiempo $P(B) = 0,7$
- De mal tiempo $p(M) = 0,3$

La probabilidad total de:

$$P(pb) = P(pb|B) \times P(B) + P(pb|M) \times P(M) = 0,9 \times 0,7 + 0,7 \times 0,3 = 0,84$$

$$P(pm) = 1 - P(pb) = 1 - 0,84 = 0,16$$

Aplicando el teorema de Bayes

$$P(B|pb) = \frac{P(pb|B) \times P(B)}{P(pb)} = \frac{0,9 \times 0,7}{0,84} = 0,75$$

$$P(M|pb) = 1 - P(B|pb) = 0,25$$

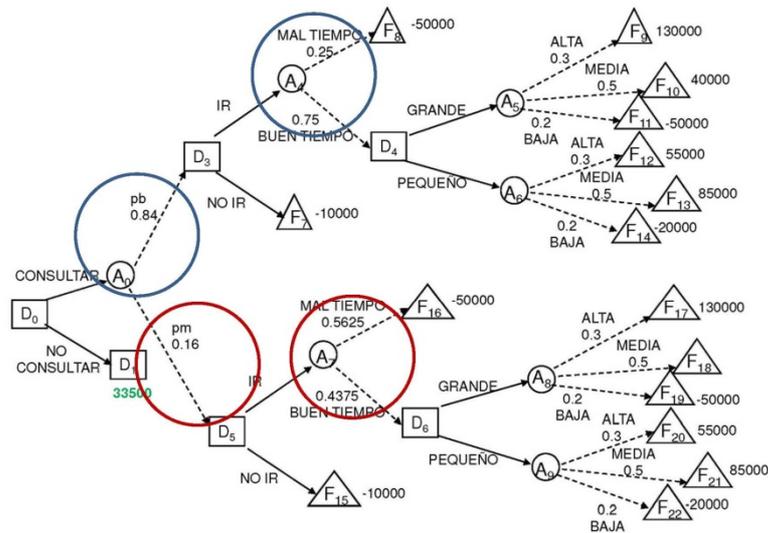
$$P(B|pm) = \frac{P(pm|B) \times P(B)}{P(pm)} = \frac{0,1 \times 0,7}{0,16} = 0,4375$$

$$P(M|pm) = 1 - P(B|pm) = 0,5625$$

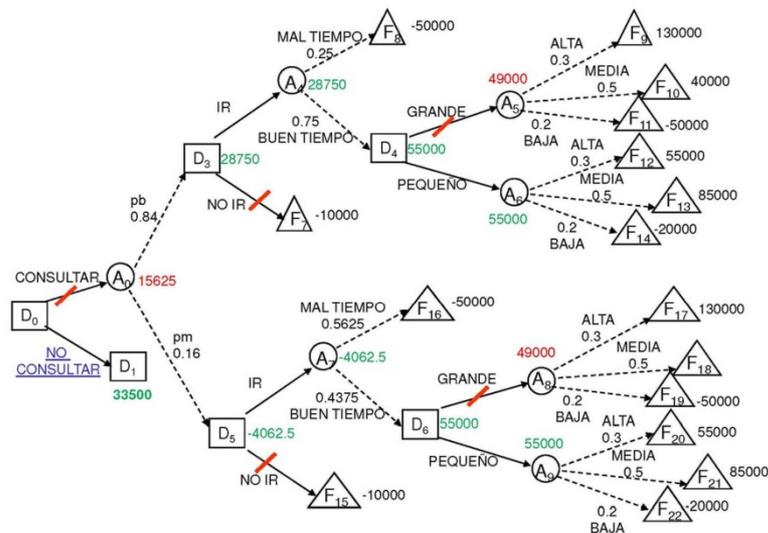
Calculemos ahora la MUE

- $MUE(A|E) = \sum_i P(\text{Resultado}_i(A) | \text{Realizar}(A), E) * U(\text{Resultado}_i(A))$

Procesos de decisión multi-etapa



Procesos de decisión multi-etapa



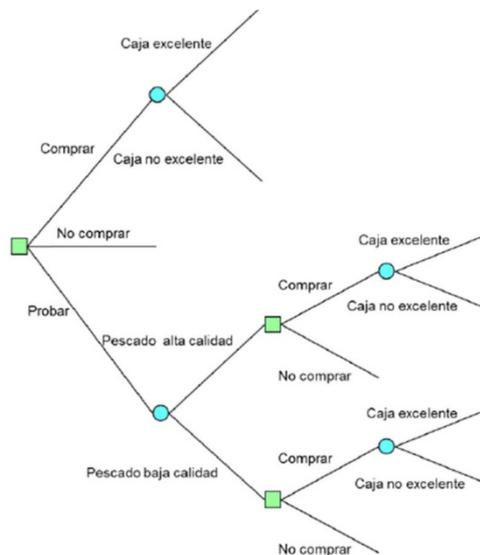
Ejemplo 3. Un empresario adquiere pescado fresco en el mercado central para su posterior venta. Cada caja de pescado la identifica como excelente o no excelente en función del porcentaje de pescado que se considere de calidad excelente.

- Una caja de pescado excelente contiene un 90 % de pescado de alta calidad, y genera un beneficio de 100 €.
- Una caja de pescado no excelente contiene un 20 % de pescado de alta calidad y causa pérdidas de 100 € por mala imagen.

Antes de comprar una caja el empresario puede comprobar la calidad de la misma extrayendo un ejemplar de pescado con el objetivo de verificar si se trata o no de pescado de alta calidad.

Establezca la estrategia que debe seguir el empresario, así como el coste de la información.

Árbol de decisión



Necesitamos calcular las probabilidades *a posteriori*, por lo que debe utilizarse el teorema de Bayes.

Empecemos:

- La probabilidad *a priori* de que una caja sea o no de excelente calidad es obviamente del 50 %.

- Las probabilidades condicionales son:
 - $P(\text{pescado sea de alta calidad} \mid \text{caja excelente}) = 0,9.$
 - $P(\text{pescado sea de baja calidad} \mid \text{caja excelente}) = 0,1.$
 - $P(\text{pescado sea de alta calidad} \mid \text{caja no excelente}) = 0,2.$
 - $P(\text{pescado sea de baja calidad} \mid \text{caja no excelente}) = 0,8.$

- De donde, la probabilidad *a priori* de cada uno de los acontecimientos:

$$P(\text{pescado sea de alta calidad}) = P(\text{caja excelente}) \times P(\text{pescado sea de alta calidad} \mid \text{caja excelente}) + P(\text{caja no excelente}) \times P(\text{pescado sea de alta calidad} \mid \text{caja no excelente}) = [0,5 \times 0,9] + [0,5 \times 0,2] = 0,55$$

$$P(\text{pescado sea de baja calidad}) = P(\text{caja excelente}) \times P(\text{pescado sea de baja calidad} \mid \text{caja excelente}) + P(\text{caja no excelente}) \times P(\text{pescado sea de baja calidad} \mid \text{caja no excelente}) = [0,5 \times 0,1] + [0,5 \times 0,8] = 0,45$$

Aplicando Bayes:

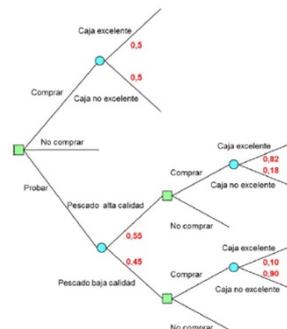
- $P(\text{caja excelente} \mid \text{pescado sea de alta calidad}) = \frac{P(\text{caja excelente}) \times P(\text{pescado sea de alta calidad} \mid \text{caja excelente})}{P(\text{pescado sea de alta calidad})} = \frac{0,5 \times 0,9}{0,55} = 0,82$

- $P(\text{caja no excelente} \mid \text{pescado sea de alta calidad}) = \frac{P(\text{caja no excelente}) \times P(\text{pescado sea de alta calidad} \mid \text{caja no excelente})}{P(\text{pescado sea de alta calidad})} = \frac{0,5 \times 0,2}{0,55} = 0,18$

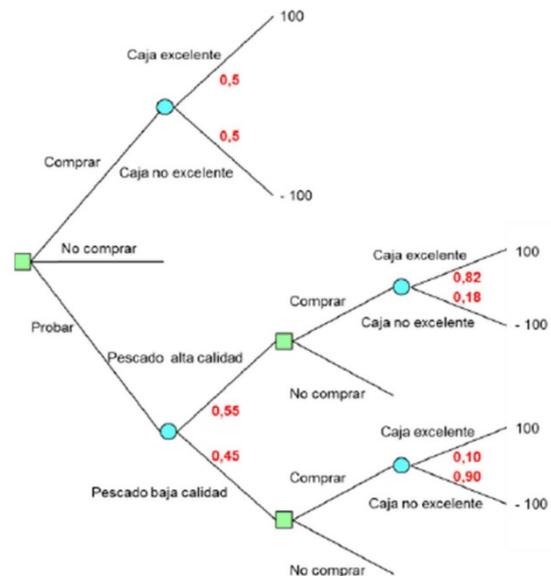
- $P(\text{caja excelente} \mid \text{pescado sea de baja calidad}) = \frac{P(\text{caja excelente}) \times P(\text{pescado sea de baja calidad} \mid \text{caja excelente})}{P(\text{pescado sea de baja calidad})} = \frac{0,5 \times 0,1}{0,45} = 0,10$

- $P(\text{caja no excelente} \mid \text{pescado sea de baja calidad}) = \frac{P(\text{caja no excelente}) \times P(\text{pescado sea de baja calidad} \mid \text{caja no excelente})}{P(\text{pescado sea de baja calidad})} = \frac{0,5 \times 0,8}{0,45} = 0,90$

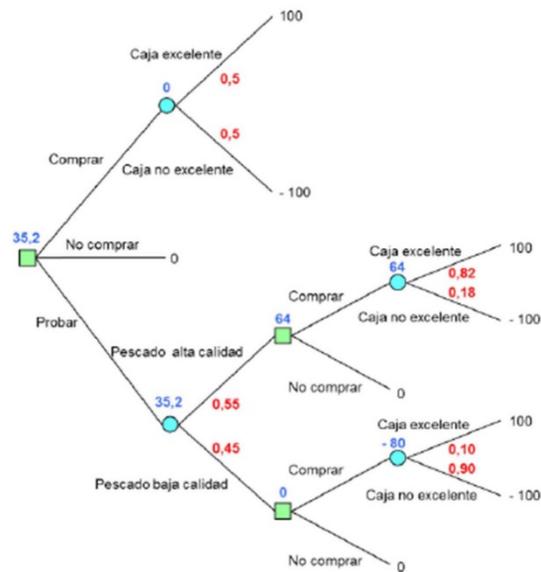
Probabilidades "a posteriori"



Coste de cada rama



- Dado que la etapa final es probabilista, se debe aplicar el criterio de la media para determinar el beneficio esperado de cada alternativa de decisión.
 - $(100 \times 0,82) + (-100 \times 0,18) = 64 \text{ €}$
 - $(100 \times 0,10) + (-100 \times 0,90) = -80 \text{ €}$
 - $(100 \times 0,50) + (-100 \times 0,50) = 00 \text{ €}$
 - $(64 + 0) = 64 \text{ €}$
 - $(-80 + 0) = -80 \text{ €}$
 - $(64 \times 0,5) + (0 \times 0,45) = 35,2 \text{ €}$



Conclusión:

La estrategia que debe seguir el empresario es la de extraer un ejemplar de pescado con el objetivo de verificar si se trata o no de pescado de alta calidad, en el caso de que el pescado extraído sea de alta calidad, debe comprar la caja de pescado. Por el contrario, si el pescado extraído es de baja calidad, no debe comprar la caja de pescado. Con esta estrategia el beneficio esperado es de 35,2 €. Si por llevar a cabo este control de calidad le cobraran más de 35,2 €, no interesaría llevarlo a cabo.

Las redes de decisión dinámicas incorporan nodos de decisión y de utilidad a las redes bayesianas dinámicas. Normalmente se tienen una serie de decisiones en el tiempo y una cierta utilidad en el futuro. Este tipo de redes son equivalentes a lo que se conoce como *procesos de decisión de Markov* o a un *proceso de decisión de Markov parcialmente observables*.

En el caso de los **procesos de decisión de Markov**, el estado es completamente observable (conocido en todo momento), y existen diferentes acciones cuyas acciones vienen descritas mediante una distribución de probabilidad (un modelo de transición para cada acción). Además, se introduce la noción de *recompensa* en un estado. En este caso hay que tomar una decisión, es decir, seleccionar las acciones para maximizar una recompensa futura.

	Estado observable	Estado oculto (observaciones)
No hay acciones (sólo transiciones estocásticas)	Cadenas de Markov (Markov Chains)	Modelos Ocultos de Markov (HMMs)
Hay acciones (que producen transiciones estocásticas)	Procesos de Decisión de Markov (MDPs)	Procesos de Decisión de Markov Parcialmente Observables (POMDPs)

La nomenclatura usual es la siguiente:

- S es un conjunto de estados.
- A es un conjunto de acciones.
- T es un conjunto de probabilidades condicionales de transición entre estados, se representa como $T(s, a, s') = P(s' | a, s)$.
- $R = S \times A \rightarrow$ Conjunto de los números reales es la función de recompensa que especifica el «valor» de ejecutar cierta acción a en el estado s , es decir, $r(s, a)$.
- $\gamma \in [0, 1]$ es el factor de descuento, que permite r más las recompensas cercanas que las lejanas.

El conjunto de posibles estados puede ser finito o infinito y puede considerarse horizonte finito (lo que significa que a partir de un instante determinado ya no existen más transiciones entre estados ni recompensas) o infinito.

En este contexto, una solución no puede ser una secuencia de acciones, ya que el efecto de cada acción es incierto. Más bien buscamos una política que en cada posible estado recomiende una acción a aplicar: por cada estado que pasemos, aplicamos la acción que nos recomienda esa política. Una política indica la acción que se debe ejecutar dado el estado (o probabilidad del estado), $\pi(s) = a$. Una misma política puede generar secuencias de acciones distintas (aunque unas con más probabilidad que otras).

Se considera que las probabilidades de transición entre estados solo dependen del estado actual por lo que son procesos markovianos.

La utilidad de un estado depende de la secuencia de acciones tomadas a partir de dicho estado (s_i) de acuerdo con la política establecida (π).

En principio, se puede obtener como la utilidad esperada de todas las posibles secuencias de acciones $U(s_0, s_1, s_2, \dots, s_n) = R(s_0) + R(s_1) + \dots + R(s_n)$. Para evitar utilidades infinitas (en el caso de horizontes infinitos) se consideran recompensas depreciadas. $U(s_0, s_1, s_2, \dots, s_n) = R(s_0) + \gamma \times R(s_1) + \dots + \gamma^n \times R(s_n)$, con $\gamma \in [0, 1]$.

En cada periodo de tiempo, el medio ambiente se encuentra en un estado. El sistema realiza una acción y se produce una transición a otro estado. Al mismo tiempo el sistema realiza una observación. Por último, el agente recibe una recompensa. El objetivo es que el agente debe elegir las acciones en cada paso de tiempo que maximice su recompensa futura. El factor de descuento favorece más las recompensas cercanas que las lejanas.

Dada la condición de separabilidad, la utilidad de un estado se puede obtener en forma iterativa maximizando la utilidad del siguiente estado.

$$U(s_i) = R(s_i) + \max_a \sum_j P(s_j | s_i, a) U(s_j)$$

Entonces, la utilidad de un estado vendrá dada por

$$U(s_i) = R(s_i) + \gamma \times \max_a \sum_j T(s_j, s_i, a) U(s_j)$$

La política óptima está dada por la acción que de mayor utilidad

$$\pi^*(s_i) = \arg \max_a \sum_j P(s_j | s_i, a) U(s_j)$$

Entonces, la mejor acción es aquella que maximiza la utilidad esperada

$$\pi^*(s_i) = \arg \max_a \sum_j T(s_j, s_i, a) U(s_j)$$

Conclusión

- La política óptima π^* busca proporcionar la mejor acción en cada caso.
- La mejor acción en cada estado depende de la utilidad de cada estado.
- La utilidad en cada estado es aquella que se obtiene siguiendo la acción óptima.

Entonces para cada estado tenemos una familia de ecuaciones que depende de otros estados con ecuaciones similares

$$U(s_i) = R(s_i) + \gamma \times \text{máx.}_a \sum_j T(s_j, s_i, a) U(s_j)$$

Dado un problema de N estados «s» se generará un sistema de N ecuaciones $U(s)$ con n incógnitas $U(s')$, que será no lineal debido al «máx.» que se debería resolver simultáneamente.

El algoritmo iterativo sería

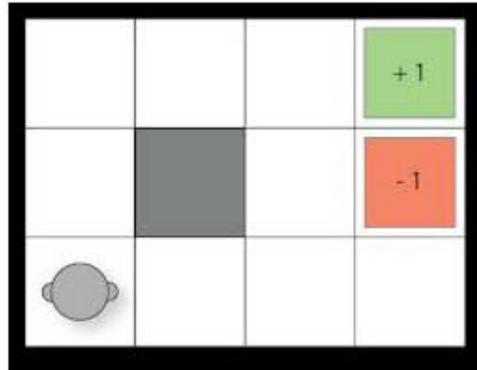
- $U(s) = R(s)$ en $t = 0$.
-
- $U_{t+1}(s) = R(s) + \gamma \times \text{máx.}_a \sum_j P(s_j | s_i, a) U(s_j)$.
- ...
- Cuando $t \rightarrow \infty$, los valores de utilidad convergen a un valor estable (propiedad de Markov):
 - Fijado un ϵ , se para cuando converge, es decir, $|U_n - U_{n+1}| < \epsilon$, y se denominará a esta utilidad $U^*(s)$.

Cálculo de la política óptima:

- Habiendo obtenido $U^*(s)$, se tiene que
$$\pi^*(s_i) = \arg \text{máx.}_a \sum_j T(s_i, a, s_j) U^*(s_j)$$

Ejemplo: Cuadrícula

En la siguiente cuadrícula, vista como espacio de estados, los *estados* son las casillas, las acciones son las descritas anteriormente y el objetivo es llegar al estado final +1, evitando el -1

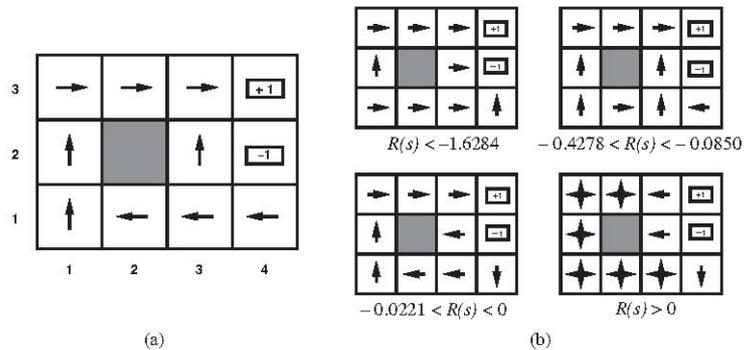


Ejemplo: Cuadrícula

- Si el mundo fuera determinista, una solución sería: *arriba, arriba, izquierda, izquierda, izquierda*.
 - Sin embargo, esa secuencia sólo sigue el camino deseado con probabilidad $(0.8)^5$. Podría ocurrir también (con probabilidad más baja) que esa secuencia no le llevara al objetivo deseado.
- Otra novedad respecto a espacio de estados: cada estado tiene una **recompensa** asociada.
 - En el ejemplo: suponemos que todos los estados tienen recompensa -0.04, excepto los dos estados terminales, que tienen recompensa +1, y -1, respectivamente. En los estados terminales, ya no se puede aplicar ninguna acción.
- El objetivo es decidir en cada estado qué acción aplicar, de manera que se maximice el *total de recompensas* de los estados por donde se pasa.
 - Más adelante matizaremos qué queremos decir con el *total de recompensas*



Ejemplo de políticas en la cuadrícula



◀ ▶ ⏪ ⏩ 🔍 🔄

Valoración de secuencia de estados en el tiempo

- Supongamos que mediante aplicación de una secuencia de acciones, se ha generado una secuencia de estados $q_0 q_1 q_2 \dots$
- ¿Cómo valoramos una secuencia de estados? Idea: a partir de las recompensas, pero *penalizando* el largo plazo
- Valoración mediante **recompensa con descuento**:

$$V([q_0 q_1 q_2 \dots]) = R(q_0) + \gamma R(q_1) + \gamma^2 R(q_2) + \dots$$

donde γ es el llamado **factor de descuento**

- En el ejemplo de la cuadrícula:
 - Con $\gamma = 0.8$, la secuencia de estados $(1, 1), (2, 1), (3, 1), (3, 2), (3, 3), (3, 4)$ tiene una valoración $-0.04 - 0.04 \cdot 0.8 - 0.04 \cdot 0.8^2 - 0.04 \cdot 0.8^3 - 0.04 \cdot 0.8^4 + 1 \cdot 0.8^5 = 0.193216$

◀ ▶ ⏪ ⏩ 🔍 🔄

Valoraciones de secuencias: observaciones

- Suponemos horizonte infinito
 - No hay un plazo fijo de terminación
 - Procesos estacionarios: la política óptima a partir de un momento sólo depende del estado en ese momento
- ¿Esto implica que las valoraciones pueden ser infinitas?
En general, no:
 - Si hay estados terminales, los asimilamos a estados a partir del cual las recompensas son cero.
 - Aún con posibles secuencias infinitas, si las recompensas están acotadas por una cantidad R_{max} y $\gamma < 1$, entonces la valoración de una secuencia no puede ser mayor de $R_{max}/(1 - \gamma)$ (¿por qué?)



Valoración de estados respecto de una política

- Dada una política π y un estado s , podemos valorar s respecto de π teniendo en cuenta la valoración de las secuencias de estados que se generan si se sigue dicha política a partir de s
- Ejemplo: si en la cuadrícula estamos en el estado $(1, 4)$ y aplicamos la política (a) del gráfico anterior, podríamos generar distintas secuencias, cada una con una probabilidad y una valoración. Entre otras:
 - $(1, 4), (1, 3), (1, 2), (1, 1), (2, 1), (3, 1), (3, 2), (3, 3), (3, 4)$, con probabilidad $0.8^8 = 0.168$ y valoración **0.0013** (siendo $\gamma = 0.8$ y $R = -0.04$)
 - $(1, 4), (1, 3), (2, 3), (3, 3), (3, 4)$ con probabilidad $0.8^3 \cdot 0.1 = 0.0512$ y valoración 0.291
 - $(1, 4), (1, 3), (2, 3), (2, 4)$ con probabilidad $0.8 \cdot 0.1^2 = 0.008$ y valoración -0.609
 - $(1, 4), (2, 4)$ con probabilidad 0.1 y valoración -0.84
 - ...



Valoración de estados respecto de una política

- Idea: valorar un estado s respecto de una política π como la media esperada (es decir, ponderada por su probabilidad) de las valoraciones de todas las secuencias que se podrían obtener.
 - En el ejemplo anterior:
 $0.168 \cdot 0.0013 + 0.0512 \cdot 0.291 - 0.008 \cdot 0.609 - 0.1 \cdot 0.84 + \dots$
- La valoración de un estado respecto de una política π la notamos por $V^\pi(s)$



Cálculo de valoración respecto de una política

- Afortunadamente, para obtener $V^\pi(s)$ *no necesitaremos calcular todas las posibles secuencias* que podrían iniciarse en s
- La clave está en la siguiente propiedad de V^π , que relaciona la valoración de cada estado con la de sus “vecinos”:

$$V^\pi(s) = R(s) + \gamma \cdot \sum_{s'} (P(s'|s, \pi(s)) \cdot V^\pi(s'))$$

- Calcular V^π es resolver ese sistema de ecuaciones lineales:
 - Las incógnitas son los $V^\pi(s)$, una por cada estado s
 - Hay tantas ecuaciones como estados



Cálculo de valoración respecto de una política (ejemplo)

En la política (a) de la cuadrícula en la figura anterior, éstas serían alguna de las ecuaciones que salen:

- $V^\pi(1, 1) = -0.04 + \gamma \cdot (0.8 \cdot V^\pi(2, 1) + 0.1 \cdot V^\pi(1, 1) + 0.1 \cdot V^\pi(1, 2))$
- $V^\pi(1, 2) = -0.04 + \gamma \cdot (0.8 \cdot V^\pi(1, 1) + 0.2 \cdot V^\pi(1, 2))$
- $V^\pi(1, 3) = -0.04 + \gamma \cdot (0.8 \cdot V^\pi(1, 2) + 0.1 \cdot V^\pi(1, 3) + 0.1 \cdot V^\pi(2, 3))$
- ...
- $V^\pi(3, 4) = +1$

Resolviendo este sistema, obtenemos V^π



Políticas óptimas y valoraciones de estados

- La **valoración de un estado** s , notada $V(s)$, se define como:

$$V(s) = \max_{\pi} V^\pi(s)$$

- Es decir, la mejor valoración que una política pueda conseguir a partir de dicho estado.
- Podemos además definir una **política óptima** π^* de la siguiente manera:

$$\pi^*(s) = \operatorname{argmax}_{a \in A(s)} \sum_{s'} (P(s'|s, a) \cdot V(s'))$$

- Es decir, en cada estado, aplicar la acción que lleve a la mejor valoración esperada en el estado siguiente
- Objetivo: calcular V y π^*



Los métodos principales para resolver este tipo de problemas son:

- **Iteración de valor (Bellman):**

$$V^*(s) = \max_a \{ R(s, a) + \gamma \sum_{s'} P(s'|s, a) V^*(s') \}$$

Cadenas de Markov

Modelos Ocultos de Markov

Procesos de Decisión de Markov

Ecuaciones de Bellman

- De manera análoga a V^π , podemos describir $V(s)$ en función de las valoraciones de los estados “vecinos”. Son las llamadas **ecuaciones de Bellman**:

$$V(s) = R(s) + \gamma \cdot \max_{a \in A(s)} \sum_{s'} (P(s'|s, a) \cdot V(s'))$$

- Nuevamente, es un sistemas de ecuaciones (una ecuación por estado).
- La solución a este sistema de ecuaciones nos da la valoración de cada estado y a partir de ésta, la política óptima

◀ ▶ ⏪ ⏩ ⏴ ⏵ ⏶ ⏷ ⏸ ⏹ ⏺ ⏻ ⏼ ⏽ ⏾ ⏿ 🔍 ↻

Cadenas de Markov

Modelos Ocultos de Markov

Procesos de Decisión de Markov

Ecuaciones de Bellman (ejemplo)

- Por ejemplo, en la cuadrícula, la ecuación correspondiente al estado de la casilla (1, 1) es:

$$V(1, 1) = -0.04 + \gamma \cdot \max \left[\begin{aligned} &0.8V(1, 2) + 0.1V(2, 1) + 0.1V(1, 1), \\ &0.9V(1, 1) + 0.1V(1, 2), \\ &0.9V(1, 1) + 0.1V(2, 1), \\ &0.8V(2, 1) + 0.1V(1, 2) + 0.1V(1, 1) \end{aligned} \right]$$

- Las demás ecuaciones, son similares, una por cada estado

◀ ▶ ⏪ ⏩ ⏴ ⏵ ⏶ ⏷ ⏸ ⏹ ⏺ ⏻ ⏼ ⏽ ⏾ ⏿ 🔍 ↻

Ecuaciones de Bellman (ejemplo)

La siguiente figura muestra la valoración de cada estado en el problema de la cuadrícula (para $\gamma = 1$, y $R(s) = -0.04$), obtenida solucionando las correspondientes ecuaciones de Bellman.

3	0.812	0.868	0.918	+1
2	0.762		0.660	-1
1	0.705	0.655	0.611	0.388
	1	2	3	4

◀ ▶ ⏪ ⏩ ⏴ ⏵ ⏶ ⏷ ⏸ ⏹ ⏺ ⏻ ⏼ ⏽ ⏾ ⏿ 🔍 ↻

Cálculo de política óptima (ejemplo)

A partir de la valoración anterior, podemos calcular la política óptima, usando la fórmula

$\pi^*(s) = \underset{a \in A(s)}{\operatorname{argmax}} \sum_{s'} (P(s'|s, a) \cdot V(s'))$. Por ejemplo, en el estado (1,1):

- Acción *arriba*:
 $0.8 \cdot 0.762 + 0.1 \cdot 0.705 + 0.1 \cdot 0.655 = 0.7456$
- Acción *abajo*: $0.8 \cdot 0.705 + 0.1 \cdot 0.705 + 0.1 \cdot 0.655 = 0.7$
- Acción *izquierda*:
 $0.8 \cdot 0.705 + 0.1 \cdot 0.705 + 0.1 \cdot 0.762 = 0.7107$
- Acción *derecha*:
 $0.8 \cdot 0.655 + 0.1 \cdot 0.705 + 0.1 \cdot 0.762 = 0.6707$

Luego la acción óptima en el estado (1, 1) es moverse hacia *arriba*. El resto se muestra en la figura anterior (a)

◀ ▶ ⏪ ⏩ ⏴ ⏵ ⏶ ⏷ ⏸ ⏹ ⏺ ⏻ ⏼ ⏽ ⏾ ⏿ 🔍 ↻

Iteración de valores

- ¿Cómo resolvemos las ecuaciones de Bellman?
 - Cada $V(s)$ es una incognita en el sistema de ecuaciones.
 - Hay tantas ecuaciones como estados
 - Problema: *no es un sistema lineal* (debido al *max*)
- Se aplica un método *iterativo*
 - Comenzamos con un valor arbitrario $V_0(s)$, para cada estado s
 - En cada iteración se aplican las ecuaciones de Bellman para *actualizar* los valores que se tienen hasta el momento:

$$V_{i+1}(s) \leftarrow R(s) + \gamma \cdot \max_{a \in A(s)} \sum_{s'} (P(s'|s, a) \cdot V_i(s'))$$



Iteración de valores: propiedades

- Se puede demostrar que los valores que se van calculando en las diferentes iteraciones convergen asintóticamente hacia la solución (única si $\gamma < 1$) de las ecuaciones de Bellman
- La convergencia es rápida para valores de γ pequeños
- Además, podemos acotar con bastante precisión el error que cometemos . Más precisamente:
 - Sea $\|V_{i+1} - V_i\| = \max_s |V_{i+1}(s) - V_i(s)|$
 - Se tiene que si $\|V_{i+1} - V_i\| < \epsilon \cdot (1 - \gamma)/\gamma$, entonces $\|V_{i+1} - V\| < \epsilon$, donde V es la solución exacta a las ecuaciones de Bellman
 - Éste será el criterio de parada en las iteraciones



Algoritmo de iteración de valores

- **Entrada:**
 - Un proceso de decisión de Markov: conjunto de estados S , $A(s)$, modelo de transición $P(s'|s, a)$, recompensa $R(s)$ y descuento γ
 - $\epsilon > 0$, cota de error máximo permitido
- **Salida:**
 - Una valoración $V(s)$ para cada estado
- **Procedimiento:**
 - Inicio: Sean V, V' funciones sobre los estados, inicialmente ambas con valor 0 para cada estado
 - Repetir:
 - Hacer V' igual a V y $\delta = 0$
 - Para cada $s \in S$ hacer:
 - $V(s) \leftarrow R(s) + \gamma \cdot \max_{a \in A(s)} \sum_{s'} (P(s'|s, a) \cdot V'(s'))$
 - Si $|V(s) - V'(s)| > \delta$ entonces hacer $\delta = |V(s) - V'(s)|$
 - hasta que $\delta < \epsilon \cdot (1 - \gamma) / \gamma$
 - Devolver V



- **Iteración de política (Howards):**

$$\pi^*(s_i) = \arg \max_a \{ R(s, a) + \gamma \sum_{s'} P(s'|s, a) V^*(s') \}$$

siendo γ un factor de descuento.

Iteración de políticas

- Existe una manera alternativa de obtener V , comenzando con una política inicial π_0 cualquiera:
 1. Calcular V^{π_i}
 2. A partir de V^{π_i} , calcular una nueva política π_{i+1} que en cada estado recomiende la acción que mejor valoración espera, respecto de V^{π_i}
- Los dos pasos anteriores se iteran hasta que π_j y π_{j+1} coinciden, en cuyo caso hemos conseguido la política óptima, y su valoración asociada es precisamente V .



Iteración de políticas: observaciones

- El proceso anterior termina (pues hay un número finito de políticas distintas) y devuelve la valoración buscada (ya que se llega a una solución de las ecuaciones de Bellman).
- ¿Cómo calcular V^{π_j} en el paso 1? Dos opciones:
 - Como ya hemos visto, resolviendo un sistema de ecuaciones lineal (diapositivas 58 y 59).
 - Aplicando método iterativo similar al que se usa con las ecuaciones de Bellman (pero ahora más fácil, ya que no hay máximos):

$$V_{j+1}^{\pi_j}(s) \leftarrow R(s) + \gamma \cdot \sum_{s'} (P(s'|s, \pi_j(s)) \cdot V_j^{\pi_j}(s'))$$

- ¿Cómo calcular π_{j+1} en el paso 2? Ya lo hemos visto:

$$\pi_{j+1}(s) = \operatorname{argmax}_{a \in A(s)} \sum_{s'} (P(s'|s, a) \cdot V_j^{\pi_j}(s'))$$



Algoritmo de iteración de políticas

- **Entrada:**
 - Un proceso de decisión de Markov: conjunto de estados S , $A(s)$, modelo de transición $P(s'|s, a)$, recompensa $R(s)$ y descuento γ
 - Un número k de iteraciones
- **Salida:** Una política óptima π y su valoración asociada
- **Procedimiento:**
 - Inicio: π una política aleatoria, asignando una acción a cada estado
 - Repetir
 - Calcular V^π (iterando k veces la actualización descrita en la diapositiva anterior)
 - Hacer *actualizada* igual a *Falso*
 - Para cada estado s :
 - Si $\max_{a \in A(s)} \sum_{s'} (P(s'|s, a) \cdot V^\pi(s')) > \sum_{s'} (P(s'|s, \pi(s)) \cdot V^\pi(s'))$, entonces hacer $\pi(s)$ igual a $\operatorname{argmax}_{a \in A(s)} \sum_{s'} (P(s'|s, a) \cdot V^\pi(s'))$ y *actualizada* igual *Verdad*
 - Devolver π y V^π



	Estado observable	Estado oculto (observaciones)
No hay acciones (sólo transiciones estocásticas)	Cadenas de Markov (Markov Chains)	Modelos Ocultos de Markov (HMMs)
Hay acciones (que producen transiciones estocásticas)	Procesos de Decisión de Markov (MDPs)	Procesos de Decisión de Markov Parcialmente Observables (POMDPs)

En un **proceso de decisión de Markov parcialmente observable**, el estado es parcialmente observable (no se conoce, pero se dispone de observaciones), existen diferentes acciones (un modelo de transición para cada acción) y un modelo de recompensa. Es una generalización de un proceso de decisión de Markov.

Una solución exacta a un problema de este tipo produce la acción óptima que maximiza o minimiza la recompensa esperada o coste del agente en un horizonte posiblemente infinito. La secuencia de acciones óptimas se conoce como *política óptima del agente* para interactuar con su entorno.

La nomenclatura usual es la siguiente:

- S es un conjunto de estados.
- A es un conjunto de acciones.
- T es un conjunto de probabilidades condicionales de transición entre estados.
- $R = S \times A \rightarrow$ Conjunto de los números reales es la función de recompensa.
- Ω es un conjunto de observaciones.
- O es un conjunto de probabilidades condicionales de observación. $P(O|S)$.
- $\gamma \in [0, 1]$ es el factor de descuento.
- Una distribución de probabilidad inicial para los estados $P(S)$.

En cada periodo de tiempo, el medio ambiente se encuentra en un estado. El sistema realiza una acción y se produce una transición a otro estado. Al mismo tiempo el sistema realiza una observación. Por último, el agente recibe una recompensa. El objetivo es que el agente debe elegir las acciones en cada paso de tiempo que maximice su recompensa futura. El factor de descuento favorece más las recompensas cercanas que las lejanas.

El enfoque exacto para resolver un problema de este tipo requiere considerar toda la historia de observaciones y acciones. Lo que es equivalente a considerar la distribución de probabilidad sobre los estados y, según esta, determinar las decisiones óptimas.

El problema es que el espacio de estados se vuelve infinito y la solución exacta es muy compleja. Por ello se buscan soluciones aproximadas. Para ello, se consideran un número finito de pasos y se modela el problema como una red de decisión dinámica, la aproximación depende del número de estados que se ven hacia adelante.

Como ejemplo considérese otra vez el mundo de 4x3 celdas, pero ahora supóngase que el agente no tiene sensores de ninguna clase y que no tiene ni idea de dónde se encuentra.

Supóngase que cualquiera de los nueve estados terminales tiene la misma probabilidad de ser el estado inicial. ¿Qué hará el agente? Una respuesta posible es que el agente, en un primer momento, actuaría con el fin de reducir la incertidumbre y solo entonces trataría de dirigirse hacia la salida +1.

Un estado de creencia b es una distribución de probabilidad sobre todos los posibles estados, por ejemplo $[1/9, 1/9, \dots, 1/9, 0, 0]$.

Mediante $b(s)$ se denotará la probabilidad asignada al estado actual (s) por el estado de creencia b . El sistema puede calcular su estado de creencia actual como la distribución de probabilidad condicionada sobre los estados actuales, conocida la secuencia de observaciones y acciones realizadas hasta ahora.

$$b'(s') = \alpha O(s', o) \sum_s T(s, a, s') b(s)$$

Donde α es una constante de normalización que garantiza que la suma de los componentes del estado de creencia suma 1.

La idea fundamental, necesaria para entender los procesos de decisión de Markov parcialmente observables, es que: la acción óptima depende exclusivamente del estado de creencia actual del agente. Es decir, la política óptima puede describirse mediante una correspondencia $\pi^*(b)$ que asocie estados de creencias con acciones. Por lo tanto, no depende del estado actual en el que se encuentre el agente. Esto es una buena cosa, ya que el agente no sabe cuál es su estado actual y todo lo que conoce es su estado de creencia. Por lo tanto, el ciclo de decisión de un sistema PDMPO es el siguiente:

- Dado el estado de creencia actual b , ejecutar la acción $a = \pi^*(b)$.
- Recibir la observación « o ».
- Establecer el estado de creencia actual con el valor devuelto por

$$b'(s') = \alpha O(s', o) \sum_s T(s, a, s') b(s)$$

y volver al paso 1.

Los detalles del algoritmo se escapan al ámbito de este documento. Los PDMPO con apenas una docena de estados son a menudo intratables.

Los caminos que se pueden seguir se basan en la utilización de redes bayesianas dinámicas.

A pesar de todo lo que se ha visto, no siempre se puede resolver la incertidumbre mediante la reconstrucción de un modelo interno del mundo. Los efectos de las respuestas a la incertidumbre no siempre pueden resolverse mediante una actualización bayesiana exitosa. Ahí radica la amenaza de la incertidumbre.

Algoritmos de aprendizaje

Véase <https://es.wikipedia.org/wiki/Aprendizaje_autom%C3%A1tico>

El aprendizaje automático o aprendizaje de máquinas (Machine Learning en inglés) es una rama de la inteligencia artificial cuyo objetivo es desarrollar técnicas que permitan que las computadoras aprendan.

Se dice que un sistema aprende cuando su forma de resolver problemas mejora con la experiencia; es decir, cuando la habilidad no estaba presente en su nacimiento. De forma más concreta, los investigadores del aprendizaje de máquinas buscan algoritmos y heurísticas para convertir muestras de datos en programas de computadora, sin tener que escribirlos explícitamente.

Se dice que un programa de computadora aprende de una experiencia E , con respecto a alguna tarea T y alguna medida de la calidad del resultado P , si su comportamiento sobre T , tal y como lo mide P , aumenta con la experiencia E (Tom Mitchell, Carnegie Mellon University).

En muchas ocasiones el campo de actuación del aprendizaje automático se solapa con el de la estadística inferencial, ya que las dos disciplinas se basan en el análisis de datos. Sin embargo, el aprendizaje automático incorpora las preocupaciones de la complejidad computacional de los problemas. Muchos problemas son de clase NP-hard, por lo que gran parte de la investigación realizada en aprendizaje automático está enfocada al diseño de soluciones factibles a esos problemas.

El aprendizaje automático es un proceso de inducción del conocimiento. El quid de la cuestión es definir de forma clara el objetivo.

- Cuando se busca clasificar, el resultado es una clase, entre un número limitado de clases. Con clases hacemos referencia a categorías arbitrarias según el tipo de problema. Muchos de los algoritmos de este tipo dan los resultados de clasificación con probabilidades.
- Cuando se busca como resultado un número dentro de un conjunto infinito de posibles resultados, se denomina *algoritmo de regresión*. Existe una técnica denominada de *regresión logística*, pero puede inducirnos a pensar que puede usarse en problemas de regresión, sin embargo, solo funciona para problemas de clasificación.
- Aunque hay algunas técnicas que son específicas de clasificación y otras de regresión, la mayoría de las técnicas funcionan con ambos.

Cada modelo de aprendizaje automático intenta resolver un problema con un objetivo diferente utilizando un conjunto de datos diferente y, por lo tanto, es

importante comprender el contexto y seleccionar la métrica que nos indica la calidad de los resultados concretos obtenidos. Existen métricas para los diferentes algoritmos. Para saber más véanse:

<<https://sitiobigdata.com/2019/05/27/aprendizaje-automatico-seleccionando-metricas-regresion/>>

<<https://sitiobigdata.com/2019/05/27/modelos-de-machine-learning-metricas-de-regresion-mse-parte-2/>>

<<https://sitiobigdata.com/2019/01/19/machine-learning-metrica-clasificacion-parte-3/>>

Ejemplos:

- Problemas de clasificación, ejemplo detección de correo basura o *spam*.
- Predecir una categoría de destino:
 - Clasificación binaria, se dividen los datos en dos categorías.
 - Clasificación multiclase, se dividen los datos en tres o más categorías.
- Búsqueda de datos inusuales, estos algoritmos detectan anomalías identificando datos que están fuera de los parámetros definidos como «lo normal».
- Predicción de valores, estos algoritmos de regresión predicen el valor de un nuevo dato en función de datos históricos.
- Detección de similitudes, estos algoritmos de *clustering* dividen los datos en varios grupos determinando el nivel de similitud entre los datos.
- ...

En muchas áreas del saber, el conocimiento se ha venido obteniendo por el clásico método hipotético-deductivo de la ciencia positiva. En él es fundamental el paso inductivo inicial: a partir de un conjunto de observaciones y de unos conocimientos previos, la intuición del investigador le conduce a formular la hipótesis. Esta «intuición» resulta inoperante cuando no se trata de observaciones aisladas y casuales, sino de millones de datos almacenados en soporte informático. En el fondo de todas las investigaciones sobre inducción en bases de datos subyace la idea de automatizar ese paso inductivo.

En general, el aprendizaje puede plantearse del modo siguiente. Dada una determinada tarea por resolver, y una clase de funciones F , el aprendizaje consiste en utilizar un conjunto de observaciones para encontrar una función $f^* \in F$ de manera que esta resuelve el problema planteado por la tarea de alguna forma óptima.

La exigencia de optimalidad requiere la definición de una función coste $C: F \rightarrow \mathbb{R}$ tal que, para la solución óptima f^* , verifica que su coste es el menor que el coste de ninguna otra solución $f \in F$.

Esa función de coste es muy importante en el aprendizaje, ya que en cada momento la bondad de una solución particular f en relación con la tarea por resolver.

El campo del aprendizaje automático depende de la teoría de probabilidades y estadística, con un uso creciente de la probabilidad bayesiana. Algunos aprendizajes usan redes neuronales, otros IA simbólica complementada con potentes algoritmos estadísticos.

Tipos de algoritmos de aprendizaje

El aprendizaje automático tiene tres grandes categorías: aprendizaje supervisado, aprendizaje no supervisado y aprendizaje por refuerzo.

- En el supervisado, el programador «entrena».
- En el no supervisado, los programadores no necesitan saber qué patrones / agrupaciones existen en los datos, el sistema los encuentra por sí mismo.
- En el aprendizaje por refuerzo se rige por principios análogos al castigo y la recompensa: mensajes de notificación que le dicen al sistema si lo que acaba de hacer está bien o mal, o se categoriza numéricamente el resultado.

Supervisado

En este tipo de aprendizaje, se dispone previamente de una serie de pares (x, y) (datos, categoría) de ejemplos que permiten definir una función $f: X \rightarrow Y$, que relaciona las X con las Y . La función de coste está relacionada con la falta de coincidencia entre nuestro mapeo y los datos, y contiene implícitamente el conocimiento previo adquirido a partir de los ejemplos sobre el dominio del problema.

El algoritmo produce una función que establece una correspondencia entre las entradas y las salidas deseadas del sistema. Se caracteriza por contar con información que especifica qué conjuntos de datos son satisfactorios para el objetivo del aprendizaje.

Un ejemplo de este tipo de algoritmo es el problema de clasificación, donde el sistema de aprendizaje trata de etiquetar (clasificar) una serie de vectores utilizando una entre varias categorías (clases). La base de conocimiento del sistema está formada por ejemplos de etiquetados anteriores. Este tipo de aprendizaje puede llegar a ser muy útil en problemas de investigación biológica, biología computacional y bioinformática. Por ejemplo, un *software* podría llegar a reconocer si una imagen dada es o no la imagen de un rostro.

Se suele usar en problemas de clasificación, también conocidos como *problemas de reconocimiento de patrones* en los (la variable objetivo, es de tipo categórico) o de regresión, también conocidos como *aproximación de funciones* (la variable es de tipo numérico).

Los algoritmos más habituales que utilizan el aprendizaje supervisado son:

- Árboles de decisión:
 - Este tipo de aprendizaje usa un árbol de decisiones como modelo predictivo. Se mapean observaciones sobre un objeto con conclusiones sobre el valor final de dicho objeto.
 - Los árboles son estructuras básicas en la informática. Los árboles de atributos son la base de las decisiones.
 - Los algoritmos más usados son: árboles de clasificación y regresión (CART), árbol condicional, random forest.

- Redes bayesianas:
 - Una red bayesiana es un modelo probabilístico que representa una serie de variables de azar y sus independencias condicionales a través de un grafo acíclico dirigido.
 - Los algoritmos más utilizados son: Naïve Bayes, Gaussian Naive Bayes, Multinomial Naive Bayes, Bayesian Network.

- Algoritmos de regresión:
 - En las tareas de regresión, el programa de aprendizaje automático debe estimar y comprender las relaciones entre las variables utilizando una medida de error que se intentará minimizar en un proceso iterativo para poder realizar predicciones. Los hay lineal, mínimos cuadrados, logística.

- Máquinas de vectores de soporte (Support vector Machines [SVM]).
 - Las MVS son una serie de métodos de aprendizaje supervisado usados para clasificación y regresión. Los algoritmos de MVS usan un conjunto de ejemplos de entrenamiento clasificado en dos categorías para construir un modelo que prediga si un nuevo ejemplo pertenece a una u otra de dichas categorías. Trabajan con un hiperplano de dimensión (N-1) siendo N la dimensión de los atributos, que separa los puntos.

- Métodos de «ensemble» (conjuntos de clasificadores).

- Aprendizaje inductivo. Espacio de versiones. Es un método basado en la lógica que pretende encontrar el subconjunto de todas las hipótesis que son consistentes con los ejemplos de entrenamiento.

No supervisado

En el aprendizaje no supervisado, se dan algunos datos (x) y la función de coste que depende de la tarea y del conocimiento *a priori* que se tenga de la misma.

Todo el proceso de modelado se lleva a cabo sobre un conjunto de ejemplos formado tan solo por entradas al sistema. No se tiene información sobre las categorías de esos ejemplos. Por lo tanto, en este caso, el sistema tiene que ser capaz de reconocer patrones para poder etiquetar las nuevas entradas. Por ello, tienen un carácter exploratorio.

Un *software* no sería capaz de decirnos si una imagen dada es un rostro o no, pero sí podría, por ejemplo, clasificar las imágenes entre aquellas que contienen rostros humanos, de animales o las que no contienen. La información obtenida por un algoritmo de este tipo debe ser interpretada por una persona para darle utilidad.

