RecTour 2019

ACM RecSys Workshop on Recommenders in Tourism

RECSYS 2019

Re-CoSKQ: Towards POIs Recommendation Using Collective Spatial Keyword Queries

Ramón Hermoso, **Sergio Ilarri**, Raquel Trillo-Lado

University of Zaragoza, I3A

Introduction and goals

- ❑ Interest of recommender systems in mobile computing scenarios
- ❑ The location is a key spatial attribute:
  - Can techniques from the field of spatial databases help?

→ Explore the potential use of Collective Spatial Keyword Querying (CoSKQ)

Proposal: Re-CoSKQ for the recommendation of POIs

- ❑ Semantic coverage of the query keywords (no exact match req.)
- ❑ Minimize the cost:
  - Distance to get to the POIs
  - Similarity between the query and the descriptions of items

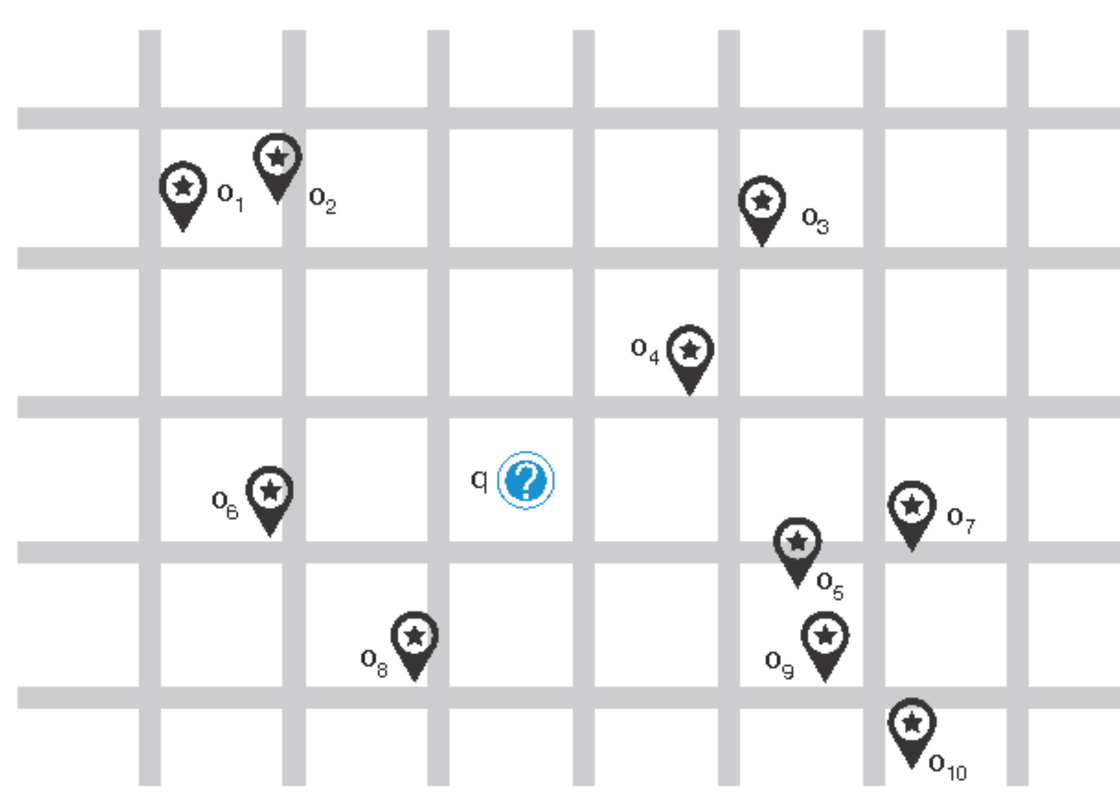
$U = \{u_1, \ldots, u_n\}$ → users

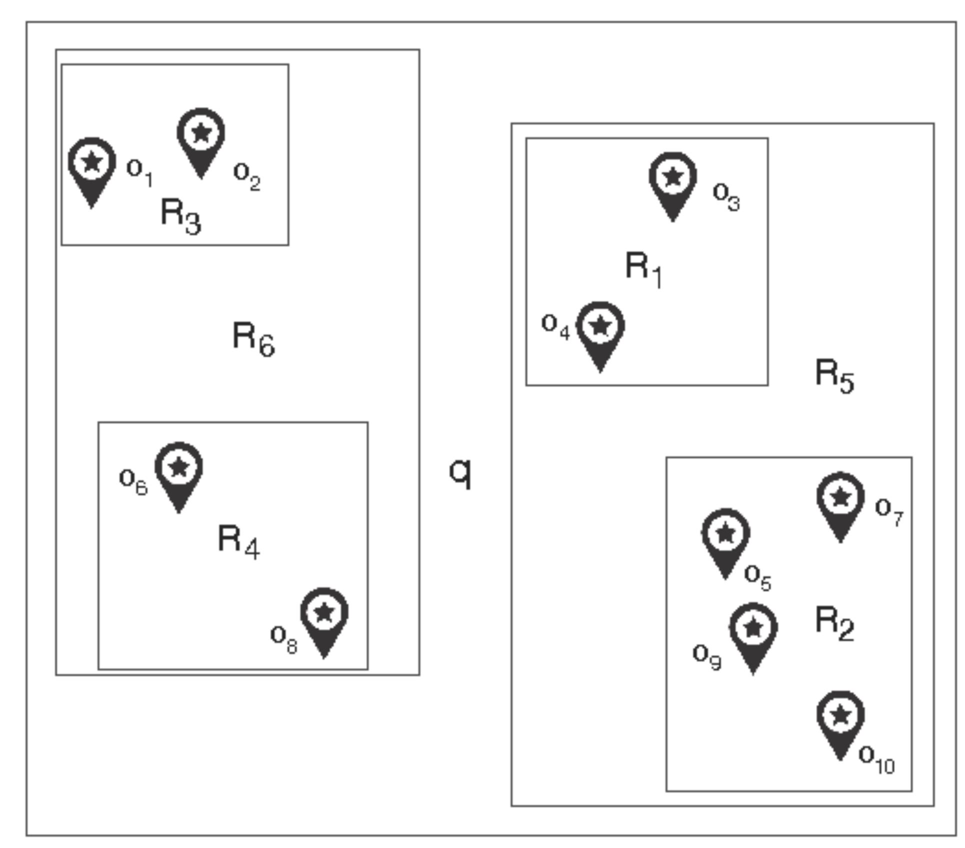
$O = \{o_1, \ldots, o_m\}$ → POIs

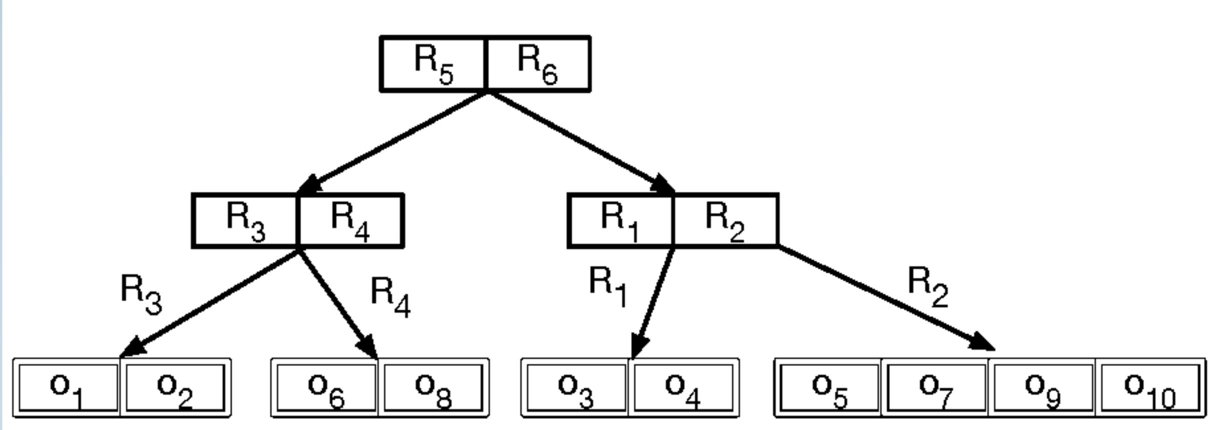
$o_i.\kappa = \{k_1, \ldots, k_j\}$ → keywords describing POI $o_i \in O$

Collective Spatial Keyword Querying (CoSKQ)

