

Algoritmia para problemas difíciles

Elvira Mayordomo

Universidad de Zaragoza

18 de septiembre de 2017

- Elvira Mayordomo Cámara
Despacho 1.06 elvira@unizar.es
<http://webdiis.unizar.es/~elvira>
Tutorías: Lunes (9-12h), Viernes (9-10h, 12-14h)

Bounded Pushdown dimension vs Lempel Ziv information density. P. Albert, E. Mayordomo, and P. Moser. Lecture Notes in Computer Science (2017)

[...] We then study the relationships between BPD (Bounded PushDown) compression, and the standard Lempel-Ziv (LZ) compression algorithm, and show that in contrast to the finite-state compressor case, LZ is not universal for bounded pushdown compressors in a strong sense: we construct a sequence that LZ fails to compress significantly, but that is compressed by at least a factor 2 by a BPD compressor.

Computing Absolutely Normal Numbers in Nearly Linear Time. Jack H. Lutz, E. Mayordomo. Submitted (2017)

A real number x is *absolutely normal* if, for every base $b \geq 2$, every two equally long strings of digits appear with equal asymptotic frequency in the base- b expansion of x . This paper presents an explicit algorithm that generates the binary expansion of an absolutely normal number x , with the n th bit of x appearing after $n \text{polylog}(n)$ computation steps. (The fastest previous algorithm takes $n^2 \text{polylog}(n)$ steps.) [...]

Machine learning classifier for identification of damaging missense mutations exclusive to human mitochondrial DNA-encoded polypeptides. A. Martín-Navarro, A. Gaudioso-Simón, J. Álvarez-Jarreta, J. Montoya, E. Mayordomo, and E. Ruiz-Pesini. BMC Bioinformatics (2017)

BACKGROUND: Several methods have been developed to predict the pathogenicity of missense mutations but none has been specifically designed for classification of variants in mtDNA-encoded polypeptides. [...] Because mtDNA sequencing of patients suffering mitochondrial diseases is revealing many missense mutations, it is needed to prioritize candidate substitutions for further confirmation. [...]

RESULTS: We have developed a SVM classifier (Mitoclass.1) specific for mtDNA missense variants. [...] Our classifier has performed better than other web-available tested predictors. [...] Our method has an improved sensitivity and better specificity in relation to PolyPhen-2. [...]

¿Por qué estudiar algoritmia?

- El mejor algoritmo es mucho mejor que el mejor computador (Rico ignorante vs pobre experto en algoritmos)
- Los algoritmos son una poderosa lente para ver la informática:
 - ▶ Los grandes cambios en los estándares de Internet se pueden ver como debates sobre deficiencias de un algoritmo de camino más corto y ventajas relativas de otro.
 - ▶ Los grandes avances de empresas líder como google son algorítmicos (ranking, maps, self-driving car ...)
- No es exclusivo para informáticos:
 - ▶ Muchas definiciones novedosas en biología molecular/genética son algorítmicas: similitudes entre genes o genomas
 - ▶ En economía: la viabilidad de subastas combinatorias está ligada a la intratabilidad de problemas relacionados ...
- Es divertido ...
- Las empresas buscan estudiantes que sepan algorítmica

No es exclusivo para informáticos: Elecciones

- ¿Cuál es el **mejor sistema** para elegir la opción más popular?
 - ▶ pluralidad, múltiples rondas, opciones 2 a 2, ...
- ¿cómo evitamos la **manipulación** o los sobornos?
 - ▶ manipular: mentir en las preferencias para conseguir un resultado más deseable
- ¿cómo conseguimos propiedades como unanimidad o independencia de opciones irrelevantes?
 - ▶ unanimidad: si todo el mundo prefiere A a B entonces en el resultado A debe estar por delante de B
 - ▶ independencia de opciones irrelevantes: si las opciones ganadoras son A y B, cualquier cambio de opinión en el resto de las opciones no afecta al resultado
- Las soluciones son todas **algorítmicas**
 - ▶ algoritmos eficientes para sistemas restringidos con buen comportamiento
 - ▶ intratable saber cómo de cerca estamos de uno de esos sistemas restringidos
- *The Handbook of Computational Social Choice*, Brandt, Conitzer, Endriss, Lang and Procaccia. Cambridge University Press 2016

- En *Algoritmia Básica* hemos estudiado las principales técnicas para diseñar algoritmos
- ¿Qué otras técnicas quedan?
- ¿Qué problemas difíciles?