Las principales aplicaciones del aprendizaje no supervisado son:

- Segmentación de conjuntos de datos por atributos compartidos.
- Detección de anomalías que no encajan en ningún grupo.
- Simplificación de conjuntos de datos agregando variables con atributos similares.
- Estimación de distribuciones estadísticas.
- Compresión de datos.
- ...

En resumen, el objetivo principal es estudiar la estructura intrínseca y normalmente oculta de los datos.

Este tipo de aprendizaje se suele usar en problemas de *clustering* y algoritmos de reducción de dimensión.

Los algoritmos más habituales que utilizan el aprendizaje no supervisado son:

- Algoritmos de *clustering* o algoritmos de agrupamiento:
 - El análisis por agrupamiento (*clustering* en inglés) es la clasificación de observaciones en subgrupos —*clusters*— para que las observaciones en cada grupo se asemejen entre sí según ciertos criterios.
 - Las técnicas de agrupamiento hacen inferencias diferentes sobre la estructura de los datos; se guían usualmente por una medida de similitud

- específica y por un nivel de compactamiento interno (similitud entre los miembros de un grupo) y la separación entre los diferentes grupos.
- El agrupamiento es un método de aprendizaje no supervisado y es una técnica muy popular de análisis estadístico de datos.
 - Los algoritmos más usados: K-Medias, clusterización jerárquica, agrupación espacial basada en la densidad del ruido, agrupamiento gaussiano.
- Algoritmos de reducción de dimensión:
 - Buscan explorar la estructura existente de manera no supervisada para simplificar los datos y reducirlos o comprimirlos. Suelen ser útiles para simplificar el conjunto de variables que luego pueda usar un algoritmo supervisado.
 - Los algoritmos más utilizados son:
 - Análisis de componentes principales (PCA): Principal Componente Analysis (PCA). Es un procedimiento estadístico que usa una transformación ortogonal para convertir un conjunto de observaciones de variables posiblemente correlacionadas en un conjunto de valores linealmente no correlacionadas llamadas *componentes principales*.
 - Análisis de componentes independientes (ICA). Es una técnica estadística que permite revelar los factores ocultos que subyacen a conjuntos de variables aleatorias.
 - Incrustación estocástica de vecinos (SNE) y su corrección t-SNE.
 - ...

Por refuerzo

No todos los algoritmos de aprendizaje se pueden clasificar de alguna de las dos maneras previas. Hay una tierra de nadie que es donde encajan las técnicas de aprendizaje por refuerzo.

En este tipo de aprendizaje, los datos (X) por lo general no se dan, pero se pueden ir obteniendo de la interacción del sistema con el medio ambiente. En cada instante de tiempo (t), el agente realiza una acción (y_t) y el medio ambiente genera una observación (x_t) y un coste instantáneo (c_t). El objetivo es descubrir una política para la selección de las acciones que minimice una cierta medida de un costo a largo plazo. La dinámica del entorno y el coste a largo plazo para cada política general son desconocidos, pero pueden ser estimados.

El algoritmo aprende observando el mundo que le rodea. Su información de entrada es el *feedback* o retroalimentación que obtiene del mundo exterior como respuesta a sus acciones. Por lo tanto, el sistema aprende a base de ensayo-error.

El aprendizaje por refuerzo es el más general entre las tres categorías. En vez de que un instructor indique al sistema qué hacer, el agente inteligente debe aprender cómo se comporta el entorno mediante recompensas (refuerzos) o castigos, derivados del éxito o del fracaso, respectivamente. El objetivo principal es aprender la función de valor que le ayude al agente inteligente a maximizar la señal de recompensa y así optimizar sus políticas de modo a comprender el comportamiento del entorno y a tomar buenas decisiones para el logro de sus objetivos formales.

Los principales algoritmos de aprendizaje por refuerzo se desarrollan dentro de los métodos de resolución de problemas de decisión finitos de Markov, que incorporan las ecuaciones de Bellman y las funciones de valor. Los tres métodos principales son: la programación dinámica, los métodos de Monte Carlo y el aprendizaje de diferencias temporales.

Las principales aplicaciones del aprendizaje por refuerzo son:

- Problemas de control.
- Juegos.
- Tareas secuenciales.
- ...

Otras técnicas

- Algoritmos genéticos.

Los algoritmos genéticos son procesos de búsqueda heurística que simulan la selección natural. Usan métodos tales como la mutación y el cruzamiento para generar nuevas clases que puedan ofrecer una buena solución a un problema dado.

- Redes neuronales artificiales.

Las redes de neuronas artificiales (RNA) son un paradigma de aprendizaje automático inspirado en las neuronas de los sistemas nerviosos de los animales. Se trata de un sistema de enlaces de neuronas que colaboran entre sí para producir un estímulo de salida. Las conexiones tienen pesos numéricos que se adaptan según la experiencia. De esta manera, las redes neuronales se adaptan a un impulso y son capaces de aprender. La importancia de las redes neuronales cayó durante un tiempo con el desarrollo de los vectores de soporte y clasificadores lineales, pero volvió a surgir a finales de la década de 2000 con la llegada del aprendizaje profundo.

Los algoritmos más usados son: Compuerta XOR, Perceptron, Back-Propagation, Hopfield Network, Multi layered Perceptron (MLP)

- Algoritmos de aprendizaje profundo.

Son la evolución de las redes neuronales artificiales. De hecho, son redes neuronales interconectadas en diversas capas que pueden ejecutarse en paralelo.

Los algoritmos más usados son: Convolutional Neural Networks, Long Short Memory Neuronal Networks.

Problemas de los algoritmos de aprendizaje

Overfitting y underfitting

Las principales causas del mal funcionamiento de los algoritmos del aprendizaje de máquina son el *overfitting* o el *underfitting* de los datos.

Las máquinas que aprenden deben intentar ser capaces de generalizar conceptos. Es decir, cuando se entrena a un algoritmo con un conjunto de datos, si este es capaz de generalizar, cuando se le presenta un conjunto de datos desconocido, este debe ser capaz de sintetizarlo, comprenderlo y devolvernos un resultado fiable.

La clave para conseguirlo es el conjunto de datos de entrenamiento. Si los datos son pocos, la máquina no será capaz de generalizar el conocimiento y estará incurriendo en *underfitting*. La razón es que no es capaz de alcanzar un conocimiento sólido. Tiene que ver con el bias (sesgo) estadístico. Resumiendo:

- Se produce error durante el entrenamiento.
- Se produce error durante las pruebas.
- Se necesita mejorar el aprendizaje.

Las mejoras evidentes para disminuir el bias alto pasarían por:

- Agregar nuevos atributos.
- Usar algoritmos más complejos.
- Aumentar parámetros en el modelo del algoritmo utilizado.

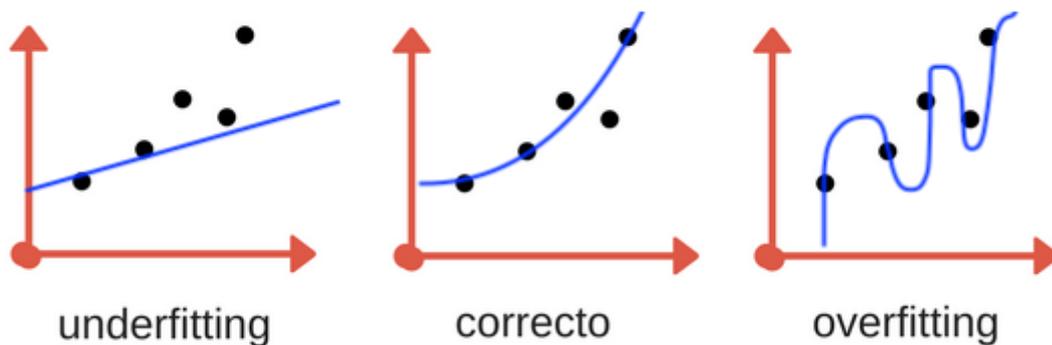
Por el contrario, si se entrena con un exceso de datos, puede que el conocimiento adquirido sea muy riguroso y un nuevo conjunto de datos sea incapaz de reconocerlo por no cumplir exactamente con las características de lo aprendido. En este caso se trata de un problema de *overfitting*. Tiene que ver con la varianza estadística.

Resumiendo:

- Se produce un bajo error durante el entrenamiento.
- Se produce un alto error durante las pruebas.
- Se necesita mejorar el aprendizaje.

Las mejoras evidentes para disminuir la varianza alta pasarían por:

- Que los datos tengan clases variadas y equilibradas en cantidad.
- Eliminar atributos.
- Obtener más ejemplos.
- Disminuir el número de parámetros en el modelo del algoritmo utilizado.



Se aconseja para alcanzar el equilibrio en el aprendizaje subdividir el conjunto de datos de entrada en dos: uno para entrenamiento y otro para comprobar. Esta división suele hacerse el 80 % para entrenar y el 20 % para comprobar.

Otra técnica un poco más elaborada consiste en trabajar con tres conjuntos de datos. Uno para entrenamiento (el 60 %), otro para comprobación (el 20 %) y el resto (20 %) para comprobar el error de generalización.

Además, en muchas técnicas de aprendizaje automático, el aprendizaje consiste en encontrar los coeficientes que minimizan una cierta función. La técnica de regularización consiste en añadir una penalización a dicha función, de manera que se generen modelos que generalizan mejor.

Ejemplo: Más adelante nos surgirá la siguiente función que habrá que minimizar:

$$J(a, b) = \frac{1}{2m} \sum_{i=1}^m [a x_i + b - y(x_i)]^2$$

La regularización consistiría en cambiar dicha expresión por la siguiente

$$J(a, b) = \frac{1}{2m} \sum_{i=1}^m [a x_i + b - y(x_i)]^2 + \lambda a \text{ con } \lambda \in [0, \infty]$$

- Si $\lambda = 0$ no hay regularización, y si alcanza valores bajos disminuye el bias.
- Si $\lambda = \infty$ hay mucha regularización, y si alcanza valores altos disminuye la varianza.

Para saber más:

<<https://iartificial.net/regularizacion-lasso-l1-ridge-l2-y-elasticnet/#:~:text=En%20muchas%20t%C3%A9cnicas%20de%20aprendizaje,m%C3%A1s%20simples%20que%20generalizan%20mejor>>

Sesgos

Los algoritmos de aprendizaje automático a menudo pueden verse afectados por el sesgo que puedan tener los datos. Por ejemplo, no se podrán clasificar todas aquellas entradas de las que no se haya recibido ninguna información en la fase de entrenamiento. De hecho, cuando el entrenamiento se realiza con datos clasificados por el ser humano, el aprendizaje automático tiende a crear los mismos sesgos que hay en la sociedad.

Algunos ejemplos de esto son cuando en 2015 el algoritmo de Google photos identificaba algunas personas negras con gorilas, o en 2016 cuando el *bot* de Twitter de Microsoft desarrolló comportamientos racistas y machistas a base de observar el tráfico de datos en dicha red social. Por este motivo, en los últimos años ha habido una tendencia a desarrollar métodos para aumentar la equidad, es decir, para reducir el sesgo en este tipo de algoritmos por parte de los expertos en IA.

Problema de la cola larga

La larga estela o larga cola fue una expresión popularizada por Chris Anderson en un artículo de la revista *Wired* de octubre de 2004. Y hace referencia al hecho de que existen distribuciones de frecuencia de aparición de un suceso o probabilidad de aparición de un suceso como la que se muestra en la gráfica



La larga estela, en color amarillo, puede comprender un área incluso mayor que la de la primera parte de la función. Gráfica de User:Husky - Trabajo propio, Dominio público, <<https://commons.wikimedia.org/w/index.php?curid=1449504>>

La zona verde muestra los sucesos que tienen una gran frecuencia de aparición, que viene seguida por la zona amarilla que muestra los sucesos que tienen una baja frecuencia de aparición. En muchos casos, los acontecimientos de baja frecuencia o escasa amplitud —la larga cola— pueden abarcar la mayor parte del gráfico.

Prácticamente en todos los dominios de la IA aparece dicha distribución, lo que hace que aquellos algoritmos que se basan en la probabilidad de aparición no tengan en cuenta esa larga cola y los sistemas de aprendizaje basados en ellos fallen con bastante facilidad al generalizar el conocimiento aprendido.

Problema de la caja negra

Hubo un momento en que podíamos rastrear o entender las decisiones que una máquina tomaba de acuerdo con algún tipo de algoritmo. Sin embargo, los algoritmos que se han desarrollado para lo que se conoce como *aprendizaje automático* no permiten realizar esa tarea. Los algoritmos se entrenan para que funcionen, pero dichos algoritmos son una gran masa de números de los que no se pueden extraer las razones lógicas de su funcionamiento. La realidad es que ese es el motivo de que generen desconfianza, ya que en realidad son un sistema de caja negra. Para que, con este tipo de algoritmos, los humanos trabajen de manera efectiva en cooperación con robots y sistemas impulsados por la IA, es esencial que se establezca la confianza. Pero ¿cómo puede existir la confianza si esa IA toma una decisión que los humanos no comprenden?

En respuesta al problema de la caja negra, el hecho de que no siempre sabremos cómo la IA deriva su conclusión, ¿es posible limitar el sesgo al que está sujeta la IA al exponerla a una gran variedad de conjuntos de datos? ¿Sería mejor entrenar a la IA con datos multinacionales y multiculturales para evitar que nuestros propios sesgos humanos influyan en sus decisiones? ¿Es posible establecer normas sobre cómo debe ser entrenada la IA? En resumen, ¿cómo se introduce la confianza con el uso de ese tipo de algoritmos?

Mientras tanto, lo lógico, lo ético, lo de sentido común es que mientras los sistemas basados en la IA sean tan frágiles y poco fiables, habría que tomar una actitud de prudencia.

Todo ello no significa que haya que abandonar ninguna línea de investigación, tan solo cuando se aplique conocer tanto sus pros como sus contras. En su avance, en los últimos años se han propuesto integrar capacidad de razonamiento y aprendizaje, lo que no es más que la búsqueda de la respuesta a una pregunta clave en toda actividad científica: ¿Por qué?

Problema del sentido común

Los conocimientos que el ser humano posee le vienen de nuestra experiencia en el mundo; nos permiten comportarnos inteligentemente en situaciones muy diversas. Ese conocimiento que todos los humanos tenemos sobre nuestro mundo se denomina *sentido común* y es imprescindible para alcanzar un comportamiento inteligente de tipo general. Es un conocimiento muy basto, que requiere diferentes tipos de inferencia para su uso y que se resiste a la formalización.

Se sabe que las técnicas de aprendizaje detectan correlaciones, pero no relaciones de causa y efecto. Por ejemplo, no son capaces de aprender que es la

Tierra la que gira alrededor del Sol y no al revés. En estos momentos hacemos máquinas con habilidades para hacer cosas, pero sin la capacidad de comprender.

Para saber más: Ramón López de Mántaras, «El traje nuevo de la inteligencia artificial», *Investigación y Ciencia*, 526 (julio 2020), pp. 50-59.

Tipos de algoritmos de aprendizaje

Algoritmos de aprendizaje supervisado (modelo simbólico)

Árbol de decisión en aprendizaje automático

Los árboles de decisión son representaciones gráficas de posibles decisiones y además pueden realizar tareas de clasificación (cuando los datos son discretos) o de regresión (cuando los datos son continuos).

Tal y como ya vimos en un capítulo previo, los árboles de decisión tienen un primer nodo llamado *raíz* y luego se descomponen el resto de atributos en ramas. Cada nodo interno es una pregunta sobre alguna característica y cada nodo hoja es una clase.

Al estar hablando en estos momentos de aprendizaje automático, va a ser el algoritmo quien analizando los datos y las salidas (recuerde que es supervisado) decidirá la mejor forma de hacer las divisiones.

La nomenclatura suele ser la siguiente:

- E , conjunto de ejemplos. E_x , ejemplo x .
- A , conjunto de atributos. A_y , atributo y .
- V_y , conjunto de valores del atributo y .
- C , conjunto de clases (etiquetas).

Pensemos que tenemos 10 atributos de entrada cada uno con 2 o más valores posibles, las combinaciones para decidir el mejor árbol serían muchísimas... por lo que ya no es un trabajo para hacerlo manualmente. Y ahí es donde este algoritmo cobra importancia, pues él nos devolverá el árbol óptimo para poder tomar la decisión más acertada desde un punto de vista probabilístico.

Para obtener el árbol óptimo y valorar cada subdivisión entre todos los árboles posibles y conseguir el nodo raíz y los subsiguientes, el algoritmo deberá medir de alguna manera las predicciones logradas y valorarlas para comparar entre todas y obtener la mejor. Es decir, hay que elegir el atributo que nos indique el procedimiento a seguir.

Para medir y valorar, existen numerosas funciones, siendo las más conocidas y usadas los «índice gini» y la «ganancia de información». Sin entrar en muchos detalles:

- El índice gini (se utiliza con valores discretos y continuos) mide cuán desordenados o mezclados quedan los nodos una vez que se han dividido. Este índice hay que minimizarlo.

$$gini = \sum_k p_k (1 - p_k)$$

Supongamos que tenemos 5 muestras, 3 de clase A y 2 de clase B, entonces:

$$Gini = \frac{3}{5} * \left(1 - \frac{3}{5}\right) + \frac{2}{5} * \left(1 - \frac{2}{5}\right) = 0,48$$

La distribución toma valores próximos a cero cuando la proporción p de una clase es muy pequeña (lo que implica que el conjunto de valores está muy ordenado, pues si hay pocos elementos de la clase considerada significa que hay muchos de las demás clases, o cuando es muy grande (lo que implica, nuevamente, que hay pocos elementos de las demás clases y que el conjunto está muy ordenado. A medida que aumenta el número de clases el valor del índice gini aumenta también.

- Ganancia de información (se utiliza con valores discretos). Este criterio intenta estimar la información que aporta cada atributo basado en la «teoría de la información», que fundamentalmente mide la imprevisibilidad de un valor. Al obtener la medida de entropía de cada atributo, se puede calcular la ganancia de información del árbol. Este índice hay que maximizarlo o lo que es lo mismo hay que minimizar la nueva entropía.
 - Si A es la ganancia de lo que había hasta un determinado momento y añadimos un atributo B , entonces:
la Ganancia $(A+B) = Entropía(A) - Entropía (A+B)$.
 - La entropía de $(z) = Probabilidad de que ocurra Z \times Información de Z$.
 - La Información de $Z = -\log_2 P(Z)$.
 - Entropía $(Z) = - \sum_{i=1}^{nv} P(Z) * \log_2 (P(Z))$ siendo $nv = n.$ de valores de un atributo.

Ejemplo detallado del uso de la entropía:

<[https://es.wikipedia.org/wiki/%C3%81rbol_de_decisi%C3%B3n_\(modelo_de_clasificaci%C3%B3n_ID3\)](https://es.wikipedia.org/wiki/%C3%81rbol_de_decisi%C3%B3n_(modelo_de_clasificaci%C3%B3n_ID3))>

Ejemplo detallado del índice gini:

<[https://iartificial.net/arboles-de-decision-con-ejemplos-en-python/#:~:text=Los%20%C3%A1rboles%20de%20decisi%C3%B3n%20son,\(las%20ramas\)%20representan%20soluciones](https://iartificial.net/arboles-de-decision-con-ejemplos-en-python/#:~:text=Los%20%C3%A1rboles%20de%20decisi%C3%B3n%20son,(las%20ramas)%20representan%20soluciones)>

Ejemplo detallado de árbol de decisión para regresión:

<[https://iartificial.net/arboles-de-decision-con-ejemplos-en-python/#:~:text=Los%20%C3%A1rboles%20de%20decisi%C3%B3n%20son,\(las%20Oramas\)%20representan%20soluciones](https://iartificial.net/arboles-de-decision-con-ejemplos-en-python/#:~:text=Los%20%C3%A1rboles%20de%20decisi%C3%B3n%20son,(las%20Oramas)%20representan%20soluciones)>

Ejemplo: Recomiendo visitar esta página donde se cuenta magistralmente el ejemplo de donde he sacado los datos:

<<https://www.aprendemachinelearning.com/arbol-de-decision-en-python-clasificacion-y-prediccion/>>

Supongamos una tabla con $E = 635$ filas de datos, y $A = 6$ atributos. Algunos de los atributos son categóricos (valores enteros), y otros numéricos (valores reales).

Una de esas columnas será la etiqueta (clase) que tiene como valores posibles (1) que significa objetivo alcanzado y (0) que significa objetivo no alcanzado.

- 494 no han alcanzado el objetivo (0).
- 141 han alcanzado el objetivo (1).
- *Observen que las cantidades no están balanceadas.*

Veamos cómo se reparte la etiqueta «alcanzado el objetivo» entre los diversos atributos mapeados.

Atributo 1 (estados de ánimo)			
Valor del atributo	Cantidad	N.º con etiqueta 1	Proporción
0	1	0	0
1	8	0	0
2	62	17	0,274194
3	103	15	0,145631
4	146	20	0,136986
5	156	46	0,294872
6	159	43	0,270440
	635	141	

Atributo 2 (tipo de artista)			
Valor del atributo	Cantidad	N.º con etiqueta 1	Proporción
1	95	29	0,305263
2	153	49	0,320261
3	387	63	0,162791

	635	141	
Atributo 3 (género)			
Valor del atributo	Cantidad	N.º con etiqueta 1	Proporción
0	19	2	0,105263
1	100	7	0,070000
2	113	1	0,008850
3	188	60	0,319149
4	215	71	0,330233
	635	141	

Atributo 4 (tempo)			
Valor del atributo	Cantidad	N.º con etiqueta 1	Proporción
0	53	12	0,226415
1	65	16	0,246154
2	517	113	0,218569
	635	141	

Atributo 5 (duración)			
Valor del atributo	Cantidad	N.º con etiqueta 1	Proporción
0,0	71	21	0,295775
1,0	30	10	0,333333
2,0	108	23	0,212963
3,0	168	34	0,202381
4,0	112	26	0,232143
5,0	55	8	0,145455
6,0	91	19	0,208791
	635	141	

Atributo 6 (edad)			
Valor del atributo	Cantidad	N.º con etiqueta 1	Proporción
0,0	66	17	0,257576
1,0	163	49	0,300613
2,0	142	37	0,260563
3,0	217	36	0,165899
4,0	47	2	0,042553
	635	141	

Ahora hay que utilizar algún *software* que permita calcular cuántos niveles de profundidad le asignaremos al árbol de decisión, en este caso de clasificación.

Todos estos algoritmos necesitan ser parametrizados con ciertos valores. Estos valores se obtienen por refinamiento prueba / error. Eso significa que hay que conocer muy bien el algoritmo y haberlo utilizado de forma habitual.

En este caso, el autor del estudio seleccionó una librería denominada «Python SKlearn»:

`<https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html>`

Los parámetros seleccionados:

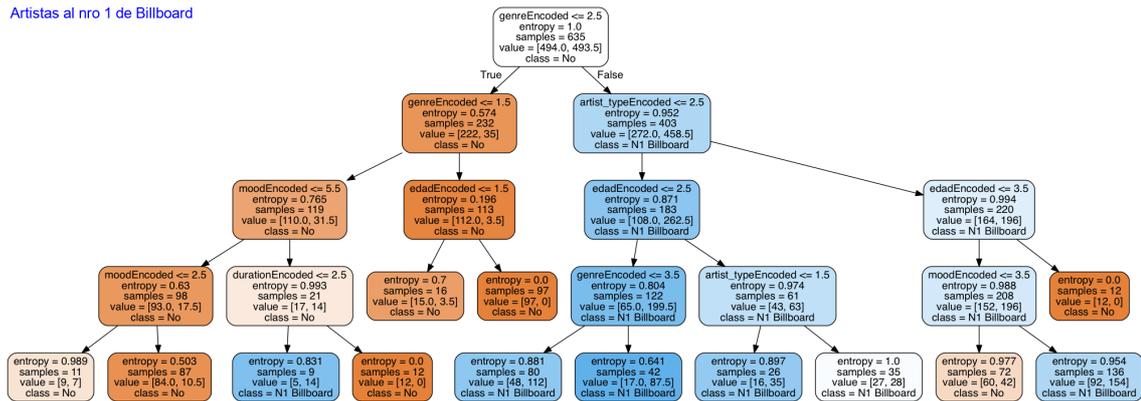
- Criterion = entropy.
- min_samples_split = 20, hace referencia a la cantidad mínima de muestras que debe tener un nodo para poder subdividir.
- min_samples_leaf= 5, hace referencia a la cantidad mínima que puede tener una hoja final.
- Class_weight = {1:3.5} Sirve para compensar los desbalances existentes. Se le asigna el peso 3,5 de peso al valor (1) «alcanzado el objetivo». El valor sale de dividir $\frac{494}{141} = 3,5$.

El resultado del cálculo es:

Profundidad del árbol	Exactitud alcanzable por el árbol
1	0,556101
2	0,556126
3	0,564038
4	0,648859
5	0,617386
6	0,614236
7	0,625124

Ahora lo que queda es crear y visualizar el árbol de 4 niveles de profundidad

Artistas al no 1 de Billboard



La precisión alcanzada en el árbol es de 64,88 %.

Vamos a aprender a leerlo:

- Cada color representa una clase:
 - El naranja para «objetivo no alcanzado».
 - El azul para «objetivo alcanzado».
- El color más intenso indica la seguridad de la clasificación.
- Los nodos blancos evidencian la falta de certeza.
- Hay dos tipos de nodos:
 - Nodos de decisión: Tienen una condición al principio y tienen más nodos debajo de ellos.
 - Nodos de predicción: No tienen ninguna condición ni nodos debajo de ellos (nodos hijo).
- La información de cada nodo es la siguiente:
 - Condición: Si es un nodo donde se toma alguna decisión.
 - Entropy: Función de valorización.
 - *Samples*: Número de muestras que satisfacen las condiciones necesarias para llegar a ese nodo.
 - *Value*: Cuántas muestras de cada clase llegan a este nodo.
 - *Class*: Qué clase se les asigna a las muestras que llegan a este nodo.

Conclusiones y análisis del árbol

A medida que se baja de nivel se observa que los valores de entropía se aproximan más a 1 cuando el nodo tiene más muestras «objetivo alcanzado (1)» en azul y se acerca a 0 cuando hay mayoría de muestras «objetivo no alcanzado (0)» en naranja. Se ven algunas hojas naranjas que finalizan antes de llegar al último nivel: esto es porque alcanzan un nivel de entropía cero, o porque quedan con una cantidad de muestras menor al mínimo permitido para bifurcar que es de 20.

Ahora se puede utilizar la librería (para calcular la certeza) y el árbol para predecir el resultado:

- Suponga que se le proporcionan los siguientes datos:
 - Atributo 1 (5) estado de ánimo.
 - Atributo 2 (1) tipo de artista.
 - Atributo 3 (4) género.
 - Atributo 4 (2) tempo.
 - Atributo 5 (3) duración.
 - Atributo 6 (0) edad.

Resultado, tipo «alcanzará el objetivo» con un 83 % de certeza.

- Suponga que se le proporcionan los siguientes datos:
 - Atributo 1 (4).
 - Atributo 2 (3).
 - Atributo 3 (1).
 - Atributo 4 (2).
 - Atributo 5 (3).
 - Atributo 6 (2).

Resultado, tipo «no alcanzará el objetivo» con un 88 % de certeza.

Los árboles de decisión, ventajas e inconvenientes:

- Ventajas:
 - Son fáciles de entender y explicar.
 - Se adaptan a cualquier tipo de datos.
 - Descubren cuáles son los atributos relevantes.
- Desventajas:
 - Requiere muchos ejemplos.
 - No se garantiza la solución óptima.
 - No extrapolan bien fuera del rango de entrenamiento.
 - Tienen la tendencia a sobre ajustar, sobre todo si no se regularizan. Esto significa que, para obtener el menor error posible, pueden crear tantos nodos como necesiten. Esto puede dar lugar en un modelo muy complejo que funcione muy bien en el conjunto de entrenamiento, pero muy mal con datos nuevos. La regularización consiste en limitar de alguna manera las capacidades del modelo para obtener un modelo de aprendizaje automático que sea más simple y generalice mejor.

Árbol de decisión, Random Forest (Bosque aleatorio), en aprendizaje automático

Si a un árbol de decisión se le da la profundidad suficiente, el árbol tiende a memorizar las soluciones en vez de generalizar el aprendizaje. Es decir, se produce *overfitting*. La idea es trabajar con distintos árboles y ante una pregunta que cada árbol proporcione un voto, y la opción más votada será la respuesta del bosque. Véase <https://es.wikipedia.org/wiki/Random_forest>

Cada árbol del bosque se construye del modo siguiente:

- Sea N el número de casos de prueba, y M el número de atributos en el clasificador.
- Sea m el número de variables que se va a usar para determinar la decisión de un árbol siendo $m \ll M$.
- Se elige un conjunto de entrenamiento para cada árbol y se usa el resto de los casos de prueba para estimar el error.
- Se le pide a cada árbol que haga una misma clasificación. Se guardan los resultados de cada árbol obteniendo n salidas.
- Se calculan los votos obtenidos para cada clase seleccionada y se considera la más votada.
- Tanto la selección de los « m » atributos de cada árbol, como de la cantidad de muestras para entrenar cada árbol, se seleccionan aleatoriamente.

Los bosques aleatorios, ventajas e inconvenientes:

- Ventajas:
 - Se adaptan a cualquier tipo de datos.
 - Al utilizar múltiples árboles se reduce considerablemente el riesgo de *overfitting*.
 - Se mantiene estable con nuevas muestras, puesto que al utilizar cientos de árboles sigue prevaleciendo el promedio de sus votaciones.
- Desventajas:
 - Es mucho más costoso de crear y de ejecutar.
 - Puede requerir muchísimo tiempo de entrenamiento.
 - No funciona bien con conjuntos de datos pequeños.
 - Es muy difícil poder interpretar los árboles de un bosque si se quiere explicar su comportamiento.

Véase:

<<https://www.aprendemachinlearning.com/random-forest-el-poder-del-ensamble/>>

Algoritmos de aprendizaje basados en redes bayesianas

Una red bayesiana es un modelo probabilístico que representa una serie de variables de azar y sus independencias condicionales a través de un grafo acíclico dirigido. Los algoritmos más utilizados son: Naïve Bayes, Gaussian Naive Bayes, Multinomial Naive Bayes, Bayesian Network.

Los modelos Naïve Bayes son un tipo de algoritmos de clasificación de aprendizaje automático basados en el teorema de Bayes. En ellos se supone que todas las evidencias (variables o atributos) son visibles y son totalmente independientes.

Recordemos que el teorema de Bayes permite calcular la probabilidad «posterior» de que ocurra un cierto evento A , dadas algunas probabilidades de eventos anteriores.

$$P(A|R) = \frac{P(R|A)P(A)}{P(R)}$$

Siendo:

- $P(A)$: La probabilidad de A .
- $P(R|A)$: La probabilidad de que se de R dado A .
- $P(R)$: La probabilidad de R .
- $P(A|R)$: La probabilidad posterior de que se de A dado R .

Es decir:

$$\text{Posterior} = \frac{\text{Anterior} \times \text{Probabilidad}}{\text{Evidencia}}$$

Este algoritmo trabaja bien con pocas clases, muchos atributos (cientos, miles), y cada clase será padre de todos los atributos. Puede utilizarse con variables discretas, pero también con variables continuas.

Ejemplo: véase <<https://medium.com/datos-y-ciencia/algoritmos-naive-bayes-fundamentos-e-implementaci%C3%B3n-4bcb24b307f>>

Supongamos que Alicia y Bruno trabajan en la misma oficina y en intervalos semanales de 4 días.

Información previa:

- Alicia viene a la oficina 3 días a la semana $P(\text{Alicia}) = 3/4 = 0,75$.
- Bruno viene a la oficina 1 día a la semana $P(\text{Bruno}) = 1/4 = 0,25$.

Un día vemos pasar delante de nosotros a alguien y dada su rapidez no sabemos si es Alicia o Bruno. Lo que sí detectamos es que llevaba una chaqueta roja.

También sabemos que la semana en la oficina es de 5 días y

- Alicia viste de rojo 2 veces a la semana $P(\text{Rojo}|Alicia) = 2/5 = 0,4$.
- Bruno viste de rojo 3 veces a la semana $P(\text{Rojo}|Bruno) = 3/5 = 0,6$.

Por lo tanto, la pregunta es ¿a quién vimos pasar?

Lo conocido son las probabilidades de que:

- Alicia vista de rojo.
- Bruno vista de rojo.

Lo desconocido son las probabilidades de que la persona que viste de rojo sea:

- Alicia.
- Bruno.

Por lo tanto:

- $P(\text{Alicia}) = 0,75$.
- $P(\text{R}|A) = 0,4$ $P(\neg\text{R}|A) = 0,6$.
- $P(A|R) = \frac{P(\text{R}|A)P(A)}{P(R)} = \frac{0,4 * 0,75}{P(R)} = \frac{0,3}{P(R)}$.
- $P(\text{Bruno}) = 0,25$
- $P(\text{R}|B) = 0,6$ $P(\neg\text{R}|B) = 0,4$.
- $P(B|R) = \frac{P(\text{R}|B)P(B)}{P(R)} = \frac{0,25 * 0,6}{P(R)} = \frac{0,15}{P(R)}$.
- $P(R) = P(\text{R}|A)P(A) + P(\text{R}|B)P(B) = 0,3 + 0,15 = 0,45$.

Por lo tanto:

- $P(A|R) = \frac{0,3}{0,45} = 0,67$.
- $P(B|R) = \frac{0,15}{0,45} = 0,33$.

Puntos fuertes de este algoritmo:

- Es sencillo de implementar.
- Cuanto más se acerque a la presunción de independencia, el algoritmo se comportará muy bien.

Puntos débiles:

- Cuando el conjunto de datos de prueba tiene una característica que no ha sido observada en el conjunto del entrenamiento, el modelo le asignará una probabilidad cero y será inútil realizar predicciones. Uno de los principales métodos para evitar esto es la técnica de suavizado, siendo la estimación de Laplace una de las más populares, aunque existen otras como la interpolación lineal.

Véase otro ejemplo detallado, titulado: «Proyecto de implementación de un detector de spam» en

<https://medium.com/datos-y-ciencia/algoritmos-naive-bayes-fundamentos-e-implementaci%C3%B3n-4bcb24b307f>

Los modelos **Naive Bayes Gaussianos** son un tipo de algoritmos de clasificación de aprendizaje automático basados en el teorema de Bayes. En ellos se supone que todas las evidencias (variables o atributos) son visibles y son totalmente independientes. En este caso los modelos probabilísticos son continuos, una hipótesis típica es que los valores continuos asociados con cada clase se distribuyen según una distribución normal. Véase

https://es.wikipedia.org/wiki/Clasificador_bayesiano_ingenuo#:~:text=En%20teor%C3%A1da%20de%20la%20probabilidad,y%20algunas%20hip%C3%B3tesis%20simplificadoras%20adicionales

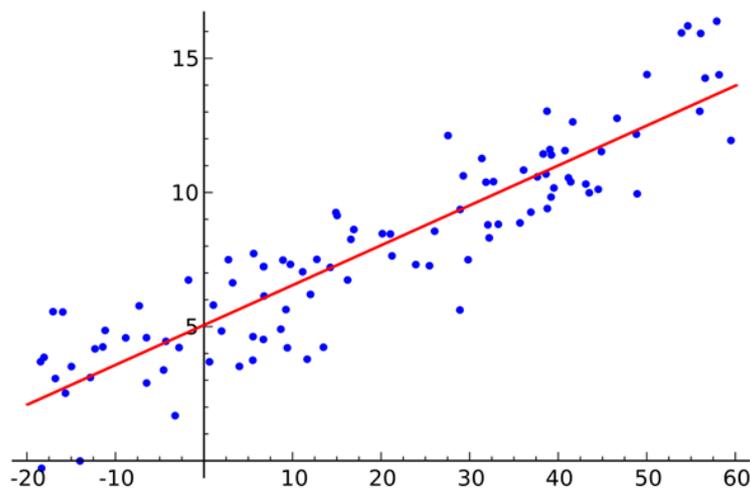
Algoritmos de regresión

Ya se dijo previamente que cuando se usa aprendizaje automático, se pueden realizar tareas de clasificación o de regresión. La diferencia está en el tipo de resultado que se espera que el algoritmo de aprendizaje produzca. En el caso de los algoritmos de regresión se busca un número dentro de un conjunto infinito de posibles resultados. Hay algoritmos lineales, de mínimos cuadrados, con función logística. El propósito en todos los casos es establecer un modelo que relacione un cierto número de características y una variable objetivo-continua.

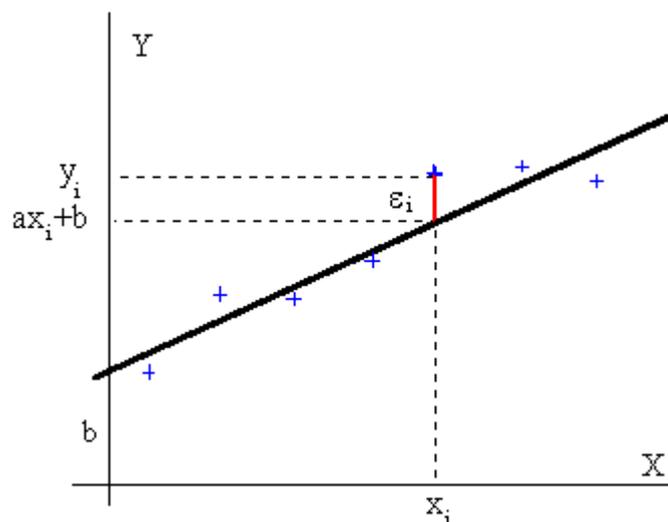
La **regresión lineal simple** ajusta la relación de dependencia entre una sola característica específica independiente (un valor de la variable independiente x) y el valor «resultado» correspondiente (un valor de la variable dependiente y), esta variable también se denomina *predictora*.

Esta relación se realiza estableciendo una línea recta arbitraria y calculando la distancia de la recta a los «puntos datos reales» correspondientes a los valores m valores de partida (x,y) .

Simple Regression \rightarrow $y = mx + b$



El objetivo es sustituir la nube de puntos discreta (x_i, y_i) por una línea recta continua que denominaremos y_r de manera que esa recta $y_r(x) = a x + b$ se aproxime lo máximo posible a los ejemplos utilizados para su entrenamiento.



Se denomina $error(x_i)$ a la diferencia entre el valor real de su y_i asociada, y el valor de $y_r(x_i)$ dada por la recta de regresión.

$$Error(x_i) = y_r(x_i) - y(x_i) = a x_i + b - y(x_i)$$

Como el error puede ser en unas ocasiones positivo y en otras negativo (como en el caso de la figura), se va a elevar al cuadrado para que todos los errores sean positivos

$$Error^2(x_i) = [y_r(x_i) - y(x_i)]^2 = [a x_i + b - y(x_i)]^2$$

Ahora se propone sumar todos los errores al cuadrado y calcular el promedio teniendo en cuenta que si tenemos m parejas de datos (x_i, y_i) la expresión que se obtendría que la denominaremos $J(a, b)$ sería

$$J(a, b) = \frac{1}{2m} \sum_{i=1}^m [a x_i + b - y(x_i)]^2$$

$J(a, b)$ recibe el nombre de *error cuadrático medio*.

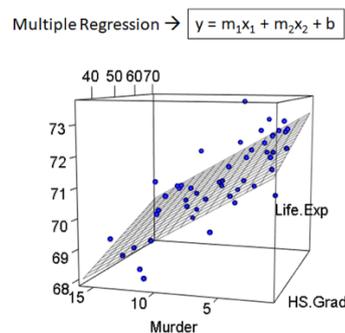
Esa ecuación depende de dos valores « a » y « b » que son desconocidos. Por lo tanto, podríamos encontrar infinitos valores de « a » y de « b ». Con lo cual no habríamos conseguido resolver el problema de encontrar los parámetros que definirían nuestra recta de regresión. Ahora bien, si imponemos alguna condición más al problema, es posible que lográsemos el objetivo propuesto. En este y otros muchos casos análogos, esos valores pueden calcularse exigiendo que $J(a, b)$ sea mínimo, o lo que es lo mismo que los coeficientes de la recta de regresión sean aquellos que minimicen la suma del error que aporta cada una de las parejas dato. Los detalles concretos del cálculo por mínimos cuadrados los puede encontrar en

https://es.wikipedia.org/wiki/Error_cuadrático_medio

Una vez obtenida la expresión $y_r(x) = a x + b$, se puede calcular el valor de $y_r(x)$ para cualquier valor de x , diferente de los valores dato.

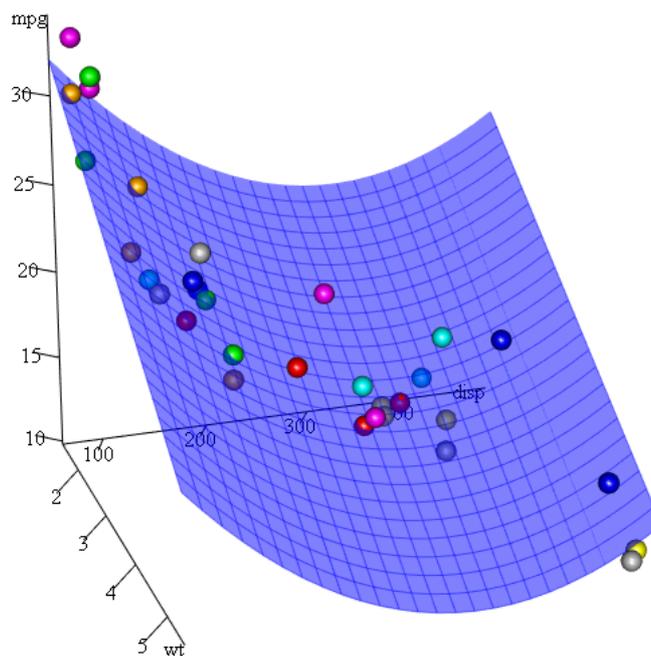
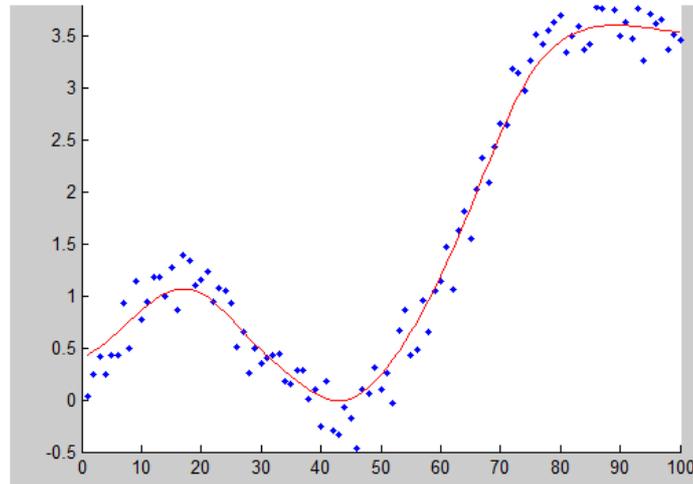
Es más, el algoritmo de regresión puede ir evolucionando con la adquisición de nuevas parejas del tipo «datos reales», recalculando la recta con cada interacción, buscando aquella que mejor ajuste a los puntos de datos (x, y) , o en otras palabras, la línea con el menor error (la más cercana al máximo número de puntos).

Cuando se tiene más de una característica, significa que hay más dimensiones y entonces habrá más de un valor de las variables independientes x_1, x_2, \dots, x_{n-1} y el valor resultado correspondiente, un valor de la variable dependiente y predictora. En estos casos se ajustan planos o hiperplanos. Este tipo de regresión se denomina *regresión lineal múltiple*.



La **regresión polinómica** es un caso especial de análisis de regresión en la que la relación entre «x» e «y» se modela como un polinomio de grado «n» en «x», es típica cuando la distribución de datos es más compleja que la lineal.

Polynomial Regression → $y = W_1x^3 + W_2x^2 + W_3x + W_4$



En cualquiera de los modelos descritos debemos comprobar si se comporta correctamente, para ello es preciso aplicar métricas de evaluación. Esta métrica de evaluación es el error calculado entre el valor proporcionado por la figura generada (línea recta, plano, hiperplano, curva polinomial, superficie polinomial o hipersuperficie polinomial) a cada uno de los puntos reales. Dicha función de error es la que hay que minimizar. Algunas de las expresiones más utilizadas son:

- Error absoluto medio.
- Error cuadrático medio.
- La raíz cuadrada del error cuadrático medio.
- ...

Máquinas de vectores de soporte (Support vector Machines [SVM])

Las MVS son una serie de métodos de aprendizaje supervisado usados para clasificación y regresión. Los algoritmos de MVS usan un conjunto de ejemplos de entrenamiento clasificado en «dos» categorías para construir un modelo que prediga si un nuevo ejemplo pertenece a una u otra de dichas categorías. Trabajan con un hiperplano de dimensión (N-1) siendo N la dimensión de los atributos, que separa los puntos. El vector definido entre dos puntos, de las dos clases, más cercanos es el que recibe el nombre de *vector soporte*.

Las MVS buscan un hiperplano que se pare de forma «óptima» a los puntos de una clase de la de otra. Este tipo de algoritmos buscan el hiperplano que tenga la máxima distancia con los puntos que estén más cerca de él mismo. Desde un punto de vista algorítmico, el problema de optimización del margen geométrico representa un problema de optimización cuadrático con restricciones lineales que puede ser resuelto mediante técnicas estándar de programación cuadrática.

En la literatura se llama *atributo* a la variable predictora y *característica* al que define el hiperplano.

Supongamos un conjunto separable de ejemplos $S = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$, donde x_i es de dimensión «d» e $Y_i \in \{+1, -1\}$. Se puede definir un hiperplano de separación como una función lineal que es capaz de separar dicho conjunto sin error:

$$D(x) = (w_1x_1 + \dots + w_dx_d) + b \text{ (ecuación de un hiperplano de dimensión «d»)}.$$

$$D(x) = w_1x_1 + b \text{ (ecuación de un hiperplano de dimensión «1», una recta)}.$$

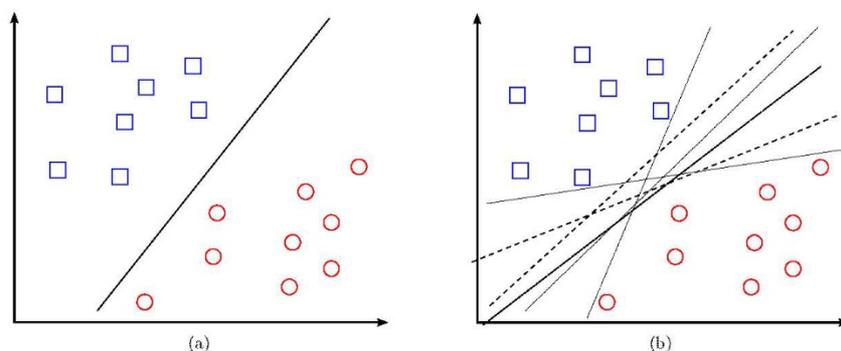


Figura 1: Hiperplanos de separación en un espacio bidimensional de un conjunto de ejemplos separables en dos clases: (a) ejemplo de hiperplano de separación (b) otros ejemplos de hiperplanos de separación, de entre los infinitos posibles.

El hiperplano de separación cumplirá las siguientes restricciones para todo x_i del conjunto de ejemplos

$$y_i D(\mathbf{x}) = y_i [(w_1 x_1 + \dots + w_d x_d) + b] \geq 0, \text{ tanto si } y_i \text{ vale } (+1) \text{ o } (-1)$$

Tal y como se ha visto en la figura anterior, el hiperplano que puede separar no es único, de hecho, existen infinitos hiperplanos separables. Por lo tanto, surge la pregunta sobre si es posible establecer algún criterio adicional que permita definir un hiperplano de separación óptimo. Para ello:

- Se define el concepto de margen de un hiperplano de separación, denotado por τ , como la mínima distancia entre dicho hiperplano y el ejemplo más cercano de cualquiera de las dos clases.
- Un hiperplano de separación se denominará *óptimo* si su margen es de tamaño máximo.