- ❑ Technique from the spatial databases field
- ❑ Goal: retrieve a group of spatial objects that collectively match the user preferences given:
  - Specific locations (of the user and also of the objects)
  - A set of keywords
- ❑ Use of IR-tree data structures (balanced trees that allow indexing objects and keywords)







Internal node:
- pointers to the child nodes
- a Minimum Bounding Rectangle (MBR) covering its subtree
- the set of all keywords in the subtree

Leaf node:
- items o (POI objects) in the node
- a bounding rectangle for each o
- a pointer to an inverted file with the keywords that describe each POI

Examples of distance functions

- ❑ Location distance:
  - Euclidean
  - L1-Norm / Manhattan
  - Geodesic distance (shortest path)
- ❑ Term distance:
  - Similarity based on concept closeness (relatedness)

$$sim(k_1, k2) = 1 - \frac{sp(k_1, k_2)}{2D}$$

  - Similarity based on closeness and concept depth

$$sim(k_1, k_2) = \begin{cases} e^{-\alpha l} \frac{e^{\beta h} - e^{-\beta h}}{e^{\beta h} + e^{-\beta h}} & if \quad k_1 \neq k_2 \\ 1 & otherwise \end{cases}$$

- l: shortest path
- d:depth of the least common subsumer
- α, β > 0: weights

Examples of cost functions

$$cost(q, \mathbb{O}') = \alpha \cdot \max_{o \in \mathbb{O}'} [dist(q.\lambda, o.\lambda)] + \beta \cdot \max_{o_1, o_2 \in \mathbb{O}'} [dist(o_1, o_2)] + \omega \cdot \max_{k_1 \in q.\kappa, k_2 \in \cup_{o \in \mathbb{O}'} o.\kappa} [dist(k_1, k_2)]$$

← TYPE 1 – COMB

$$cost(q, \mathbb{O}') = \max \left\{ \alpha \cdot \max_{o \in \mathbb{O}'} [dist(q.\lambda, o.\lambda)], \beta \cdot \max_{o_1, o_2 \in \mathbb{O}'} [dist(o_1, o_2)], \omega \cdot \max_{k_1 \in q.\kappa, k_2 \in \cup_{o \in \mathbb{O}'} o.\kappa} [dist(k_1, k_2)] \right\}$$

← TYPE 2 – MAX

$$cost(q, \mathbb{O}') = \alpha \cdot \min_{o \in \mathbb{O}'} [dist(q.\lambda, o.\lambda)] + \beta \cdot \max_{o_1, o_2 \in \mathbb{O}'} [dist(o_1, o_2)] + \omega \cdot \max_{k_1 \in q.\kappa, k_2 \in \cup_{o \in \mathbb{O}'} o.\kappa} [dist(k_1, k_2)]$$

← TYPE 3 – MIN-MAX

$$cost(q, \mathbb{O}') = \left[\left(\alpha \cdot \left(\sum_{o \in \mathbb{O}'} (dist(q.\lambda, o.\lambda))^{\phi_1} \right)^{\frac{1}{\phi_1}} \right)^{\phi_2} + \left(\beta \cdot \max_{o_1, o_2 \in \mathbb{O}'} dist(o_1, o_2) \right)^{\phi_2} + \left(\omega \cdot \max_{k_1 \in q.\kappa, k_2 \in \cup_{o \in \mathbb{O}'} o.\kappa} dist(k_1, k_2) \right)^{\phi_2} \right]^{\frac{1}{\phi_2}}$$

← TYPE 4 – UNIFIED COST FUNCTION

Evaluation proposal

- Define a representative set of queries
- Annotate a dataset of POIs with predefined categories based on the keywords → ground truth → precision, recall, … + performance and tuning
- Also interesting: user-centered evaluation, DataGenCARS

Acknowledgements

- Government of Aragon (Group Reference T35_17D, COSMOS group) and co-funded with Feder 2014-2020 "Construyendo Europa desde Aragón".
- Project TIN2016-78011-C4-3-R (AEI/FEDER, UE).

Contact

- Dr. Sergio Ilarri (silarri@unizar.es)
- Dr. Ramón Hermoso (rhermoso@unizar.es)
- Dra. Raquel Trillo-Lado (raquelt@unizar.es)

Universidad Zaragoza

COS²MOS COMPLEXITY MANAGEMENT