- Algunos ejemplos de problemas tratados en Algoritmia Básica:
 - ▶ Coloreado de grafos
 - ▶ El viajante de comercio
 - ▶ Ciclos hamiltonianos
 - ▶ Mochila 0-1
 - ▶ ...
- ¿Qué tienen en común estos problemas?
- Los intentos que hicimos (algoritmo voraz, programación dinámica, búsqueda con retroceso, ...) no eran satisfactorios
- O bien los algoritmos **no eran eficientes** o bien encontrar una solución (o una solución óptima) **no estaba garantizado**

- En esta asignatura trataremos sobre todo problemas:
 - ▶ NP-completos
 - ▶ NP-difíciles
- Son problemas que se consideran intratables porque nadie ha conseguido encontrar algoritmos **eficientes** y que **resuelvan completamente el problema**
- ¿Qué podemos hacer con uno de estos problemas?
- Primero convencernos de que hay pruebas claras para considerarlo **intratable**
- Segundo utilizar técnicas **imaginativas** que nos acerquen a la solución eficiente (aunque no lo consigan completamente)

- **Algoritmos aproximados:** Nos dan una solución cercana al óptimo con una estimación del error cometido
- **Heurísticas:** Son ideas felices que funcionan bien en algunos casos, sin garantías
- **Algoritmos probabilistas:** Algoritmos que resuelven el problema pero necesitan utilizar el azar
- **Análisis amortizado:** Que el algoritmo sea eficiente cuando lo utilizamos masivamente, no para cada entrada

Ideas felices versus garantías

- ¿Qué diferencia hay entre una heurística y un algoritmo aproximado?
- Es la misma diferencia que entre una cuenta corriente y una cartera de acciones
- Una **heurística** es como en una cartera de acciones, puedes ganar mucho dinero pero no hay garantías
- Un **algoritmo aproximado** es como una cuenta corriente, el interés que consigues se conoce a priori
- ¿Qué es mejor? Depende del objetivo, del riesgo ...
- **Según otras opiniones, una heurística deja de serlo cuando se convierte en un algoritmo eficiente (conseguimos análisis de eficiencia) o en un algoritmo aproximado (conseguimos análisis de prestaciones)**

- 1 Introducción. Los problemas NP-difíciles.
- 2 Algoritmos aproximados. Concepto. Diseño de algoritmos. Garantías y límites.
- 3 Heurísticas. Simulated annealing (templado simulado).
- 4 Algoritmos genéticos.
- 5 Algoritmos probabilistas: Las Vegas y Montecarlo. Análisis. Generadores pseudoaleatorios.
- 6 Análisis amortizado. Estructuras de datos avanzadas.

- S.S. Skiena. The algorithm Design Manual, Springer, 1997
- J. Hromkovic. Algorithms for Hard Problems, Springer, 2001 (*En la biblioteca*)
- T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein. Introduction to Algorithms (3rd edition), The MIT Press, 2009 (*En la biblioteca*)
- G. Brassard, P. Bratley. Fundamentos de Algoritmia, Prentice Hall, 1997 (*En la biblioteca*)

- J. Kleinberg, E. Tardos. Algorithm Design, Addison-Wesley, 2005 (*En la biblioteca*)
- S. Dasgupta, C. Papadimitriou, U. Vazirani. Algorithms, McGraw-Hill, 2008 (*En la biblioteca*)
- V. V. Vazirani. Approximation Algorithms, Springer, 2001 *En la biblioteca y versión electrónica*

- Información en <http://webdiis.unizar.es/asignaturas/APD>
- Twitter [@APDunizar](https://twitter.com/APDunizar) <https://twitter.com/APDunizar>
- Apuntes, ejercicios y guiones de prácticas

Evaluación: opción recomendada

- Prácticas de laboratorio (en parejas) **durante el cuatrimestre** (2 prácticas): 20 %
- Ejercicios de trabajo individual **durante el cuatrimestre** (3 ejercicios): 20 %

***Importante:** Hay que mandar un mail **antes** al profesor para pedir la reserva de un ejercicio*

- Prueba escrita intermedia: 20 %
- Examen final (realizar sólo una parte): 40 %

Evaluación exclusivamente por exámenes finales

- Examen práctico: 20 %
- Examen final completo: 80 %

Fecha de la prueba intermedia

- Viernes 10 de noviembre

Fechas de los exámenes finales

- Lunes 29 de enero (tarde) ????????????
- Miércoles 5 de septiembre (mañana)

Nota:

- El lunes 30 de octubre tiene horario de miércoles.
- El martes 5 de diciembre tiene horario de viernes.

Me gustaría cambiar y suspender las clases de los días:

- lunes 8 de enero
- viernes 12 de enero

Las tutorías esos dos días serán electrónicas.

Mi propuesta es trasladar alguna de ellas a un martes de octubre de 12 a 14h.

- martes 17 de octubre de 12 a 14h.
- martes 24 de octubre de 12 a 13h.