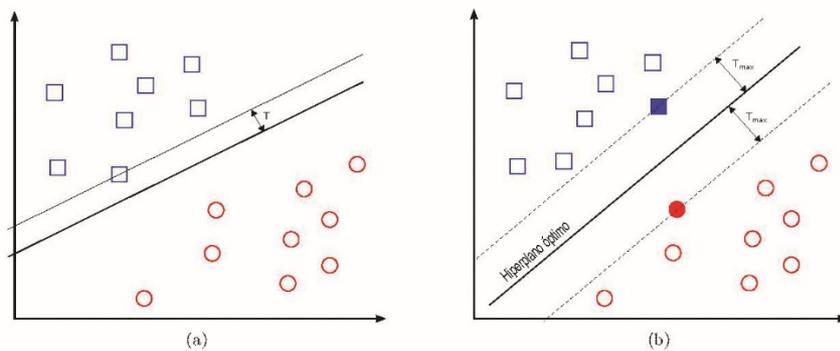


Figura 2: Margen de un hiperplano de separación: (a) hiperplano de separación no-óptimo y su margen asociado (no máximo) (b) hiperplano de separación óptimo y su margen asociado (máximo).

- Una propiedad inmediata de la definición de hiperplano de separación óptimo es que este equidistará del ejemplo más cercano de cada clase.

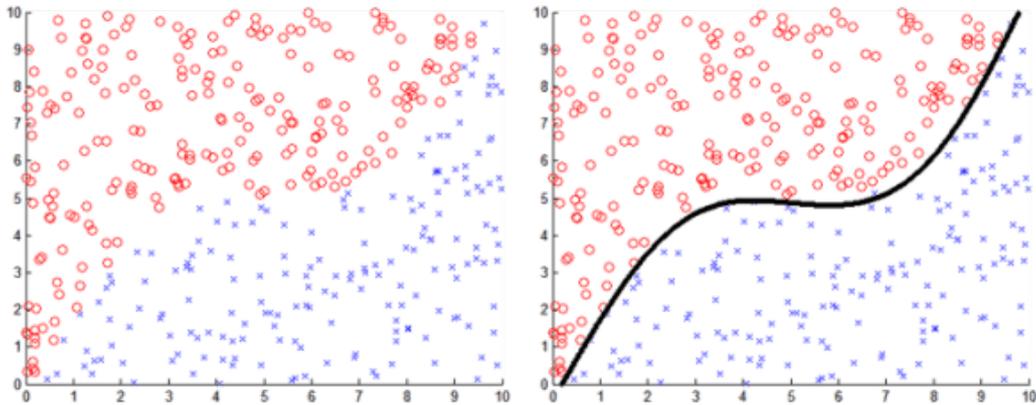
Por geometría se puede deducir que la búsqueda del hiperplano óptimo para el caso separable puede ser formalizado como el problema de encontrar el valor w y b que:

$$\begin{aligned} &\text{Minimiza el funcional } f(\mathbf{w}) = \frac{1}{2} \|\mathbf{w}\|^2 \\ &\text{Sujeto a las restricciones } y_i(D(\mathbf{x}) - 1) \geq 0, i = 1, 2, \dots, n \end{aligned}$$

Siendo $|\cdot|$ el operador valor absoluto y $\|\cdot\|$ el operador norma de un vector.

Este problema de optimización con restricciones corresponde a un problema de programación cuadrática y es abordable mediante la teoría de la optimización.

Existen generalizaciones de esa teoría básica para cuando los ejemplos son cuasi separables, es decir, haya ruido o no sea perfecta la separación lineal. Y también para cuando las funciones de separación no pueden ser lineales.



Las máquinas de vectores soporte pueden también adaptarse para resolver problemas de regresión SVR (Support Vector Regression).

Comentario: ¿Cuáles son las ventajas de la regresión lineal respecto de los árboles de decisión? Todo depende de los datos, para el caso de la regresión lineal, es ideal cuando tus datos pueden ser clasificados perfectamente usando una división más simple, que podría ser lineal, viéndolo gráficamente, sin embargo, para el caso de árboles de decisión, la clasificación es más compleja.

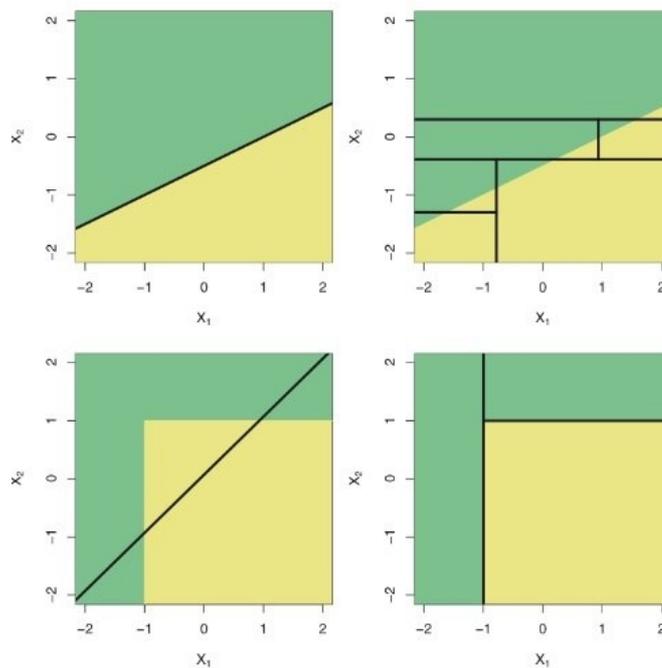


Imagen de Yesenia Alejandro.

En este gráfico se puede ver en la primera gráfica, para el comportamiento de las variables, que la clasificación puede ser perfectamente modelada usando una regresión lineal y no mediante un árbol de decisión, como lo muestra la figura superior derecha, sin embargo, para el comportamiento de las variables inferiores, se observa lo contrario, la regresión lineal no aporta un buen ajuste, pero el árbol de decisión es perfecto.

Para saber más:

Véase <[http://www.ia.uned.es/~ejcarmona/publicaciones/\[2013-Carmona\]%20SVM.pdf](http://www.ia.uned.es/~ejcarmona/publicaciones/[2013-Carmona]%20SVM.pdf)>

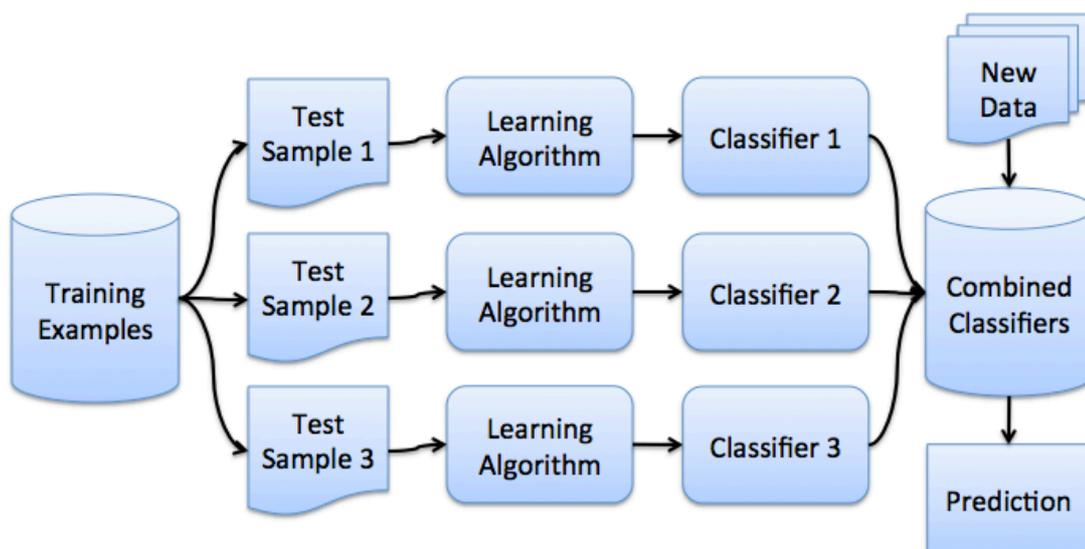
Software sobre SVM:

- LIBSVM <<http://www.csie.ntu.edu.tw/~cjlin/libsvm/>>
- SVM light <<http://svmlight.joachims.org/>>

Métodos de «ensemble» (conjuntos de clasificadores). Véase

<<http://www.cs.us.es/~fsancho/?e=106>>

En el campo del aprendizaje automático, los métodos combinados (**métodos de ensemble**) utilizan múltiples algoritmos de aprendizaje para obtener un rendimiento predictivo que mejore el que podría obtenerse por medio de cualquiera de los algoritmos de aprendizaje individuales que lo constituyen.



Como ya vimos, los algoritmos de aprendizaje supervisado se describen normalmente como la tarea de buscar a través de un espacio de hipótesis para encontrar la más adecuada que haga buenas predicciones con un problema en

particular. En general, esta tarea es muy complicada y, ni siquiera teniendo la certeza de que en el espacio completo existe una buena solución, podemos estar seguros de encontrarla.

La idea de los métodos combinados es considerar múltiples hipótesis simultáneamente para formar una hipótesis que, con suerte (y la ayuda de algunos teoremas esenciales), se comporte mejor. El término de *métodos de ensemble* se suele reservar para aquellas combinaciones que hacen uso de múltiples hipótesis pertenecientes a una misma familia, mientras que se usa el término más general de *sistemas de aprendizaje múltiples* cuando las hipótesis que se combinan provienen de diversas familias.

Evidentemente, debido a que los métodos combinados hacen uso de varias hipótesis simultáneas, se produce una elevación en los costes computacionales, por lo que suele ser habitual utilizar algoritmos rápidos como espacio de hipótesis base, como son los árboles de decisión.

Empíricamente, se ha comprobado que cuando existe una diversidad significativa entre los modelos individuales, las combinaciones tienden a obtener mejores resultados, por lo que muchos de los métodos existentes buscan promover la diversidad entre los modelos que se combinan, y ello provoca a veces que se usen como modelos aquellos que hacen un uso fuerte de la aleatoriedad, en vez de modelos más dirigidos y que funcionan mejor individualmente.

Puede ver aplicaciones en

<https://en.wikipedia.org/wiki/Ensemble_learning#:~:text=in%20anomaly%20detection.-,Ensemble%20theory,from%20which%20it%20is%20built>

En la literatura se pueden encontrar métodos de este tipo:

- Agregación Bootstrap o Bagging

Véase <<https://unipython.com/ensemble-methods-metodos-conjunto/>>

- Boosting

Véase <<https://studylib.es/doc/8452073/boosting--m.-c%C3%A1rdenas-montes->>

- Subespacios aleatorios

Véase <<https://olmo.usal.es>>

En general:

<https://link.springer.com/chapter/10.1007/3-540-45014-9_1#:~:text=Ensemble%20methods%20are%20learning%20algorithms,coding%20%20Bagging%2C%20and%20boosting>

Espacio de versiones

Ejemplo 1, dada una lista de pares atributo-valor, que se denominará *conjunto de entrenamiento D*

Origen	Marca	Color	Década	Tipo	Clase	Abreviatura
Asia	Hyunday	Negro	2000	Eco	+	AHNOE
Asia	Kia	Rojo	1990	Sport	-	AKR9S
Asia	Kia	Negro	2010	Eco	+	AKN1E
UE	Dacia	Negro	2000	Eco	-	EDNOE
Asia	Hyundai	Blanco	2000	Eco	+	AHBOE

Notación y terminología:

- Una instancia X (ejemplo) es un conjunto de valores que se corresponden con cada uno de los atributos.
- Un concepto $c(x)$ es un subconjunto de ejemplos de X , que toman uno de los valores (Sí / No).
- Función objetivo es lo que se desea aprender. Solo se conoce el valor de c para algunas instancias (ejemplos).
- En el conjunto de entrenamiento dado hay 3 ejemplos que dan a $c(x)$ el valor positivo y dos que dan a $c(x)$ el valor negativo.
- Espacio de hipótesis H : Conjunto de funciones $h: X \rightarrow \{1, 0\}$ que en el proceso de aprendizaje se pueden considerar como posibles definiciones del concepto objetivo. Usualmente, en H no están todos los posibles conceptos. Hay sesgo inductivo.
- Objetivo del aprendizaje. Encontrar $h \in H$ tal que para cualquier ejemplo $\langle x, c(x) \rangle \in D$, se tenga $h(x) = c(x)$, hipótesis consistente con los ejemplos.

Representación compacta:

- Se usará « i » para representar un valor cualquiera, «0» para representar que ningún valor es posible.

El espacio de versiones es un concepto del aprendizaje automático que representa el subconjunto de todas las hipótesis que son consistentes con los ejemplos de entrenamiento (Tom M. Mitchell, 1997). Este conjunto contiene todas las hipótesis que no han sido eliminadas por provocar conflictos con los datos observados. Veamos la implementación denominada «Algoritmo de eliminación de candidatos»:

- Sea G el conjunto de elementos de máxima generalidad de H .
- Sea S el conjunto de elementos de máxima especificidad de H .
- Para cada ejemplo d del conjunto de entrenamiento D :

- Si d es un ejemplo positivo, entonces:
 - Eliminar de G cualquier hipótesis inconsistente con d .
 - S se va ampliando con cada ejemplo positivo, para que los incluya.
- Si d es un ejemplo negativo, entonces:
 - Eliminar de S cualquier hipótesis inconsistente con d .
 - G se va restringiendo con cada ejemplo negativo, para que los excluya.

Sean S y G los conjuntos obtenidos por eliminación de candidatos:

- Si S y G son no vacíos, resultan ser, respectivamente, la cota específica y la cota general del espacio de versiones (respecto del conjunto de entrenamiento).
- Si $S = G = \{h\}$, entonces h es la única hipótesis de H consistente con todos los ejemplos.
- Si $S = G = \emptyset$, no existe h consistente con los ejemplos.

Se produce convergencia hacia el concepto objetivo, siempre que:

- El conjunto de entrenamiento es suficientemente grande.
- Hay ausencia de ruido en los ejemplos, es decir, no hay errores.
- El concepto objetivo está en H .

Ejemplo 1 con la tabla inicial

Origen	Marca	Color	Década	Tipo	Clase	Abreviatura
Asia	Hyunday	Negro	2000	Eco	+	AHNOE
Asia	Kia	Rojo	1990	Sport	-	AKR9S
Asia	Kia	Negro	2010	Eco	+	AKN1E
EU	Dacia	Negro	2000	Eco	-	EDNOE
Asia	Hyundai	Blanco	2000	Eco	+	AHBOE

- Paso 0: $S_0 = \{ \langle 0,0,0,0,0 \rangle \}$ $G_0 = \{ \langle ?,?,?,?,? \rangle \}$
- Paso 1:
 - Ejemplo positivo $\langle \text{AHNOE} \rangle$.
 - Nada que eliminar de G_0 .
 - Generalización de S_0 :
 - $\langle \text{AHNOE} \rangle$.
 - Luego:
 - $S_1 = \{ \langle \text{AHNOE} \rangle \}$.
 - $G_1 = \{ \langle ?,?,?,?,? \rangle \}$.
- Paso 2:
 - Ejemplo negativo $\langle \text{AKR9S} \rangle$.
 - Nada que generalizar, es decir, nada que eliminar de S_1 .
 - Especialización de G_1 :
 - $\langle \neg A ? ? ? ? \rangle$ no incluye S_1 .
 - $\langle ? \neg K ? ? ? \rangle$ puede ser:
 - $\langle ? H ? ? ? \rangle$, incluye S_1 .
 - $\langle ? D ? ? ? \rangle$, no incluye S_1 .
 - $\langle ? ? \neg R ? ? \rangle$ puede ser:
 - $\langle ? N ? ? ? \rangle$, incluye S_1 .
 - $\langle ? B ? ? ? \rangle$, no incluye S_1 .
 - $\langle ? ? ? \neg 9 ? \rangle$ puede ser:
 - $\langle ? ? ? 0 ? \rangle$, incluye S_1 .
 - $\langle ? ? ? 1 ? \rangle$, no incluye S_1 .
 - $\langle ? ? ? ? \neg S \rangle$ Puede ser:
 - $\langle ? ? ? ? E \rangle$ incluye S_1 .
 - Por lo tanto:
 - $G_2 = \{ \langle ? H ? ? ? \rangle, \langle ? ? N ? ? \rangle, \langle ? ? ? 0 ? \rangle, \langle ? ? ? ? E \rangle \}$.
 - $S_2 = \{ \langle \text{AHNOE} \rangle \}$.

- Paso 3:
 - Ejemplo positivo <AKN1E>.
 - Generalizar S_2 , es decir, $\{A ? N ? E\}$.
 - Especializar $G_2 = \{<? H ? ? ?>, <? ? N ? ?>, <? ? ? 0 ?>, <? ? ? ? E>\}$:
 - <? H ? ? ?> eliminar, no hay equiparación.
 - <? ? N ? ?> mantener, hay equiparación.
 - <? ? ? 0 ?> eliminar, no hay equiparación.
 - <? ? ? ? E>] mantener, hay equiparación.
 - Por lo tanto:
 - $G_3 = \{<? ? N ? ?>, <? ? ? ? E>\}$.
 - $S_3 = \{< A H N O E \>\}$.
- Paso 4:
 - Ejemplo negativo <EDNOE>.
 - Nada que generalizar en S_3 , es decir, $\{A ? N ? E\}$, se mantiene.
 - Especializar $G_3 = \{<? ? N ? ?>, <? ? ? ? E>\}$
 - <? ? N ? ?>:
 - <-E ? N ? ? > puede ser:
 - <A ? N ? ? >, incluye S_3 .
 - <? -D N ? ? > puede ser:
 - <? H N ? ? >, no incluye S_3 .
 - <? K N ? ? >, no incluye S_3 .
 - <? ? N -0 ? > puede ser:
 - <? ? N 9 ? >, no incluye S_3 .
 - <? ? N 1 ? >, no incluye S_3 .
 - <? ? N ? -E> puede ser:
 - <? ? N ? S>, no incluye S_3 .
 - <? ? ? ? E>:
 - <-E ? ? ? ? E > puede ser:
 - <A ? ? ? ? >, incluye S_3 .
 - <? -D ? ? E > puede ser:
 - <? H ? ? E >, no incluye S_3 .
 - <? K ? ? E >, no incluye S_3 .
 - <? ? -N ? E> puede ser:
 - <? ? R ? E>, no incluye S_3 .
 - <? ? B ? E>, no incluye S_3 .
 - <? ? ? -0 E > puede ser:
 - <? ? ? 9 E >, no incluye S_3 .
 - <? ? ? 1 E >, no incluye S_3 .
 - Por lo tanto:
 - $G_4 = \{<A ? N ? ?>, <A ? ? ? ? E>\}$.
 - $S_4 = \{<A ? N ? E>\}$.

- Paso 5:
 - Ejemplo positivo $\langle AHBOE \rangle$.
 - Generalizar S_4 , es decir, $\{A \ ? \ ? \ ? \ E\}$.
 - Especializar $G_4 = \{\langle A \ ? \ N \ ? \ ? \ \rangle, \langle A \ ? \ ? \ ? \ E \rangle\}$:
 - $\langle A \ ? \ N \ ? \ ? \ \rangle$ eliminar, no hay equiparación.
 - $\langle A \ ? \ ? \ ? \ E \rangle$ mantener, hay equiparación.
 - Por lo tanto:
 - $G_5 = \{\langle A \ ? \ ? \ ? \ E \rangle\}$.
 - $S_5 = \{\langle A \ ? \ ? \ ? \ E \rangle\}$.
- Solución $G_5 = S_5 = \langle A \ ? \ ? \ ? \ E \rangle = \langle \text{Asia} \ ? \ ? \ ? \ \text{Eco} \rangle$.

Clasificación de una nueva instancia, utilizando S y G obtenidos previamente:

- $S = G = \{\langle A \ ? \ ? \ ? \ E \rangle\}$
- Si la nueva instancia es consistente con todo S , entonces positivo.
- Si la nueva instancia no es consistente con ninguno de G , negativo.
- En otro caso, voto mayoritario o simplemente no se clasifica.

Nuevas instancias:

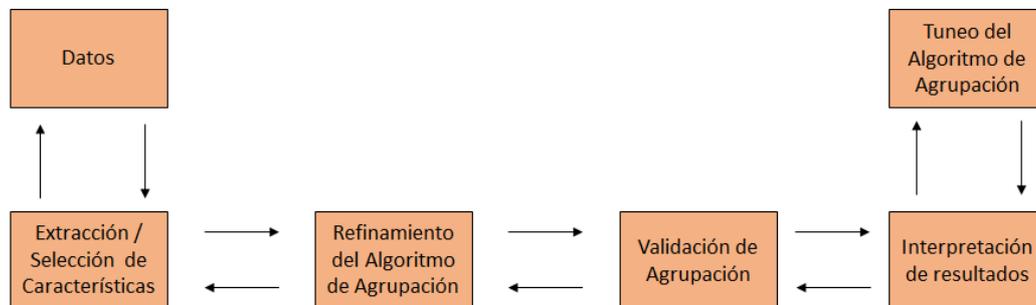
- $\langle \text{Asia, Kia, Blanco, 2000, Eco} \rangle \rightarrow \text{Sí}$.
- $\langle \text{Europa, Kia, Blanco, 2000, Sport} \rangle \rightarrow \text{No}$.
- $\langle \text{Europa, Kia, Blanco, 2000, Eco} \rangle \rightarrow \text{Empate, sí o no}$.

Para ver otro ejemplo con más detalle véase

<https://www.studocu.com/es/document/uned/aprendizaje-automatico/apuntes/espacio-de-versiones/6883533/view>

Algoritmos de aprendizaje no supervisado (modelo simbólico)

El proceso general que se suele seguir al desarrollar un modelo de aprendizaje no supervisado se resume en la siguiente imagen



Este tipo de aprendizaje se suele usar en problemas de: *clustering*, agrupamientos de co-ocurrencias y perfilado o *profiling*.

Clustering

- En términos básico, el objetivo de la agrupación es encontrar diferentes grupos dentro de los elementos de los datos.
- Los algoritmos más usados son:
 - K-Medias (*K-Means*).
 - Clusterización jerárquica (*Hierarchical Clustering*).
 - *Density Based Scan Clustering* (DBSCAN).
 - Agrupamiento gaussiano.

K-Medias

Véase

<<https://medium.com/datos-y-ciencia/aprendizaje-no-supervisado-en-machine-learning-agrupaci%C3%B3n-bb8f25813edc>>

Son algoritmos fáciles de implementar y muy eficientes computacionalmente. Este algoritmo tiene como objetivo encontrar y agrupar en clases los puntos de datos numéricos que tienen una alta similitud entre ellos. Es decir, la similitud se mide como la opuesto de la distancia entre puntos de datos. Cuanto más cerca estén los puntos de datos, más similares y con más probabilidad de pertenecer al mismo clúster serán.

Conceptos clave:

- Distancia cuadrada euclídea en un espacio m-dimensional

$$d(\mathbf{x}, \mathbf{y})^2 = \sum_{j=1}^m (x_j - y_j)^2 = \|\mathbf{x} - \mathbf{y}\|_2^2$$

- Inercia de los clústeres. Es decir, la suma de errores cuadrados

$$SSE = \sum_{i=1}^n \sum_{j=1}^k w^{(i,j)} \|\mathbf{x}^{(i)} - \boldsymbol{\mu}^{(j)}\|_2^2$$

Siendo:

μ_j el centroide del clúster j.

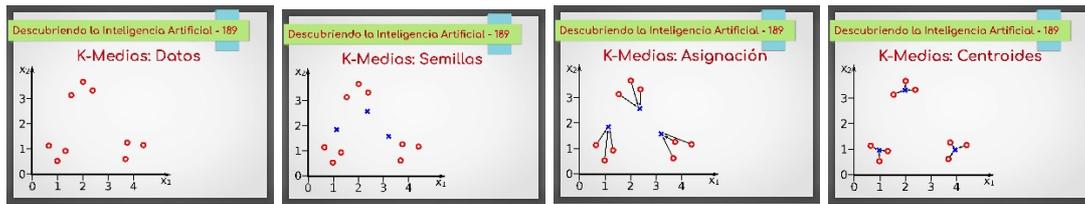
$w(i,j)$ es 1 si la muestra $x(i)$ está en el clúster j y 0 en caso contrario.

K-Media se puede entender como un algoritmo que intenta minimizar el factor de inercia del clúster.

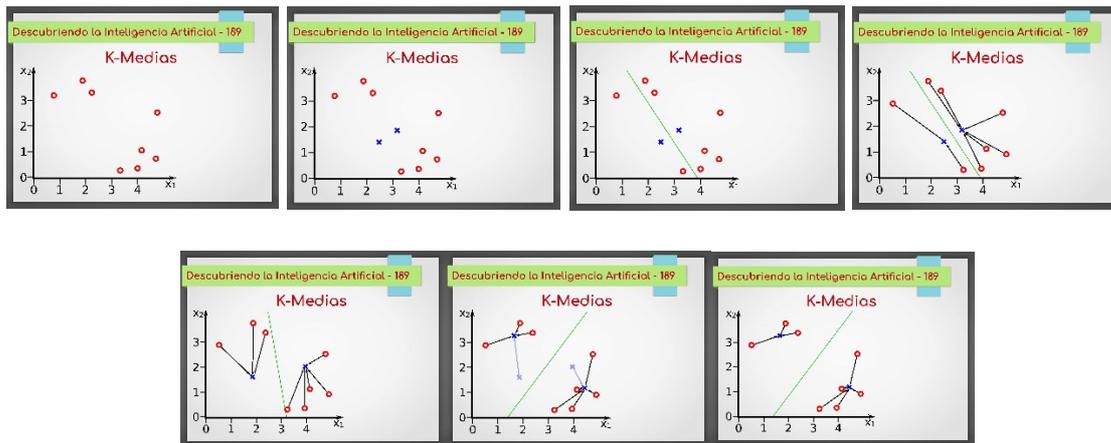
Los pasos del algoritmo son los siguientes:

- Es necesario seleccionar un valor k, que representa al número de clústeres que se desea encontrar.
- El algoritmo selecciona aleatoriamente los centroides de cada grupo.
- Se asignará cada punto de datos al centroide más cercano utilizando la distancia euclídea.
- Se calcula la inercia del conglomerado.
- Se calculan nuevos centroides como la media de los puntos que se han agrupado previamente. Es decir, se calcula el error cuadrático mínimo de los puntos de datos al centro de cada clúster, moviendo el centro hacia ese punto.
- Se vuelve al paso tercero.

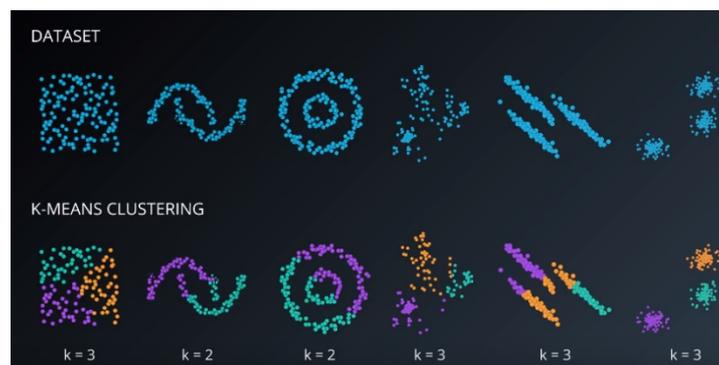
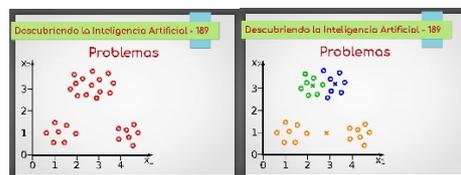
Ejemplo 1



Ejemplo 2



Problemas



El algoritmo depende de las semillas de partida, del número fijo de clases iniciales y es solo para valores numéricos.

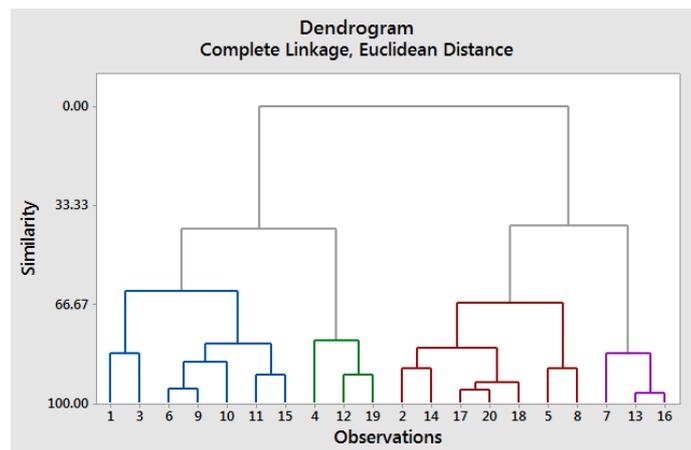
Este algoritmo se suele utilizar como punto de referencia para evaluar el rendimiento de otros métodos de agrupación.

Agrupación jerárquica

La principal ventaja de este tipo de algoritmos es que no es necesario especificar el número de agrupaciones, lo encuentra por sí mismo. Además, permite el trazado de dendrogramas.

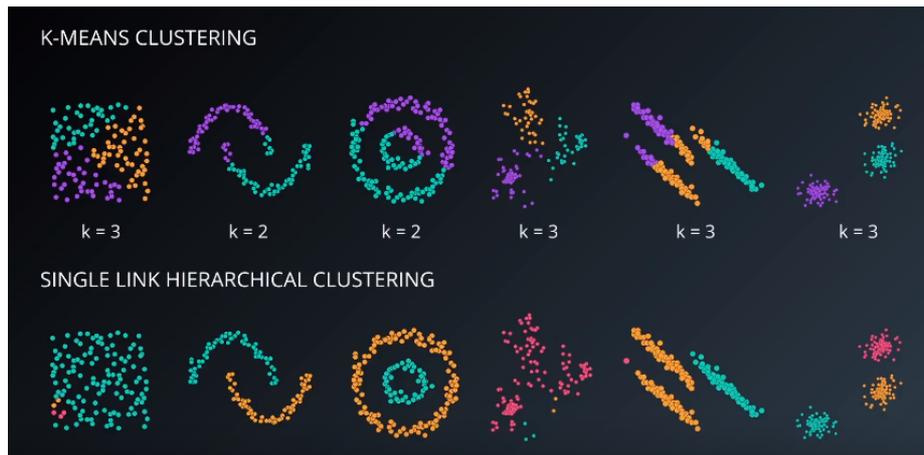
Un dendrograma es un tipo de representación gráfica que organiza los datos en subcategorías que se van dividiendo en otros hasta llegar al nivel de detalle deseado. Este tipo de representación permite apreciar claramente las relaciones de agrupación entre los datos e incluso entre grupos de ellos, aunque no las relaciones de similitud o cercanía entre categorías. También podríamos referirnos al dendrograma como la ilustración de las agrupaciones derivadas de la aplicación de un algoritmo de clustering jerárquico. Véase

<<https://es.wikipedia.org/wiki/Dendrograma#:~:text=Un%20dendrograma%20es%20un%20tipo,van%20dividiendo%20en%20otras%20sucesivamente>>



Tipos de agrupaciones jerárquicas:

- Divisivo: Este método comienza por englobar todos los puntos de datos en un solo grupo. Posteriormente, irá dividiendo el grupo iterativamente en otros más pequeños hasta que cada uno contenga solo una muestra.
- Aglomerativo: Este método comienza con cada muestra siendo un grupo diferente y luego los va fusionando por cercanía hasta que solo hay un grupo. Existen algunos algoritmos aglomerativos:
 - Acoplamiento simple: Supone que cada punto inicial es un conglomerado. Luego calcula distancias entre cada par de cúmulos y fusiona los dos cúmulos para los cuales la distancia entre los miembros es la más pequeña.



- Acoplamiento completo: La distancia o similitud entre dos clústeres se mide atendiendo a sus elementos más dispares, es decir, por la máxima distancia (o mínima similitud) entre sus componentes.
- Ventajas:
 - Suelen ser muy informativas.
 - Son potentes cuando el conjunto de datos contiene relaciones jerárquicas reales.
- Desventajas:
 - Son muy sensibles a los valores atípicos, disminuyendo el rendimiento.
 - Son muy exigentes computacionalmente.

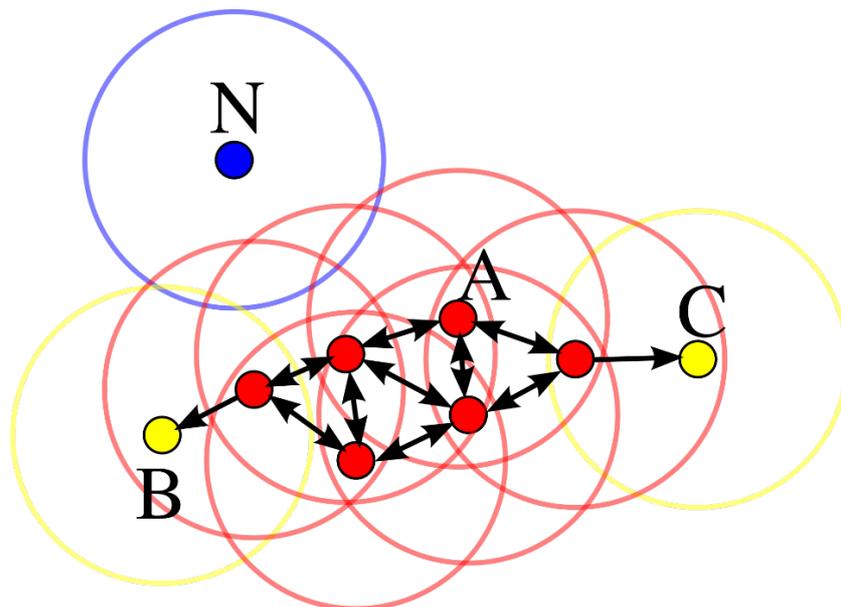
Para saber más: Véase <<https://www.ugr.es/~gallardo/pdf/cluster-3.pdf>>

Agrupación espacial basada en la densidad del ruido (Density Based Scan Clustering) (DBSCAN): Véase

<<https://es.wikipedia.org/wiki/DBSCAN#:~:text=El%20agrupamiento%20espacial%20basado%20en,y%20Xiaowei%20Xu%20en%201996>>

Considere un conjunto de puntos a ser agrupados en un espacio determinado. La técnica de agrupación DBSCAN clasifica los puntos como puntos núcleo, puntos (densamente-)alcanzables, o ruido de la siguiente forma:

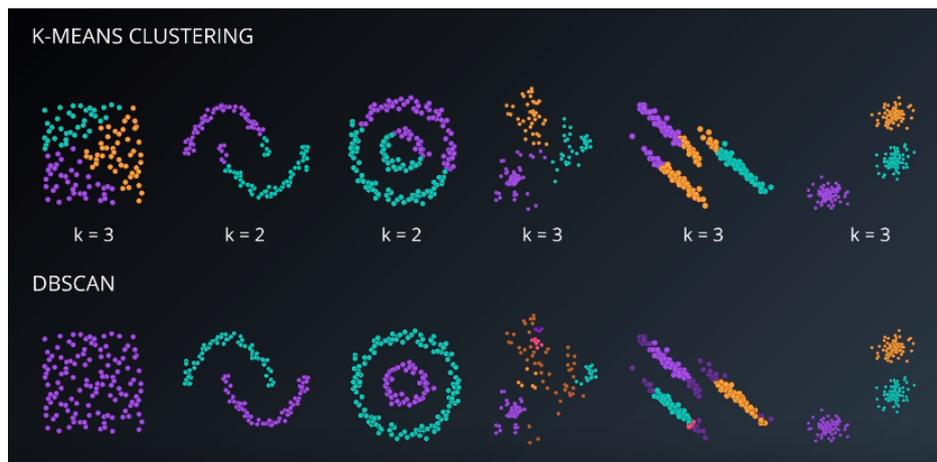
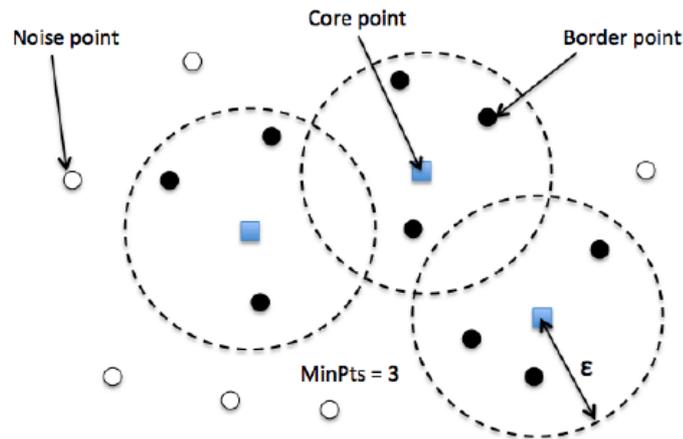
- Un punto p es un punto núcleo si al menos minPts puntos están a una distancia ϵ de él y esos puntos son directamente alcanzables desde p . No es posible tener puntos directamente alcanzables desde un punto que no sea un núcleo.
- Un punto q es alcanzable desde p si existe una secuencia de puntos $p_1 = p, p_2, p_3, \dots, p_{n-1}, p_n = q$, tal que p_{i+1} es directamente alcanzable desde p_i ; es decir, todos los puntos de la secuencia deben ser puntos núcleo, con la posible excepción de q .
- Un punto que no sea alcanzable desde cualquier otro punto se considera ruido.



Los puntos marcados como A son puntos núcleo. Los puntos B y C son *densamente alcanzables* desde A y *densamente conectados* con A, y pertenecen al mismo clúster. El punto N es un punto ruidoso que no es núcleo ni densamente alcanzable. (MinPts=3 o MinPts=4) (Autor Chire) [CC BY-SA 3.0](https://creativecommons.org/licenses/by-sa/3.0/).

El algoritmo sigue la lógica de:

- Identificar un punto central y hacer un grupo para cada uno, o para cada grupo conectado de puntos centrales.
- Identificar y asignar puntos fronterizos a sus respectivos puntos centrales.



- Ventajas:
 - No es necesario especificar el número de grupos.
 - Existe una gran flexibilidad en las formas y en los tamaños que pueden adoptar los grupos.
 - Es muy útil para identificar y tratar con datos de ruido y valores atípicos.
- Desventajas:
 - Se enfrenta a dificultades a la hora de tratar los puntos de encuentro que son alcanzables por dos grupos.
 - No encuentra bien racimos de densidades variables.

Modelos de mezcla Gaussiana o Agrupamiento Gaussiano.

Véanse:

<<https://es.wikipedia.org/wiki/DBSCAN#:~:text=El%20agrupamiento%20espacial%20basado%20en,y%20Xiaowei%20Xu%20en%201996>>

<<https://sites.google.com/site/jenavarrob/home/projects/support-vector-machines-para-reconocimiento-de-patronos>>

Estos modelos probabilísticos asumen que todas las muestras se generan a partir de una mezcla finita de distribuciones gaussianas con parámetros desconocidos (media y desviación típica) de cada gaussiana. Es una generalización del método K-Medias.

El objetivo del algoritmo es tomar una serie de datos de cualquier dimensión y asignarle una serie de funciones gaussianas de probabilidad a los clústeres que encuentra en estos datos. Para esto el algoritmo obtiene las distancias que hay entre los datos y las medias de las gaussianas, de esta manera obtiene un peso para cada dato, este peso varía de acuerdo con la probabilidad encontrada para cada dato de que pertenezca a una gaussiana, de esta manera en cada iteración del algoritmo se van modificando las medias y las matrices de covarianzas de manera que se van acercando a posibles agrupaciones encontradas en los datos.

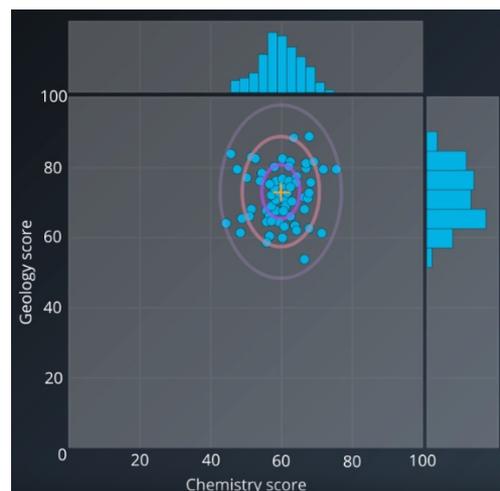
Cada punto dato pertenece con diferentes niveles de pertenencia a cada grupo.



El algoritmo funciona de la manera siguiente:

- Se inicializan las distribuciones gaussianas. Lo hace con los valores μ (media) y σ (desviación estándar) obtenidos aleatoriamente. Se puede aplicar *a posteriori* de haber aplicado K_Medias, partiendo ya de ciertas agrupaciones.
- Paso E (Expectation), todos los puntos de dato se asignarán a cada grupo con su respectivo nivel de pertenencia. Es decir, se calculan las distancias de los datos con respecto a las medias de cada gaussiana.
- Paso M (Maximization), se comprueban las expectativas y se calculan nuevos parámetros para las funciones gaussianas, es decir, nuevos parámetros de μ y σ .
- Se repiten los pasos E y M hasta la convergencia.

Ejemplo:



- Ventajas:
 - Permite una gran flexibilidad en el número y en la forma de los grupos.
- Desventajas:
 - Es muy sensible a los valores iniciales que condicionan en gran medida el rendimiento.
 - Puede converger a un mínimo local, lo que generaría una solución no óptima.
 - Suele requerir regularización de las covarianzas entre los puntos dados.

Algoritmos de reducción de dimensión

Buscan explorar la estructura existente de manera no supervisada para simplificar los datos y reducirlos o comprimirlos. Suelen ser útiles para disminuir el conjunto de variables que luego pueda usar un algoritmo supervisado, de manera que al trabajar con un espacio reducido se puede obtener más precisión que en el espacio original y se necesitan menos cálculos.

Esa transformación de datos puede ser lineal, como en el caso de análisis de componentes principales (PCA), pero también existen otras técnicas de reducción no lineal.

Los algoritmos más utilizados son:

- Análisis de componentes principales (PCA): Principal Componente Analysis (PCA).
- Análisis de componentes independientes (ICA).
- Incrustación estocástica de vecinos (SNE) y su corrección t-SNE.
- ...

Análisis de componentes principales (ACP)

Véase:

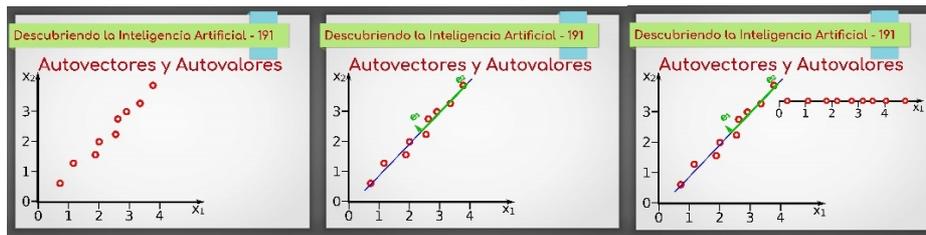
<[https://es.wikipedia.org/wiki/An%C3%A1lisis_de_componentes_principales#:~:text=En%20estad%C3%ADstica%2C%20el%20an%C3%A1lisis%20de,%C2%ABcomponentes%C2%BB\)%20no%20correlacionadas.&text=El%20ACP%20se%20emplea%20sobre,y%20para%20construir%20modelos%20predictivos](https://es.wikipedia.org/wiki/An%C3%A1lisis_de_componentes_principales#:~:text=En%20estad%C3%ADstica%2C%20el%20an%C3%A1lisis%20de,%C2%ABcomponentes%C2%BB)%20no%20correlacionadas.&text=El%20ACP%20se%20emplea%20sobre,y%20para%20construir%20modelos%20predictivos)>

El ACP construye una transformación lineal que escoge un nuevo sistema de coordenadas para el conjunto original de datos en el cual la varianza de mayor tamaño del conjunto de datos se captura en el primer eje (primera componente principal), la segunda varianza más grande es el segundo eje, y así sucesivamente. Para construir esta transformación lineal existen dos formas básicas de obtenerla:

- O se construye la matriz de coeficientes de correlación, si los datos no son dimensionalmente homogéneos o el orden de magnitud de las variables aleatorias medidas no es el mismo.
- O se construye la matriz de covarianzas, si los datos son dimensionalmente homogéneos y presentan valores medios similares.

La transformación queda definida por los valores y vectores propios de la matriz.

Ejemplo 1



x_1 y x_2 son atributos, e_1 y e_2 son los autovectores y su tamaño son los autovalores. De la gráfica se deduce que e_1 es despreciable, por lo tanto, el atributo x_2 no es necesario.

Ejemplo 2

Un análisis de 11 indicadores socioeconómicos de 96 países, reveló que los resultados podían explicarse en alto grado a partir de solo dos componentes principales, el primero de ellos tenía que ver con el nivel de PIB total del país y el segundo con el índice de ruralidad.

Análisis de componentes independientes (ACI)

Véanse:

<https://es.wikipedia.org/wiki/An%C3%A1lisis_de_componentes_independientes>

<<http://bibing.us.es/proyectos/abreproy/11088/fichero/Proyecto+Fin+de+Carrera+%252F5.pdf>>

El análisis de componentes independientes (ACI) es una generalización del análisis de componentes principales (ACP), en ambos casos se practica una transformación lineal de los datos originales, aunque la diferencia básica es que el ACI no requiere que las variables originales tengan una distribución gaussiana.

Incrustación estocástica de vecinos (SNE) y su corrección t-SNE

Véanse:

<<https://www.deeplearningitalia.com/incrustacion-estocastica-de-vecinos-sne-y-su-correccion-en-t-sne/>>

<<https://sitiobigdata.com/2018/08/27/algorithmo-t-sne-con-python/>>

Es una técnica no lineal basada en dos conceptos: La entropía y las divergencias de probabilidad.

Para saber más ver:

<https://es.wikipedia.org/wiki/Reducci%C3%B3n_de_dimensionalidad>

Algoritmos de aprendizaje por refuerzo (modelo simbólico)

Véanse:

<https://es.wikipedia.org/wiki/Aprendizaje_por_refuerzo#:~:text=Aprendizaje%20por%20refuerzo%20o%20Aprendizaje,noci%C3%B3n%20de%20%22recompensa%22%20o%20premio>

<<http://www.cs.us.es/~fsancho/?e=109>>

<<https://rubenlopezg.wordpress.com/2015/05/12/q-learning-aprendizaje-automatico-por-refuerzo/>>

Aprendizaje por refuerzo (en inglés *reinforcement learning*) es un área del aprendizaje automático inspirada en la psicología conductista, cuyo objetivo es determinar qué acciones debe escoger un agente de *software* en un entorno dado con el fin de maximizar alguna noción de «recompensa» o premio acumulado. Las diferencias con otro tipo de aprendizaje son:

- No se le presentan pares entrada-salida.
- El sistema tiene que obtener experiencia útil acerca de los estados, acciones, transiciones y recompensas de manera activa para poder actuar de manera óptima.
- La evaluación del sistema ocurre a la vez que el aprendizaje.

El modelo básico de aprendizaje por refuerzo consiste en:

- Un conjunto de estados S del entorno que verifican la propiedad de Markov.
- Un conjunto de acciones A .
- Reglas de transición entre estados.
- Reglas que determinan la recompensa inmediata escalar de una transición.
- Reglas que describen lo que observa el agente.

Por ejemplo, en el juego de Mario Bros:

Véase:

<<https://sitiobigdata.com/2019/12/31/reinforcement-learning-con-mario-bros-con-mario-bros-parte-1/>>

- El sistema es el algoritmo que controla a Mario.
- El entorno es el nivel del juego (los enemigos en la pantalla y los bloques que conforman el mundo).
- El objetivo de Mario es interactuar con el entorno de tal manera que obtenga una recompensa.
- Las acciones son: saltar, agacharse, avanzar, disparar... Cada una de esas acciones alterará el entorno. Mario puede observar el cambio, usarlo como una señal de retroalimentación y aprender de él.
- Esos cambios que Mario observa son cambios de estado del entorno.

- El nuevo estado que observa puede generar una señal de recompensa.
- Combinando la acción tomada, el cambio de estado y la recompensa potencial recibida del cambio de estado, Mario comienza a construir un modelo de trabajo para el entorno que está explorando.
- La recompensa se expresa mediante puntos y el mantenimiento de su vida. Si Mario aprende eso cuando salta y aterriza en un enemigo, recibe un aumento de puntos y ya no puede ser asesinado por dicho enemigo. ¡Eso es algo bueno que aprender!
- También podría aprender que, si Mario cae en un hoyo, el juego termina y no hay oportunidad futura de ganar más puntos o de ganar nivel. Estas son cosas que el agente puede aprender con el tiempo, cuanto más interactúa con el entorno más aprende.

Este tipo de algoritmos deben aprender a decidir qué es una buena acción en un entorno lleno de incertidumbre. La retroalimentación se recibe como una señal de recompensa demorada en el tiempo. Para poder razonar sobre por qué se le otorgó o no una recompensa, el sistema debe tener claras las acciones, los objetivos son como definimos la señal de recompensa (puntos, completar nivel, acciones buenas o malas) y la información recibida del entorno.

Los elementos subyacentes del algoritmo son: la política, la señal de recompensa, la función de valor y el modelo óptimo del entorno.

- La política es el corazón del algoritmo. Es lo que le dice qué acción elegir dado el estado del entorno. De alguna manera es lo que ha ido aprendiendo a hacer con el paso del tiempo.
- La señal de recompensa es la forma en que medimos el éxito de nuestro sistema. Es una medida numérica de cómo nos acercamos a la meta. Puede ser positiva o negativa o neutra.
- La función de valor es la forma que se tiene de planificar a largo plazo para «maximizar» el éxito de la tarea. Es lo que denominamos *aprender a jugar*.

Una política π es una relación entre parejas (estado «s», acción «a») que establece cómo de probable es tomar cada acción desde un estado. El valor de un estado «s» bajo una política π se denota $V^\pi(s)$ y es la recompensa que se espera obtener cuando se sigue la política π de ahí en adelante.

De manera similar se puede definir el valor de tomar una acción «a» cuando se está en el estado «s» siguiendo la política π que se denota $Q^\pi(s, a)$, como la recompensa esperada comenzando en «s», tomando la acción «a» y siguiendo la política π a partir de entonces.

El valor de V y de Q se puede estimar a partir de la experiencia. Por ejemplo, si el agente sigue una política y para cada estado guarda una media de las recompensas que se han obtenido a partir de él, esos valores terminarán convergiendo a los de la función V . Si esas medias se separan para cada una de las acciones posibles que se pueden tomar en cada estado, entonces tendríamos un valor asociado a cada acción a tomar desde cada estado, y esos valores se irían aproximando a los de la función Q , a medida que el agente evoluciona mediante experiencia.

En general el ambiente es no-determinístico (tomar la misma acción en el mismo estado puede dar resultados diferentes). Sin embargo, se supone que el ambiente es estacionario (las probabilidades de cambio de estado no cambian o cambian muy lentamente). Las reglas son a menudo estocásticas. La observación implica la recompensa inmediata al escalar asociado con la última transición.

En cada instante de tiempo « t », el agente recibe una observación o_t , que normalmente incluye la recompensa r_t . Se elige entonces una acción a_t , del conjunto de acciones, que actúa sobre el entorno. El entorno transita al nuevo estado s_{t+1} , y se determina la recompensa r_{t+1} , asociada con la transición (s_t, a_t, s_{t+1}) . El objetivo de un agente de aprendizaje por refuerzo es recoger tanta recompensa como sea posible. El aprendizaje por refuerzo es especialmente adecuado para los problemas que incluyen un razonamiento a largo plazo frente a uno a corto plazo.

El objetivo del aprendizaje por refuerzo es extraer qué acciones deben ser elegidas en los diferentes estados para maximizar la recompensa. En cierta forma, buscamos que el agente aprenda lo que se llama una **política**, que formalmente podemos verla como una aplicación que dice en cada estado qué acción tomar. Dividiremos la política del agente en dos componentes: por una parte, cómo de buena cree el agente que es una acción sobre un estado determinado y, por otra, cómo usa el agente lo que sabe para elegir una de las acciones posibles.

En aprendizaje por refuerzo, se tiene que decidir en cada estado la acción que realizar. Este proceso de decisión secuencial se puede caracterizar como un proceso de decisión de Markov. La dinámica del sistema está determinada por una función de transición de probabilidad que mapea estado y acciones a otros estados. Si la secuencia de estados tiene un punto terminal, se denominan *tareas episódicas*, si no se tienen se llaman *tareas continuas*.

Los modelos de recompensas pueden ser de horizonte finito, en el que el agente trata de optimizar su recompensa esperada en los siguientes h pasos sin preocuparse de lo que ocurra después; o puede ser de horizonte infinito (la más utilizada), las recompensas se reducen geométricamente de acuerdo con un factor de descuento positivo pero menor que 1.

El aprendizaje por refuerzo difiere del aprendizaje supervisado en que los pares de entradas / salidas correctas nunca se presentan. Además, en el aprendizaje por

refuerzo hay que encontrar un equilibrio entre la exploración (de un territorio desconocido) y la explotación (de los conocimientos adquiridos hasta el momento).

Cuándo aprender, el sistema puede aprender en cada instante temporal del juego o puede aprender al final de cada episodio.

Los enfoques algorítmicos son de dos tipos. Aquellos que se basan en funciones de valor de los estados y aquellos que se basan en políticas de las parejas (estado, acción). Es decir, aquellos que buscan estimaciones cada vez mejores de las funciones de recompensa a largo plazo y sus optimizaciones. O aquellos que buscan construir una política compleja con una estructura de decisión que le permita intentar tomar la acción óptima en cualquier situación dada.

No todos los sistemas de aprendizaje por refuerzo modelan su entorno. Algunos sistemas aprenden mediante prueba y error, y construyen un modelo de entorno un tanto implícito mediante una función de valor y una combinación de políticas. Otros sistemas crean explícitamente un modelo interno de un entorno, permitiendo al agente predecir los estados resultantes y las recompensas en función de las acciones que desee tomar directamente. No es el caso habitual, ya que en entornos altamente complejos es muy difícil construirlo.

Los principales algoritmos de aprendizaje por refuerzo se desarrollan dentro de los métodos de resolución de problemas de decisión finitos de Markov, que incorporan las ecuaciones de Bellman y las funciones de valor. Los tres métodos principales son: la programación dinámica para agentes de aprendizaje pasivo, los métodos de Monte Carlo y el aprendizaje de diferencias temporales.

Véanse

<https://es.wikipedia.org/wiki/Aprendizaje_por_refuerzo>

<<https://ccc.inaoep.mx/~emorales/Cursos/Aprendizaje2/Acetatos/refuerzo.pdf>>

Programación dinámica: Son formas de descubrir políticas óptimas para resolver problemas, partiendo de un modelo del entorno y representando el problema como un proceso de decisión de Markov.

Métodos de Monte Carlo: Son formas de descubrir políticas óptimas para resolver problemas, y están basados en obtener información a partir de la experiencia. No requieren conocer un modelo del entorno y su dinámica.

Diferencias temporales: Estos métodos combinan los dos anteriores, estiman funciones de valor como en el caso de Monte Carlo y usan otras estimaciones de Q como las proporcionadas por la programación dinámica. Diferencias temporales frente a la programación dinámica no requiere un modelo de entorno, ni de sus recompensas ni la probabilidad de los estados siguientes. Y sobre los métodos de

Monte Carlo es que no necesitan esperar a terminar el episodio completo, sino que se implementan de forma más incremental.

Véase

<https://es.qwe.wiki/wiki/Temporal_difference_learning>

Para cualquier proceso de decisión de Markov finito, hay varias formas de implementar estos procesos de aprendizaje. En este momento nos centraremos en lo que se conoce como *Q-learning*, una forma de aprendizaje por refuerzo en la que el agente aprende a asignar valores de bondad a los pares (estado, acción). Véase

<<https://es.wikipedia.org/wiki/Q-learning>>

Si el agente supiera *a priori* los valores Q de todos los posibles pares (estado, acción) podría usar esta información para seleccionar la acción adecuada para cada estado. Este método permite construir la política expresándola como una tabla exhaustiva de acciones y estados, así como las recompensas acumuladas asociadas a cada pareja. Al principio dicha tabla es completamente desconocida y se trata de ir construyéndola mediante prueba y error. Es decir, el aprendizaje se realiza por prueba y error.

En el algoritmo Q Learning, el **valor** Q de un par (estado, acción) contiene la suma de todas estas posibles recompensas. El problema es que esta suma podría ser infinita en caso de que no haya un estado terminal que alcanzar y, además, es posible que no queramos dar el mismo peso a las recompensas inmediatas que a las recompensas futuras, en cuyo caso se hace uso de lo que se llama un **refuerzo acumulado con descuento**: las recompensas futuras quedan multiplicadas por un factor, $\gamma \in [0,1]$, de forma que cuanto mayor sea este factor, más influencia tienen las recompensas futuras en el valor Q del par analizado.

Matemáticamente, podemos formalizar el cálculo de los valores Q por medio de la siguiente ecuación:

$$Q(s_t, a_t) = r(s_t, a_t) + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1})$$

Esto es, el valor de Q óptimo para un par (estado, acción) es la suma de la recompensa recibida cuando se aplica la acción junto al valor descontado del mejor valor Q que se puede conseguir desde el estado alcanzado al aplicar esa acción. Para aproximar este cálculo, al principio del aprendizaje los valores Q se establecen a un valor fijo (puede ser aleatorio), a continuación, el agente va tomando pares de estado, acción y anota cuánta recompensa recibe en ellos, entonces, actualiza el valor almacenado del valor Q de cada par considerando como ciertas las anotaciones tomadas de los otros pares (algunas de las cuales habrán sido aproximadas en pasos anteriores). Esa función recibe el nombre de *actualización determinista*.

Ejemplo:

Proceso de Decisión de Markov
Aprendizaje por Refuerzo
Aprendizaje Por Refuerzo
Generalización
Ejemplos de Aplicación

Ejemplo

- Suponer el siguiente MDP determinista

- Tabla Q Inicial:

Q(s,a)	a1	a2	a3
s1	0	0	0
s2	0	0	0
s3	0	0	0
s4	0	0	0

- El agente ejecuta el siguiente episodio o secuencia de acciones: $s_1 \rightarrow^{a_1} s_2 \rightarrow^{a_2} s_3 \rightarrow^{a_2} s_4$
- Actualizaciones en la tabla Q:
 - $Q(s_1, a_1) = R(s_1, a_1) + \gamma \arg \max Q(s_2, a) = 0 + \gamma \cdot 0 = 0$
 - $Q(s_2, a_2) = R(s_2, a_2) + \gamma \arg \max Q(s_3, a) = 0 + \gamma \cdot 0 = 0$
 - $Q(s_3, a_2) = R(s_3, a_2) + \gamma \arg \max Q(s_4, a) = 1 + \gamma \cdot 0 = 1$
- Tabla Q resultante:

Q(s,a)	a1	a2	a3
s1	0	0	0
s2	0	0	0
s3	0	0	1
s4	0	0	0

Fernando Fernández y Daniel Borrajo
Aprendizaje Automático
Fernando Fernández y Daniel Borrajo
Aprendizaje Automático

Proceso de Decisión de Markov
Aprendizaje por Refuerzo
Aprendizaje Por Refuerzo
Generalización
Ejemplos de Aplicación

Ejemplo

- Segundo episodio de aprendizaje: $s_1 \rightarrow^{a_2} s_3 \rightarrow^{a_2} s_2 \rightarrow^{a_2} s_4$
- Actualizaciones en la tabla Q:
 - $Q(s_1, a_2) = R(s_1, a_2) + \gamma \arg \max Q(s_3, a) = 0 + \gamma \max(0, 1) = \gamma = 0,5$
 - $Q(s_2, a_2) = R(s_2, a_2) + \gamma \arg \max Q(s_3, a) = 0 + \gamma \cdot 0 = 0$
 - $Q(s_3, a_2) = R(s_3, a_2) + \gamma \arg \max Q(s_4, a) = 1 + \gamma \cdot 0 = 1$
- Tabla Q resultante:

Q(s,a)	a1	a2	a3
s1	0	0,5	0
s2	0	1	0
s3	0	0	1
s4	0	0	0

- Tabla Q óptima:

Q*(s,a)	a1	a2	a3
s1	0,5	0,5	0,25
s2	0,25	1	0,5
s3	0,5	0,5	1
s4	0	0	0

- Política óptima:
 - $\pi^*(s_1) = \arg \max Q(s_1, a) = a_3$
 - $\pi^*(s_2) = a_2$
 - $\pi^*(s_3) = a_1$
- Otra política óptima: igual que la anterior pero con $\pi^*(s_1) = a_2$

Fernando Fernández y Daniel Borrajo
Aprendizaje Automático
Fernando Fernández y Daniel Borrajo
Aprendizaje Automático

Proceso de Decisión de Markov
Aprendizaje por Refuerzo
Aprendizaje Por Refuerzo
Generalización
Ejemplos de Aplicación

Exploración vs. Explotación

- Métodos de balancear la exploración/explotación
 - Estrategias de selección de acciones:
 - ϵ -greedy:
 - Ejecuta $\arg \max Q(s, a)$ con probabilidad ϵ
 - Ejecuta una acción aleatoria con probabilidad $1 - \epsilon$
 - Softmax:

$$P(a) = \frac{e^{Q(s,a)/\tau}}{\sum_{a' \in \mathcal{A}} e^{Q(s,a')/\tau}}$$
 - Inicialización de la función Q
 - Seguir la selección de acciones con conocimiento del dominio adicional

Fernando Fernández y Daniel Borrajo
Aprendizaje Automático

Una variante de la ecuación anterior suele ser la siguiente:

$$Q'(s_t, a_t) = (1-\nu) Q(s_t, a_t) + \nu [r(s_t, a_t) + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1})]$$

Esta segunda ecuación intenta que la actualización de la función sea más gradual, no permitiendo cambios en una determinada dirección de forma tan brusca, para ello, introduce un factor de aprendizaje, ν , que controla la variación de Q . El nuevo valor de Q es la combinación ponderada del antiguo valor de Q y la nueva información que el agente debe creer. Esa función recibe el nombre de *actualización no determinista*.

- Después en cada jugada se selecciona una acción « a_t », observa una recompensa « $r(s_t, a_t)$ », introduce un estado nuevo « s_{t+1} », que depende del estado anterior « s_t » y de la acción seleccionada, y Q se actualiza haciendo una media ponderada del valor antiguo y la información nueva:

Siendo:

- $Q'(s_t, a_t)$ el nuevo valor.
- $Q(s_t, a_t)$ el valor antiguo.
- $r(s_t, a_t)$ recompensa.
- γ factor de descuento.
- ν velocidad de aprendizaje.
- $\max_{a_{t+1}} Q(s_{t+1}, a_{t+1})$ estimación del valor futuro óptimo.

Entonces:

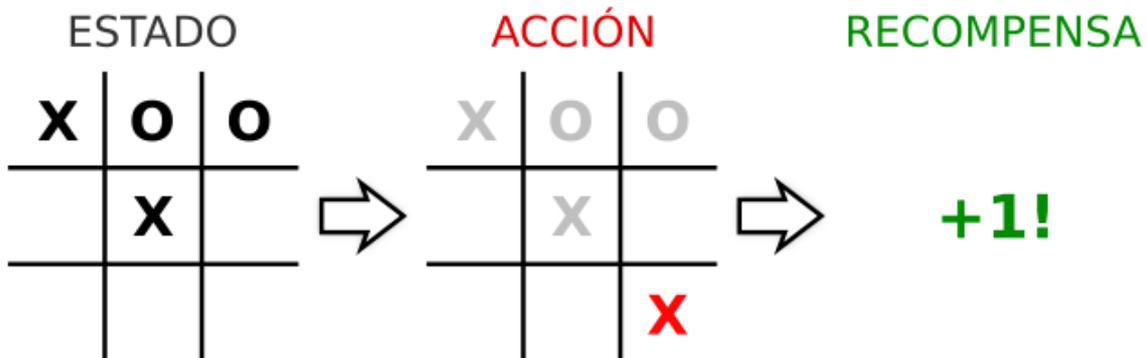
$$Q'(s_t, a_t) = (1-\nu) Q(s_t, a_t) + \nu [r(s_t, a_t) + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1})]$$

Un aspecto importante del aprendizaje por refuerzo es la restricción de que solo se actualizan los valores Q de acciones que se ejecutan sobre estados por los que se pasa, no se aprende nada de acciones que no se intentan. Pero puede ser interesante que, sobre todo al principio del aprendizaje, el agente intente una gama más amplia de acciones, para hacerse una idea de lo que funciona y lo que no. Más concretamente, en cualquier momento el agente puede elegir entre:

- Seleccionar una acción con el valor Q más alto para ese estado, lo que se denomina *explotación*, o
- seleccionar una acción al azar, lo que se denomina *exploración*.

Se puede comprobar que estos valores Q convergen al valor óptimo que estamos buscando.

Ejemplo el 3 en raya



Vamos a intentar aprender cuál es la mejor jugada para cada estado del tablero. Para eso, definamos una matriz cuyas filas corresponden a estados del tablero, y cuyas columnas corresponden a acciones. Cada celda de esta tabla contendrá la recompensa recibida para una combinación (estado, acción).

Si ahora analizamos muchas partidas de 3 en raya, la matriz contendrá todas las recompensas posibles.

Tabla de recompensas directas al ejecutar una acción (A) en un estado (E). Se muestran solo algunas filas y columnas, pero la tabla es mucho mayor...

A \ E	x	x	...	x	x
	0	0	...	0	0
x	0	0	...	0	0
...
x x	0	1	...	0	0
x o	0	0	...	0	1
...

Cada casilla del tablero puede estar en tres estados (vacía, con nuestra X, con la O del oponente). Como tenemos 9 casillas, por lo tanto, hay 3^9 estados, es decir, 19 683.

En cualquier situación un jugador puede poner su ficha en un máximo de 9 casillas.

Por lo tanto, como máximo la matriz de recompensas tendría $19\ 683 \times 9$, aproximadamente 200 000 celdas. En ella están contenidas muchos estados

imposibles, y no todas las acciones son válidas, ya que habrá casillas ocupadas. Ya hablaremos de eso más tarde.

En los estados en que es posible recibir una recompensa, es fácil decidir cuál es la mejor acción: la que obtiene la mayor recompensa. Si ahora intentamos utilizar esta tabla para jugar una partida, vemos que nos falta mucha información. Solo conocemos las recompensas recibidas cuando ganamos una partida, pero ¿qué hacemos en todas las jugadas intermedias?

Rellenar la tabla para las jugadas intermedias es precisamente el objetivo del algoritmo Q-learning. En una jugada intermedia, ¿qué hace que una acción sea mejor que las demás? Q-learning utiliza un concepto genérico y fácil de calcular: la recompensa a largo plazo. La idea es construir una tabla que, en lugar de recompensas directas, contenga una mezcla de la recompensa directa (a corto plazo) y las recompensas a largo plazo.

La recompensa a largo plazo es la que se espera acumular a la larga si en cada estado realizamos la mejor acción posible. La recompensa mixta (*discounted reward*) es una combinación de la recompensa a largo plazo con la recompensa directa, y se calcula de forma voraz (*greedy*).

Veamos un ejemplo de lo que sucede en el algoritmo, empezando por el final de la partida y volviendo a la jugada anterior.

$$r\left(\begin{array}{|c|c|c|} \hline x & o & o \\ \hline & x & \\ \hline & & \\ \hline \end{array}, \begin{array}{|c|c|c|} \hline x & o & o \\ \hline & x & \\ \hline & & x \\ \hline \end{array}\right) = 1$$

Recompensa directa para un movimiento ganador

Un posible estado previo podría ser

$$r\left(\begin{array}{|c|c|c|} \hline x & o & \\ \hline & & \\ \hline & & \\ \hline \end{array}, \begin{array}{|c|c|c|} \hline x & o & \\ \hline & & x \\ \hline & & \\ \hline \end{array}\right) = 0$$

Recompensa directa para un estado intermedio

En ese estado intermedio, realizar la acción indicada en rojo no tiene ninguna recompensa directa, pero acabamos de ver que a largo plazo vamos a recibir una recompensa.

Ahora que tenemos dos estados consecutivos, podemos introducir el concepto de *experiencia*. Una experiencia incluye un estado, la acción realizada, la recompensa recibida, y como novedad, el estado siguiente tras realizar esta acción.

Ojo, porque en un juego por turnos este concepto del estado siguiente es un poco confuso. En principio podríamos pensar que, partiendo de un tablero vacío, si el algoritmo pone una X, el estado siguiente es un tablero con una casilla cubierta. Sin embargo, esto no es así. Q-learning solo se preocupa por aprender qué hacer cuando es su turno. Tras cubrir el tablero con una X, es el turno del rival. Cuando el rival haya colocado una O, volverá a ser el turno del algoritmo. Este es el estado del tablero que debemos usar como estado siguiente en las experiencias.



Experiencia (estado, acción, recompensa, estado siguiente)

Es decir, partiendo del estado de la izquierda, realizando la acción en rojo, no recibimos recompensa, y tras jugar nuestro rival, nos encontramos con el estado que figura a la derecha (vuelve a ser nuestro turno). Es cierto que en este caso el rival está realizando una mala jugada, pero sirve para ejemplificar de manera sencilla el algoritmo.

Con esa información, y la matriz que ya tiene algunas recompensas directas, Q-learning ya puede trabajar para rellenar más celdas de la matriz. Como se menciona al principio de esta sección, ya sabemos cuál es la mejor jugada cuando nos encontramos con el estado de la derecha, y sabemos qué recompensa produce. Utilizaremos esa recompensa directa en el estado de la derecha para aprender algo sobre la recompensa a largo plazo asociada al estado de la izquierda y la acción en rojo.

Para un estado y una acción es posible tener múltiples experiencias. En el 3 en raya, las experiencias que encontraremos dependen de lo que decida hacer el rival. Por eso, hay que usar las experiencias con cautela y aprender solo un poco de cada una.

Veamos ahora el algoritmo. Todo lo que se necesita almacenar en memoria durante el aprendizaje es una tabla como la mencionada anteriormente con las recompensas

acumuladas para estados y acciones. Esta tabla va a contener la recompensa mixta para cada celda. En realidad, tiene nuestra mejor estimación de esta recompensa, que al principio será un desastre, y a medida que aprende de distintas experiencias, se va volviendo más y más precisa.

Algoritmo Q-learning:

- Inicializa $Q(s, a)$ arbitrariamente.
- Repetir:
 - Inicializa «s».
 - Repetir:
 - Elija una acción «a» desde «s» mediante la política que depende de Q .
 - Ejecute la acción «a». Observe la recompensa «r» y el siguiente estado «s'».
 - Recalcule $Q(s, a) = (1-\nu) Q(s, a) + \nu[r(s, a) + \gamma \max_{a'} Q(s', a')]$.
 - $s = s'$.
 - hasta que «s» sea terminal.
- Hasta que no haya más episodios.

El algoritmo necesita dos parámetros que debemos ajustar en función del problema que estemos resolviendo:

- Velocidad de aprendizaje (*learning rate*). Es un valor entre 0 y 1 que indica cuánto podemos aprender de cada experiencia. 0 significa que no aprendemos nada de una nueva experiencia, y 1 significa que olvidamos todo lo que sabíamos hasta ahora y nos fiamos completamente de la nueva experiencia.
- Factor de descuento (*discount factor*). Es también un valor entre 0 y 1 que indica cuán importante es el largo plazo. 0 significa que solo nos importan los refuerzos inmediatos, y 1 significa que los refuerzos inmediatos no importan, solo importa el largo plazo. Ojo, porque valores muy cercanos a 1 tienden a diverger. Este factor nos ayuda a mezclar recompensas directas con recompensas a largo plazo y producir la recompensa mixta.

En ambos parámetros los extremos son poco útiles. La velocidad de aprendizaje se puede ajustar en función de la incertidumbre respecto a los estados siguientes en las experiencias. Por ejemplo, en el 3 en raya la incertidumbre viene de las posibles jugadas del rival, que puede realizar hasta 9 acciones diferentes. Por tanto, una velocidad de aprendizaje superior a $1/9$ da un peso excesivo a las nuevas experiencias y olvida demasiado rápido las anteriores.

El factor de descuento establece un balance entre el refuerzo inmediato y el refuerzo a largo plazo. En el caso del 3 en raya, solo recibimos refuerzos a corto plazo cuando se acaba la partida, así que este factor no importa mucho, cualquier

valor entre 0,1 y 0,9 funcionará muy bien. En otros problemas donde se puedan recibir refuerzos intermedios, debemos decidir dónde poner la balanza.

Una vez finalizado el aprendizaje, después de haber jugado un número adecuado de partidas, ya se puede utilizar la matriz para tomar decisiones.

La gran pregunta es cómo encontrar un equilibrio razonable entre explorar y hacer caso de la tabla. Existen múltiples opciones, pero una bastante común consiste en usar los refuerzos que podemos encontrar en la tabla para las diferentes acciones como una distribución de probabilidad. De esa forma, las acciones que produzcan un mayor refuerzo serán elegidas más a menudo que las acciones que produzcan un menor refuerzo, pero todas las acciones tendrán su oportunidad en algún momento.

Y ahora, un truco. Si no se quiere pasar innumerables horas jugando contra el ordenador al 3 en raya para que aprenda, ¡ponedlo a jugar contra sí mismo! En menos de un minuto se puede jugar un millón de partidas, más que suficiente para que aprenda a jugar como un maestro a este juego tan simple.

- Ventajas:
 - Es eficaz en problemas pequeños. Si no hay demasiados estados, estos se van a visitar a menudo, lo que actualizará la tabla Q a menudo, llegando antes a la solución óptima.
 - En juegos más complejos en los que se puedan recibir recompensas no solo al final de la partida, sino también durante la partida (por ejemplo, en ajedrez se pueden comer fichas al rival), es importante combinar ambos tipos de recompensa, a corto y largo plazo. Por cierto, si alguien está pensando en aplicar esta técnica al ajedrez, cuidado con el número de estados, aproximadamente un 2 seguido de 40 ceros. Como anécdota, ni todos los discos duros del mundo juntos tienen tanta capacidad de almacenamiento.
- Desventajas:
 - Una de las mayores limitaciones aparece cuando intentamos diseñar la representación del estado. Si la representación es muy general, puede que no contenga la información necesaria para dotar al sistema de la información adecuada. Por el contrario, si es demasiado detallado, habrá muchos estados y la actualización de Q será más lenta.

En general, los algoritmos de aprendizaje por refuerzo escalan bastante mal, es decir, cuando el número de estados y acciones posibles es muy elevado, el aprendizaje resulta extremadamente lento.

Otro hándicap que los caracteriza es que siempre hay que alcanzar un compromiso entre lo que ya se sabe y lo que todavía queda por aprender a la hora de tomar decisiones para encontrar una estrategia mejor.

Para saber más: Véase

<<https://ccc.inaoep.mx/~emorales/Cursos/Aprendizaje2/Acetatos/refuerzo.pdf>>

Sutton, R.S., Barto, A. G.: *Introduction to Reinforcement Learning*. MIT Press, Cambridge, MA, USA, 1st edn. (1998).

Para saber más: Véase

<<http://dit.upm.es/~anto/tesis/html/stateart.html#31479>>

El campo de los juegos de tablero, basados en el aprendizaje por refuerzo.

Tipos de algoritmos bio-inspirados

Durante bastantes años, las líneas de trabajo en inteligencia artificial se focalizaron en crear algoritmos que mostraran habilidades humanas, pero durante su proceso evolutivo se habían dejado de lado todos aquellos aspectos relacionados con la inteligencia basada en comportamientos biológicos diferentes del humano, como, por ejemplo, los sistemas evolutivos basados en la genética, los sistemas celulares, los sistemas neuronales, el sistema inmunológico, los sistemas colectivos, los fractales... A todo ello es lo que se ha dado en llamar la *inteligencia artificial bio-inspirada*. Veamos algunos de ellos a continuación.

Redes neuronales artificiales (*modelo conexionista*)

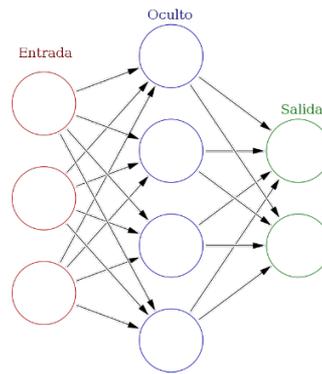
Con anterioridad a los éxitos de las redes neuronales y, sobre todo, de las redes neuronales profundas, el modelo dominante de la IA había sido el simbólico. No hay más que ver todo lo que se ha contado hasta ahora.

Cuando se habla de IA simbólica se hace referencia a un conjunto de símbolos que, mediante una serie de reglas, pueden combinarse para formar estructuras mayores y transformarse en otras. Tales procedimientos permiten crear símbolos nuevos, generar y modificar las relaciones entre ellos, almacenarlos, determinar si dos símbolos son iguales o no... Tales símbolos son además físicos, en el sentido de que su existencia depende de que haya un sustrato material (electrónico en el caso de los computadores, o químico-biológico en el caso de los seres vivos).

Se presupone (no sin discusión) que todo eso es lo necesario para llevar a cabo acciones inteligentes. En este caso, el objetivo de la IA general no es otro que intentar demostrar esa hipótesis en el contexto de los computadores, en el otro contexto está ya demostrado experimentalmente.

Las redes neuronales cambian el enfoque, y se basan en la hipótesis de que la inteligencia general emerge a partir de la actividad distribuida de un gran número de unidades interconectadas (las neuronas artificiales). Las redes neuronales, también procesan símbolos, pero estos no son explícitos, sino que se encuentran repartidos en la red. Esa es la razón de que la IA simbólica pueda explicar el funcionamiento, mientras que la IA neuronal obtiene resultados, pero no puede explicar cómo ha llegado a las conclusiones que llega.

Las redes neuronales ([Lippmann, 87], [Freeman y Skapura, 93], [Hertz *et alii*, 91]), incluidas dentro de los modelos conexionistas, son sistemas formados por un conjunto de sencillos elementos de computación llamadas *neuronas artificiales*.



Este modelo computacional está vagamente inspirado en el comportamiento observado en su homólogo biológico. Consiste en un conjunto de unidades llamadas *neuronas artificiales*, conectadas entre sí para transmitirse señales. La información de entrada atraviesa la red neuronal (donde se va sometiendo a diversas operaciones) generando unos valores de salida.

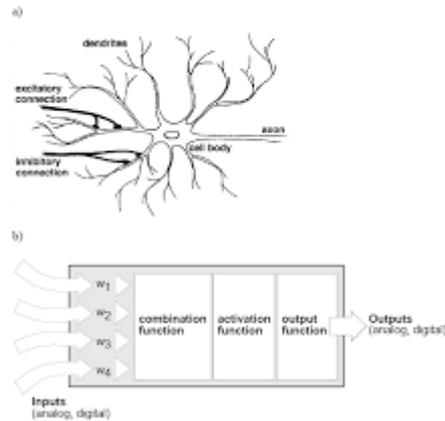
Comparadas con el cerebro, las redes neuronales artificiales (RNA) son demasiado:

- Claras, porque priorizan la elegancia matemática y la potencia, mientras que el cerebro no.
- Simples, porque una sola neurona real (hay aprox. 30 clases diferentes) es más compleja.
- Escasas, porque incluso las RNA con millones de unidades son minúsculas comparadas con el cerebro humano.
- Pobres, porque los investigadores ignoran además de los factores temporales como frecuencias neuronales en su punto máximo y sincronías, la biofísica de las espinas dendríticas, los neuromoduladores, las corrientes sinápticas y el paso de iones.

Aunque todas estas limitaciones se están reduciendo con el paso del tiempo.

De manera sencilla el diagrama de bloques de una neurona estaría formado por:

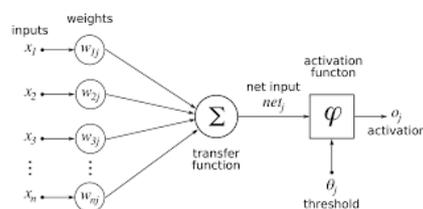
- Señales de entrada.
- Función de combinación de estas señales.
- Función de activación, que permite activar o inhibir el estado de las neuronas adyacentes.
- Función de salida, con una función limitadora o umbral que modifica el valor resultado o impone un límite que no se debe sobrepasar antes de propagarse a otra neurona.



Estas neuronas están interconectadas a través de unas conexiones con unos pesos asociados, que representan el conocimiento en la red. Cada neurona calcula la suma de sus entradas, ponderadas por los pesos de las conexiones, le resta un valor umbral y le aplica una función no lineal (por ej., una sigmoide); el resultado sirve de entrada a las neuronas de la capa siguiente (en redes como el perceptrón multicapa).

Redes neuronales artificiales

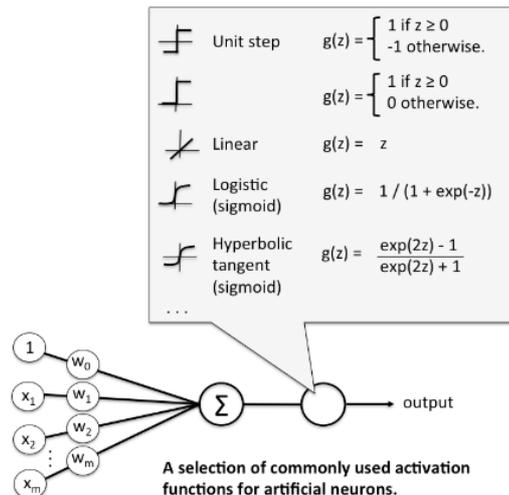
El perceptrón simple, creado por Frank Rosenblatt, es la unidad básica de las redes neuronales y tiene la siguiente estructura



Donde:

- $x_0 = 1$ y $w_0 = b$ valor del umbral de inhibición de la neurona.
- x_1, x_2, \dots, x_n conjunto de entradas a la neurona.
- w_1, w_2, \dots, w_n conjunto de pesos que se irán ajustando de forma automática a medida que la red neuronal vaya aprendiendo.
- Σ función de agregación de las entradas ponderadas por sus pesos.
- φ o g función de activación que se encarga de mantener el conjunto de valores de salida en un rango determinado, normalmente (0, 1) o (-1, 1).

Existen diferentes funciones de activación.



- $g(z)$ Es el valor de salida del perceptrón que viene definido dependiendo de la función de activación:
 - Definiendo $z = (w_1 x_1 + w_2 x_2 + \dots + w_n x_n) + b$.
 - Si $g(z)$ es la función salto:
 - $g(z) = 1$ si $z > 0$.
 - $g(z) = 0$ en cualquier otro caso.
 - Si $g(z)$ es la función lineal:
 - $g(z) = z$.
 - Si $g(z)$ es la función sigmoide:
 - $g(z) = \frac{1}{(1 + e^z)}$.
 - ...

Funcionamiento del perceptrón simple

Ya hemos dicho que el perceptrón es la red neuronal más básica que existe y es un ejemplo de aprendizaje supervisado. El entrenamiento del perceptrón consiste en determinar los valores «w» (pesos) y el valor umbral « x_0 » que mejor hagan que la entrada se ajuste a la salida «z».

El algoritmo que sigue es el siguiente:

- Inicialización de pesos y umbral, normalmente con valores aleatorios.
- Se realiza un bucle para todos los ejemplos hasta que el resultado de los pesos sea aceptable:
 - Leer valores de entrada «x».
 - Calcular el valor de salida «z».
 - Calcular el error entre lo que se obtiene «z» y el valor asociado «y» a los valores de entrada «x».
 - Actualizar los pesos y el umbral en función del error (Aprendizaje).

Las reglas de aprendizaje pueden ser, por ejemplo, alguna de las siguientes:

- Tipo 1

$$\text{error} = (y_{\text{esperado}} - z_{\text{perceptron}})$$

$$w_{\text{jnuevo}} = w_{\text{janterior}} + \text{error} \cdot x_j$$

- Tipo 2

$$\text{error} = (y_{\text{esperado}} - z_{\text{perceptron}})$$

$$w_{\text{jnuevo}} = w_{\text{janterior}} + \alpha \cdot \text{error} \cdot x_j, \text{ siendo } 0 < \alpha < 1 \text{ «tasa de aprendizaje»}$$

Un ejemplo

- Diseñar un perceptrón que aprenda a realizar la función binaria NAND definida mediante la siguiente tabla de verdad

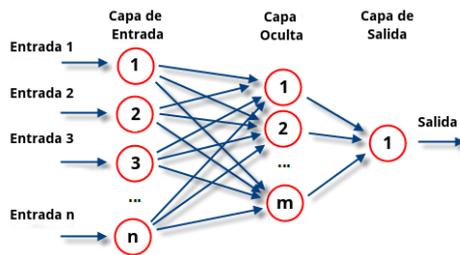
x_1	x_2	$x_1 \text{ nand } x_2$
0	0	1
0	1	1
1	0	1
1	1	0

- $x_0 = 1$.
- $b = 0$.
- umbral (t) = 0,5.
- tasa de aprendizaje $\alpha = 0,1$.
- Conjunto de entrenamiento, que consiste en cuatro muestras con la siguiente estructura $((x_0, x_1, x_2), y_{\text{esperado}})$:
 - $\{[(1, 0, 0), 1], [(1, 0, 1), 1], [(1, 1, 0), 1], [(1, 1, 1), 0]\}$

Las cuentas y un código en Python los pueden encontrar en extraído de
<https://es.wikipedia.org/wiki/Perceptr%C3%B3n>

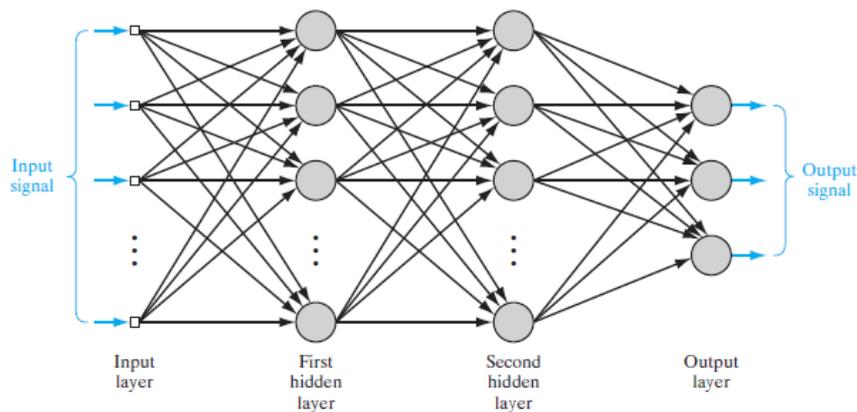
La principal desventaja del perceptrón simple es que solo sirve para clasificar problemas linealmente separables, cosa que ya se podía hacer mediante métodos estadísticos, y de una forma mucho más eficiente.

El siguiente paso consistió en intentar resolver problemas más complejos para lo que se diseñó lo que se conoce como *perceptrón multicapa*, que se puede interpretar como un apilamiento de perceptrones.



Dicho perceptrón puede diseñarse de varias maneras:

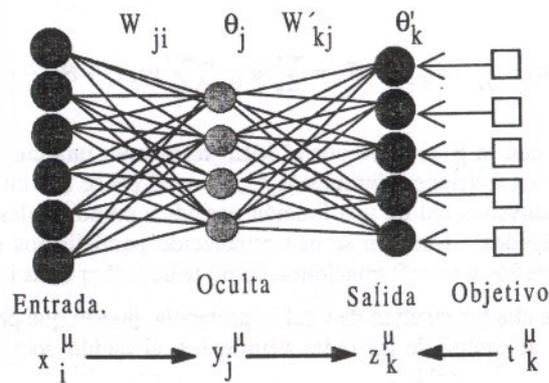
- Cada salida de una neurona de la capa oculta «i» es una entrada para todas las neuronas de la capa «i+1».
- Cada salida de una neurona de la capa oculta «i» es entrada de unas ciertas neuronas de la capa «i+1».



El algoritmo es bastante análogo al del perceptrón salvo el paso de aprendizaje.

Cada patrón de entrenamiento se propaga a través de la red y sus parámetros para producir una respuesta en la capa de salida, la cual se compara con los patrones objetivo o salidas deseadas para calcular el error en el aprendizaje, este error marca el camino más adecuado para la actualización de los pesos y los umbrales que al final del entrenamiento producirán una respuesta satisfactoria a todos los patrones de entrenamiento, lo que se logra «minimizando» el error medio cuadrático en cada iteración del proceso de aprendizaje.

Supóngase una sola capa oculta



Notación:

- « x_i » las « n » entradas a la red dadas por un patrón de aprendizaje.
- « y_j » las « o » salidas de la capa oculta.
- « z_k » las « m » salidas de la capa final.
- « c_k » las « m » salidas objetivo a alcanzar.
- « w_{ij} » los pesos de la capa oculta.
- « θ_j » sus umbrales correspondientes.
- « w'_{ij} » los pesos de la capa de salida.
- « θ'_k » sus umbrales respectivos.
- Función de activación para la capa final es la sigmoide.
- Función de activación para la capa oculta, la lineal.

Funcionamiento:

- Paso 1. En $t = 0$, inicialización de pesos y umbral, normalmente con valores aleatorios.
- Paso 2. Para cada patrón « r » del conjunto de entrenamiento \mathbf{x}^r ($r = 1, \dots, N$)
 - Calcular z_k^r las « m » salidas de la capa final

$$z_k^r = (w'_{k1} y_1^r + w'_{k2} y_2^r + \dots + w'_{ko} y_o^r) - \theta'_k$$

Como a su vez cada y_j es la salida de la capa de entrada, entonces

$$z_k = (w'_{k1} f(w_{11} x_1^r + w_{12} x_2^r + \dots + w_{1n} x_n^r - \theta_1) + \dots + w'_{ko} f(w_{o1} x_1^r + w_{o2} x_2^r + \dots + w_{on} x_n^r - \theta_o)) - \theta'_k$$

siendo f la función de activación sigmoidea

- Paso 3. Se calcula el error total cometido sumando cada uno de los N errores cometidos con cada patrón, que a su vez tiene « m » errores de cada salida z_k^r y las « m » salidas objetivo a alcanzar c_k^r :
 - Suponiendo que se calcula el error cuadrático medio

$$\text{Error} = \frac{1}{2} \{ [(c_1^1 - z_1^1)^2 + \dots + (c_m^1 - z_m^1)^2] + \dots + [(c_1^N - z_1^N)^2 + \dots + (c_m^N - z_m^N)^2] \}$$

Obsérvese que, a su vez, esa expresión depende de los diferentes valores «w, w' de la capa de salida y de la capa oculta» y de los diferentes valores «θ y θ' de los umbrales correspondientes» como aparecen en la expresión del paso 2 a través de los valores «z». Es decir, la función error depende de

$$\text{Error} (w, w', \theta, \theta')$$

- Paso 4. En este momento se exige que dicha función «Error» sea mínima, con respecto a todos esos parámetros (w, w', θ, θ'). Para ello se puede aplicar el método del gradiente. Lo que permitirá calcular la variación de los pesos de w', de w, θ y θ'.

Las expresiones obtenidas llevan implícito el concepto de *propagación hacia atrás*, ya que en primer lugar se calcula la señal de error proporcional al error de la salida actual de la red, a partir de la cual se calculan las señales de error de la capa oculta.

- Paso 5. Se actualizan los pesos y los umbrales.
- Paso 6. Se calcula el error total, se pasa de t a t+1, y se vuelve al paso 2.

El algoritmo se puede extender a arquitecturas con más de una capa oculta siguiendo este esquema de retropropagación del error. Véase

<<http://www.sc.ehu.es/ccwbayes/docencia/mmcc/docs/t8neuronales.pdf>>

Las funciones de activación suelen ser las ya conocidas:

- Sigmoides, que se suele usar solo en la capa de salida.
- TanH, alternativa a la sigmoide en las capas ocultas, no en la salida.
- ReLU (lineal), la más usada en las capas ocultas.

Ventajas:

- Estos sistemas aprenden y se forman a sí mismos, en lugar de ser programados de forma explícita, y sobresalen en áreas donde la obtención de soluciones es difícil de expresar con la programación convencional.

- Una de las grandes ventajas de este tipo de redes es que pueden aprovechar su naturaleza paralela para reducir el tiempo requerido por un procesador secuencial para determinar la correspondencia entre unos patrones dados. A pesar de ello, para aquellas aplicaciones que deban explorar un gran espacio de entrada o que intenten correlacionar todas las permutaciones posibles de un conjunto de patrones muy complejo, el tiempo de computación necesario se hace bastante grande.
- Las tasas de error de las redes neuronales son equivalentes a las de las reglas generadas por los métodos de aprendizaje simbólicos, aunque son algo más robustas cuando los datos son ruidosos.
- Si las funciones de activación son no lineales, la salida final de la red es una función no lineal de sus entradas. Debido a esto, la mayoría de las redes neuronales usan funciones de activación no lineales.

Desventajas:

- Una desventaja es que no existe una técnica para determinar el número de capas ocultas, ni el número de neuronas que debe contener cada una de ellas para un problema específico, esta elección queda a elección del diseñador y de su posible experiencia, así como de técnicas de optimización que permiten optimizar parámetros en función de análisis de errores durante el proceso de entrenamiento y diseño.
- El aprendizaje es bastante más lento que en un sistema de aprendizaje simbólico.
- El conocimiento obtenido por las mismas no es representable en forma de reglas inteligibles, sino que lo forma el conjunto de pesos de las conexiones interneuronales.
- Además, es difícil incorporar conocimiento de base o interacción del usuario en el proceso de aprendizaje de una red neuronal.
- Las redes neuronales actuales suelen contener desde unos miles a miles de millones de unidades neuronales. Lo cual requiere grandes bases de datos con patrones clasificados y una gran potencia de cálculo.

Las redes neuronales han sido utilizadas con éxito en diferentes tipos de problemas:

- Autoasociación: la red genera una representación interna de los ejemplos aportados, y responde con el más aproximado a su «memoria». Ejemplo: máquina de Boltzman.

- Clasificación de patrones: la red es capaz de clasificar cada entrada en un conjunto predefinido de clases. Ej.: *back-propagation*.
- Detección de regularidades: la red se adapta a los ejemplos de entrada, tomando de ellos varias características para clasificarlos; en este caso, el conjunto de clases no está definido de antemano, por lo que el aprendizaje es no supervisado. Ej.: red MAXNET, ART1, mapas de Kohonen, red de Oja, etc.

Comentario. Véase el teorema de aproximación universal

<https://en.wikipedia.org/wiki/Universal_approximation_theorem>

Este teorema afirma que no existe ninguna razón teórica para utilizar más de dos capas ocultas, de hecho, la mayoría de los problemas prácticos se resuelven con una sola capa oculta con un número finito de neuronas. Estas pueden ser entrenadas para aproximar cualquier función arbitraria al azar cuando se les dan los pesos adecuados. Ahora bien, no se indica cómo construir los pesos, tan solo establece que tal construcción es posible.

Lo que sí se ha demostrado experimentalmente es que se aprende mejor en la práctica con múltiples capas ocultas, es decir, redes más profundas.

La importancia de las capas ocultas es que cada capa almacena la representación interna abstracta de los datos abstractos almacenados en la capa oculta anterior, y así hasta llegar a la capa de los datos.

El número de neuronas en las redes multicapa sigue una forma piramidal, con un número decreciente de neuronas de la entrada hacia la salida. Existen excepciones cuando las redes requieren que el número de entradas sea igual al número de salidas.

Por ejemplo, para una red de tres capas (con una sola capa oculta), el número inicial de neuronas en la capa oculta es igual a la raíz cuadrada del producto del número de neuronas de salida por el número de neuronas de entrada.

Por ejemplo, una red neuronal de 3 capas con 8 entradas y 2 salidas. La capa oculta sería de 4 neuronas.

Esta regla, no es más que una aproximación burda del tamaño de las capas ocultas...

Las redes neuronales se pueden adjectivar como redes neuronales *feed-forward* (prealimentada) que no es más que una red neuronal artificial donde las conexiones entre las unidades no forman ciclo, es decir, la información se mueve en una única dirección. Posteriormente se verán las redes neuronales recurrentes que utilizan los ciclos en su definición.

No hay que confundir una red *feed-forward*, con un método de aprendizaje de redes que se denomina *back propagation* o propagación hacia atrás. En este caso lo que se propagan son los errores durante el cálculo del gradiente en el caso de algoritmos de aprendizaje supervisado utilizados para entrenar redes neuronales artificiales.

Introducción

El actual entusiasmo por la IA se debe a los recientes logros de lo que se conoce como *aprendizaje profundo* que se ha aplicado con éxito al reconocimiento de imágenes, los juegos de tablero y el procesamiento del lenguaje.

Al parecer, todo comenzó en 1912 con el éxito del diseño de lo que se llama *red neuronal convolucional*, que fueron introducidas en 1980 a partir de los estudios sobre el sistema visual de los animales. En ese año, dicha red consiguió un éxito del 85 % al clasificar, entre mil categorías posibles, 150 000 imágenes de la base de datos ImageNet. Una propiedad interesante de esas redes es que son invariantes en escala y traslación, lo que significa que no importan ni el tamaño ni la posición de los objetos presentes en la imagen. Sin embargo, no son invariantes frente a rotaciones, por lo que fallan a la hora de reconocer la misma imagen boca abajo. De hecho, este problema permanece y es objeto de investigación.

Este gran avance hizo que aparecieran titulares como el siguiente

Los ordenadores superan a los humanos
en el reconocimiento y clasificación de imágenes.
The Guardian, 13 de mayo de 2015.

Lo que no se dice es que no se ha hecho ningún estudio serio comparativo entre la capacidad del ser humano frente a la capacidad de la máquina en este ámbito. Y, por otro lado, se oculta el enorme esfuerzo que se requiere para preparar una red antes de que pueda empezar a aprender, es decir, etiquetar la base de datos de entrenamiento, definición y refinamiento del gran conjunto de parámetros que requiere la red para funcionar... destrezas que cuesta mucho tiempo adquirir basadas en un trabajo de prueba y error. Además, si se cambia el objetivo de la tarea de aprendizaje, hay que volver a empezar.

La capacidad de análisis de una imagen por parte de los humanos no puede desligarse del resto de las facultades que conforman nuestra inteligencia general; en particular, la capacidad de abstracción, de entender el lenguaje y de razonar con sentido común. Nada de eso parece que puede aprenderse a partir de las imágenes de una base de datos, sino que requiere interaccionar con el mundo real. De hecho, se cree que el aprendizaje supervisado no es el mejor camino para alcanzar la IA general. Además, recuerde que los algoritmos de aprendizaje supervisado se basan en la probabilidad de ocurrencia de los sucesos y aquellos que tienen muy poca probabilidad de aparición no serán reconocidos.

Otra técnica que ha generado una gran popularidad es la de los sistemas de aprendizaje por refuerzo. Este enfoque se ha combinado con las redes convolucionales profundas para desarrollar programas que han aprendido a jugar diferentes juegos, llegando a superar a los mejores jugadores humanos. Esta combinación conocida como *aprendizaje profundo con refuerzo*, también ha contribuido de manera significativa a la reciente fiebre de la IA.

Ejemplos comerciales:

- **2010** — Google compra DeepMind por ~450 M \$. **DeepMind** es el referente entre las compañías dedicadas al I+D en el campo de la inteligencia artificial.
- **2014** — DeepMind desarrolla un *software* basado en aprendizaje por refuerzo capaz de aprender a jugar videojuegos (**Atari**) de manera similar a los humanos.
- **2015** — El *software* **AlphaGo**, desarrollado por DeepMind, vence al actual campeón del mundo del juego de estrategia *go*.
- **2017** — El programa **AlphaZero** de DeepMind logra en 24 horas un nivel de juego sobrehumano en los juegos: ajedrez y su equivalente asiático, *shogi* y *go*, derrotando a los respectivos campeones del mundo.
- **2019** — Otro programa desarrollado por DeepMind, **AlphaStar**, consigue ganar a un jugador profesional de uno de los videojuegos de estrategia más complejos: StarCraft II.

Este tipo de problemas se caracteriza por que cada situación que se plantea en el desarrollo del juego es perfectamente observable. Pero qué ocurre cuando se quieren extrapolar este tipo de algoritmos a situaciones cuando el conocimiento que tiene el sistema es imperfecto, lo que es habitual cuando se diseñan sistemas que se tienen que desenvolver en el mundo real, con sistemas de percepción con error... Por el momento no está claro el poder de generalización de este tipo de algoritmos.

Otro de los problemas que se plantean en los algoritmos de aprendizaje por refuerzo es cómo alcanzar el compromiso entre aprovechar lo ya aprendido hasta un momento determinado o seguir explorando para aprender más con la esperanza de encontrar una estrategia mejor.

Por último, el mayor problema de esta técnica reside en su escalabilidad. Cuando el número de estados y acciones posibles es muy elevado, el aprendizaje resulta extremadamente lento y hace falta extender los algoritmos con el concurso de otras ideas.

Otro ámbito de la IA con espectaculares progresos es el del procesamiento del lenguaje natural. El uso de las redes recurrentes profundas ha permitido el avance

tanto en la traducción automática como en los asistentes personales capaces de responder.

Estos grandes avances han hecho que aparezcan titulares como los siguientes

El nuevo servicio de Google traduce idiomas casi tan bien como las personas.
MIT Technology Review, 27 de septiembre de 2016.

Watson de IBM, ya habla con fluidez nueve idiomas (y subiendo).
Wired, 6 de octubre de 2016.

Nuestras redes neuronales han desarrollado un asombroso sentido de la comprensión.
Gereon Frahling,
presidente de la compañía de traducción automática DeepI,
20 de marzo de 2018.

En el momento en el que estoy escribiendo este texto ha surgido un nuevo modelo de lenguaje desarrollado por OpenAI, denominado *GPT-3*. OpenAI es la famosa organización sin ánimo de lucro enfocada en la investigación sobre inteligencia artificial fundada por Elon Musk, y en la que empresas como Microsoft ha invertido mucho.

Se acaba de publicar uno de sus proyectos más impresionantes hasta la fecha. Se trata de su nuevo modelo de lenguaje llamado *GPT-3*, que analiza texto o datos para ofrecer predicciones de palabras en función de todas las palabras anteriores. Es lo que se usa en aplicaciones de procesamiento natural del lenguaje.

Este en particular es el más poderoso hasta ahora debido a su tamaño. *GPT-3* cuenta con 175 000 millones de parámetros. Es tan masivo que su versión anterior, *GPT2*, tenía solo 1500 millones de parámetros.

El resultado es extremadamente básico, es algo impresionante para un modelo de lenguaje, pero no reemplaza el trabajo de un humano profesional que puede producir algo mucho mejor, y sin mucha diferencia de tiempo. Su utilidad actual en el mundo real no es mucha, no va a cambiarlo todo de un día para otro, pero no deja de ser fascinante y estar extremadamente cargado de potencial.

Para generar sus predicciones, *GPT-3* se basa en un modelo probabilístico generado básicamente porque se ha «comido» todo el texto disponible en Internet, es decir, se ha leído todo lo escrito sobre cualquier tema y alguna vez se haya publicado en la red. De ahí que cuando se le inyecta un texto, el modelo predice lo que debería venir después. Por ello le puedes preguntar sobre cualquier cosa y

siempre tiene una respuesta. Básicamente, puede generar una respuesta, pero esto no quiere decir que la entienda, de hecho, conforme se alarga la respuesta, el texto va perdiendo consistencia y sentido. Como carece de razonamiento abstracto, es fácil que cometa errores que una persona no cometería jamás.

La comprensión del lenguaje natural es una tarea endemoniadamente difícil, ya que el lenguaje no solo es ambiguo y dependiente del contexto, sino que su uso presupone una gran cantidad de conocimientos generales. Por ello, a pesar de los éxitos, seguimos estando muy lejos del nivel humano.

Descripción

Una red neuronal profunda (DNN) es una red neuronal artificial (ANN) con varias capas ocultas entre las capas de entrada y salida. Al igual que en las ANN poco profundas, las DNN pueden modelar relaciones no lineales complejas. Las redes neuronales se utilizan ampliamente en el aprendizaje supervisado y en los problemas de aprendizaje por refuerzo. En el aprendizaje profundo, el número de capas ocultas, en su mayoría no lineales, puede ser grande; digamos unas 1000 capas.

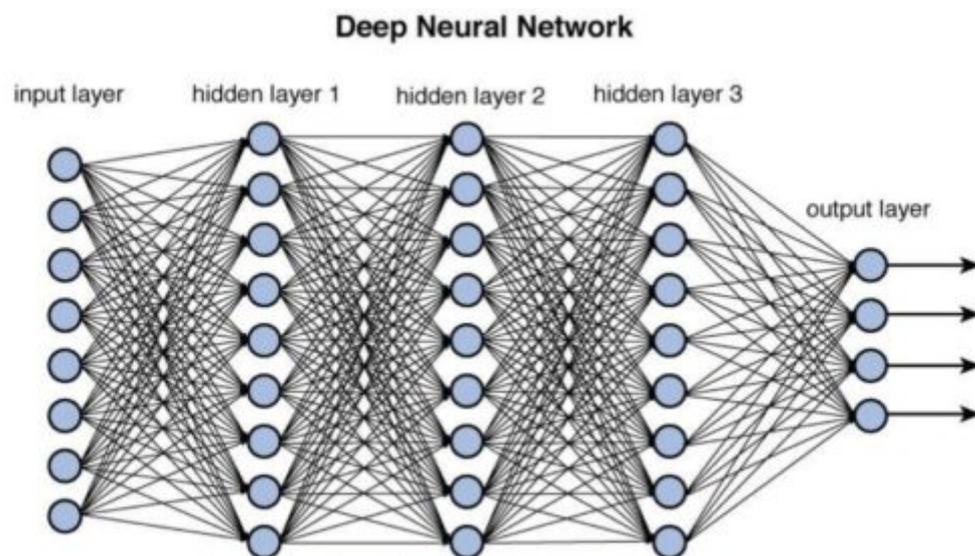


Figure 12.2 Deep network architecture with multiple layers.

Principalmente se utiliza el método del gradiente para optimizar la red y minimizar la función de coste. Además, el entrenamiento se realiza utilizando el algoritmo de *Backpropagation*.

El algoritmo en este caso viene dado por:

Notación:

- « x_i » las « n » entradas a la red dadas por un patrón de aprendizaje.
- « y_j » las « o » salidas de la capa oculta.
- « z_k » las « s » salidas de la capa final.
- « c_k » las « s » salidas objetivo a alcanzar.
- « w_{ij} » los pesos de la capa oculta.
- « θ_j » sus umbrales correspondientes.
- « w'_{ij} » los pesos de la capa de salida.
- « θ'_k » sus umbrales respectivos.
- Función de activación para la capa final es la sigmoide.
- Función de activación para la capa oculta, la lineal.

Funcionamiento

- Paso 1. En $t = 0$, inicialización de pesos y umbral, normalmente con valores aleatorios.
- Paso 2. Para cada patrón « r » del conjunto de entrenamiento \mathbf{x}^r ($r = 1, \dots, N$)
 - Calcular z^r_k las « s » salidas de la capa final

$$z^r_k = (w'_{k1} y^r_1 + w'_{k2} y^r_2 + \dots + w'_{ko} y^r_o) - \theta'_k$$

Como a su vez cada y_j es la salida de la capa de entrada, entonces

$$z_k = (w'_{k1} f(w_{11} x^r_1 + w_{12} x^r_2 + \dots + w_{1n} x^r_n - \theta_1) + \dots + w'_{ko} f(w_{o1} x^r_1 + w_{o2} x^r_2 + \dots + w_{on} x^r_n - \theta_o)) - \theta'_k$$

siendo f la función de activación sigmoidea.

- Se calcula el error cometido con ese patrón. Suponiendo que se calcula el error cuadrático medio

$$\text{Error} = \frac{1}{2} \{ [(c^r_1 - z^r_1)^2 + \dots + (c^r_s - z^r_s)^2] \}$$

Obsérvese que, a su vez, esa expresión depende de los diferentes valores « w, w' de la capa de salida y de la capa oculta» y de los diferentes valores « θ y θ' de los umbrales correspondientes» como aparecen en la expresión del paso 2 a través de los valores « z ». Es decir, la función error depende de

$$\text{Error} (w, w', \theta, \theta')$$

- En este momento se exige que dicha función «Error» sea mínima, con respecto a todos esos parámetros. Para ello se puede aplicar el método

del gradiente. Lo que permitirá calcular la variación de los pesos de w' , de w , θ y θ' .

- Las expresiones obtenidas llevan implícito el concepto de *propagación hacia atrás*, ya que en primer lugar se calcula la señal de error proporcional al error de la salida actual de la red, a partir de la cual se calcula las señales de error de la capa oculta.
- Se actualizan pesos y umbrales generados por ese patrón «r».
- Paso 3. Se calculan el incremento total actual de pesos (w , w') y umbrales (θ , θ'), sumando respectivamente todos los valores obtenidos con todos los patrones.
- Paso 4. Se actualizan pesos y umbrales.
- Paso 5. Se calcula el error total, se pasa de t a $t+1$ y se vuelve al paso 2 si todavía el error total no está satisfecho.

Resumiendo, se inicia siempre con pesos iniciales aleatorios pequeños, tanto positivos como negativos. En el esquema presentado se lleva a cabo una fase de ejecución para todos y cada uno de los patrones del conjunto de entrenamiento, se calcula la variación de los pesos y umbrales debida a cada patrón, se acumulan, y a continuación se efectúa la actualización de los pesos. Este esquema se acostumbra a denominar *aprendizaje por lotes*.

Problemas que pueden aparecer en el proceso del cálculo.

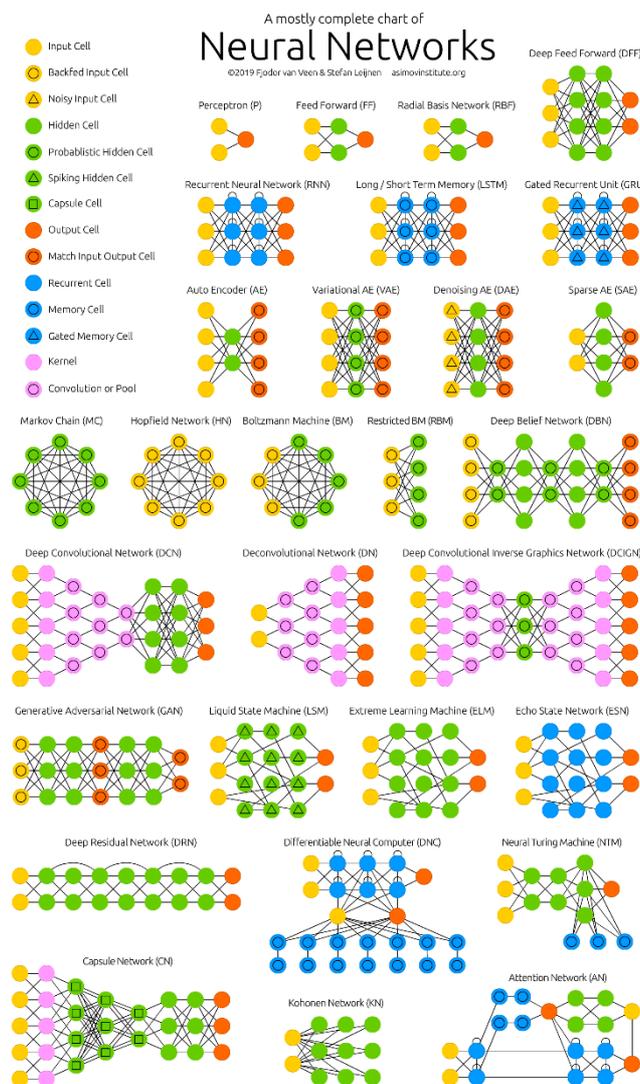
- Problema del desvanecimiento de gradiente. Como se ha visto cada uno de los pesos de la red neuronal recibe una actualización proporcional a la derivada parcial de la función de error con respecto al peso actual en cada iteración de entrenamiento. El problema surge cuando esos valores van haciéndose cada vez más pequeños, y al multiplicar «n» (n capas) de esos números pequeños el error disminuye exponencialmente, lo que puede impedir el entrenamiento.
- Problema del gradiente explosivo. Justo el problema contrario al indicado previamente.

Redes profundas y redes poco profundas

No hay un umbral claro de profundidad que divida el aprendizaje superficial del aprendizaje profundo; pero en general se acepta para el aprendizaje profundo que tiene múltiples capas no lineales.

Si existe el problema del reconocimiento de patrones simples, una máquina de vectores de soporte (svm) o un clasificador de regresión logística puede hacer el trabajo bien, pero a medida que aumenta la complejidad del patrón, no hay más remedio que buscar redes neuronales profundas.

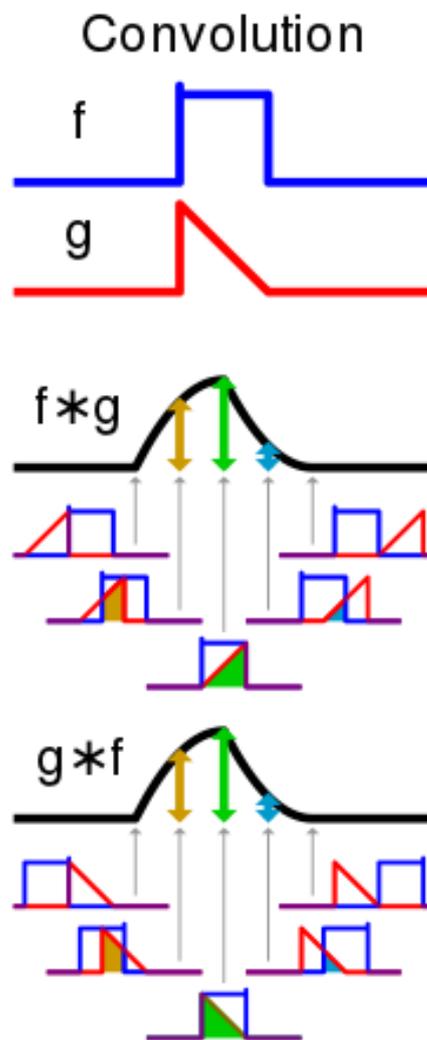
Redactar una lista completa de los diferentes tipos de redes neuronales es prácticamente imposible, ya que constantemente se inventan nuevas arquitecturas. Entonces, voy a proporcionar una lista que encontré en la dirección web <<https://www.asimovinstitute.org/neural-network-zoo/>> que puede brindar una visión general de las mismas. Teniendo en mente que dicha lista no es exhaustiva.



Redes neuronales profundas convolucionales - CNNs

CNN = (Convolutional Neural Network)

Estas redes se denominan por el concepto matemático *convolución*, que es una transformación lineal de dos funciones [«f» y «g»] en una tercera que en cierto sentido representa el área de superposición en «f» de una versión trasladada en invertida de «g». Véase la gráfica



Definición matemática

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\gamma)g(t - \gamma)d\gamma$$

Veamos qué hace esa ecuación cuando trabaja en dos dimensiones como es una imagen digital de pixeles.

Imagen

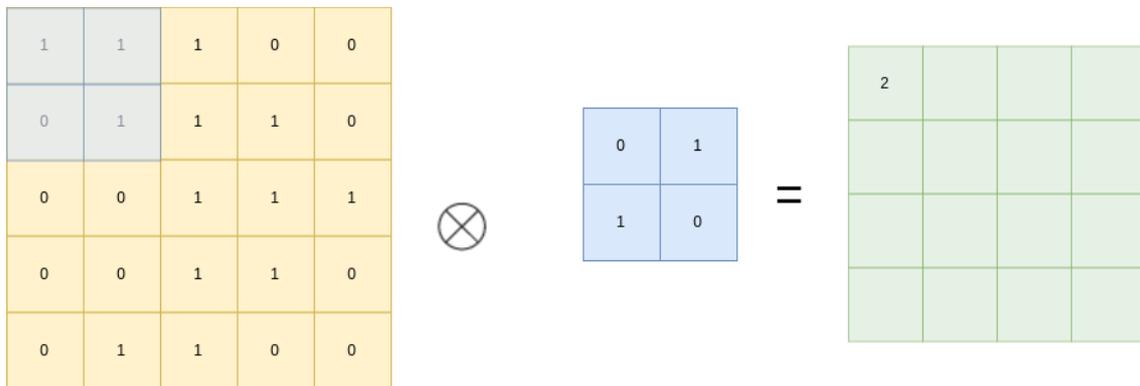
1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

Detector de propiedades

0	1
1	0

La operación de convolución no es más que una multiplicación elemento por elemento de la zona coloreada de la imagen por el detector de propiedades. Aquí se ha seleccionado una matriz (2x2), dependiendo del problema el tamaño y los valores pueden cambiar

$$(1 \times 0 + 1 \times 1) + (0 \times 1 + 1 \times 0) = 2$$



A continuación se desplaza una cierta cantidad la zona coloreada de la imagen, supongamos que el desplazamiento es de valor 1 en horizontal, mientras no se salga de la zona de la imagen.

Posteriormente se desplazará un valor 1 en vertical y se volverá a hacer el recorrido completo en horizontal.

Se sigue así hasta que la zona coloreada alcance el margen inferior derecho, sin salirse.

El resultado es la imagen convolución, también denominada *matriz final de propiedades* que en este caso será:

2	2	1	1
1	2	2	1
0	1	2	1
0	2	2	0

Obsérvese que hemos partido de una matriz de datos de 5x5 y hemos obtenido una matriz de propiedades de 4x4. El valor de la reducción depende del valor de desplazamiento utilizado. Y, además, la información que ha quedado es algo que está relacionado con alguna característica que había en la imagen de partida.

Veamos unos ejemplos extraídos de

<<https://docs.gimp.org/2.8/en/plug-in-convmatrix.html>>

Blur

0	0	0	0	0
0	1	1	1	0
0	1	1	1	0
0	1	1	1	0
0	0	0	0	0

Filtro



Resultado

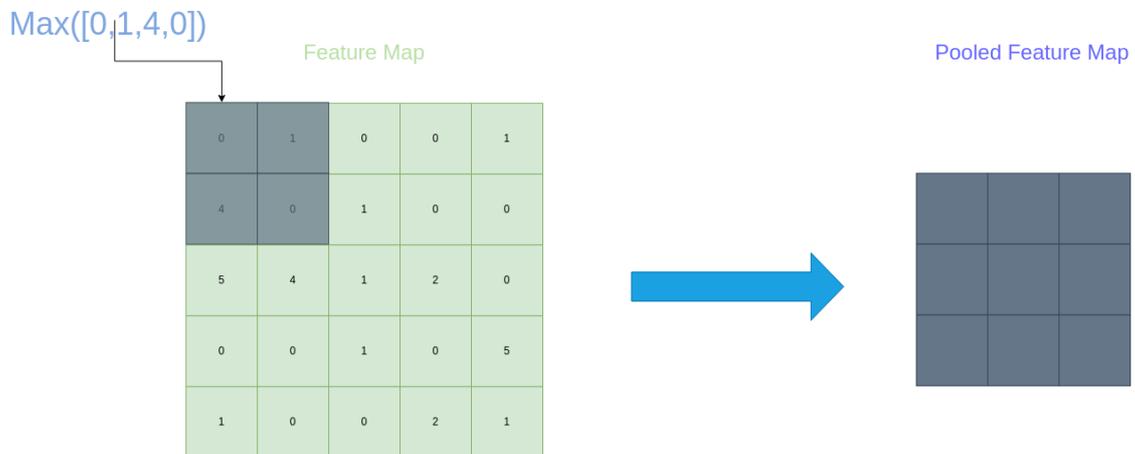
En las redes convolucionales profundas, cada neurona de una capa no recibe conexiones entrantes de todas las neuronas de la capa anterior, sino solo de algunas. Esto hace que una neurona se especialice en un dominio de la capa anterior «reduciendo» el número de pesos que se deben calcular.

Las capas convolucionales funcionan en base a borrar cierta información de la imagen y maximizar otro tipo de información que puede ser parte integral del resultado. Cada capa se especializa en alguna característica que hay que extraer.

Posteriormente a esta fase de cálculo se aplica la función rectificadora lineal (ReLU) de manera que aumente la no linealidad del resultado obtenido en cada una de las imágenes convolucionada. La razón para ello es que en las imágenes hay muchos elementos no lineales que nosotros como seres humanos sabemos distinguir, pero a un algoritmo de entrada le cuesta mucho más.

Y ahora debemos extraer de cada matriz de propiedades generada cada una con una matriz de características diferentes, los valores mayores. La forma de hacerlo es análoga al del cálculo de la convolución. Para ello seleccionaré una zona de la matriz de características, por ejemplo, una zona de (2x2) y la iré desplazando en este caso de 2 en 2 en horizontal y de 1 en 1 en vertical.

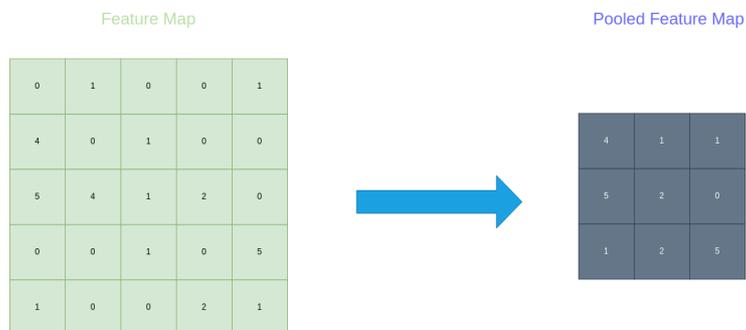
Sea la matriz de características



MaxPooling = (2,2)

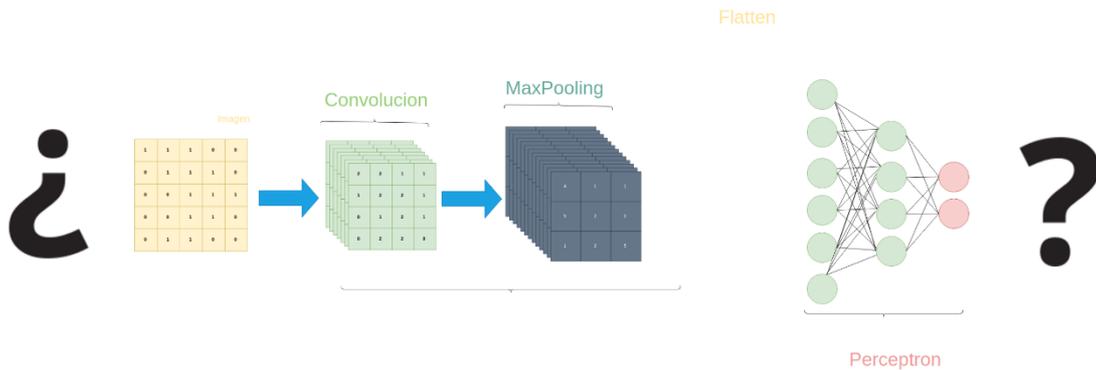
La matriz 3x3 resultante (*pooled Feature Map*) sería

MaxPooling = (2,2)
Strides = 2

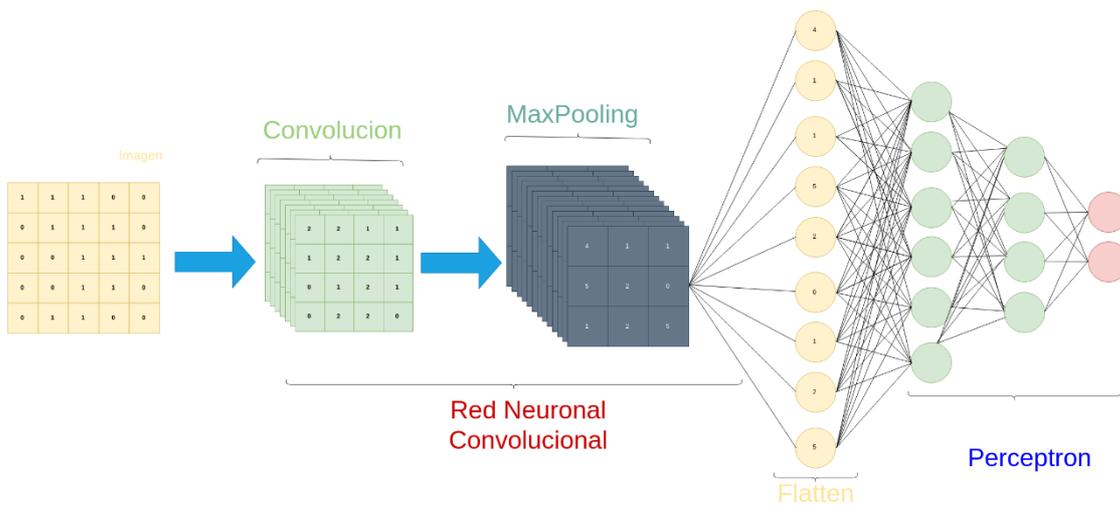


En este caso nos hemos desecho del 75 % de la imagen.

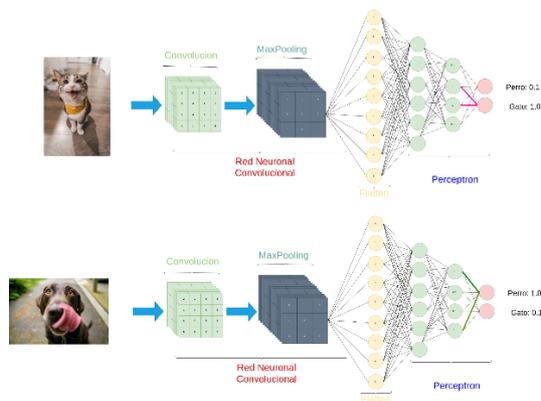
Si se resume el trabajo descrito, queda reflejado en la imagen siguiente. Ahora el problema consiste en traspasar esto a una red neuronal:



La forma de solucionar el problema es colocar las capas de MaxPooling en una columna, con el fin de aplanarla (Flattering) para que esto alimente a un perceptrón que se encargará de clasificar estos valores y encasillarlos en categorías que le hemos pasado junto con el conjunto de ejemplos (patrones). Lo normal es tener una neurona perceptrón por cada categoría a clasificar. Las fases de propagación hacia adelante y de propagación hacia atrás (esta en la fase de aprendizaje) actúan sobre todas las capas de la red neuronal.



Ejemplo de una red neuronal que clasifica perros y gatos

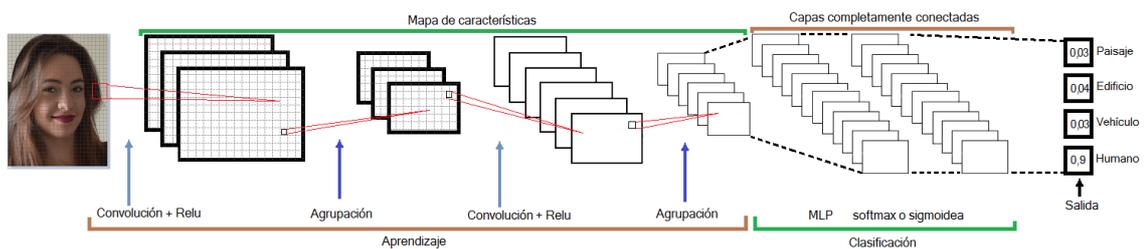


Para saber más, véase

<<https://medium.com/@jcrispis56/introducci%C3%B3n-al-deep-learning-parte-2-redes-neuronales-convolucionales-f743266d22a0>>

Resumiendo mucho. La estructura general de una red convolucional se construye con tres tipos de capas distintas:

- Una capa convolucional o varias.
- Una capa de agrupación /reducción entre capas convolucionales.
- Una capa final clasificadora completamente conectada.



Para saber más, véanse

<<http://numerentur.org/convolucionales/>>

<<https://www.aprendemachinlearning.com/como-funcionan-las-convolutional-neural-networks-vision-por-ordenador/>>

<https://es.wikipedia.org/wiki/Redes_neuronales_convolucionales>

Es evidente que las aplicaciones de las redes neuronales convolucionales son:

- Clasificación de imágenes (2 dimensiones).
- Clasificación de señales de audio (1 dimensión).
- Clasificación de datos volumétricos (3 dimensiones).
- ...

En pocas palabras, las redes neuronales convolucionales (CNN) son redes neuronales de múltiples capas. Las capas a veces tienen hasta 17 o más y asumen que los datos de entrada son imágenes.

Redes neuronales recurrentes - RNNs (Recurrent Neuronal Network)

<<https://medium.com/@jcrispis56/introducci%C3%B3n-al-deep-learning-parte-3-redes-neuronales-recurrentes-7da543c3b181>>

<<http://numerentur.org/rnr/>>

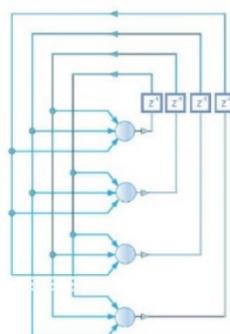
El concepto básico que subyace a las RNN es utilizar información secuencial. En una red neuronal normal, se supone que todas las entradas y las salidas son independientes entre sí. Si queremos predecir la siguiente palabra en una oración, tenemos que saber qué palabras vinieron antes de ella.

Son redes neuronales que permiten hacer aprendizaje supervisado procesando datos secuenciales de longitud variable y con la capacidad de incrementar la longitud de las secuencias. Proporcionar modelos con memoria y permitirles modelar la evolución temporal de las señales es un factor clave en muchas tareas de clasificación y traducción de secuencias en las que los RNN sobresalen, como la traducción automática, el modelado del lenguaje o el reconocimiento de voz, entre muchas otras áreas, donde la secuencia de datos y su dinámica temporal que conecta los datos a menudo es más importante que el contenido espacial (de los píxeles) de cada dato (imagen) individual.

Se caracterizan porque a lo largo del tiempo integran bucles de realimentación, permitiendo que a través de ellos que la información persista durante algunos pasos o épocas de entrenamiento y de esta forma se tiene una cierta memoria a corto plazo.

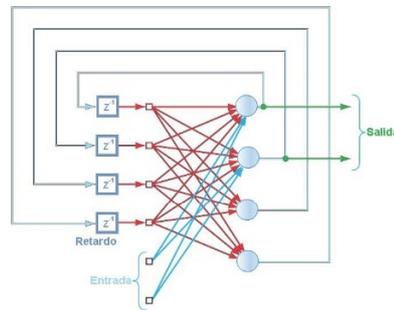
Existen dos topologías básicas:

- Una capa. Es decir, una red con una sola capa de neuronas, pero cada neurona alimenta con su señal de salida a las entradas de las otras neuronas. No hay proceso de autorretroalimentación.



RNR de 1 - capa

- Con capa oculta. Es decir, existe una red en la que aparecen las neuronas ocultas y las conexiones de retroalimentación provienen también de las salidas de las neuronas ocultas.



RNR de capa oculta

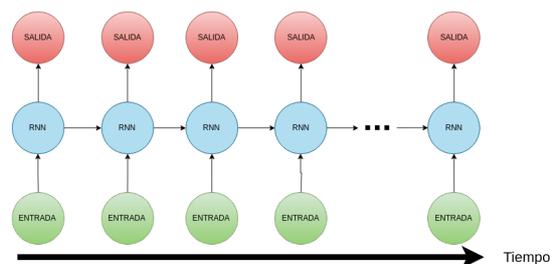
Cada neurona recurrente tiene dos conjuntos de parámetros, uno que lo aplica a la entrada de datos que recibe de la capa anterior y otro conjunto que lo aplica a la entrada de datos correspondiente al vector de salida del instante anterior.

Dado que la salida de una neurona recurrente en un instante de tiempo determinado es una función de entradas de los instantes de tiempo anteriores, se podría decir que una neurona recurrente tiene en cierta forma memoria. La parte de una red neuronal que preserva un estado a través del tiempo se suele llamar *célula de memoria* o *memoria interna*.

En función del modelo de conexión las RNN pueden ser:

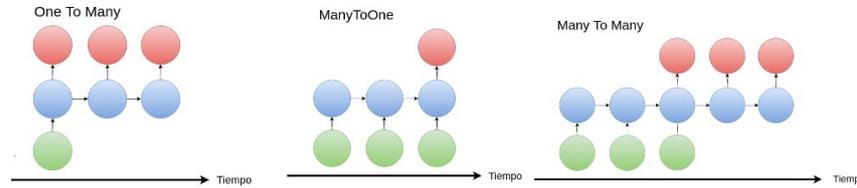
- Totalmente recurrentes. Cada neurona puede estar conectada a cualquier otra y sus conexiones recurrentes son variables.
- Parcialmente recurrentes. Son aquellas que sus conexiones recurrentes son fijas.

Para visualizar dicho proceso se «desenrolla» la red en tantas capas como pasos temporales o de datos se dispone en la secuencia temporal de entrenamiento, como si fuese una red no recurrente típica. Cada capa desenrollada tiene los mismos pesos para acelerar el proceso.



En esa representación cada círculo es una capa completa de neuronas. Por otro lado, las arquitecturas posibles son numerosas.

Una entre muchas salidas, muchas a una o muchas a muchas



Cuanta más larga sea la secuencia temporal que analizar, mayor será el número de capas que se deben desarrollar, por lo que puede aparecer el problema del desvanecimiento de gradiente. Esto se soluciona incorporando capas del tipo «Long Long-Term Memory» (LSTM) o «Gated Recurrent Units» (GRU) que permiten la propagación hacia atrás a través del tiempo conectando eventos muy alejados en los datos de entrada, para que su peso se diluya entre las capas.

Las LSTM son una extensión de las redes neuronales recurrentes, que básicamente amplían su memoria para aprender de experiencias importantes que han pasado hace mucho tiempo. De esta manera se pueden recordar sus entradas durante un periodo largo de tiempo. Una neurona de una LSTM puede leer, escribir y borrar información en su memoria. Las GRU son LSTM que combinan la puerta de entrada y la de borrar.

Ejemplos puede verse en:

<<https://torres.ai/redes-neuronales-recurrentes/>>
<<https://medium.com/@prvnk10/long-short-term-memory-lstm-and-gated-recurrent-units-gru-240d8a62db9>>

Mapas autoorganizados (Self-Organizing Maps SOM).

<<https://medium.com/@jcrispis56/introducci%C3%B3n-al-deep-learning-parte-4-soms-665c2c0bb223>>

<<http://halweb.uc3m.es/esp/Personal/personas/jmmarin/esp/DM/tema5dm.pdf>>

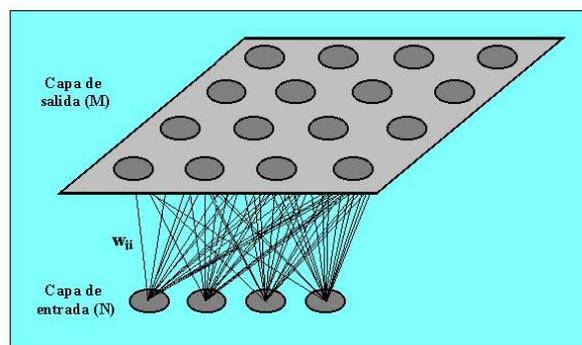
Esta técnica permite realizar aprendizaje no supervisado. Estos mapas, también conocidos como *mapas de Kohonen*, se utilizan para pasar de un espacio de mayor dimensión a otro de menor dimensión.

Se pasa de una entrada multidimensional de un conjunto de datos formado por miles de columnas y miles de filas, y transforma esa matriz en un mapa de 2 dimensiones, que representa a todo el conjunto de datos.

El algoritmo aprende simplemente a partir de los datos de entrada, para ello busca las similitudes de forma autónoma.

Los SOM operan en dos modos: entrenamiento y mapeo. Durante el entrenamiento construye el mapa usando ejemplos entrantes, mientras que en el mapeo clasifica una nueva entrada.

Un modelo SOM está compuesto por dos capas de neuronas. La capa de entrada formada por N neuronas, una por cada variable de entrada (x_1, x_2, \dots, x_N), que se encarga de recibir y transmitir a la capa de salida la información procedente del exterior. La capa de salida, formada por M neuronas, es la encargada de procesar la información y formar el mapa. Normalmente, las neuronas de la capa de salida se organizan en forma de mapa bidimensional.



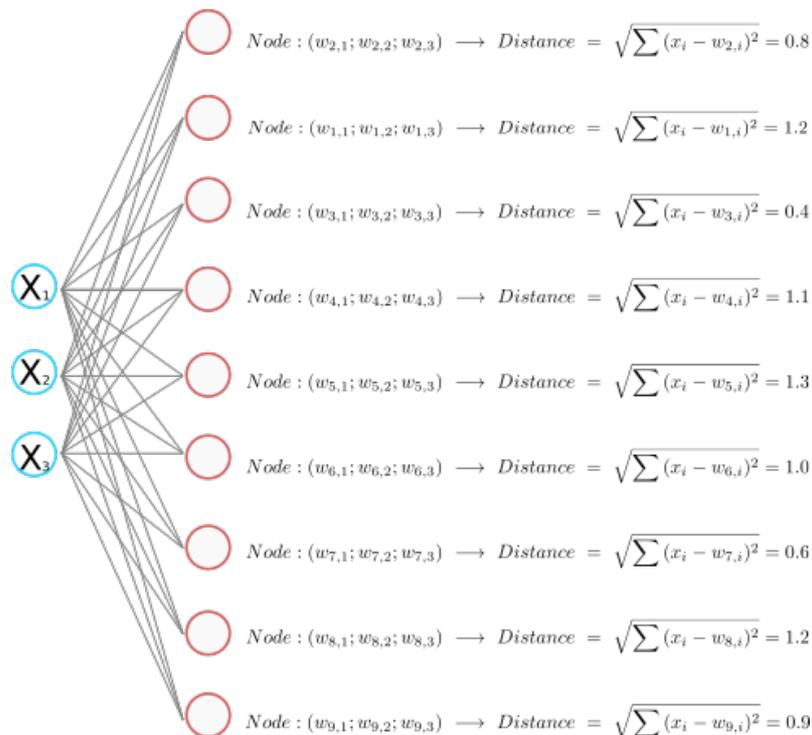
Las conexiones entre las dos capas que forman la red son siempre hacia adelante. Cada neurona de entrada « i » está conectada con cada una de las neuronas de salida « j » mediante un peso w_{ij} . De esta forma, las neuronas de salida tienen asociado un vector de pesos W_j llamado *vector de referencia*, $\{w_{ij}\}$, dada la neurona de salida « j » el vector de referencia será de N componentes, uno por cada neurona « i » de entrada.

Si estamos con la neurona de salida $j=9$, y hay 3 capas de entrada, entonces, $W_9 = \{w_{19}, w_{29}, w_{39}\}$.

Entre las neuronas de la capa de salida, existe una cierta influencia sobre sus vecinas. Dada la neurona «j» hay V_j neuronas vecinas, dicha topología se establece desde el principio.

El algoritmo del SOM en el proceso de aprendizaje es el siguiente:

- Paso 1. Se selecciona al azar del conjunto de datos un vector y se calcula su distancia (similitud) a los vectores de referencia de cada neurona de salida. Se suele utilizar la distancia euclídea



Ahora se selecciona aquella neurona cuya distancia es la menor, en este caso es la 3, cuyo vector es el $W_3 = \{w_{13}, w_{23}, w_{33}\}$, que en este caso recibe el nombre de BMU (*Best Matching Unit*).

- Paso 2. Ahora hay que ajustar el resto de los vectores de referencia con respecto al BMU seleccionado.

Dado un vector «j» cualquiera, la regla de actualización de su vector de referencia viene dada por

$$W_j(t+1) = W_j(t) + \alpha(t)(x(t) - W_j(t)) \quad \text{si } j \text{ pertenece al entorno } V_j$$

$$W_j(t+1) = W_j(t) \quad \text{si } j \text{ no pertenece al entorno } V_j$$

- Se repiten los pasos 1 y 2 hasta que el entrenamiento termina. El número de pasos se debe fijar *a priori*, para calcular la tasa de convergencia de la función de vecindad y la tasa de aprendizaje (α).
- Una vez terminado el entrenamiento, el mapa ha de ordenarse en sentido topológico y visualizarse. Normalmente se representa como una rejilla regular de neuronas.

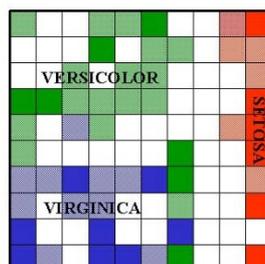
Ejemplo: Se tiene una muestra de 150 plantas donde cada planta tiene cuatro características, el objetivo es determinar el tipo de cada planta según esas características. Hay 50 ejemplares de cada uno de los tres tipos de planta existentes. Las tres categorías son desconocidas para la red y las tiene que poner de manifiesto la red.

Datos para la red:

- Criterio de similitud de la distancia euclídea en la etapa de entrenamiento.
- El mapa de salida es de 10 x 10 neuronas.

Entrenamiento:

- Se entrena un total de 10 modelos, cada uno con una configuración inicial de pesos diferentes con rangos de valor comprendidos entre [0-1].
- Todos van a seguir el mismo esquema de aprendizaje.
- El entrenamiento de cada mapa se organiza en dos fases:
 - Fase 1 para organizar el mapa. Se utiliza una tasa de aprendizaje alta igual a 1, y un radio de vecindad grande igual al diámetro del mapa, es decir, de valor 10. A medida que avanza el aprendizaje, tanto la tasa de aprendizaje como el radio de vecindad se van reduciendo de forma lineal hasta alcanzar unos valores mínimos, 0,05 y 1, respectivamente.
 - Fase 2 para ajustar mejor el mapa. Se utiliza una tasa de aprendizaje pequeña y constante igual a 0,05 y un radio de vecindad constante y mínimo igual a 1.
- Una vez entrenados los 10 modelos, se calcula para cada uno de ellos el error promedio y se selecciona el modelo cuyo error sea más pequeño.
- Finalmente, se muestra el mapa del modelo finalmente seleccionado.



En la figura, los colores representan categorías. Los colores blancos no representan ninguna categoría.

Si la intensidad del color es baja indica que hay pocos patrones asociados a la neurona. Si la intensidad es alta, indica que hay más.

Funcionamiento:

Una vez entrenada la red, se le alimenta con nuevos datos y se observan los resultados. Véase una implementación en

<<http://halweb.uc3m.es/esp/Personal/personas/jmmarin/esp/DM/tema5dm.pdf>>

Modelos generativos

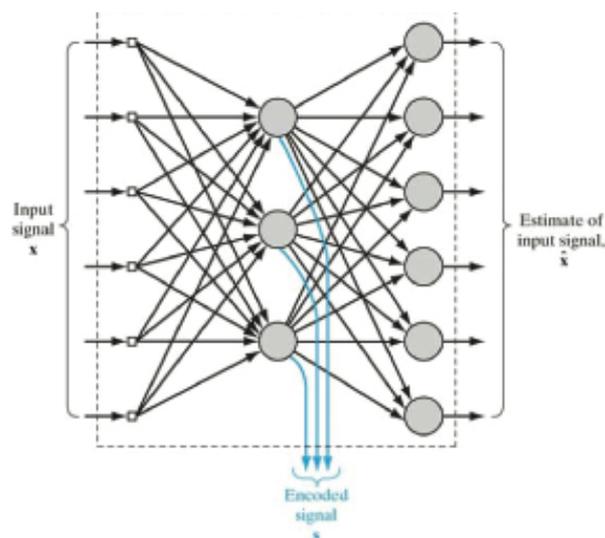
Recordemos que los algoritmos discriminatorios, como los presentados hasta ahora, tratan de clasificar los datos de entrada. Es decir, dadas unas características de unos datos, predicen una etiqueta o categoría a la que pertenecen esos datos. Es decir, correlacionan características con etiquetas.

En cambio, los algoritmos generativos hacen lo contrario; en lugar de predecir una etiqueta con ciertas características, intentan predecir características con una etiqueta determinada.

El método de aprendizaje del *Backpropagation* no funciona bien con redes que tengan varias capas ocultas (salvo en el caso de las redes convolucionales). De hecho, los problemas que tienen son:

- Requieren datos etiquetados, pero muchas veces eso no es factible.
- No resulta demasiado escalable.
- Es lento con múltiples capas ocultas.
- Se puede quedar atascado en óptimos locales.

Los modelos generativos profundos representan modelos de aprendizaje no supervisado que tienen como objetivo aprender una distribución similar a la que tienen los datos del conjunto de entrenamiento y, una vez logrado, ser capaces de generar nuevas muestras con ligeras variaciones, sin olvidar su interés generativo. Mientras que un modelo discriminatorio es un modelo de probabilidad condicionada $P(Y|X)$. Los modelos generativos permiten calcular la distribución de probabilidad conjunta entre un observable X y una variable Y , $P(X, Y)$.



Autoencoders

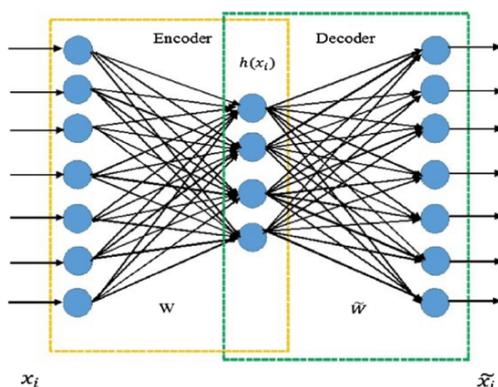
<<https://www.toptal.com/machine-learning/an-introduction-to-deep-learning-from-perceptrons-to-deep-networks>>

Un *autoencoder* es una red neuronal que tiene como objetivo aprender una codificación comprimida y distribuida de un conjunto de datos de manera no supervisada. Conceptualmente, la red está entrenada para «recrear» la entrada, es decir, la entrada y la salida son los mismos, es decir, intenta generar lo mismo que entró, pero comprimido de alguna manera.

Ejemplo: Supongamos que los datos de entrenamiento constan de imágenes en escala de grises de 28x28 y el valor de cada píxel está sujeto a una neurona de la capa de entrada (es decir, la capa de entrada tendrá 784 neuronas). Entonces, la capa de salida tendría el mismo número de unidades (784) que la capa de entrada y el valor objetivo para cada unidad de salida sería el valor de escala de grises de un píxel de la imagen.

La intuición detrás de esta arquitectura es que la red no aprenderá un «mapeo» entre los datos de entrenamiento y sus etiquetas, sino que aprenderá la *estructura interna* y las características de los datos en sí. (Debido a esto, la capa oculta también se llama *detector de características*). Por lo general, el número de unidades ocultas es menor que las capas de entrada / salida, lo que obliga a la red a aprender solo las características más importantes y logra una reducción de dimensionalidad.

El *autoencoder* se puede ver como constituido por dos partes principales:



- Un codificador que mapea la entrada en la capa oculta:
 - Como ya sabemos $\mathbf{h} = \sigma(\mathbf{W} \mathbf{x} + \mathbf{b})$:
 - \mathbf{h} recibe el nombre de código.
 - σ es una función de activación.
 - \mathbf{W} es la matriz de pesos.
 - \mathbf{b} es el vector de umbrales.

- Un decodificador que mapea la capa oculta en la capa de salida:
 - Como ya sabemos $\mathbf{x}' = \sigma'(\mathbf{W}' \mathbf{h} + \mathbf{b}')$:
 - \mathbf{x}' es la salida.
 - σ' es una función de activación.
 - \mathbf{W}' es la matriz de pesos.
 - \mathbf{b}' es el vector de umbrales.

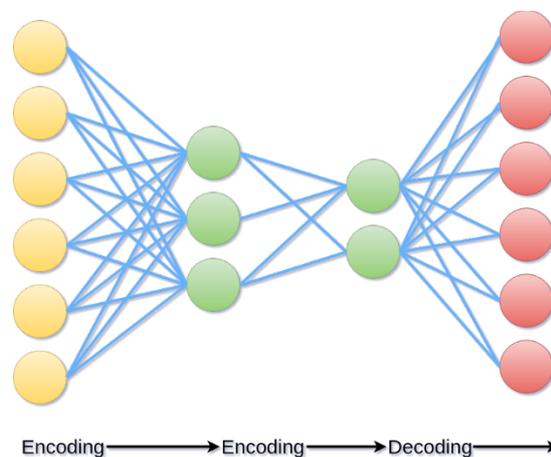
La red se entrena minimizando los errores de reconstrucción.

La capa oculta no es más que un codificador que reconstruye la entrada de forma aproximada, conservando los aspectos más relevantes de los datos en la copia.

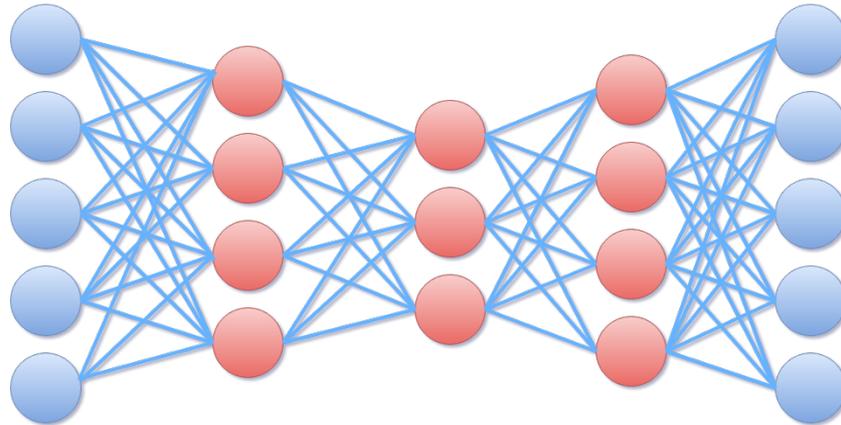
Existen distintos *autoencoders* para evitar que aprendan la función de identidad y para mejorar su capacidad para capturar información importante y aprender representaciones más ricas, como, por ejemplo:

- Codificador automático disperso.
- *Autoencoder* de reducción de ruido.
- Codificador automático contractual.
- *Autoencoder* variacional.
- ...

Autoencoders apilados, como en el caso de este ejemplo, en él se ve que hay dos fases de codificación y una fase de decodificación



Autoencoders profundos, con más de una capa oculta,



cuyas ventajas son:

- Reducir el coste computacional.
- Reducir la cantidad de datos de entrenamiento.
- Son más interesantes.

Los *autoencoders* se utilizan por ejemplo para:

- Reducción de la dimensionalidad.
- Los de una sola capa están muy relacionados con el análisis de componentes principales.
- Recuperación de información.
- Detección de anomalías.
- Procesamiento de imágenes:
 - Eliminación de ruido.
 - Reconocimiento de caras.
 - Detección de tumores...
- Buscar el significado semántico de palabras.
- ...

Para saber más, véanse

<<https://en.wikipedia.org/wiki/Autoencoder>>
<<https://medium.com/@jcrispis56/deep-learning-b%C3%A1sico-parte-final-autoencoders-3d5c6fff2966>>

Modelos estocásticos

Existe una posibilidad que es intentar ajustar los pesos maximizando la probabilidad de que un modelo generativo genere los datos de entrada, es decir, maximizar $p(x)$, no $p(y|x)$. Para ello, surgen las neuronas binarias estocásticas, que son exactamente como las neuronas sigmoideas, si bien su salida se interpreta como una probabilidad. Los modelos basados en estas ideas son:

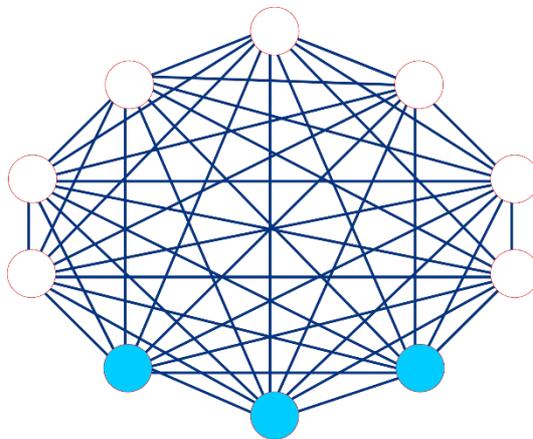
- *Autoencoders.*
- Las máquinas de Boltzmann.
- Las máquinas de Boltzmann restringidas.
- Las redes de creencias profundas.

Redes de Boltzmann restringidas - RBN

<<https://medium.com/@jcrispis56/introducci%C3%B3n-al-deep-learning-parte-5-maquinas-de-boltzmann-507b8d65fe10>>

<<https://elvex.ugr.es/decsai/deep-learning/slides/NN9%20Stochastic.pdf>>

Las máquinas de Boltzmann son un modelo que no tiene una dirección, entrada-salida, sino que tiene múltiples direcciones



Todas las neuronas están conectadas entre sí, incluso los nodos de entrada, y la información fluye en todas las direcciones. En la imagen, los nodos azules son las entradas y los otros son la capa oculta, es decir, no hay capa de salida. Además, no espera datos de entrada, los genera. Por ello se llama *modelo generativo*.

Es una red estocástica que representa la información a partir de una distribución de probabilidad. Los pesos se inician aleatoriamente y la red aprende por el método de retropropagación.

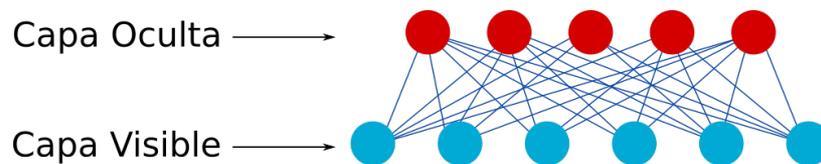
La red se utiliza para resolver dos problemas diferentes:

- Para realizar búsquedas. Los pesos en las conexiones son fijos y se utilizan para representar una función de coste que habrá que minimizar.
- Para aprendizaje. Hacen muchas pequeñas actualizaciones de los pesos y cada actualización requiere que se resuelvan problemas de búsqueda diferentes. Este procedimiento es muy rápido en el caso de que haya una sola capa (modelo restringido).

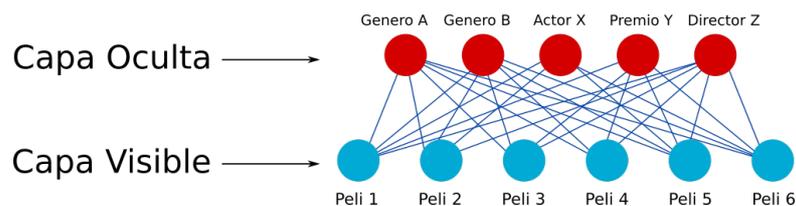
Las máquinas de Boltzmann son algo parecidas a las RNN que, en vez de utilizar las redes para almacenar recuerdos, las utilizan para construir interpretaciones de las entradas.

Las máquinas de Boltzmann restringidas RBM no admiten las conexiones entre las neuronas de la misma capa. Son redes neuronales estocásticas que pueden aprender una distribución de probabilidad sobre un conjunto de entradas.

Volvamos a recordar que a diferencia de las redes *feedforward*, las conexiones entre la capa visible y la capa oculta son bidireccionales, a veces los valores van de capa visible a capa oculta y a veces al revés.



Ejemplo



El ajuste de los pesos (fase de aprendizaje) se realiza del modo siguiente:

- Inicialmente son aleatorios y cuando le pasamos nuestra entrada se calculan los nodos ocultos y lo que va a pasar ahora es que los nodos ocultos van a usar sus pesos para calcular los nodos de entrada, van a reconstruir la entrada.
 - Fase 1. Se envían un ejemplo de datos «vector \mathbf{v} » a la capa de entrada, y a través de los pesos se obtienen los resultados de la salida «vector \mathbf{h} ».
 - Fase 2. Se reenvía el vector « \mathbf{h} » desde la capa oculta hacia la capa de entrada (con los mismos pesos), obteniéndose los nuevos datos «vector

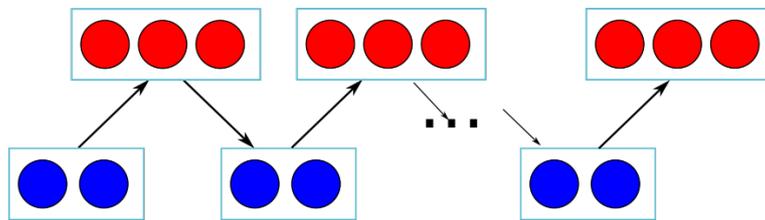
\mathbf{v}' ». Con posterioridad se vuelve a enviar esos datos hacia la capa oculta (con los mismo pesos) obteniéndose el «vector \mathbf{h}' ».

- Fase 3. Se recalculan los pesos

$$W(t+1) = W(t) + \alpha (\mathbf{v} \mathbf{h}' - \mathbf{v}' \mathbf{h}^t)$$

Donde α es la velocidad de aprendizaje.

La intuición detrás del algoritmo es que la fase 1 ($\mathbf{h} \rightarrow \mathbf{v}$) refleja la representación interna de la red de los datos del mundo real. Mientras que la fase 2 ($\mathbf{h} \rightarrow \mathbf{v}'$) la red intenta recrear los datos a partir de la representación interna. El objetivo final es generar datos que sean lo más próximos posible a los del mundo real a través de los pesos de la red. Se seguirá así hasta que se encuentre un mínimo global y finalice la fase de entrenamiento. Este proceso de entrenamiento recibe el nombre de *Contrastive Divergence*.

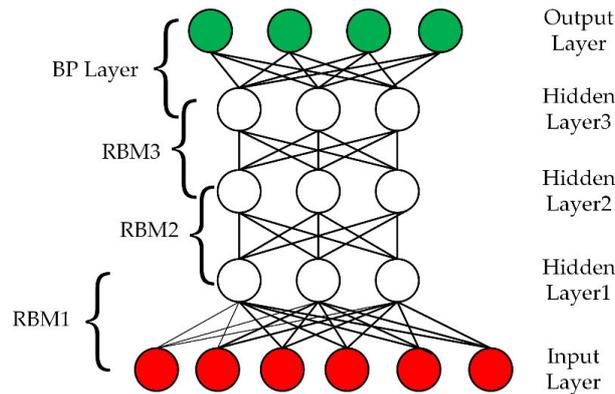


Aplicación: Sistemas de recomendación de películas como lo hace Netflix. El objetivo es predecir las evaluaciones de películas, partiendo de los datos de por ejemplo 500 000 usuarios sobre 18 000 películas en una escala de 1 a 5. Cada usuario solo evalúa una pequeña fracción de películas.

Redes de creencias profundas Deep Belief Networks DBN

<<https://www.toptal.com/machine-learning/an-introduction-to-deep-learning-from-perceptrons-to-deep-networks>>

Del mismo modo que se pueden apilar perceptrones y autocodificadores, también se pueden apilar máquinas de Boltzman restringidas (RBM) obteniéndolo que se conoce como *redes de creencias profundas* (DBN).



Tenemos un nuevo modelo que finalmente resuelve el problema de la desaparición del gradiente. Tanto las RBM como las Deep Belief Nets son alternativas a la propagación hacia atrás.

Una DBN es similar en estructura a un MLP (*Multi-layer perceptron*), pero muy diferente cuando se trata de entrenamiento. Una DBN se puede visualizar como una pila de RBM donde la capa oculta de un RBM es la capa visible de la RBM que se encuentra arriba. La primera RBM está capacitada para reconstruir su entrada con la mayor precisión posible.

La capa oculta de la primera RBM se toma como la capa visible de la segunda RBM y la segunda RBM se entrena utilizando las salidas de la primera RBM. Este proceso se itera hasta que cada capa en la red está entrenada.

Estas redes pueden aprender a reconstruir probabilísticamente las entradas, de manera que las capas actúan como detectores de características de dicha entrada. Después de este paso de aprendizaje, una DBN puede ser entrenada de nuevo de forma supervisada para mejorar la etapa de clasificación.

Para saber más, véase

<https://es.wikipedia.org/wiki/Red_de_creencia_profunda>

Redes Generativas antagónicas - GAN

<https://es.wikipedia.org/wiki/Red_generativa_antag%C3%B3nica>

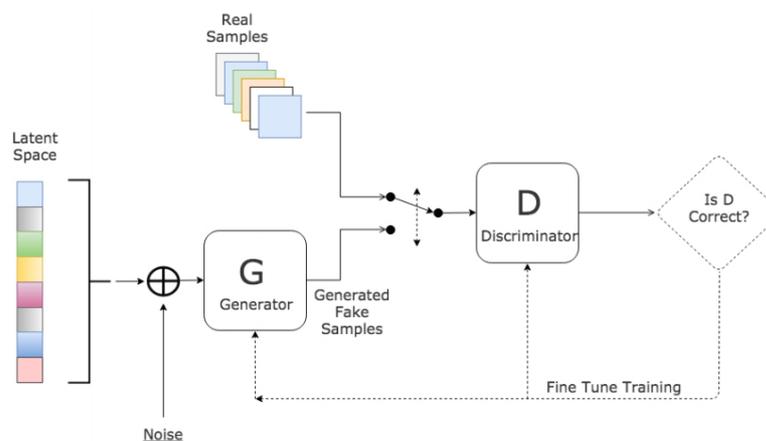
<<https://www.xataka.com/inteligencia-artificial/conceptos-inteligencia-artificial-que-gans-redes-generativas-antagonicas>>

<<https://puentesdigitales.com/2019/04/05/todo-lo-que-necesitas-saber-sobre-las-gan-redes-generativas-antagonicas/>>

<<https://iartificial.net/redes-neuronales-generativas-adversarias-gans/>>

Las redes generativas antagónicas son redes neuronales profundas que comprenden dos redes, enfrentadas una con la otra, que compiten en un constante juego de suma cero (la ganancia o la pérdida de una de las redes se compensa con la ganancia o la pérdida de la opuesta), de ahí, el nombre de «antagónicas». Es un proceso donde cada una de las redes va mejorando y aprende de su oponente. Una de las redes, la generativa, va produciendo muestras de aquello que queremos crear (imágenes, textos, sonidos...), mientras que la otra, la discriminadora, las evalúa para determinar su autenticidad, es decir, decide si lo que se ha generado es adecuado o no. Para entrenar esta arquitectura se necesita tener un conjunto de datos reales. De esta manera tanto el generador como el discriminador van aprendiendo a diferenciar cada vez mejor.

Por ejemplo, si se desea crear imágenes de coches, al comienzo, la red del generador recibe información en forma de números aleatorios y devuelve una imagen. Es evidente que al principio el generador ofrecerá imágenes parecidas a una imagen de ruido del estilo del que producía una televisión analógica desintonizada. Esta imagen generada se proporciona como entrada a la red discriminadora junto con imágenes reales y esta devuelve probabilidades, un número entre 0 y 1, donde 1 representa una predicción de autenticidad y 0 representa falsa. El entrenamiento puede requerir cientos, miles o millones de intentos antes de que la red discriminadora acepte el resultado.



El objetivo del aprendizaje de la red generativa es aumentar el índice de error de la red discriminadora produciendo nuevos elementos sintéticos que parecen provenir de la distribución de datos auténticos.

En ambas redes se aplica la técnica de retropropagación, de modo que el generador produzca imágenes progresivamente mejores, mientras el discriminador se refina cada vez más a la hora de distinguir esas imágenes sintéticas. Los generadores son normalmente redes neuronales deconvolucionales y los discriminadores son redes neuronales convolucionales.

Para mejorar la calidad de los modelos generativos se usan las GAN progresivas. La clave es ir aumentando la resolución de las imágenes. Se inicia con imágenes 4x4 tanto en generador como en discriminador y conforme se alcanzan buenos niveles de calidad se pasa a otra resolución mayor, y así sucesivamente.

Prueba las GAN de forma interactiva

<<https://poloclub.github.io/ganlab/>>

El potencial de las GAN es enorme, ya que se puede enseñar a las GAN a crear mundos paralelos sorprendentemente similares a los nuestros en cualquier dominio: imágenes, música, habla, prosa. En cierto modo, son artistas de robots y su producción es bastante impresionante.

Ejemplos de aplicación. Las GAN tienen una gran cantidad de aplicaciones, ya que pueden aprender a imitar distribuciones de datos de casi cualquier tipo:

- Generadores de rostros humanos *fake*: véase

<<https://www.xataka.com/robotica-e-ia/todas-caras-este-alucinante-video-falsas-estan-creadas-motor-inteligencia-artificial-nvidia>>

- Generadores de habitaciones, personajes manga o gatos: véase

<<https://www.xataka.com/inteligencia-artificial/prolifera-webs-fotos-generadas-mediante-ia-gatos-personajes-manga-apartamentos-airbnb>>

<<https://www.xataka.com/inteligencia-artificial/prolifera-webs-fotos-generadas-mediante-ia-gatos-personajes-manga-apartamentos-airbnb>>

<<https://futurism.com/cat-doesnt-exist-ai>>

- Conversión de garabatos en bruto en paisajes realistas: véase

<<https://www.theverge.com/2019/3/19/18272602/ai-art-generation-gan-nvidia-doodle-landscapes>>

- Generación de partituras: véase

<<https://arxiv.org/abs/1703.10847>>

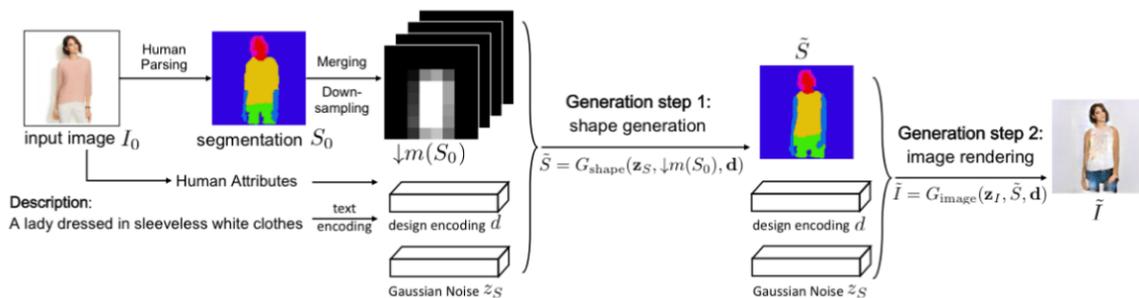
De cualquier modo, es muy difícil trabajar con las GAN, debido a su inestabilidad porque se atascan con facilidad.

Las razones:

- Colapso modal. Las distribuciones naturales de datos son altamente complejas y multimodales. Es decir, el generador para engañar al discriminador, aquel se centra en una parte de la muestra.
- Convergencia. ¿Cuándo se finaliza el entrenamiento?, como las dos redes se complementan, es muy difícil saberlo.

Las GAN también se pueden apilar. Una única GAN puede no ser suficientemente potente como para llevar a cabo su misión. En su lugar, podemos utilizar varias GAN colocadas consecutivamente, en las que cada GAN resuelva una parte sencilla del problema global. Por ejemplo: véase

<https://www.cs.toronto.edu/~urtasun/publications/zhu_etal_iccv17.pdf>



Las GAN convencionales miden la probabilidad de que los datos generados sean reales. Las GAN relativistas miden la probabilidad de que los datos generados sean «más realistas» que los datos reales.

Para saber más: véase

<<https://puentesdigitales.com/2019/04/05/todo-lo-que-necesitas-saber-sobre-las-gan-redes-generativas-antagonicas/>>

Comentario final sobre las redes neuronales artificiales

La mayoría de los estudios sobre las redes neuronales artificiales sugieren que lo único que importa de la red neuronal es su anatomía. ¿Qué unidades están conectadas a qué otras y cómo de fuertes son sus pesos? Sin embargo, la neurociencia ha demostrado recientemente que los circuitos biológicos a veces pueden alterar su función computacional gracias a algunos compuestos químicos que se difunden por el cerebro. Como, por ejemplo, el monóxido de nitrógeno, el sulfuro de hidrógeno... y con moléculas complejas neurotransmisoras como la acetilcolina (ACH), la dopamina (DA), las encefalinas y endorfinas, el ácido butírico amino-gama (GABA), la norepinefrina (NE), la serotonina (5-HT)... Es decir, la comunicación por volumen reemplaza a la comunicación punto a punto.

Optimización y algoritmos genéticos (*modelo evolutivo*)

El número de Shannon, 10^{120} , es una estimación de la complejidad del árbol de juego del ajedrez. Fue calculado por primera vez por Claude Shannon, el padre de la teoría de la información.

De acuerdo con su cálculo, se realizan una media de 40 movimientos en una partida de ajedrez, mientras que cada jugador escoge un único movimiento de unos 30 posibles. Así, tenemos que son posibles $(30 \times 30)^{40}$, i. e., 900^{40} juegos de ajedrez diferentes. De manera aproximada se dice que es igual a 10^{120} , valor que se obtiene de resolver la ecuación: $900^{40} = 10^x$. Despejando, tenemos que: $x = 40 \times \log 900$.

Actualmente, la complejidad de árbol de juego del ajedrez se calcula en torno a 10^{123} (el número de posiciones legales en una partida de ajedrez se estima entre 10^{43} y 10^{50}). Como comparación, el número de átomos que se estima que existen en el universo son entre unos 4×10^{78} a 6×10^{79} .

<https://es.wikipedia.org/wiki/N%C3%BAmero_de_Shannon#:~:text=Actualmente%20la%20complejidad%20de%20%C3%A1rbol,a%206%C3%971079>

Como dice Razvan Lagar del Consejo Superior de Investigaciones Científicas Español (CSIC), resolver el problema del ajedrez consistiría en establecer una estrategia óptima para jugar una partida; es decir, encontrar el camino que contiene las mejores jugadas tanto para las blancas como para las negras, desde el principio hasta el final. Al día de hoy, se trata de un problema abierto que ha surgido a partir del desarrollo de los programas informáticos de ajedrez. Vista la complejidad del árbol del juego, ninguna máquina actual podrá completar esta tarea. Fíjense que estoy hablando de «estrategia óptima». Sin embargo, todos hemos oído de la existencia de algoritmos de computador que pueden jugar al ajedrez y derrotar a jugadores profesionales en condiciones de torneo, y otros han vencido a muchos jugadores campeones del mundo en tiempos de control muy cortos. Si tiene curiosidad por saber cómo, véase

<https://es.wikipedia.org/wiki/Ajedrez_por_computadora>

La optimización hace referencia a la acción y efecto de optimizar. En términos generales, se refiere a la capacidad de hacer o resolver alguna cosa de la manera más eficiente posible y, en el mejor de los casos, utilizando la menor cantidad de recursos. Los problemas de optimización combinatoria surgen en el área de la matemática aplicada, en ciencias de la computación, en inteligencia artificial y en ingeniería de *software*. Existen dos tipos de problemas de optimización:

- Los continuos, que son los que pueden ser resueltos tomando un rango de valores, comúnmente utilizando los números reales. Esta familia de problemas tiene complejidad computacional polinomial (P).

- Los combinatorios, que son aquellos en los que las variables de decisión son enteras, es decir, donde el espacio de soluciones está formado por ordenaciones o subconjuntos de números naturales. Los algoritmos de optimización combinatoria resuelven instancias de problemas que en el caso general se creen que son muy difíciles, de hecho, estos problemas son de complejidad computacional no polinomial (NP). Esos algoritmos logran esto reduciendo el tamaño efectivo del espacio (aun así, muy grande), y explorando el espacio de búsqueda eficientemente, es decir, se recurre a explorar solamente un espacio de soluciones menor en lugar de todas las soluciones posibles, de esta manera se reduce el espacio de búsqueda y así se resuelven de forma eficiente.

Las técnicas aproximadas sacrifican la garantía de encontrar el resultado óptimo a cambio de obtener una buena solución en un tiempo razonable. Se distinguen tres tipos: métodos constructivos, métodos de búsqueda local y las técnicas metaheurísticas:

- Los métodos constructivos suelen ser los más rápidos. Partiendo de una solución vacía, a la que se le van añadiendo componentes, generan una solución completa. Las soluciones ofrecidas suelen ser de muy baja calidad. Su planteamiento depende en gran parte del tipo de problema. Es muy difícil encontrar métodos de esta clase que produzcan buenas soluciones, y en algunas ocasiones es casi imposible, por ejemplo, en problemas con muchas restricciones.
- Los métodos de búsqueda local usan el concepto de *vecindario* y se inician con una solución completa recorriendo parte del espacio de búsqueda hasta encontrar un óptimo local. El vecindario de una solución es el conjunto de soluciones que se pueden construir a partir de aquella aplicando un operador de modificación denominado *movimiento*. Estos métodos parten de una solución inicial, examinan su vecindario y eligen el mejor vecino continuando el proceso hasta que encuentran un óptimo local. En función del operador de movimiento utilizado, el vecindario cambia y el modo de explorar el espacio de búsqueda también, pudiendo la búsqueda complicarse o simplificarse.
- Las técnicas metaheurísticas son algoritmos no exactos que usan diferentes métodos heurísticos para explorar el espacio de búsqueda de gran tamaño, con el objetivo de encontrar una solución óptima o cercana al óptimo. Debe identificar rápidamente las regiones prometedoras del espacio de búsqueda global, y no malgastar tiempo en regiones que hayan sido exploradas y/o no contengan soluciones de alta calidad. Hay diferentes formas de clasificar estas técnicas: las que están basadas en la naturaleza (algoritmos bioinspirados) y las que no basadas en la naturaleza, sino más bien en memoria o sin memoria, con función objetivo estática o dinámica, etc.

La computación evolutiva es una rama de la inteligencia artificial que involucra problemas de optimización combinatoria. Se inspira en los mecanismos de la evolución biológica, es decir, en la capacidad de la evolución de seres o individuos para adaptarlos a los cambios de su entorno. Cada individuo representa una posible solución.

En un **algoritmo evolutivo** se define una estructura de datos que admita todas las posibles soluciones a un problema. Cada uno de los posibles conjuntos de datos admitidos por esa estructura será una solución al problema. Unas soluciones serán mejores, otras peores. Solucionar el problema consistirá en encontrar la solución óptima y, por tanto, los algoritmos evolutivos son en realidad un método de búsqueda. Pero un método de búsqueda muy especial, en el que las soluciones al problema son capaces de reproducirse entre sí, combinando sus características y generando nuevas soluciones. En cada ciclo se seleccionan las soluciones que más se acercan al objetivo buscado, eliminando el resto de las soluciones. Las soluciones seleccionadas se reproducirán entre sí, permitiendo de vez en cuando alguna mutación o modificación al azar durante la reproducción.

El funcionamiento básico de estos algoritmos es el siguiente: la población se genera de forma aleatoria. Cada individuo de la población tiene asignado un valor de su bondad con respecto al problema considerado, por medio de una función de aptitud, capacidad, adaptabilidad o estado, también denominada con bastante frecuencia por la palabra inglesa *fitness*.

El valor de la aptitud de un individuo es la información que el algoritmo utiliza para realizar la búsqueda. La modificación de la población se efectúa mediante la aplicación de tres operadores:

- Selección, recombinación (cruce) y mutación.

En estos algoritmos se pueden distinguir la fase de selección, explotación de buenas soluciones, y la fase de reproducción, búsqueda de nuevas regiones. Se debe mantener un equilibrio entre estas dos fases. La política de reemplazo permite la aceptación de nuevas soluciones que no necesariamente mejoran las existentes.

Los algoritmos más famosos de este tipo reciben el nombre de *algoritmos genéticos*:

- **Algoritmos genéticos**

Son una técnica metaheurística de búsqueda basada en la teoría de la evolución biológica y del de selección natural. Desarrollada inicialmente por Holland en 1975, su objetivo consistía en que las computadoras aprendieran por su cuenta; por tanto, los inicios de los algoritmos genéticos no fueron

planteados para la optimización combinatoria, sino para la inteligencia artificial.

Algoritmos genéticos. Un ejemplo:

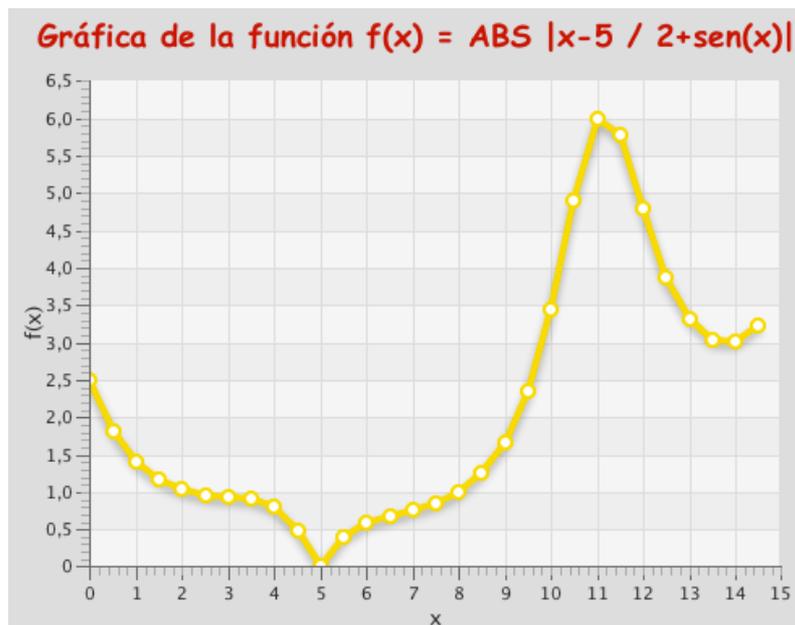
Véase: <<https://jarroba.com/algoritmos-geneticos-ejemplo/>>

Los algoritmos genéticos son un tipo de algoritmos evolutivos que sirven para resolver problemas de optimización y que se diferencia principalmente de los demás tipos de algoritmos evolutivos por la forma de representar a los individuos, que lo hacen mediante cadenas binarias.

Supóngase que se desea obtener el valor máximo de la siguiente función en el intervalo [0-15]

$$f(x) = ABS \left| \frac{x - 5}{2 + \sin(x)} \right|$$

Cuya gráfica es la siguiente



Empecemos representando a los individuos que podrán participar en las diferentes poblaciones con las que se va a buscar la solución al problema. En el caso de los algoritmos genéticos, los individuos se representan mediante cadenas binarias, dado que buscamos un número comprendido en el intervalo [0-15], la representación binaria de los números es una forma intuitiva de representar los cromosomas de cada individuo.

Como los valores van entre el 0 y el 15, su representación binaria vendría dada por la tabla siguiente

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

- Observemos que la longitud del cromosoma es 4.
- El conjunto de elementos que definen cada cromosoma son el valor 0 y el valor 1.

Ahora habría que definir lo que se conoce como *función de fitness*, que es la función que es capaz de medir la calidad de cada individuo de la población. En general no es una tarea fácil, ya que no es algo evidente, pero en este caso se puede elegir la propia función del problema. Esta función heurística tiene que dar un valor mayor cuanto más prometedor sea un individuo.

Ahora se trata de obtener la primera generación de individuos, posteriormente emparejarlos y realizar mutaciones:

- Seleccionemos el número de individuos de la población 2.
- Para la creación de la primera generación, definamos las siguientes probabilidades de aparición de los símbolos 0 y 1 de cada individuo:
 - Para el símbolo 0, se generará un número aleatorio con una probabilidad de valor $\leq 0,5$.
 - Para el símbolo 1, se generará un número aleatorio con una probabilidad de valor $\geq 0,5$.
- Se realizarán cruces con una probabilidad de 0,7.
- Se generarán mutaciones con una probabilidad de 0,3.

El conjunto de números aleatorios que se han ido generando a lo largo del algoritmo son los siguientes

0,34	0,82	0,77	0,71	0,35	0,75	0,48	0,40
0,41	0,74	0,15	0,85	0,51	0,44	0,89	0,85
0,43	0,07	0,97	0,93	0,11	0,58	0,75	0,90
0,51	0,62	0,67	0,15	0,89	0,87	0,86	0,77

- *Seleccionemos el número de individuos de la población 2*
- Para la creación de la primera generación, hemos definido las siguientes probabilidades de aparición de los símbolos 0 y 1:
 - Para el primer individuo, elijamos los primeros cuatro números de la tabla aleatoria.

- $0,34 \leq 0,5 \rightarrow 0.$
- $0,82 \geq 0,5 \rightarrow 1.$
- $0,77 \geq 0,5 \rightarrow 1.$
- $0,71 \geq 0,5 \rightarrow 1.$

Es el individuo [0,1,1,1] que representa al valor (7).

- Para el segundo individuo, elijamos los primeros cuatro números de la tabla aleatoria:
 - $0,35 \leq 0,5 \rightarrow 0.$
 - $0,75 \geq 0,5 \rightarrow 1.$
 - $0,48 \leq 0,5 \rightarrow 0.$
 - $0,40 \leq 0,5 \rightarrow 0.$

Es el individuo [0,1,0,0] que representa al valor (4).

- Evaluemos la calidad de cada individuo mediante la función de *fitness*:
 - $f(7) = 0,75.$
 - $f(4) = 0,8.$

Como se anda buscando el valor máximo, entonces $f(4)$ es el mejor por el momento.

- Se realizarán cruces con una probabilidad de 0,7, es decir:
 - Si la probabilidad está en el intervalo [0-0,7], habrá cruces.
 - Si la probabilidad está en el intervalo (0,7-1], no habrá cruces.
 Se calculan las probabilidades de cada individuo de cara al cruce

$$\text{Individuo 1} = \frac{0,75}{0,75+0,8} = 0,48, \text{ rango [0-0,48]}$$

$$\text{Individuo 2} = \frac{0,8}{0,75+0,8} = 0,52, \text{ rango (0,48-1]}$$

- Ahora hay que comprobar si los dos individuos son aptos para emparejarse en esta generación. Para ello es necesario generar dos nuevos números aleatorios, seleccionemos los dos primeros de la tabla:
 - Le asociaremos el valor 0,41 al individuo 1.
Como 0,41 está dentro del rango, entonces es apto para el emparejamiento.
 - Le asociaremos el valor 0,74 al individuo 2.
Como 0,74 está dentro del rango, entonces es apto para el emparejamiento.
- La siguiente pregunta que hay que hacerse es ¿habrá emparejamiento en esta generación? Para responder seleccionemos el siguiente número aleatorio de la tabla, es decir, 0,15. Como se ha dicho que habrá cruces cuando la probabilidad generada esté en el intervalo [0-0,7], podemos afirmar entonces que habrá emparejamiento y los dos individuos se

pueden emparejar. Veamos cómo se produce el emparejamiento entre estos dos individuos.

Para ello hay que calcular el «punto de corte» sobre el cromosoma sobre el que se va a hacer el intercambio. En un cromosoma de longitud 4, solo son posibles tres puntos de corte

Cada uno con los siguientes intervalos de probabilidad

- [0-0,333]
 - (0,333-0,666]
 - (0,666-1,0]

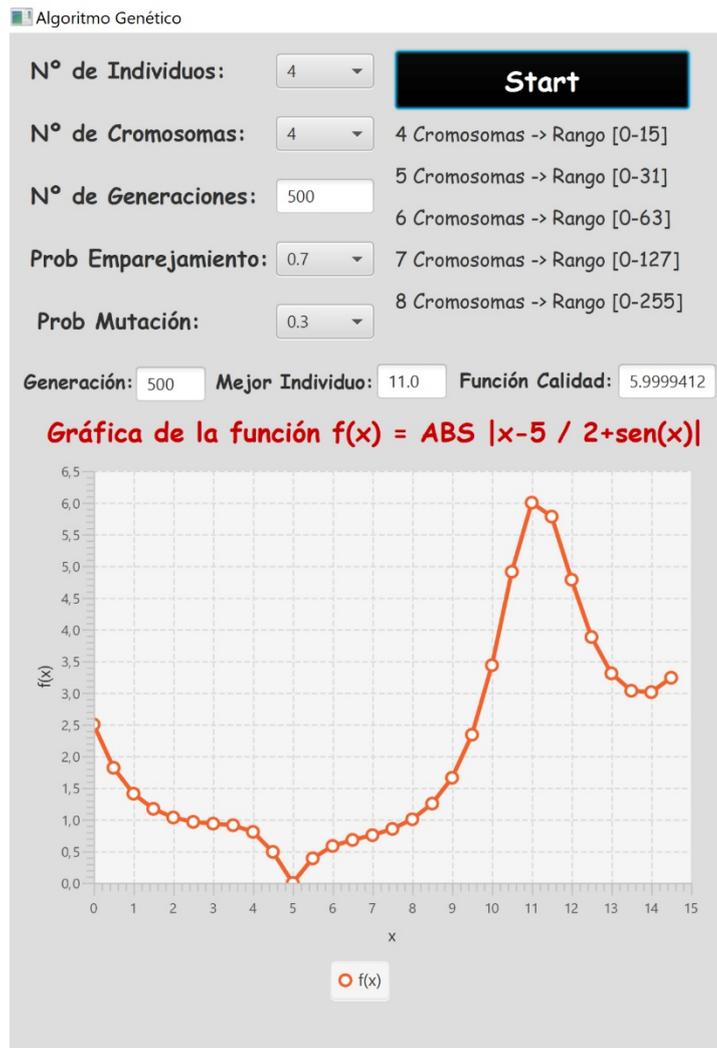
Volvamos a generar un nuevo número aleatorio, esta vez es el 0,85, por lo que el punto de corte será entre la posición 3 y 4.

- Ya se tiene todo para generar los nuevos individuos. Para ello lo que se hace es intercambiar entre los dos individuos que tenemos en este caso los valores del último bit, es decir:
 - El nuevo individuo 1 pasará de ser el [0,1,1,1] al [0,1,1,0].
 - El nuevo individuo 2 pasará de ser el [0,1,0,0] al [0,1,0,1].
- Y ahora toca ver si se producirán mutaciones que sabemos que serán con una probabilidad de 0,3, es decir, en el intervalo [0-0,3] las habrá y en el intervalo (0,3-1,0] no las habrá:
 - Seleccionemos 8 nuevos valores aleatorios, 4 para el primer individuo y 4 para el siguiente.
 - Para el primer individuo, elijamos los primeros cuatro números de la tabla aleatoria:
 - $0,51 \geq 0,3 \rightarrow$ no hay mutación y se mantiene el valor 0.
 - $0,44 \geq 0,3 \rightarrow$ no hay mutación y se mantiene el valor 1.
 - $0,89 \geq 0,3 \rightarrow$ no hay mutación y se mantiene el valor 1.
 - $0,85 \geq 0,3 \rightarrow$ no hay mutación y se mantiene el valor 0.
 - El individuo mutado es el valor [0,1,1,0] que representa al (6).
 - Para el segundo individuo, elijamos los primeros cuatro números de la tabla aleatoria:
 - $0,43 \geq 0,3 \rightarrow$ no hay mutación y se mantiene el valor 0.
 - $0,07 \leq 0,3 \rightarrow$ hay mutación y se cambia de valor al 0.
 - $0,97 \geq 0,3 \rightarrow$ no hay mutación y se mantiene el valor 0.
 - $0,93 \geq 0,3 \rightarrow$ no hay mutación y se mantiene el valor 1.
 - El individuo mutado es el valor [0,0,1,0] que representa al (2).

- En este momento ya se tienen los dos nuevos individuos de la nueva población, el individuo 1 [0,1,1,0], y el individuo 2 [0,0,1,0].

Y ya se está en disposición de volver a realizar una nueva iteración, así hasta que alcance la convergencia.

En https://github.com/jarroba/Algoritmo_Geneticov2, se puede encontrar un código en JAVA en el que se puede resolver este problema poniendo los individuos que se quieran, las generaciones que se quieran, en más rangos de la función, es decir, con individuos representados con más cromosomas y cambiando las probabilidades de emparejamiento y mutación. También se pueden ver todos los detalles y los pasos que da el algoritmo genético en cada generación realizada. A continuación se puede ver una imagen del programa que yo he utilizado, tras una ejecución con los parámetros que aparecen. Después de 500 generaciones, pueden ver que el máximo de la función se encuentra en el valor $f(11) = 6000$, y el código genético ha convergido.



El procedimiento arriba descrito es la base de la mayoría de las aplicaciones de los algoritmos genéticos. Como se puede observar, hay muchos detalles que son importantes de los que depende el algoritmo como, por ejemplo, cuál ha de ser el tamaño de la población, cuáles las probabilidades de cruce y de mutación. De esos «detalles» dependerá, en gran parte, el éxito o el fracaso del algoritmo genético que estemos aplicando. Asimismo, existen otros algoritmos genéticos mucho más complejos (y eficaces) que el que se ha utilizado, como aquellos que trabajan sobre otras representaciones además de las cadenas de bits, o los que emplean otros operadores de cruce y mutación, pero con este ejemplo se ha pretendido mostrar una primera aproximación del lector al campo de los algoritmos genéticos, no desarrollar un completo estudio de este.

Algunas aplicaciones de los algoritmos genéticos son:

- Optimización.
- Programación automática.
- Aprendizaje máquina.
- Economía.
- Simulación de sistemas.
- Simulación de genética de poblaciones.
- Simulación de sistemas sociales.
- ...

Ventajas:

• Los algoritmos genéticos son intrínsecamente paralelos, es decir, operan de forma simultánea con varias soluciones, en vez de trabajar de forma secuencial como las técnicas tradicionales. Esto significa que mientras técnicas tradicionales solo pueden explorar el espacio de soluciones hacia una solución en una dirección al mismo tiempo, y si la solución que descubren resulta subóptima, no se puede hacer otra cosa que abandonar todo el trabajo hecho y empezar de nuevo. Sin embargo, los algoritmos genéticos simplemente desechan esta solución subóptima y siguen por otros caminos.

• Cuando se usan para problemas de optimización resultan menos afectados por los máximos locales (falsas soluciones) que las técnicas tradicionales. Muchos algoritmos de búsqueda pueden quedar atrapados en los óptimos locales, ejemplo: si llegan a lo alto de una colina del paisaje adaptativo, descubrirán que no existen soluciones mejores en las cercanías y concluirán que han alcanzado la mejor de todas, aunque existan picos más altos en algún otro lugar del mapa, situación que no sucede para algoritmos genéticos. Otra ventaja es su habilidad para manipular muchos parámetros simultáneamente. Resulta interesante en caso de tener varios objetivos que resolver.

• No necesitan conocimientos específicos sobre el problema que intentan resolver. Realizan cambios aleatorios en sus soluciones candidatas y luego utilizan la función de aptitud para determinar si esos cambios producen una mejora o no.

- Resulta sumamente fácil ejecutarlos en las modernas arquitecturas masivas en paralelo.

- Usan operadores probabilísticos, en vez de los típicos operadores determinísticos de las otras técnicas.

Desventajas:

- La definición de una representación del problema es bastante heurística.

- Pueden tardar mucho en converger, o no converger en absoluto, dependiendo en cierta medida de los parámetros que se utilicen tamaño de la población, número de generaciones... Pero existe evidencia empírica de que se encuentran soluciones de un nivel aceptable en un tiempo competitivo comparado con otros algoritmos de optimización combinatoria. Aunque para ciertos problemas existen algoritmos específicos que pueden superar a los algoritmos genéticos.

Para saber más: véase

https://es.wikipedia.org/wiki/Algoritmo_gen%C3%A9tico

Dentro de lo que es la computación evolutiva, se engloban diferentes estrategias para la resolución de problemas de optimización. Estas estrategias reciben los nombres siguientes:

- Procesos de búsqueda evolutiva. Propuesta por Alan Turing en 1948.
- Estrategias evolutivas (EE). Propuesta por Rechenberg en 1964.
 - Representan a los individuos con vectores de números reales.
- Programación evolutiva (PE). Propuesta por Fogel en 1965.
 - Utilizan máquinas de estados finitos.
- Algoritmos genéticos. Propuesta por Holland en 1975.
 - Representan a los individuos mediante cadenas binarias.
- Programación genética. Propuesta por Koza en 1992.
 - Utilizan árboles de funciones.

La diferencia entre esos algoritmos propuestos radica en la forma en la que representan a los individuos. Aspecto que no deja de ser bastante irrelevante.

Algoritmos de comportamiento colectivo

- **Algoritmos basados en colonias de hormigas**

Estos algoritmos se inspiran en el comportamiento de las hormigas en la búsqueda de alimento. Inicialmente, las hormigas exploran un área cercana al hormiguero de forma aleatoria. Cuando una hormiga encuentra comida la lleva de vuelta al hormiguero. En el camino, la hormiga va depositando una sustancia química denominada *feromona* que guía al resto de hormigas a encontrar la comida. El rastro de feromona sirve a las hormigas para encontrar el camino más corto entre el hormiguero y la comida. Cuantas más veces una hormiga tras conseguir éxito pase por un camino dejando el rastro de hormona, más fuerte es dicho rastro dando a entender al resto de hormigas que ese camino lleva a una fuente mayor de comida; o, dicho de otro modo, la probabilidad de encontrar comida siguiendo ese rastro es mayor que el de otro con menos intensidad del rastro.

- **Algoritmos basados en nubes o enjambres de partículas**

Estos algoritmos son técnicas metaheurísticas inspiradas en el comportamiento del vuelo de las bandadas de aves o en el movimiento de los bancos de peces (cardumen). La toma de decisión por parte de cada individuo o partícula se realiza teniendo en cuenta una componente social y una componente individual, mediante las que se determina el movimiento de esta partícula para alcanzar una nueva posición. Es una técnica relativamente reciente desarrollado originalmente en Estados Unidos por el sociólogo James Kennedy y por el ingeniero Russ C. Eberhart en 1995.

Sistemas cognitivos (modelo corpóreo)

En todo lo que se ha visto hasta el momento, tanto de IA simbólica como la IA neuronal, no se ha requerido de la existencia de un cuerpo situado en un entorno real. Hay muchos investigadores que afirman que un sistema inteligente necesita de un cuerpo que le permita tener experiencias directas con el entorno. No basta con que un programador simule la inteligencia proporcionándole descripciones abstractas de ese entorno codificadas en un lenguaje de representación como es el caso de la IA simbólica o con millones de datos de entrenamiento como ocurre con las redes neuronales artificiales. Por ello, la IA corpórea debería modelar todo ese tipo de aspectos.

Antes de introducir el modelo corpóreo, reflexionemos sobre la siguiente paradoja asociada al concepto de *simulación*:

- ¿Por qué los meteorólogos no se mojan durante la simulación de una tormenta?
- ¿Por qué los astrofísicos pueden simular un agujero negro sin miedo a ser engullidos por el espacio / tiempo deformado simulado en su ordenador?

La respuesta es que una simulación no tiene el poder causal de hacer que el vapor de agua atmosférico se condense en agua o que el espacio tiempo se curve.

Sin embargo, sería posible alcanzar un nivel humano de consciencia si vamos más allá de las simulaciones y desarrollamos *hardware* con propiedades neuromórficas, basado en una arquitectura construida a imagen y semejanza del sistema nervioso. Debo aclarar que «estamos empezando» a ser capaces de concebir como podrían ser las respuestas. Pero aprovecho para insistir en que es un buen tema de investigación.

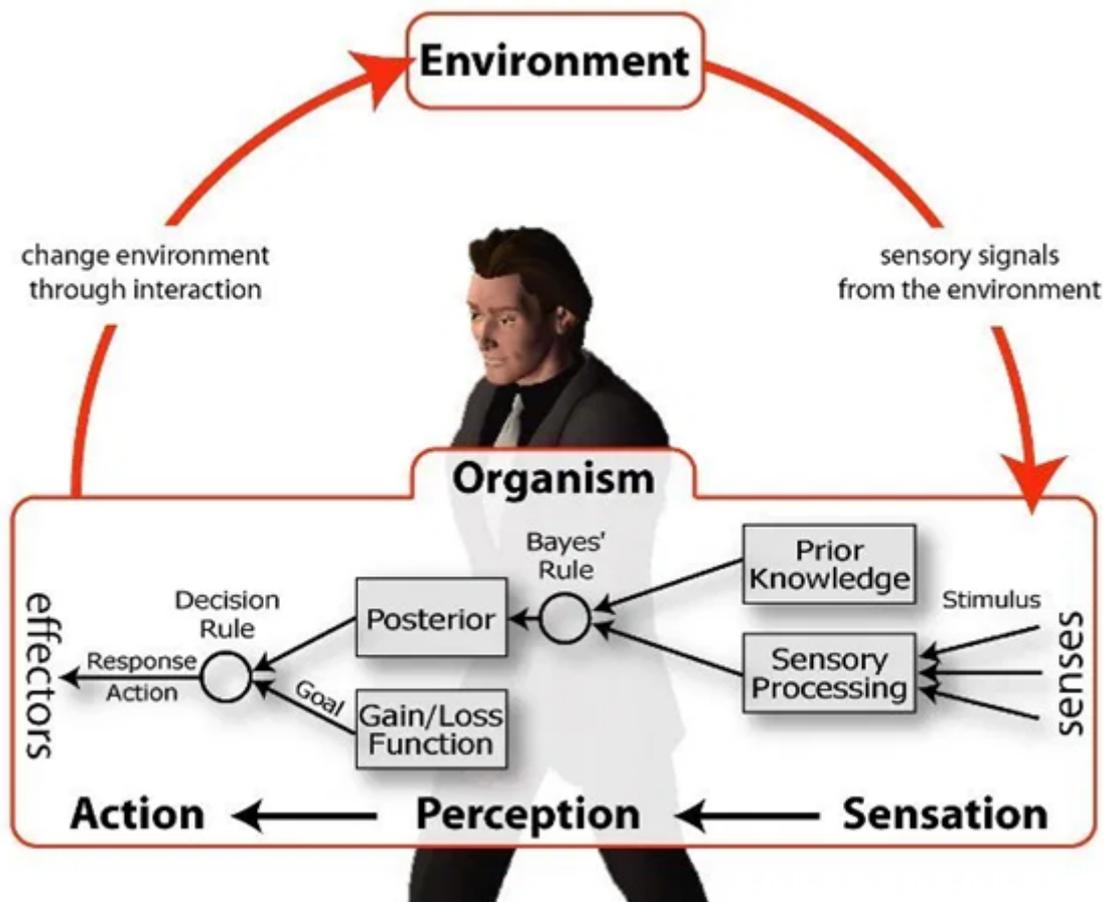
Fueron los investigadores Daniel Bobrow y Allan Collins los primeros en utilizar y divulgar la denominación de «ciencia cognitiva». En la obra colectiva (con diecisiete autores), editada por ellos en 1975 titulada *Representation and Understanding. Studies in Cognitive Science*, dicen en el prefacio:

Este libro contiene estudios en un nuevo campo que llamamos ciencia cognitiva. La ciencia cognitiva incluye elementos de psicología, ciencia de la computación, lingüística, filosofía y educación, pero es más que la intersección de estas disciplinas. Su integración ha producido un nuevo conjunto de instrumentos para ocuparse de un amplio campo de cuestiones. En los años recientes, las interacciones entre los que han trabajado en estos campos han conducido a excitantes nuevos desarrollos en nuestra comprensión de los sistemas inteligentes y al desarrollo de una ciencia de la cognición. El grupo de trabajadores se ha dedicado a problemas que no parecían resolubles desde dentro de una disciplina aislada.

Partamos nosotros de una afirmación incuestionable, los seres vivos del mundo animal pueden sobrevivir en entornos con una complejidad alta y a veces muestran

comportamientos sorprendentes. Los términos *adaptación*, *comportamiento* y *generación de diversidad comportamental* por su propia naturaleza implican la existencia de un cuerpo que interactúa con un entorno. Para conseguirlas, entre muchas otras cosas, son necesarias, habilidades motoras, perceptuales que requieren de un cierto tipo de inteligencia. Y a ello va unido otro hecho incuestionable, que es la idea de que la inteligencia de los seres vivos deriva del hecho de estar situados en un ambiente con el que interactúan. El proceso evolutivo natural sería un ejemplo de que sin necesidad de un diseñador los sistemas naturales han alcanzado capacidades impresionantes.

Por lo dicho, las arquitecturas cognitivas intentan resolver el problema de cómo integrar los distintos componentes de la inteligencia explorando hipótesis acerca de la naturaleza de la inteligencia y de las posibles interacciones entre dichos componentes.



De alguna manera intentan resolver el problema planteado por lo que se conoce como *falacia mereológica*, que tiene que ver con el estudio de las relaciones entre partes, tanto de las partes con el todo como de las partes con otras partes. En el tema que nos interesa, el cerebro por sí solo no puede hacer todo lo que se le atribuye, de hecho, no puede hacer mucho sin el resto del cuerpo y de sus partes. Así, caer en la falacia mereológica es atribuir al cerebro, o a algunas de sus partes, propiedades y acciones que en realidad son realizadas por las personas. Del mismo

modo que sería absurdo decir que no es el halcón sino sus alas las que vuelan, sería falaz decir que el cerebro piensa, reflexiona o decide. Nos dejamos llevar frecuentemente por estas asunciones simplemente porque nos es más simple comprender cómo funciona la mente si nos dejamos llevar por el reduccionismo, y no porque investigaciones científicas hayan demostrado que este conjunto de órganos razone o piense al margen del resto del cuerpo.

La inteligencia

Si uno observa a la naturaleza en relación con la manifestación en los seres vivos de la inteligencia, se pueden sacar unas pocas conclusiones claras:

- La naturaleza actúa de forma fundamentalmente paralela, lo que se refleja en la arquitectura de los organismos vivos.
- Comportamientos tremendamente complejos pueden obtenerse de conjuntos de componentes con funciones extremadamente sencillos.

Por lo tanto, para obtener comportamientos complejos, sería suficiente con definir (modelar) una serie de comportamientos básicos, establecer relaciones entre ellos, de cooperación, de exclusión, de competencia, y observar si el comportamiento «emergente» de esas interacciones en un contexto determinado es lo que se esperaba del sistema.

Ahora introduzcamos el concepto de *emergencia*. Piense en dos habitaciones. En la primera, puede observar extendidas sobre el suelo todas las piezas que componen una bicicleta, y en la segunda, se encuentra la bicicleta montada. En las dos están las mismas piezas, pero hay una sutil diferencia, en la segunda están interconectadas (relacionadas) de una manera adecuada lo que permite que interactúen entre sí. Gracias a ese simple hecho «emerge» la propiedad que ofrece la bicicleta de poder desplazar a una persona que se monte en ella y sepa mantener el equilibrio cuando pedalea. Desde un punto de vista de categorización de sistemas, lo que hay en la primera habitación es un sistema complicado y lo que hay en la segunda es un sistema complejo. La diferencia entre ambos radica en que el comportamiento de cualquier sistema complicado se deduce entendiendo cómo funciona cada una de las partes que lo componen, pero para comprender el funcionamiento de cualquier sistema complejo es necesario añadir a lo anterior el conocimiento sobre cómo las partes se relacionan entre sí. De las interacciones entre las partes emergen siempre propiedades nuevas, que no pueden explicarse a partir de las propiedades de los elementos aislados.

Empezaré afirmando que normalmente en sus inicios, los sistemas tienden a ser sistemas complicados, pero aquellos dirigentes que conocen la potencialidad de los sistemas complejos saben que deben tratar de hacer tres cosas, la primera mejorar las piezas o sus relaciones de la manera más adecuada, la segunda esperar a ver qué emerge de la nueva estructura, ya que normalmente no es evidente el resultado, y la tercera es volver al punto primero. Como ve se genera un bucle de actuaciones que

hará que el sistema evolucione y del que se espera que vaya ofreciendo poco a poco estructuras mejores, más eficaces y con mayor capacidad de resolución del problema para el que se diseñaron. En el caso de la bicicleta, desde que se inventó allá por 1817, esta ha ido evolucionando y ofreciendo modelos más adecuados para el transporte humano.

El porqué de la necesidad de ese comportamiento en bucle radica en que los sistemas complejos son dinámicos, no lineales, y no predecibles con exactitud. Sabemos que el avance no puede surgir nunca de un sistema complicado, lo hace siempre desde un sistema complejo exitoso, lo que puede que no ocurra siempre. Pero una vez conseguido por primera vez, si no se entra en el bucle, no hay mucho más que esperar. Piense qué hubiera pasado si nos hubiéramos conformado con el primer modelo de bici.

La forma clásica de pensar sobre la inteligencia es que los humanos somos procesadores de información que manipulamos símbolos y que la inteligencia está localizada en el cerebro. En el momento actual estas afirmaciones hay que completarlas / modificarlas con la necesidad de interacción de un cuerpo con un entorno.

Los procedimientos tradicionales desarrollan modelos de cálculo de las funciones mentales que tienen que ver con pensar, razonar y resolver problemas abstractos. Con esta aproximación el cerebro humano se ve como el soporte de la inteligencia, es decir, un sistema que recibe entradas del mundo exterior a través de los sentidos. Estas entradas se procesan y dependiendo de su estado interno y sus motivaciones, se genera la intención o un plan de actuación. De esta manera, el control del comportamiento supone una jerarquía en la cual hay niveles de control. Se emplean el principio de búsqueda de objetivos, de acuerdo con el cual el comportamiento resulta de una comparación entre una representación del objetivo y el estado actual. Pero esta noción de *jerarquía* también implica centralización y secuenciación. Esta idea general de control basada en objetivos y controles de comportamiento jerárquico se conceptualiza en función del concepto de *realimentación negativa*, es decir, la diferencia entre el estado presente del agente y el comportamiento inteligente que controla los objetivos. Y para ser capaz de planificar una secuencia de acciones, el agente debe tener:

- Un modelo interno que le permita estimar el resultado de una acción sin esperar a la realimentación sensorial o incluso sin realizar la acción.
- Una noción de distancia que mida la diferencia entre el objetivo y la situación actual.

Lo dicho no es lo que se está buscando en la aproximación cognitiva denominada *embodied cognitive science (ECG)*. En ella hay un desplazamiento del foco desde el procesado de la información a la corporeización y la coordinación sensorial / motora,

y se defiende que el comportamiento inteligente se puede obtener de manera emergente, es decir, no hay necesidad de que esté programada internamente.

Una de las suposiciones básicas de la aproximación basada en la ciencia cognitiva es que la inteligencia se debe estudiar en el contexto de la interacción sistema / entorno. El principio establece que todo el comportamiento inteligente se concibe como una coordinación sensorial / motora que sirve para estructurar la entrada.

Ese fenómeno de la emergencia se propone / comprueba que surge cuando existen un gran número de procesos paralelos acoplados ligeramente que funcionan de manera preponderante de forma asíncrona y paralela, y que están conectados al aparato sensorial / motor del agente. Y se postula que en todo comportamiento inteligente lo importante es la coordinación sensorial / motora y buscar inducir en el comportamiento del sistema bajo diseño regularidades que simplifiquen el aprendizaje como almacenamiento de información.

El aspecto de la coordinación que se acaba de citar está relacionado con la idea de que ni el sistema elige, ni lo hace el entorno. El sistema se construye de tal manera que las elecciones correctas emergerán de las interacciones entre sus movimientos, sus acciones y el entorno. La literatura ha sugerido dos aproximaciones principales, no excluyentes, a los procesos de coordinación, a saber:

- La coordinación de procesos competitivos, de tal manera que solo un proceso actúa sobre el exterior, el resto se desactiva o inhibe mientras el proceso seleccionado está activo.
- La coordinación cooperativa, de tal manera que la salida de dos o más procesos se combina para generar la misma actividad.

Por lo tanto, el arte de diseñar este tipo de sistemas es diseñar para la emergencia. La emergencia aparece funcionalmente a través de la interacción entre los procesos y no se diseñan con algún tipo de función particular en mente. Esta es una de las ideas centrales que subyacen en el principio de procesos paralelos ligeramente acoplados, el comportamiento inteligente emerge de ellos. En este paradigma, ni el sistema elige, ni lo hace el entorno. El sistema se construye de tal manera que las elecciones correctas emergerán de las interacciones entre sus acciones y el entorno.

El procesamiento interno necesario es mínimo y las señales sensoriales requeridas se aplican casi directamente a las señales motoras. Este tipo de arquitectura conduce a un ajustado acoplo entre el sistema y el entorno. Con este punto de vista, la inteligencia emerge de la interacción del organismo con su entorno.

En la IA tradicional los modelos utilizados son programas de computador (aproximación analítica), en el caso de la ECG los modelos toman la forma de sistemas autónomos corporeizados que hay que construir (aproximación sintética).

Principios de diseño de sistemas corpóreos

Estos principios representan solo una primera aproximación hacia una teoría más compleja de los sistemas corpóreos.

Hay tres componentes que intervienen en el diseño de este tipo de sistemas:

- El nicho ecológico o entorno en el que está inmerso el sistema.
- Las tareas y los comportamientos deseados.
- El propio sistema corporeizado.

Hay tres aspectos que hay que considerar en el diseño de un sistema:

- La morfología.
- La arquitectura.
- Los mecanismos.

Un sistema completo significa:

- Autónomo.
- Autosuficiente.
- Situado.
- Corporeizado.
- Diseñado para un nicho ecológico particular.

Definición del nicho ecológico (metaprincipio):

No existe universalidad en el mundo real. Los sistemas se diseñan siempre para un nicho particular, lo que se conoce como nicho ecológico, que no es más que el entorno en el cual el sistema va a operar. Las características del nicho es lo que restringe las condiciones de diseño.

Características de los sistemas que se pretenden construir:

Un sistema es autosuficiente, cuando tienen la habilidad de mantenerse activo durante un periodo amplio de tiempo en un mundo que tiene sus propias mecánicas. Para ello:

- Deben realizar múltiples tareas y diferentes comportamientos.
- Actuar en entornos cuyas condiciones pueden cambiar a lo largo del tiempo.

- El control del comportamiento. Debido a que los agentes tienen muchas posibles tareas que hacer, hay que resolver el problema del control del comportamiento: algunos de esos comportamientos serán compatibles, otros mutuamente exclusivos y el diseño debe tener en cuenta que las decisiones se deben tomar conforme los comportamientos se van necesitando a lo largo del tiempo. Es decir, hay que hacer lo correcto en el instante adecuado. Por lo tanto, se necesita un mecanismo para decidir qué acción elegir para ejecutarla en algún momento del tiempo, es decir, qué módulo interno hay que ejecutar. En otras palabras, debe resolverse el problema de seleccionar la acción.
- La autonomía. Cuando se habla de agentes autónomos, queremos significar que tienen cierto grado de autonomía. La autonomía significa independencia de control. No existen agentes totalmente autónomos. La controlabilidad y la capacidad de adquirir su propia historia van correlacionadas. Cuanto más controla un agente su propia historia, menos controlable es. Con respecto a la autonomía existen dos aspectos que considerar:
 - Con relación a la dependencia del entorno. No hay atributos intencionales.
 - Con relación a la dependencia de otros agentes. Hay atributos intencionales. En general los agentes pueden ser influenciables, y pueden depender de otros, pero no suelen ser completamente controlables.
- La situabilidad. Podemos decir que un agente está situado si adquiere información sobre su entorno solo cuando sus sensores interactúan con el entorno, de manera que tiene el potencial de adquirir su propia historia para lo que va a requerir su apropiado mecanismo de aprendizaje. Este tipo de agente es más autónomo que un agente reactivo preprogramado.
- La corporeizabilidad. Esto significa que existe una entidad física en el mundo real que está sujeto a las fuerzas físicas, a la disipación de energía, al daño, en general a cualquier influencia en el entorno. También puede existir en un entorno virtual, aunque requieren de un gran esfuerzo de modelado. Hay quien piensa que solo en este tipo de agentes puede surgir la inteligencia.
- La adaptabilidad. Este concepto significa la capacidad o habilidad para autoajustarse al entorno. También se denomina *homeostasis*, significando que ciertas variables esenciales permanecen dentro de unos límites que permiten al agente sobrevivir. Existen diversos tipos de adaptación que además trabajan sobre diferentes escalas temporales:

- Adaptación evolutiva, que tiene que ver con la forma mediante la cual las especies se ajustan genéticamente al cambio ambiental a lo largo de largos periodos de tiempo.
- Adaptaciones fisiológicas, que están relacionadas con los ajustes al cambio climático, alimentación...
- Adaptación sensorial, mediante la cual los órganos sensoriales se ajustan a los cambios que se producen en la intensidad de las estimulaciones para los que están diseñados sus detectores.
- Adaptación mediante aprendizaje, proceso mediante el cual los animales son capaces de ajustarse a una amplia variedad de tipos diferentes de cambios ambientales.

Como se ha demostrado, su diseño es un proceso muy complejo que envuelve consideraciones que van más allá de los problemas técnicos con los que hay que enfrentarse. Ello incluye las consideraciones de seguridad, sociales, ambientales, estéticas, psicológicas, mercado, política corporativa, prácticas y legales. Por lo tanto, el proceso de diseño se puede formalizar solo para dominios bien definidos.

En el paradigma corporeizado se diseña para que se produzca el fenómeno de la emergencia. Es decir, sistemas que deben actuar en un cierto nicho ecológico y realizar un cierto tipo de comportamientos.

De una manera concisa, los principios que hay que considerar en el diseño para alcanzar lo que intuitivamente denominamos *sistema cognitivo corporeizado, autónomo, autosuficiente y situado* son:

- El principio de procesos paralelos ligeramente acoplados.
- El principio de la coordinación sensorial / motora.
- El principio del diseño parsimonioso, bueno y barato. Es decir, hay que explotar: la física de la interacción sistema / entorno, las restricciones del nicho ecológico y que el sistema sea lo más simple y sencillo (parsimonia = navaja de Occam).
- El principio de redundancia en los canales sensoriales, es decir, el principio establece que los sensores tienen que posicionarse en el agente de tal manera que exista un cierto solape en la información adquirida a partir de los diferentes canales sensoriales.

- El principio del balance ecológico, que establece que la complejidad del agente no puede ser mayor que la del nicho ecológico y los comportamientos deseados.
- El principio de valorización que establece que el agente debe estar equipado con un sistema que permita valorar y autosupervisar su mejora incremental del aprendizaje empleando principios de autoorganización. El sistema de valoración puede ser explícito, si los valores de las señales que modulan el aprendizaje se generan como una consecuencia de su comportamiento. El sistema de valoración puede ser implícito cuando se obtiene mediante mecanismos que seleccionan interacciones con el entorno que conducen a un aumento de la adaptabilidad. En aprendizaje basado en la valoración, la modulación del aprendizaje ocurre *a posteriori*. Los sistemas de valoración se activan solo después de que un agente ha realizado un comportamiento, y tienen carácter evaluativo.
- ...

Los problemas fundamentales de la IA clásica y de las ciencias cognitivas

Desde este punto de vista, la construcción de sistemas de IA o CG está inspirada en los seres vivos naturales, animales y humanos, que son capaces de sobrevivir en el mundo real. Son completos porque incorporan todo lo necesario para realizar su comportamiento natural. De la naturaleza han emergido y emergen criaturas capaces de sobrevivir en el mundo real. Un pequeño recordatorio:

- 3,5 miles de millones de años las células simples.
- 500 millones de años peces y vertebrados.
- 450 millones de años insectos.
- 370 millones de años reptiles.
- 330 millones de años dinosaurios.
- 250 millones de años mamíferos.
- 120 millones de años primates.
- 18 millones de años grandes monos.
- 2,5 millones de años homínidos.

Apliquemos el sentido común, tomemos como referencia un ser humano y veamos si está de acuerdo con lo que se dice en los párrafos siguientes. Alrededor de nosotros giran conceptos como *inteligencia, pensamiento, conciencia, aprendizaje, memoria, intuición, creatividad, emociones, sentido común...*

Si se habla sobre inteligencia, se considera que no es una característica que está presente o no, sino que está presente de forma gradual. Las personas parece que podemos distinguir de manera clara niveles de inteligencia. De hecho, tenemos una tendencia a ordenar a los seres vivos en función de su inteligencia.

Muchas personas coinciden en que la inteligencia no es una componente del cerebro o una cosa, sino más bien una propiedad de un organismo que emerge de la interacción de un determinado conjunto de subsistemas como son: la percepción, la memoria, la planificación, el aprendizaje...

Cuando nos referimos al pensamiento, se incluye entre sus capacidades la resolución de problemas, el razonamiento lógico y también formas de actividad mental tales como las que utilizamos en nuestras vidas todos los días para desenvolvernos. Entre los posibles tipos de formas de pensar, destaca el pensamiento abstracto... y la conciencia, que se considera como la experiencia subjetiva del control ejecutivo de la mente.

La capacidad de aprender es también una habilidad bastante popular. Muchas personas ven el aprendizaje como la propiedad central de la inteligencia. Pero el aprendizaje *per se* no hace a la gente inteligente. Lo que sí parece una propiedad interesante es aprender a aprender. En el proceso de aprender, la memoria se

considera también importante, aunque lo importante es la transferencia de conocimiento no el mero almacenamiento. De hecho, la memoria podría ser la habilidad mejorada para categorizar y generalizar asociativamente, no el almacenamiento de características y atributos de los objetos en una lista. Y en todo ello, la capacidad para comunicarse en lenguaje natural se considera el sello de calidad de la inteligencia. Lo que es muy interesante es que categorizamos el mundo conforme vamos aprendiendo. Pero como el mundo es desconocido y las categorías son desconocidas, Los procesos humanos de aprendizaje son incrementales y duran toda la vida.

La intuición se toma como el medio para llegar a conclusiones sin un hilo conductor de pensamiento lógico que pueda trazarse desde el principio. Y la creatividad incluye no solo la individual sino la reconocida por la sociedad. No parece provenir de un individuo aislado, sino más bien se discute con respecto los criterios de una sociedad particular. Se considera como la forma más alta de la inteligencia humana.

La inteligencia emocional se refiere a la habilidad para reconocer emociones en otros, usar las emociones para soportar el pensamiento y las acciones, comprender las emociones y regularlas. De hecho, una emoción es un estado mental y psicológico asociado con una amplia variedad de sentimientos, pensamientos y comportamientos. Las emociones son experiencias subjetivas, a menudo asociadas con el estado anímico, el temperamento, la personalidad y la disposición.

El ser humano interpreta el sentido común desde el contexto, saber leer entre líneas. Únicamente conociendo los significados de las palabras y la gramática no es suficiente. Y es que el lenguaje que los humanos utilizamos convencionalmente está plagado de elementos extralingüísticos (gestos, entonaciones, conocimiento previo de intenciones), así como otros netamente contextuales. A todos estos aspectos incontrolables se los ha denominado de diversas formas: conocimiento implícito, sentido común o, simplemente, contexto.

Se podría seguir con las descripciones de conceptos relacionados con la inteligencia, pero en estos momentos vamos a analizar rápidamente los aspectos que influyen de manera importante en la inteligencia de una persona:

- La naturaleza (*nature*) con sus cualidades innatas (factores genéticos).
- La educación (*nurture*) con las experiencias personales vividas en los diferentes entornos desde que se nace hasta el momento presente.

Además, si hacemos una pequeña introspección sobre cómo nos comportamos cuando nos consideramos como sistema podríamos decir lo siguiente:

- Somos autosuficientes y tenemos la habilidad de mantenernos vivos durante un periodo amplio de tiempo en un mundo que tiene sus propias mecánicas.

- Tenemos un cierto grado de autonomía, ya que esta propiedad depende de nuestras relaciones con los «otros».
- Tenemos capacidad de adaptación, es decir, nos ajustamos al entorno en el que estamos inmersos. También se denomina *homeostasis*, es decir, que ciertas variables esenciales para la supervivencia permanecen dentro de unos límites adecuados.
- Tenemos la capacidad de realizar múltiples tareas y mostrar diferentes comportamientos unas veces compatibles entre sí y otras veces incompatibles.
- Intentamos hacer lo correcto en el instante adecuado, pero no suele estar claro cuál es el mejor comportamiento, es decir, cuáles son las acciones adecuadas que hay que ejecutar.
- Somos robustos y capaces de actuar en entornos con:
 - Tolerancia al ruido.
 - Tolerancia a los fallos, es decir, a veces podemos seguir funcionando, aunque algún elemento deje de funcionar bien. Es decir, ante muchas situaciones, somos redundantes.
 - Capacidad de reaccionar, casi siempre, de manera adecuada ante situaciones completamente nuevas.
- Procesamos la información en tiempo real. El mundo tiene una dinámica, y operamos en él reaccionando rápidamente para poder sobrevivir y realizar las tareas que hayamos planificado. También es verdad que la resolución de problemas complejos nos lleva a fases de reflexión antes de tomar las decisiones que se consideren.
- Como sistema vivo que somos, procesamos la información de manera masivamente paralela.

A pesar de todo lo dicho, piense que no existe el «animal universal» capaz de desenvolverse en cualquier nicho ecológico o social, ni por lo tanto puede existir el «sistema universal». Y además cumplimos el principio del balance ecológico, es decir, nuestra complejidad y comportamientos no son mayores que la del nicho ecológico que ocupamos. De ahí se deduce que, si se diseña un sistema, su complejidad (sensores, sistema motor y sistema cognitivo) tiene que casar con la complejidad de las tareas que tiene que realizar en su entorno.

Ahora bien, cuando nos enfrentamos al problema de querer construir un sistema artificial que intente comportarse de manera análoga a cómo nos comportamos nosotros, entonces surge un conjunto de problemas cuya resolución es fundamental:

- El problema del marco. Este problema es intrínseco a cualquier modelo que se utilice de un mundo cambiante. La pregunta es cómo los procesos cognitivos determinan qué información, de entre toda la disponible, es relevante dada una tarea determinada, es decir, qué es relevante y qué no lo es, entre una inmensa cantidad de información recibida. Por ejemplo, cuando se presta atención a una película, su cerebro filtra cualquier información que no esté relacionada con la visión y la audición, aunque el sistema perceptual está enviando muchísima más información. Piense en un robot en la misma situación, ¿cómo sabe qué tiene que filtrar y qué no? En la vida real, las actuaciones siempre son dependientes del contexto y dependen de tantas condiciones posibles que de hecho ni siquiera se pueden enumerar. Algunos problemas concretos son el:
 - Problema de cualificación. Nunca se sabe a ciencia cierta, si una regla específica trabajará en un instante determinado, ni que reglas se deben ignorar en una situación dada. Las modificaciones del entorno pueden hacer que esas reglas no sean las adecuadas por obsolescencia o puede que se requieran nuevas reglas que todavía no existen.
 - Problema representacional. Es la dificultad de generar certezas sobre el entorno en un momento determinado. Por ejemplo, ¿qué significa arriba y abajo?
 - Problema inferencial. En algunos momentos se hace necesario inspeccionar en el entorno todas las cosas que estén cambiando. Pero otras veces tan solo hay que inspeccionar aquellas cosas que pueden variar en un entorno local determinado.
 - Problema de ramificación. Cuando se comete un error de actuación, cómo hay que modificar el comportamiento. En nuestro mundo, existen demasiadas acciones posibles, cada una de las cuales puede tener muchas consecuencias implícitas.
 - Problema predictivo. Este problema está relacionado con los beneficios de una predicción. ¿Cambiará de manera positiva el entorno? Si no es así, las leyes o la descripción utilizada para predecir no son perfectas.
 - El problema de la corporeización. Los algoritmos abstractos no interactúan con el mundo real, por ello los sistemas inteligentes necesitan

estar corporeizados, es decir, deben estar sujetos a las leyes físicas, químicas, biológicas, al daño... Aunque también pueden existir en un entorno virtual siempre que estén simuladas esas interacciones.

- El problema de la situación. Este problema tiene que ver con la adquisición de información del exterior a través de los sensores que están en interacción con el entorno. Un agente situado tiene su propia perspectiva y no necesita observadores externos. El problema es mantener el modelo interno del entorno al día sin necesidad de que sea demasiado exhaustivos y detallados.

Siguiendo a Minsky, los seres humanos realizamos una curiosa actividad que ninguna otra criatura parece capaz de hacer: podemos ponernos a pensar sobre nuestros propios pensamientos y, si este pensamiento reflexivo nos muestra que nos hemos equivocado, esto nos puede ayudar a inventar modos de pensar nuevos y más poderosos. Sabemos todavía muy poco sobre el modo en que nuestro cerebro consigue hacer tal cosa. Y de hecho todavía no hemos resuelto preguntas como:

- ¿Qué son las emociones y los pensamientos?
- ¿Cómo forman nuestras mentes ideas nuevas?
- ¿Cuáles son las bases de nuestras creencias?
- ¿Cómo aprendemos a partir de la experiencia?
- ¿Cómo razonamos y pensamos?
- ¿Cómo funciona la imaginación?
- ¿Cuáles son las causas de la consciencia?
- ¿Qué son las emociones, los sentimientos y las ideas?
- En definitiva, ¿cómo nos las arreglamos para pensar?
- ¿Cuál es la naturaleza de la consciencia?
- ¿Cómo se relacionan nuestros cuerpos con nuestras mentes?
- ¿Qué es lo que forma nuestros valores, objetivos e ideales?

De hecho, tenemos problemas con la mayoría de los términos utilizados en psicología con son:

- Atención.
- Emoción.
- Percepción.
- Consciencia.
- Pensamiento.
- Sentimiento.
- Yo.
- Inteligencia.
- Placer.
- Dolor.
- Felicidad.

Ya que, en distintos momentos, cada una de esas palabras se refiere a distintos tipos de procesos, y de hecho cada uno de ellos puede ser el resultado de amplias redes de procesos que tienen lugar dentro de nuestros cerebros.

¿Por qué no nos damos cuenta de la complejidad de los procesos que realizamos? La razón es que la mayoría de las tareas se llevan a cabo en el interior de ciertas partes del cerebro cuyos procesos internos permanecen ocultos para el resto de

este. La razón para ello es que posiblemente nuestra mente no evolucionó para ser un instrumento que se observara a sí mismo, sino para resolver problemas prácticos como la alimentación, la defensa y la reproducción.

La experiencia que se tiene de cómo puede evolucionar eso que denominamos *mente* vendría a ser algo de este estilo:

- Cuando se nace, se viene provisto de diversas reacciones instintivas. Y ya desde el inicio se empiezan a añadir a las anteriores unas reacciones aprendidas para sobrevivir en entornos nuevos. (El cerebro reactivo)
- Posteriormente, con el paso del tiempo, añadimos de manera progresiva más modos deliberativos de razonar, imaginar y planificar para el futuro. Procesos necesarios para alcanzar objetivos más complejos, ya que necesitamos hacer unos planes más elaborados utilizando todos los conocimientos que hemos adquirido a partir de acciones realizadas en el pasado; y son esas actividades mentales internas las que nos proporcionan capacidades exclusivamente humanas. (El cerebro deliberativo)
- Más tarde, construimos un nuevo estrato en el que comenzamos a realizar un pensamiento reflexivo. No se reacciona ante sucesos externos, sino ante los que se producen dentro del cerebro. Estarían incluidas aquellas predicciones que resultaron erróneas, los planes que encontraron obstáculos y los fracasos a la hora de acceder a los conocimientos que uno necesita. A lo que habría que añadir el proceso de autorreflexión que es la reflexión sobre nuestros propios pensamientos. Este nivel trabaja más que el nivel reflexivo, ya que no solo toma en consideración algunos pensamientos recientes, sino que también piensa sobre la entidad que tuvo esos pensamientos. (El cerebro reflexivo)
- Finalmente, empezamos a pensar de una manera más autoconsciente sobre qué cosas podemos considerar correctas o erróneas. (El cerebro consciente)

Escritos de Santiago Sánchez-Migallón

El mundo de la inteligencia artificial es multidisciplinar, este documento tiene un planteamiento en el que evidentemente han intervenido de manera directa o indirecta: la Lógica, la Probabilidad, la Computación, la Economía, las Neurociencias, la Psicología, la Cibernética... pero no así la Filosofía.

Para dar idea de que sí que interviene en este mundo, he buscado referencias y me he encontrado sorprendentemente con el blog de Santiago Sánchez-Migallón Jiménez

Véase: <<https://es.linkedin.com/in/santiago-s%C3%A1nchez-migall%C3%B3n-jim%C3%A9nez-587ba727>>

El autor es un profesor de Filosofía, blogger, ensayista, conferenciante. Y enamorado de la filosofía de la mente y de la inteligencia artificial.

He seleccionado algunos de sus escritos por que complementan muy bien este documento desde el punto de vista de un profesor de Filosofía.

Lectura 1. «Soy»

Véase <<https://vonneumannmachine.wordpress.com/>>

Lectura 2. «Textos sobre los seres humanos»

Véase <<https://vonneumannmachine.wordpress.com/>>

Lectura 3. «Textos sobre la consciencia»

Véase <<https://vonneumannmachine.wordpress.com/>>

Lectura 4. «Textos sobre la creatividad»

Véase <<https://vonneumannmachine.wordpress.com/>>

Categorías de definiciones sobre IA

He dejado para el final algunas definiciones de inteligencia artificial, organizadas en cuatro categorías que aparecen en el libro de Russell...

Sistemas que piensan como humanos	Sistemas que piensan racionalmente
<p>«El nuevo y excitante esfuerzo de hacer que los computadores piensen... máquinas con mentes, en el más amplio sentido literal». (Haugeland, 1985)</p> <p>«[La automatización de] actividades que vinculamos con procesos de pensamiento humano, actividades como la toma de decisiones, resolución de problemas, aprendizaje...» (Bellman, 1978)</p>	<p>«El estudio de las facultades mentales mediante el uso de modelos computacionales». (Charniak y McDermott, 1985)</p> <p>«El estudio de los cálculos que hacen posible percibir, razonar y actuar». (Winston, 1992)</p>
Sistemas que actúan como humanos	Sistemas que actúan racionalmente
<p>«El arte de desarrollar máquinas con capacidad para realizar funciones que cuando son realizadas por personas requieren de inteligencia». (Kurzweil, 1990)</p> <p>«El estudio de cómo lograr que los computadores realicen tareas que, por el momento, los humanos hacen mejor». (Rich y Knight, 1991)</p>	<p>«La Inteligencia Computacional es el estudio del diseño de agentes inteligentes». (Poole <i>et al.</i>, 1998)</p> <p>«IA... está relacionada con conductas inteligentes en artefactos». (Nilsson, 1998)</p>

Figura 1.1 Algunas definiciones de inteligencia artificial, organizadas en cuatro categorías.

Las leyes de la robótica de Asimov

Las tres leyes de la robótica son un conjunto de normas elaboradas por el escritor de éxito Isaac Asimov que se aplican a la mayoría de los robots de sus obras y que están diseñados para cumplir órdenes. En ese universo, las leyes son «formulaciones matemáticas impresas en los senderos positrónicos del cerebro» de los robots (líneas de código del programa que regula el cumplimiento de las leyes guardado en la memoria principal de aquellos). Aparecidas por primera vez en el relato «Círculo vicioso» (*Runaround*, 1942), y establecen lo siguiente:

Las tres leyes robóticas (*Manual de Robótica*. Edición LVI. Año 2058):

- Un robot no debe dañar a un ser humano o, por su inacción, dejar que un ser humano sufra daño.
- Un robot debe obedecer las órdenes que le son dadas por un ser humano, excepto cuando estas se opongan a la primera ley.
- Un robot debe proteger su propia existencia, hasta donde esta protección no esté en conflicto con la primera o segunda leyes.

Así comienza el famosísimo *Yo robot* de Isaac Asimov (Ed. Edhasa, Barcelona, Col. Nebulae, n.º 1). Y en muchos de los cuentos del autor y de otros autores, la referencia a estas leyes es constante.

Esta redacción de las leyes es la forma convencional en la que los humanos de las historias las enuncian; su forma real sería la de una serie de instrucciones equivalentes y mucho más complejas en el cerebro del robot.

Dicho sistema legal asimoviano ha sido ampliado varias veces mediante las siguientes leyes:

- La «ley cero de la robótica» es una variación introducida en las leyes de la robótica que aparece por primera vez en la novela de Isaac Asimov *Robots e Imperio* (1985). «Un robot no hará daño a la Humanidad o, por inacción, permitir que la Humanidad sufra daño». Derivándose de la denominación «cero» que las otras tres leyes de la robótica quedan subordinadas jerárquicamente a esta nueva ley.
- Fuller (1999) propone una cuarta ley: «Un robot podrá realizar el trabajo de un ser humano, pero no debe dejar a esta persona sin empleo».
- Andrés H. Una inteligencia artificial está obligada a identificarse como tal si un humano se lo pregunta.

Las tres leyes de Asimov ahora se han visto aumentadas o mejoradas por la Unión Europea. El Parlamento ha decidido regular nuestro futuro y para ello propone seis

leyes que normalicen la convivencia con los robots. Todavía no son definitivas, ya que esta normativa tiene que pasar por el filtro de la Comisión Europea, pero su principal objetivo es reducir el impacto que supondrá la implantación en la sociedad de estas máquinas, ya que uno de los mayores efectos será la pérdida de trabajo.

- 1. Los robots deberán tener un interruptor de emergencia.
- 2. Los robots no podrán hacer daño a los seres humanos.
- 3. No podrán generarse relaciones emocionales con los robots.
- 4. Los que sean más grandes o puedan generar más riesgo deberán tener un seguro obligatorio.
- 5. Los robots tendrán derechos y obligaciones, de tal manera que sean ellos los que asuman las consecuencias de sus actos junto con sus propietarios o creadores.
- 6. Tendrán la obligación de pagar impuestos.

Estas seis son ejemplos de posibles normas que podrían imponerse a los robots. Pero, probablemente, no serán las últimas. Una nueva era de convivencia con la robótica ha comenzado, obligándonos a todos a adaptarnos a este nuevo entorno que se nos presenta. Renovarse o morir. No hay más.

Y como otra norma sólo para robots, les informo sobre otra cuarta ley de la que tuve noticia a través de la revista *NOVATICA*, vol. IX, núm. 51/52, en un cuento firmado por Carles Pol, titulado «If...»:

- No importa cuán estúpida sea una máquina: Siempre habrá un ser humano dispuesto a superarla. (Anónimo) En el texto aparece reformulada del modo siguiente «Todo robot, ante el espectáculo de la incompetencia del ser humano, no interferirá en sus acciones, ni actuará para evitarlo, salvo que, con ello, ponga en peligro a otro ser humano».

Sentido común

Según Marvin Minsky (1927-2016), hoy en día los programas pueden ganar a los seres humanos en partidas de ajedrez. Otros saben diagnosticar cierto tipo de enfermedades, otros pueden reconocer imágenes de rostros humanos, montar automóviles en las fábricas, e incluso pilotar barcos y aviones. Pero todavía no hay máquina alguna que sea capaz de hacer una cama, leer un libro o cuidar un niño.

¿Qué hace que nuestros ordenadores sean incapaces de hacer el tipo de cosas que la mayoría de la gente puede llevar a cabo?

¿A caso necesitan más memoria, más velocidad o una mayor complejidad?

¿Utilizan un conjunto de instrucciones que no son del tipo adecuado?

¿Se deben sus limitaciones al hecho de que solo utilizan unos y ceros?

¿O es que a las máquinas les falta algún atributo mágico que solo un cerebro humano puede poseer?

La respuesta es que ninguno de los programas actuales cuenta con el conocimiento basado en el sentido común. Ninguno de los programas actuales está provisto de más conocimiento que el necesario para resolver algunos problemas concretos.

Los programas actuales no tienen objetivos explícitos. Hoy día solo decimos a los programas que hagan determinadas cosas, pero no les contamos por qué queremos que las hagan.

Tampoco tienen una variedad suficiente de recursos. Un programa típico se limita a abandonar la tarea cuando le falta algún conocimiento que necesita, o cuando el método que está utilizando falla, mientras que una persona busca otras vías para continuar.

No solemos reconocer lo intrincados que son los procesos que utilizamos en cada minuto de nuestra vida cotidiana. Muchas de las cosas «de sentido común» que llevamos a cabo son en realidad mucho más complejas que gran número de las técnicas especializadas que llaman más nuestra atención y suscitan un mayor respeto.

¿Qué queremos decir al hablar de sentido común?

Nos referimos a aquello que esperamos que los demás conozcan y consideren obvio. Esto depende de los conocimientos, las creencias y los modos de pensar en los que nos hemos educado. Por lo tanto, siempre puede haber diferencias, por lo que puede ser difícil distinguir entre lo que cada persona sabe y lo que otros consideran como obvios, por lo que puede resultar arduo predecir cómo pensará otra persona.

- Conocimientos basados en el sentido común, es decir, los hechos y los conceptos que la mayoría de nosotros conoce.
- Las técnicas del razonamiento basado en el sentido común, que la gente utiliza para aplicar su saber.

Gran parte del significado de una información se deriva del contexto en que se codifica o decodifica. La comunicación es más fácil si dos seres pensantes comparten un rico contexto común.

Por otra parte, cualquier objeto o idea puede considerarse como algo que tiene múltiples significados. A veces llamamos a estos «ambigüedades» y los consideramos como defectos del modo en que nos expresamos o comunicamos.

El conocimiento se puede clasificar en dos tipos:

- Saber qué.
- Saber cómo.

La primera actuación a gran escala para catalogar el conocimiento basado en el sentido común fue el proyecto CYC de Douglas Lenat puesto en marcha en 1984.

¿Cuánto sabe una persona corriente?

- No demasiadas cosas.

¿Podríamos construir una máquina bebé? Es decir, una máquina que empiece por aprender mediante procedimientos sencillos y desarrolle posteriormente métodos más poderosos, hasta convertirse en inteligente. Normalmente estos experimentos paran su progreso porque no se consigue desarrollar unos procedimientos nuevos, efectivos y de calidad para representar el conocimiento. Y no se pueden aprender cosas que uno no es capaz de representar.

Además, hay que tener en cuenta:

- La paradoja de la optimización: Cuanto mejor funciona un sistema, mayor probabilidad hay de que cualquier cambio lo empeore, por lo tanto, le resulta más difícil encontrar modos de mejorarse a sí mismo.
- El principio de inversión: Cuanto mejor funciona un proceso, mayor tendencia tendremos a confiar en él, y nos sentiremos menos inclinados a desarrollar alternativas nuevas, especialmente si una técnica nueva no va a producir buenos resultados hasta que hayamos adquirido destreza para aplicarla.

- La barrera de la complejidad: Cuanto mayor sea la interacción entre las partes de un sistema, más probable es que cualquier cambio tenga efectos colaterales inesperados.

Todo esto sugiere que a cualquier máquina le resultará difícil seguir evolucionando, salvo que al principio desarrolle modos de protegerse contra aquellos cambios que les ocasionen efectos colaterales negativos. Una manera excelente de conseguir esto, tanto en ingeniería como en biología, ha sido fragmentar todo el sistema en partes que luego puedan evolucionar de un modo más independiente. Esta es seguramente la razón por la que todos los seres vivos evolucionaron para convertirse en ensamblajes de partes separadas (que llamamos órganos), cada una de las cuales tiene comparativamente pocas conexiones con otras partes.

En particular esta podría ser la razón por la que los recursos que se encuentran dentro de nuestros cerebros evolucionaron hasta llegar a estar «organizados» en centros y niveles más o menos separados.

El mejor modo de resolver un problema es conocer de antemano una solución, y esta es la razón por la cual el conocimiento basado en el sentido común resulta útil.

Historia de la inteligencia artificial

La inteligencia general es la capacidad de resolver cualquier problema, en lugar de encontrar una solución a un problema en particular. La inteligencia general artificial (o «AGI») es un programa que puede aplicar la inteligencia a una amplia variedad de problemas, de la misma manera que los humanos.

Ben Goertzel y otros argumentaron a principios de la década de 2000 que la investigación de IA había renunciado en gran medida al objetivo original del campo de crear inteligencia artificial general. La investigación AGI se fundó como un subcampo separado y para 2010 había conferencias académicas, laboratorios y cursos universitarios dedicados a la investigación AGI, así como consorcios privados y nuevas empresas. La inteligencia artificial general también se conoce como «IA fuerte», «IA completa», o inteligencia sintética en oposición a la «IA débil» o «IA estrecha». (Las fuentes académicas reservan «IA fuerte» para referirse a máquinas capaces de experimentar la conciencia).

Además, históricamente han existido dos aproximaciones distintas a la inteligencia artificial:

- La que está basada en las ideas de Wiener, la idea es «utilicemos datos», es conocida como *escuela conexionista*, de aproximación *bottom-up*, originalmente conocida como *scruffy*. Esta aproximación propone que la inteligencia artificial debía inspirarse en la biología, aprendiendo a partir de la observación y de la interacción con el mundo físico, esto es, de la experiencia. Según este enfoque, si aspiramos a crear IA debemos proporcionar a los computadores observaciones de las que aprender. Esto conlleva entrenar algoritmos a partir de miles de ejemplos de lo que queremos que aprendan. En la escuela *bottom-up* surge la percepción computacional que abarca el procesamiento de imágenes, vídeos, texto, audio y datos de todo tipo de sensores; el aprendizaje automático estadístico *machine learning*; el aprendizaje con refuerzo; los métodos de búsqueda, de texto, imágenes, vídeos; los sistemas de agentes; la robótica; el razonamiento con incertidumbre; la interacción humano-IA; los sistemas de recomendación y personalización; y las inteligencias social y emocional computacionales.
- La que está basada en las ideas de McCarthy, utilicemos un enfoque simbólico-lógico, de aproximación *top-down*, originalmente denominada *neat*. Postula que las máquinas, para razonar debían seguir un conjunto de reglas predefinidas y unos principios de la lógica. La idea es programar en la máquina el conocimiento que poseemos los humanos, de forma que después, aplicando las reglas que también han sido enseñadas previamente, el computador pueda derivar conocimiento nuevo. La escuela simbólico-lógica incluye, entre otras, áreas como la teoría de juegos; la lógica; la optimización; el razonamiento y la

representación del conocimiento; la planificación automática; y la teoría del aprendizaje.

Como seres humanos que somos, de siempre ha existido cierto enfrentamiento y competencia por los fondos entre esas dos escuelas de pensamiento, y en ambas un optimismo desproporcionado guiado a ciegas por lo que conocemos hoy en día por las actuales técnicas de *marketing*.

Véase:

<https://ellisalicante.org/book/historia-de-la-inteligencia-artificial?gclid=Cj0KCQjw-daUBhCIARIsALbkjSZV4Wi_EOnShO8nuWb_FBkb814fCslAp_BCaFRD8NqQjcZxErrVZKQaAqAXEALw_wcB>

El nacimiento de la inteligencia artificial, 1952-1956

En las décadas de 1940 y 1950, un puñado de científicos de diversos campos (matemáticas, psicología, ingeniería, economía y ciencias políticas) comenzaron a discutir la posibilidad de crear un cerebro artificial. Las primeras investigaciones sobre las máquinas pensantes se inspiraron en una confluencia de ideas que prevalecieron a finales de los años treinta, cuarenta y principios de los cincuenta. Investigaciones recientes en neurología habían demostrado que el cerebro era una red eléctrica de neuronas que disparaba pulsos de todo o nada. El campo de la investigación en inteligencia artificial se fundó como disciplina académica en 1956.

Los hitos más destacables fueron:

- Cibernética. La cibernética de Norbert Wiener describió el control y la estabilidad en las redes eléctricas. La teoría de la información de Claude Shannon describía las señales digitales (es decir, señales de todo o nada). La teoría de la computación de Alan Turing demostró que cualquier forma de computación podía describirse digitalmente. La estrecha relación entre estas ideas sugirió que podría ser posible construir un cerebro electrónico.

Ejemplos de trabajos en este sentido incluyen robots como las tortugas de W. Gray Walter y la Bestia de Johns Hopkins. Estas máquinas no usaban computadoras, electrónica digital o razonamiento simbólico; estaban controladas completamente por circuitos analógicos.

- Primeras redes neuronales. **Walter Pitts** y **Warren McCulloch** analizaron redes de neuronas artificiales idealizadas y demostraron cómo podrían realizar funciones lógicas simples en 1943.

Uno de los estudiantes inspirados por Pitts y McCulloch fue un joven Marvin Minsk, entonces un estudiante graduado de 24 años. En 1951 (con Dean Edmonds) construyó la primera máquina de redes neuronales, la SNARC.

- Test de Turing. En 1950, Alan Turing publicó un artículo histórico en el que especulaba sobre la posibilidad de crear máquinas que pensarán. Señaló que pensar es difícil de definir e ideó su famoso test de Turing. El test de Turing fue la primera propuesta seria en la filosofía de la inteligencia artificial.
- Surgen máquinas que desafían a los humanos en algunos juegos. En 1951 Christopher Strachey escribió un programa de damas y Dietrich Prinz escribió uno para ajedrez. El programa de damas de Arthur Samuel, desarrollado posteriormente, logró la habilidad suficiente para desafiar a un aficionado respetable.
- El razonamiento simbólico y el teórico lógico. Cuando se hizo posible el acceso a las computadoras digitales, algunos científicos reconocieron instintivamente que una máquina que podía manipular números también podía manipular símbolos y que la manipulación de símbolos bien podía ser la esencia del pensamiento humano. Este fue un nuevo enfoque para crear máquinas pensantes.

En 1955, Allen Newell y Herbert A. Simon, con la ayuda de J. C. Shaw, crearon *the Logic Theorist*, mediante el cual se probaron 38 de los primeros 52 teoremas del *Principia Mathematica* de Russell y Whitehead.

- La celebración del Taller de Dartmouth de 1956 que se considera como el nacimiento de una nueva área del conocimiento humano y el bautismo de su nombre como IA.

La primera época dorada (1956-1974) caracterizada por aproximaciones a los problemas del tipo *up-bottom*. Las razones pueden ser numerosas, pero una de ellas es que, al no haber disponibles grandes cantidades de datos, la aproximación *bottom-up* quedaba prácticamente excluida.

En esta época el dinero se ofreció con pocas condiciones. Por ejemplo, **J. C. R Licklider**, entonces director de **ARPA**, creía que su organización debería financiar personas, no proyectos, y permitió a los investigadores seguir cualquier dirección que pudiera interesarles. Esto creó una atmósfera despreocupada que, por ejemplo, en el MIT dio origen a la cultura *hacker*, pero este enfoque de no intervención no duraría.

Las aproximaciones que surgieron fueron:

- El razonamiento por búsqueda. Muchos de los primeros programas de IA usaban el mismo algoritmo básico. Para lograr algún objetivo (como ganar un juego o probar un teorema), avanzaban paso a paso hacia él (haciendo un movimiento o una deducción) como si buscaran en un laberinto, retrocediendo cada vez que llegaban a un callejón sin salida.

La principal dificultad era que, para muchos problemas, el número de caminos posibles a través del «laberinto» era simplemente astronómico (situación conocida como *explosión combinatoria*). Los investigadores reducirían el espacio de búsqueda mediante el uso de heurísticas o «reglas empíricas» que eliminarían aquellos caminos que probablemente no conducirían a una solución.

Ejemplos de algoritmos de búsqueda serían:

- El *General Problem Solver* (1959) de Newell y Simon.
 - *Geometry Theorem Prover* (1958) de Herbert Gelernter.
 - *Symbolic Automatic Integrator (SAINT)*, escrito por el estudiante de Minsky y James Slagle (1961).
 - Otros programas buscaban a través de objetivos y subobjetivos para planificar acciones, como el sistema *Stanford Research Institute Problem Solver (STRIPS)* desarrollado para controlar el comportamiento de su robot *Shakey*.
- Lenguaje natural. Un objetivo importante de la investigación de la IA es permitir que las computadoras se comuniquen en lenguajes naturales como el inglés.

Ejemplos de algoritmos con este objetivo:

- El programa *ESTUDIANTE* de Daniel G. Bobrow, que podía resolver problemas de álgebra de secundaria expresados verbalmente. Fue el resultado de su tesis doctoral (1964).
 - El primer programa de IA que usó una red semántica fue escrito por Ross Quillian en 1966, y la versión más exitosa fue la de Roger Schank (1969).
 - *ELIZA* (1964-1966) la primera *chatterbot*, de Joseph Weizenbaum.
- *Micromundos*. A finales de los años sesenta, Marvin Minsky y Seymour Papert del Laboratorio de IA del MIT propusieron que la investigación de la IA debería centrarse en situaciones artificialmente simples conocidas como micromundos. Y

Ejemplos de algoritmos con este objetivo:

- Los realizados por Gerald Sussman que trabajó en visión artificial, Adolfo Guzmán y David Waltz que trabajaron en la propagación de restricciones y especialmente, Patrick Winston.
- Minsky y Papert construyeron un brazo robótico que podía apilar bloques, dando vida al mundo de los bloques.
- El logro supremo del programa micro-mundo fue SHRDLU (1968-1970) de Terry Winograd. Podía comunicarse mediante oraciones ordinarias en inglés, planificar operaciones y ejecutarlas.
- Autómatas. En Japón, la Universidad de Waseda inició el proyecto WABOT en 1967, y en 1972 completó el WABOT-1, el primer robot humanoide «inteligente» a gran escala del mundo, también denominado *androide*.

En 1956, después de la convención de Dartmouth, Herbert Simon predijo que «en veinte años, las máquinas serán capaces de hacer el trabajo de una persona». Marvin Minsky, por su parte, declaró en 1970 a la revista *Life* que «dentro de tres a ocho años tendremos una máquina con la inteligencia general de un ser humano». Hasta mediados de los años setenta predominó el optimismo en todo lo relativo a la inteligencia artificial y su impacto.

Fueron los años en que Edward Feigenbaum, uno de los fundadores del Departamento de Informática de la Universidad de Stanford lideró el equipo que construyó el primer sistema experto, implementado en LISP. El nombre de este sistema experto era DENDRAL, y fue fruto del deseo del biólogo molecular Joshua Lederberg, también de Stanford, de disponer de un sistema que facilitara su investigación sobre compuestos químicos en el espacio.

El primer invierno (1974-1980) denominado así por la carencia de fondos.

Durante la primera época dorada (1956-1974) las capacidades de los programas de IA eran limitadas. Los investigadores de IA habían comenzado a toparse con varios límites fundamentales que no podían superarse en la década de 1970, a saber:

- Potencia informática limitada: no había suficiente memoria o velocidad de procesamiento para lograr algo realmente útil. Hans Moravec argumentó en 1976 que las computadoras todavía eran millones de veces demasiado débiles para exhibir inteligencia.
- La intratabilidad y la explosión combinatoria. En 1972, Richard Karp (basado en el teorema de 1971 de Stephen Cook) demostró que hay muchos problemas que probablemente solo pueden resolverse en tiempo exponencial (en el

tamaño de las entradas). Encontrar soluciones óptimas a estos problemas requiere cantidades inimaginables de tiempo de computadora, excepto cuando los problemas son triviales. Es casi seguro que esto significó que muchas de las soluciones de juguete desarrolladas y utilizadas por la IA en esa época probablemente nunca se convertirían en sistemas útiles.

- Conocimiento y razonamiento del sentido común. Muchas aplicaciones importantes de inteligencia artificial como la visión o el lenguaje natural requieren simplemente enormes cantidades de información sobre el mundo. Nadie en 1970 podía construir una base de datos tan grande y nadie sabía cómo un programa podía aprender tanta información.
- La paradoja de Moravec (1979-1980). Demostrar teoremas y resolver problemas de geometría es comparativamente fácil para los computadores, pero una tarea supuestamente simple como reconocer una cara o cruzar una habitación sin chocar con nada es extremadamente difícil.
- El problema del marco o problema de cualificación. Este problema cuestiona cómo los procesos cognitivos determinan qué información, de entre toda la disponible, es relevante dada una tarea determinada. Aunque postulamos una definición posible, especificar de qué trata este problema es una tarea complicada. Los investigadores de IA (como John McCarthy) que utilizaron la lógica descubrieron que no podían representar deducciones ordinarias que involucraban planificación o razonamiento predeterminado sin realizar cambios en la estructura de la lógica misma. Desarrollaron nuevas lógicas (como lógicas no **monótonas** y lógicas modales) para tratar de resolver los problemas.

Como las ambiciosas expectativas creadas durante las dos décadas anteriores no se cumplieron, los gestores de fondos retiraron una gran cantidad de fondos. Además, en 1969 fue publicado el libro *Perceptrons*, de Marvin Minsky y Seymour Papert. Esta obra contribuyó a desinflar aún más el interés por los modelos *bottom-up*, y, en particular, por las redes neuronales. A pesar de las dificultades con la percepción pública de la IA a finales de los setenta, se exploraron nuevas ideas en la programación lógica, el razonamiento de sentido común y muchas otras áreas.

Los hitos más destacables fueron:

- La lógica y el razonamiento simbólico fue introducida en 1959 por John McCarthy en su propuesta «Advice Taker» que consistía en un computador hipotético introducido en su artículo de 1959 «Programs with Common Sense».
- En 1963, J. Alan Robinson propone un método simple para implementar la deducción en las computadoras, el algoritmo de resolución y unificación. Sin embargo, las implementaciones sencillas eran especialmente difíciles: los

programas requerían una cantidad astronómica de pasos para demostrar teoremas simples.

Un enfoque más fructífero de la lógica fue desarrollado en la década de 1970 por Robert Kowalski en la Universidad de Edimburgo, y pronto esto condujo a la colaboración con los investigadores franceses Alain Colmerauer y Philippe Roussel, quienes crearon el exitoso lenguaje de programación lógica Prolog. Este tipo de trabajo brindó una base para los sistemas expertos de Edward Feigenbaum y el trabajo continuo de Allen Newell y Herbert A. Simon que conduciría a Soar y sus teorías unificadas de la cognición.

Los críticos del enfoque lógico señalaron, como lo había hecho Dreyfus, que los seres humanos rara vez usaban la lógica cuando resolvían problemas. Los experimentos de psicólogos como Peter Wason, Eleanor Rosch, Amos Tversky, Daniel Kahneman y otros proporcionaron pruebas para ello. McCarthy respondió que lo que hace la gente es irrelevante. Argumentó que lo que realmente se necesita son máquinas que puedan resolver problemas, no máquinas que piensen como las personas.

Entre los críticos del enfoque de McCarthy estaban sus colegas del MIT, Marvin Minsky, Seymour Papert y Roger Schank, que intentaban resolver problemas de manera «antilógica» como se supone que lo hacen las personas. Desafortunadamente, con esta aproximación surgen los conceptos imprecisos que son difíciles de representar. Gerald Sussman observó que *usar un lenguaje preciso para describir conceptos esencialmente imprecisos no los hace más precisos.*

Para caracterizar esas diferencias Schank describió sus enfoques antilógicos como «scruffy» (desaliñados), en oposición a los paradigmas «neat» (limpios) utilizados por McCarthy, Kowalski, Feigenbaum, Newell y Simon.

En 1974, Marvin Minsky en un artículo seminal «A Framework for Representing Knowledge», llamó a estas estructuras «frames» (marcos), y Schank usó una versión de marcos que llamó «scripts» (guiones).

La siguiente primavera (1980-1987). El interés por la inteligencia artificial, y los fondos disponibles para su desarrollo, empezaron a aumentar de nuevo a principios de los ochenta. Durante esa década llegaron al mercado los primeros sistemas expertos, con un éxito apreciable. En 1985 el gasto en sistemas de IA en las empresas era de miles de millones de dólares.

Un sistema experto es un programa que responde preguntas o resuelve problemas sobre un dominio específico del conocimiento, utilizando reglas lógicas que se derivan del conocimiento de los expertos. Los primeros ejemplos fueron desarrollados por Edward Feigenbaum y sus alumnos. Dendral, iniciado en 1965, identificó compuestos a partir de lecturas de espectrómetro. MYCIN, desarrollado en 1972, diagnosticó enfermedades infecciosas de la sangre. Demostraron la viabilidad del enfoque.

El poder de los sistemas expertos provenía del conocimiento experto que contenían. Eran parte de una nueva dirección en la investigación de IA que había ido ganando terreno a lo largo de los años setenta. «Los investigadores de IA comenzaban a sospechar, de mala gana, porque violaba el canon científico de la parsimonia, que la inteligencia bien podría basarse en la capacidad de usar grandes cantidades de conocimiento diverso de diferentes maneras», escribió Pamela McCorduck. La gran lección de la década de los setenta fue que el comportamiento inteligente dependía en gran medida de tratar con el conocimiento, a veces un conocimiento bastante detallado, de un dominio en el que se encontraba una tarea determinada. Los sistemas basados en el conocimiento y la ingeniería del conocimiento se convirtieron en un foco principal de la investigación de la IA en la década de los ochenta.

Los sistemas expertos se restringieron a sí mismos a un pequeño dominio de conocimiento específico (evitando así el problema del conocimiento de sentido común) y su diseño simple hizo que los programas fueran relativamente fáciles de construir y luego modificar una vez que estaban en su lugar. En general, los programas demostraron ser útiles: algo que IA no había podido lograr hasta este momento.

Los hitos más destacables fueron:

- En 1980, se completó un sistema experto llamado **eXpert CONfigurer (XCON)** desarrollado por John P. McDermott de **Carnegie Mellon University (CMU)** para Digital Equipment Corporation.
- En 1984, nace el proyecto de **Douglas Lenant** denominado **Cyc**, que hoy en día sigue activo en la compañía **Cycorp Inc.**
- Entre 1985 y 1988 surgen los computadores para jugar al ajedrez **HiTech** y **Deep Thought** capaces de derrotar en algunas ocasiones a maestros de ajedrez.
- En 1981, el Ministerio de Industria y Comercio Internacional de Japón inicia el proyecto de la computadora de quinta generación. Para disgusto de los desaliñados, eligieron Prolog como el lenguaje informático principal para el proyecto.

Otros países respondieron con nuevos programas propios. El Reino Unido inició el proyecto Alvey de 350 millones de libras esterlinas. Un consorcio de empresas estadounidenses formó **Microelectronics and Computer Technology Corporation** (o «**MCC**») para financiar proyectos a gran escala en inteligencia artificial y tecnología de la información. DARPA también respondió, fundando

la iniciativa informática estratégica y triplicando su inversión en IA entre 1984 y 1988.

- En 1982, surgen las redes del tipo Hopfield.
- Geoffrey Hinton y David Rumelhart popularizaron un método para entrenar redes neuronales llamado *backpropagation*, publicado por Seppo Linnainmaa (1970) y aplicado a redes neuronales por Paul Werbos (1974).

Estos dos descubrimientos ayudaron a revivir el campo del conexionismo que fue unificado e inspirado por la aparición de *Parallel Distributed Processing* en 1986, una colección de artículos en dos volúmenes editada por **David Everett Rumelhart** y el psicólogo **James McClelland**.

Además, el desarrollo de la integración a muy gran escala (VLSI) de metal-óxido-semiconductor (MOS), en forma de tecnología MOS complementaria (CMOS), permitió el desarrollo de la tecnología práctica de redes neuronales artificiales (ANN) en la década de los ochenta. Una publicación histórica en el campo fue el libro de 1989 *Analog VLSI Implementation of Neuronal Systems* de **Carver A. Mead** y **Mohammed Ismail**.

Las redes neuronales tendrían éxito comercial en la década de los noventa, cuando comenzaron a utilizarse como motores que impulsan programas como el reconocimiento óptico de caracteres y el reconocimiento de voz.

El siguiente invierno (1987-1990). Durante el congreso de 1984 de la Asociación Americana de Inteligencia Artificial, Minsky y Roger Schank alertaron de que el entusiasmo y la inversión en inteligencia artificial conducirían a una nueva decepción. En efecto, en 1987 comenzó el segundo invierno de la inteligencia artificial, que alcanzaría su momento más oscuro en 1990.

La fascinación de la comunidad empresarial por la IA aumentó y disminuyó en la década de los ochenta siguiendo el patrón clásico de una burbuja económica. El colapso se debió a que los proveedores comerciales no desarrollaron una amplia variedad de soluciones viables. Como decenas de empresas fracasaron, la percepción fue que la tecnología no era viable.

Los hitos más destacables fueron:

- El primer indicio de un cambio en el clima fue el colapso repentino del mercado de *hardware* de IA especializado en 1987. Las computadoras de escritorio de Apple e IBM habían ido ganando velocidad y potencia constantemente y en 1987 se volvieron más poderosas que las máquinas Lisp más caras fabricadas por Symbolics y otras. Ya no había una buena razón para comprarlos. Toda una

industria por valor de medio billón de dólares fue demolida de la noche a la mañana.

- Eventualmente, los primeros sistemas expertos exitosos, como XCON, resultaron demasiado costosos de mantener. Eran difíciles de actualizar, no podían aprender, eran «rágiles» (es decir, podían cometer errores grotescos cuando se les daban entradas inusuales) y eran víctimas de los siguientes problemas:
 - **The qualification problem** (el problema de calificación) que se relaciona con la imposibilidad de enumerar todas las condiciones previas requeridas para que una acción del mundo real tenga el efecto deseado.
 - **The frame problem** (problema del marco) o la necesidad de distinguir aquellas propiedades que cambian a lo largo del tiempo contra un fondo de aquellas propiedades que no lo hacen, que, por lo tanto, constituyen un marco. Dicho problema había sido identificado años antes por McCarthy y Hayes en su artículo de 1969 «Some Philosophical Problems from the Standpoint of Artificial Intelligence».

Por ello, los sistemas expertos resultaron útiles, pero solo en unos pocos contextos especiales.

- A fines de la década de los ochenta, la **Strategic Computing Initiative** de **DARPA**, recortó los fondos para la IA de una manera drástica. El nuevo liderazgo había decidido que la IA no era ya «la próxima ola» y como las expectativas habían sido mucho más altas de lo que realmente era posible, dirigió los fondos hacia proyectos que parecían más probables de producir resultados inmediatos.
- Para 1991, tampoco se había cumplido la impresionante lista de metas escritas en 1981 para el Proyecto de Quinta Generación de Japón.
- Más de 300 empresas de IA cerraron, quebraron o fueron adquiridas a fines de 1993, poniendo fin de manera efectiva a la primera ola comercial de IA.
- En 1994, **Harvey P. Newquist**, escritor con un amplio rango de tópicos, y músico, declaró en *The Brain Makers* que «El futuro inmediato de la inteligencia artificial, en su forma comercial, parece depender en parte del éxito continuo de las redes neuronales».

Sin embargo, el campo siguió avanzando a pesar de las críticas. En las décadas de los ochenta y los noventa, muchos científicos cognitivos rechazaron el modelo de procesamiento de símbolos de la mente y argumentaron que el cuerpo era esencial

para el razonamiento, e inició su andadura una teoría llamada *the embodied mind thesis*. Además, numerosos investigadores, incluidos los desarrolladores de robótica Rodney Brooks y Hans Moravec, abogaron por un enfoque completamente nuevo de la inteligencia artificial basado en la robótica. Creían que, para mostrar inteligencia real, una máquina necesita tener un cuerpo: necesita percibir, moverse, sobrevivir y lidiar con el mundo. Argumentaron que estas habilidades sensoriomotrices son esenciales para habilidades de nivel superior como el razonamiento de sentido común y que el razonamiento abstracto era en realidad la habilidad humana menos interesante o importante (véase la paradoja de Moravec). Abogaron por construir inteligencia «de abajo hacia arriba».

El enfoque revivió ideas de la cibernética y la teoría del control que habían sido impopulares desde los años sesenta. Otro precursor fue David Marr, que había llegado al MIT a finales de la década de los setenta tras una exitosa formación en neurociencia teórica para dirigir el grupo de estudio de la visión. Rechazó todos los enfoques simbólicos (tanto la lógica de McCarthy como los marcos de Minsky), argumentando que la IA necesitaba comprender la maquinaria física de la visión de abajo hacia arriba antes de que tuviera lugar cualquier procesamiento simbólico. El trabajo de Marr sería interrumpido por la leucemia en 1980.

Los hitos más destacables fueron:

- En un artículo de 1990, "Los elefantes no juegan al ajedrez", el investigador de robótica Rodney Brooks apuntó directamente a la hipótesis del sistema de símbolos físicos, argumentando que los símbolos no siempre son necesarios, ya que «el mundo es su propio mejor modelo. Es siempre exactamente al día. Siempre tiene todos los detalles que hay que saber. El truco es sentirlo apropiadamente y con la suficiente frecuencia».
- Uno de los hitos más importantes de la estrategia *bottom-up* y, en particular, del conexionismo, fue el uso del algoritmo de *backpropagation* por parte de David Rumelhart, Geoffrey Hinton y Ronald Williams en 1986, que permitió entrenar redes mucho más complejas, es decir, el perceptrón multicapa.
- Otro hito importante que destacar en el avance de las técnicas aplicables a la inteligencia artificial es el trabajo *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference* de Judea Pearl de 1988, cuando incorporó las teorías de la probabilidad y de la decisión. Algunos de los nuevos métodos propuestos incluyen modelos como las redes bayesianas y los modelos ocultos de Markov, así como la teoría de la información, el modelado estocástico y la optimización.
- También se desarrollaron los algoritmos evolutivos, inspirados en conceptos de la evolución biológica como la **reproducción**, las mutaciones, la recombinación de genes y la selección.

La siguiente primavera (1993-2011). El campo de la IA, que ahora tiene más de medio siglo, finalmente logró algunos de sus objetivos más antiguos. Comenzó a usarse con éxito en toda la industria de la tecnología, aunque algo entre bastidores. Parte del éxito se debió al aumento de la potencia de las computadoras y parte se logró centrándose en problemas aislados específicos y persiguiéndolos con los más altos estándares de responsabilidad científica.

Aun así, la reputación de la IA, al menos en el mundo de los negocios, era menos que prístina. Dentro del campo hubo poco acuerdo sobre las razones por las que la IA no cumplió el sueño de la inteligencia a nivel humano que había capturado la imaginación del mundo en la década de los sesenta. Juntos, todos estos factores ayudaron a fragmentar la IA en subcampos competitivos centrados en problemas o enfoques particulares, a veces incluso bajo nuevos nombres que disfrazaban el pedigrí empañado de la inteligencia artificial. En esta época, la AI fue más cautelosa y más exitosa que nunca. Nick Bostrom explica: «Una gran cantidad de IA de vanguardia se ha filtrado en aplicaciones generales, a menudo sin llamarse IA porque una vez que algo se vuelve lo suficientemente útil y común, ya no se etiqueta como IA».

Los hitos más destacables fueron:

- El 11 de mayo de 1997, **Deep Blue** se convirtió en el primer sistema informático de juego de ajedrez en vencer al actual campeón mundial de ajedrez, Garry Kasparov. Previamente, en febrero de 1996 se había jugado otra competición en Filadelfia (Pensilvania) que fue ganada por Kasparov.
- En 2005, un robot de Stanford ganó el **DARPA Grand Challenge** (Gran Desafío de DARPA) al conducir de forma autónoma durante 131 millas a lo largo de un sendero del desierto no ensayado. Dos años más tarde, un equipo de Carnegie Mellon University ganó el **DARPA Urban Challenge** al navegar de forma autónoma 55 millas en un entorno urbano mientras cumplía los peligros del tráfico y todas las leyes del tránsito.
- En febrero de 2011, en el concurso de exhibición **Jeopardy**, el sistema de respuesta a preguntas de IBM, Watson, derrotó a los dos mayores campeones Jeopardy!, Brad Rutter y Ken Jennings, por un margen significativo.

Estos éxitos no se debieron a algún nuevo paradigma revolucionario, sino principalmente a la tediosa aplicación de la habilidad de la ingeniería y al tremendo aumento en la velocidad y la capacidad de la computadora en los años noventa. De hecho, la computadora de Deep Blue era 10 millones de veces más rápida que la Ferranti Mark 1 a la que Christopher Strachey enseñó a jugar al ajedrez en 1951. Este espectacular aumento se justificó por la ley de Moore, que predecía que la

velocidad y la capacidad de memoria de las computadoras se duplicaría cada dos años, como resultado de la fabricación de transistores de metal-óxido-semiconductor (MOS) que se duplicaban cada dos años. El problema de la potencia bruta de la computadora se estaba superando lentamente.

- Surge un nuevo paradigma denominado *agentes inteligentes* que fue ampliamente aceptado durante la década de los noventa. Aunque los investigadores anteriores habían propuesto enfoques modulares del tipo divide y vencerás para la IA, el agente inteligente no alcanzó su forma moderna hasta que **Judea Pearl, Allen Newell, Leslie P. Kaelbling y otros** trajeron conceptos de la teoría de la decisión y la economía en el estudio de la IA. Cuando la definición como economista de un agente racional se casó con la definición informática de un objeto o módulo (Programming Paradigm Object-orientes programming), el paradigma del agente inteligente quedó completado. Por ejemplo, se esperaba que una arquitectura de agente completo (como SOAR de John Laird, Newell, Rosenbloom) algún día permitiría a los investigadores construir sistemas más versátiles e inteligentes a partir de agentes inteligentes que interactúan.
- Los investigadores de IA comenzaron a desarrollar y utilizar herramientas matemáticas sofisticadas más que nunca en el pasado. Hubo una comprensión generalizada de que muchos de los problemas que la IA necesitaba resolver ya estaban siendo trabajados por investigadores en campos como las matemáticas, la ingeniería eléctrica, la economía o la investigación de operaciones. El lenguaje matemático compartido permitió tanto un mayor nivel de colaboración con campos más establecidos y exitosos como el logro de resultados medibles y demostrables; la IA se había convertido en una disciplina científica más rigurosa. Russel y Norvig describen todo esto como nada menos que una revolución, en su conocido libro *Artificial Intelligence: A Modern Approach*, que ha evolucionado en el tiempo en sus diversas ediciones.
- El influyente libro de Judea Pearl de 1988, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference* introdujo la teoría de la probabilidad y la decisión en la IA. Entre las muchas herramientas nuevas en uso estaban las redes bayesianas, los modelos ocultos de Markov, la teoría de la información, el modelado estocástico y la optimización clásica.
- También se desarrollaron descripciones matemáticas precisas para paradigmas de inteligencia computacional como redes neuronales y algoritmos evolutivos.

Deep learning, big data: 2011-presente

En esta época se produce un avance muy significativo en las máquinas de aprendizaje estadístico que pertenecen al enfoque *bottom-up*. Las razones de ese avance hay que buscarlas en el acceso a cantidades ingentes de datos (*Big Data*); la disponibilidad de procesadores muy potentes a bajo coste; y el desarrollo de redes neuronales profundas y complejas, los modelos llamados de *deep learning* impulsaron el progreso y la investigación en el procesamiento de imágenes y vídeos, el análisis de texto e incluso el reconocimiento de voz.

De acuerdo con el teorema de aproximación universal, la profundidad en las redes neuronales no es necesaria para que pueda aproximar funciones continuas arbitrarias. Aun así, hay muchos problemas que son comunes a las redes superficiales (como el sobreajuste) que las redes profundas ayudan a evitar. Como tal, las redes neuronales profundas pueden generar modelos mucho más complejos de manera realista en comparación con sus contrapartes superficiales. Sin embargo, el aprendizaje profundo tiene sus propios problemas. Un problema común para las redes neuronales recurrentes es el *vanishing gradient* problema (problema del gradiente que desaparece), que es donde los gradientes que pasan entre las capas se reducen gradualmente y literalmente desaparecen a medida que se redondean a cero. Se han desarrollado muchos métodos para abordar este problema, como las unidades de memoria a largo plazo.

Las arquitecturas de redes neuronales profundas de última generación a veces pueden incluso rivalizar con la precisión humana en campos como la visión por computadora, específicamente en cosas como la base de datos **MNIST** y el reconocimiento de señales de tráfico.

Los motores de procesamiento de idiomas impulsados por motores de búsqueda inteligentes pueden vencer fácilmente a los humanos al responder preguntas generalistas del tipo trivial (como IBM Watson), y los desarrollos recientes en el aprendizaje profundo han producido resultados sorprendentes al competir con los humanos, en cosas como Go y Doom (que, siendo un juego de disparos en primera persona, ha desatado cierta controversia).

Epílogo

Una clasificación de los alcances realizados por la IA a la hora de resolver problemas, *leitmotiv* de este texto, es que los hay:

- Óptimos: No es posible realizarlos mejor.
- Por encima de todo los humanos: Ya que lo realizan mejor que cualquier humano.
- Por encima de algunos humanos: Ya que lo realizan mejor que algunos humanos.
- Por debajo de los humanos: Lo realizan peor que muchos humanos.

Por supuesto, los hechos nunca dejan de cambiar. En el proceso de escritura de este libro, no he parado de revisar los capítulos que ya había escrito. No me ha sido fácil mantenerme al día. Pero ahora mi trabajo ha finalizado y todo está en sus manos, queridos lectores. En vez de considerar las historias que aquí he presentado como una simple colección de datos, deberían considerarlas como un intento de abrirles a ustedes el apetito y de mantenerlos alerta ante la aparición de nuestras historias relacionadas con la IA. Sin duda se avecinan descubrimientos nuevos, y las teorías que los explicarán surgirán procedentes de muchas fuentes diferentes. Debemos usar nuestro conocimiento para desarrollar estrategias del tipo «todos ganan» que minimicen los impactos de nuestras actividades y al mismo tiempo nos permitan mejorar nuestra calidad de vida.

A principios del siglo XX, los hitos en innovación han afectado profundamente a la calidad de vida, de una manera bastante difícil de medir. Pienso que los futuros avances en I+D+i, permitirán a los seres humanos analizar cada vez mejor el complejo mundo con el que nos hemos rodeado y proporcionarán nuevas dimensiones colaborativas del tándem racionalidad-tecnología, y estos avances serán útiles a la hora de tomar decisiones en cualquier ámbito de la ciencia, la tecnología, la sociedad, el individuo. De esta manera seremos capaces de crear nuevos productos, servicios, compañías, industrias, que deberían servir para hacer más cómoda e interesante la vida del ser humano sobre la Tierra. Lo que significa que otra gran tarea a estudiar es intentar disminuir la importancia en el avance del tándem racionalidad-tecnología del parámetro que actualmente se considera más importante, «el dinero», gran desestabilizador del avance del conocimiento.

La función de la IA como generadora de aspectos del futuro se ha vuelto más complicada, más escurridiza, pero sigue siendo igual de crucial. Los cambios ahora se producen más de prisa y el futuro ya no es dentro de unas décadas, sino de unos años o incluso unos meses.

Un método de comprensión es mirar el futuro más cercano, observando lo que ocurre hoy en día, analizando el choque entre la cultura actual y la tecnología.

Aunque el futuro se abalance hacia nosotros cada vez más rápido, aunque el tiempo entre el ahora y el futuro se mida en minutos, aunque el futuro llegue y nuestros ordenadores tomen conciencia de sí mismos y nos visiten los extraterrestres, siempre contaremos cuentos de ciencia ficción. Porque en la naturaleza humana está el preguntarse qué hay «más allá». Nuestro anhelo por elevar esta cuestión y contestarla nos impulsa como especie a seguir avanzando y aprendiendo, lo cual es la clave para la supervivencia. Por eso siempre nos haremos preguntas. Siempre especularemos porque siempre habrá algo que explorar más allá.

Citando a Fei-fei Li «La IA no tiene nada de especial. Se inspira en personas, es creada por personas, y lo más importante impacta en las personas. Es una herramienta muy poderosa que tan solo hemos comenzado a entender, y esa es una gran responsabilidad». Es decir, hay diferencias fundamentales entre humanos y máquinas, y hay tareas que los ordenadores no deberían hacer, aunque técnicamente pudieran. Sus argumentos se basan en que el medio social y cultural en el que un ser humano crece, vive y trabaja, adquiriendo experiencias son distintas a las que podría llegar a adquirir una máquina.

Resulta paradójico que las computadoras sean capaces de resolver complejos teoremas matemáticos o nos machaquen jugando al ajedrez, pero que, sin embargo, no sean capaces de entender una sencillísima frase. Son endiabladamente inteligentes para muchas cosas, pero carecen de algo que hasta el humano más estúpido tiene en alguna medida: sentido común.

Citando a Searle, las máquinas carecen de intencionalidad. Término que tiene que ver con dar significado a todo lo que nos rodea.

La explicación a esta aparente contradicción hay que buscarla en la dificultad de dotar a las máquinas de conocimientos de sentido común. Sin ese tipo de conocimiento no es posible una comprensión profunda del lenguaje ni una interpretación profunda de lo que capta un cerebro. De hecho, es el requisito fundamental para conseguir IA similar a la humana en cuanto a generalidad y profundidad. Ese tipo de conocimientos son fruto de nuestras vivencias y experiencias interactuando con nuestro entorno, ya que la cognición humana es una cognición situada y corpórea. Las aproximaciones no corpóreas no permiten interacciones directas con el entorno. Las capacidades más complicadas de alcanzar son aquellas que requieren interactuar con entornos no restringidos ni previamente preparados. Diseñar este tipo de sistemas requiere integrar desarrollos de muchas áreas como percepción, representación, razonamiento, acción y aprendizaje. Sin perder de vista que se haga lo que se haga, la capacidad de explicación es una característica irrenunciable en cualquier sistema inteligente. Es necesario, además, realizar una aproximación pluridisciplinar para producir un efecto sinérgico que cambie profundamente la naturaleza de la IA e incluso nuestra comprensión de qué es la inteligencia.

En cualquier caso, por muy inteligentes que lleguen a ser las futuras inteligencias artificiales, incluidas las de tipo general, nunca serán iguales a las inteligencias humanas, pues, tal como hemos argumentado, el desarrollo mental que requiere toda inteligencia compleja depende de las interacciones con el entorno, y estas dependen a su vez del cuerpo, en particular del sistema perceptivo y del sistema motor. Ello, junto al hecho de que las máquinas no seguirán procesos de socialización y culturización como los nuestros, hace que, por muy sofisticadas que lleguen a ser, serán inteligencias distintas a las nuestras. El hecho de ser inteligencias ajenas a la humana y, por lo tanto, ajenas a los valores y necesidades humanas nos debería hacer reflexionar sobre posibles limitaciones éticas al desarrollo de la IA. En particular estamos de acuerdo con Weizenbaum (1976) en que ninguna máquina debería tomar nunca decisiones de forma completamente autónoma o dar consejos que requieran entre otras cosas, de la sabiduría, producto de experiencias humanas, y de los valores humanos, por ejemplo, no deberían permitirse por motivos éticos las armas autónomas, los *bots* que operan en bolsa tomando decisiones de compra y venta en milisegundos y las aplicaciones que atentan contra nuestra privacidad.

Aaron Sloman, 1994: La gente es demasiado impaciente. Desea una definición de tres renglones para la consciencia, y probar en cinco líneas que un sistema informático puede o no puede ser consciente. Además, lo quiere tener hoy mismo. No le apetece hacer el duro trabajo de desenmarañar los conceptos complejos y embrollados que ya conocemos, y explorar nuevas variantes que podrían surgir a partir de ciertas estructuras de los sistemas de comportamiento que están especificadas con precisión.

Para mí la consciencia y la mente son un fenómeno emergente de la materia en relación consigo misma y con el exterior. No hay dos cosas:

- Mente y cuerpo.
- Materia y alma.
- Espíritu y carne.

Solo hay una cosa, y esa cosa es Usted.

La mente humana no siempre es el precioso recipiente que tanto valoramos. Ciertas afecciones cerebrales desbaratan el funcionamiento de la mente, piense por ejemplo en el alzhéimer. Por otro lado, la anestesia para el cerebro y suprime la experiencia de estar consciente.

Como ya he dicho, la emergencia hace referencia a aquellas propiedades o procesos de un sistema no reducibles a las propiedades o procesos de sus partes constituyentes. «El todo es más que la suma de las partes».

Ejemplos de emergencia:

- La digestión emerge del sistema digestivo (proceso no consciente).
- La circulación del sistema circulatorio (proceso no consciente).
- Las hormigas juntas construyen un hormiguero (proceso no consciente).
- ...

Nuestra inteligencia y nuestras experiencias son una consecuencia ineludible de los poderes causales naturales de todo nuestro cuerpo y no de ningún poder sobrenatural.

Nuestra consciencia es el resultado de un procesador neuronal, capacitado por la selección natural para manejar algoritmos combinatorios de razonamiento causal y probabilístico que le han servido al *homo sapiens* para alcanzar los objetivos de comer, reproducirse, sobrevivir, la paternidad, la amistad, el prestigio social...

Vuelta a la ficción

Nosotros somos el (un) medio para que el cosmos se conozca a sí mismo, es decir, adquiera la (auto) consciencia.

La evolución no solo ha ocurrido y eventualmente ha conducido a que haya seres capaces de comprender el proceso, sino que seguirá ocurriendo y quizá se llegue incluso a comprender el proceso por el cual ellos lo comprenden.

A través del largo proceso de la evolución biológica, la mente se ha establecido a sí misma como una fuerza motriz en nuestro pequeño rincón del universo, en el tercer planeta de una humilde estrella de una galaxia que, de otro modo, sería indistinguible. Aquí, en este pequeño planeta, la mente ha infiltrado la materia y ha tomado el control. Me parece que la tendencia de la mente a infiltrar y controlar la materia es una ley de la naturaleza. (Freeman Dyson)

Además, casi todo el espacio y el tiempo se encuentran en el futuro. Tenemos tiempo para avanzar en nuestro conocimiento, si conseguimos ir sobreviviendo para verlo.

Como dijo Samuel Butler, coetáneo de Darwin: «Hay pocas cosas de las cuales la generación actual está más justificadamente orgullosa que los maravillosos avances que a diario ocurren en todo tipo de dispositivos mecánicos... Pero ¿qué pasaría si la tecnología continuara evolucionando?... Nosotros le estamos dando diariamente grandes poderes a las máquinas y suministrando, por todo tipo de ingeniosos dispositivos, ese poder autorregulado y automático que será para ellos lo que el intelecto ha sido para la raza humana».

Sabiendo que el futuro nunca es como se ha imaginado en el pasado, ¿a qué piensan de la factibilidad del futuro de la IA general los científicos?:

- Hay quien piensa que es imposible, aunque sea corporeizada.
- Otros creen que es posible pero no en los próximos cincuenta años.
- Y una minoría piensa que será posible antes de veinte años.

A pesar de todo, yo pienso que hay que asomarse a una de las últimas fronteras, por lo menos curiosear y tener una idea de por dónde va el mundo!

... y ... ¿ahora qué?

Celebremos que ha llegado al final de este documento y el lector tenga una visión de la IA que le permita ir avanzando por el mundo que nos va a ir surgiendo en los próximos años. Muchas gracias por su atención, espero que consciente.

24 de octubre de 2022 (el autor).

Referencias bibliográficas recomendadas para saber mucho más

En inglés:

- *Understanding Intelligence*. Rolf Pfeifer and Christian Scheier. MIT Press 2001.
ISBN 0-262-16181-8.
- *Artificial Intelligence. A modern Approach*. Third Edition. Stuart Russell, Peter Norvig. Pearson, 2010.
ISBN 10:1292153962.
- *Computers and Creativity*. Jon McCormack, Mark d'Inverno editors. Springer, 2012.
ISBN 978-3-642-31726-2.
- *The Soar Cognitive Architecture*. John E. Laird. The Mit Press, 2012.
ISBN 978-0-262-12296-2.
- *The Cambridge handbook of Artificial Intelligence*. Edited by Keith Frankish and William M. Ramsey. Cambridge University Press, 2014.
ISBN 978-0-521-87142-6.
- *Deep Learning Illustrated*. Jon Krohn. Addison-Wesley, 2020.
ISBN 13: 978-0-13-511669-2.

En castellano:

- *El robot enamorado. Una historia de la inteligencia artificial.* Félix Ares. Editorial Ariel, 2008.
ISBN 978-84-344-5369-2.
- *Aspectos básicos de la inteligencia artificial.* J. Mira, A. E. Delgado, J. G. Boticario, F. J. Díez. Editorial Sanz y Torres, 2001
ISBN 84-88667-13-2.

Sobre el autor

El profesor Dr. Francisco José Serón Arbeloa es Dr. en Ciencias Físicas, ha sido profesor titular de Universidad del Área de Matemática Aplicada, posteriormente profesor titular de Universidad del Área de Lenguajes y Sistemas Informáticos y actualmente es catedrático de Universidad del Área de Lenguajes y Sistemas Informáticos, perteneciente al Departamento de Informática e Ingeniería de Sistemas, en la Escuela de Ingeniería y Arquitectura de la Universidad de Zaragoza.

Sus áreas de actividad, en las que ha desarrollado actividades de I+D+i, han sido las de: Modelado de Agentes Inteligentes y Cognitivos, Realidades Mixtas, Informática Gráfica, Modelado 3D de estructuras terrestres, Simulación de fenómenos naturales y Modelado Sísmico, Programación paralela, cálculo numérico y Elementos finitos. En la actualidad realiza actividades relacionadas con la holografía de cara a conseguir ver mundos virtuales con la misma calidad que el mundo real.

Siempre ha estado preocupado por el fortalecimiento de los vínculos entre el sector de la investigación y la industria, tanto a nivel local, nacional e internacional, entre las que destacan: Construcciones y Auxiliar de Ferrocarriles (CAF), General Motors España (GM), Construcciones Aeronáuticas (CASA), TORRECID, IASOFT, INDRA Software, INDALUX, Lledó, RIUSA, DATINZA, SEVASA, Instituto de Robótica (Valencia), Instituto Tecnológico de Aragón (ITA), Sociedad Aragonesa de Tecnologías Aplicadas (SATA), ACESA, MONDO IBERICA, CARITAS, Técnicas Radiofísicas, AUDIOMARKET Multimedia, Grupo de Informática Médica, FOR+Vídeo Comunicación, Industrias Hidráulicas Pardo, Centro de Tecnología del Láser (Valladolid), Grupo Luz y Diseño, Eurosystems Consulting, LSLuz, INCAELEC...

En paralelo ha desempeñado actividades de formación oficial universitaria y de divulgación científica y tecnológica.

La actividad de investigación realizada se refleja en las 24 tesis doctorales que ha dirigido / codirigido en los campos de Ingeniería Informática, Ingeniería Industrial, Ciencias Físicas, Ciencias Matemáticas, Ciencias Químicas y Filosofía y Ciencias de la Educación. Tiene 94 publicaciones en revistas internacionales, 121 en congresos internacionales, y 144 publicaciones nacionales. Ha participado en 19 proyectos como investigador principal y 31 como investigador.

Hasta el momento ha dirigido / codirigido 128 proyectos fin de carrera, 34 proyectos industriales de innovación, 21 producciones audiovisuales y 11 proyectos de imágenes estáticas. Ha participado como responsable principal en 60 proyectos de desarrollo con empresas e instituciones. Ha realizado 12 informes, 5 periciales y dispone de 4 patentes en explotación.

Creo y dirigí el Grupo de Informática Gráfica Avanzada (GIGA) durante sus veinticinco años de existencia, del que han salido: el Laboratorio de Simulación de la

Luz, el *GIGA Affective Lab*, el *Graphics and Imaging Lab* y el *Interactive Systems, Adaptivity, Autonomy and Cognition (ISAAC Lab)*.

Entre las actividades de gestión ha sido subdirector de *Asuntos Generales* de la E. T. S. I. I. de la U. Z., en el periodo 9/2/1987 al 21/4/1993, vicerrector adjunto al rector para las *Tecnologías de la Información y las Comunicaciones* de la U. Z. en el periodo 21/12/2000 al 30/4/2004, vicerrector de *Prospectiva, Sostenibilidad e Infraestructuras* de la U. Z. en el periodo 14/4/2016 al 17/1/2021], y actualmente es delegado del rector para el *Plan de Recuperación, Transformación y Resiliencia* (28/1/2021-...].

C. V. completo: <<http://webdiis.unizar.es/~seron/>>
<<http://cgit.unizar.es/>>



Prensas de la Universidad
Universidad Zaragoza