K-Nearest Neighbor Temporal Aggregate Queries

Yu Sun [†] Jianzhong Qi [†] Yu Zheng [‡] Rui Zhang [†]

[†]Department of Computing and Information Systems
University of Melbourne

[‡]Microsoft Research, Beijing

March 26th 2015

Outline

Examples in Our Daily Life

- Find some walking-distance attractions
- Find a nearby club gathering lots of people now
- Find a good restaurant not far away and has few customers now

Answering Such Questions Has Wide Applications

- Foursquare or Facebook: places nearby
- Flickr or Instagram: photos taken nearby having many *Likes*
- Urban computing

No Existing Queries Can Effectively Answer Such Questions

- Ranking locations on
  - Spatial distance
  - Temporal aggregate on *visits* or *likes*
- Range aggregate does not work

No Existing Algorithms or Indexes Can Efficiently Support Such Rankings

- Characteristics of such applications
  - Visits or likes arrive continuously
  - Interested periods range from hours to years
  - Explore results of different preferences

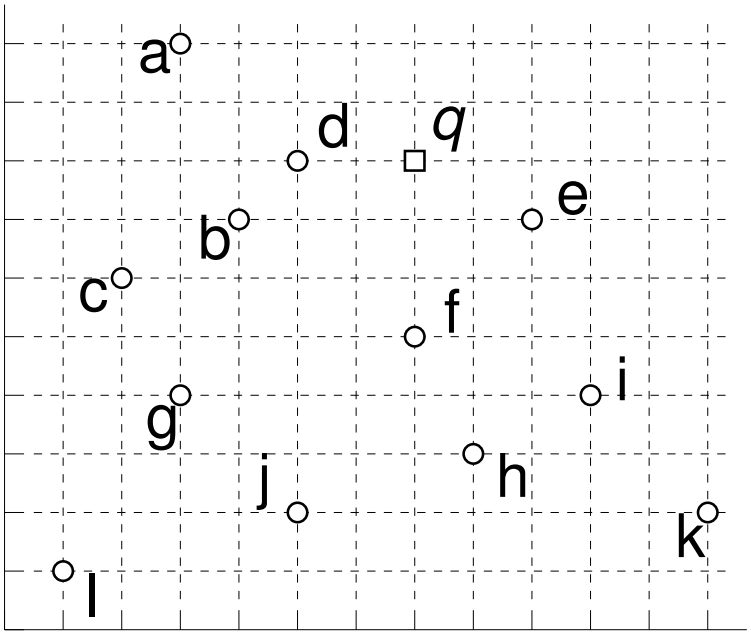
A Weighted Sum of the Spatial Distance and Temporal Aggregate

- The ranking function

$$f(p) = \alpha d(p, q) + (1 - \alpha)(1 - g(p, \mathcal{I}_q))$$

- K-Nearest Neighbor Temporal Aggregate Queries (kNNTA): returns the k locations with the minimum ranking scores

A Query Example



	$t_0\rightarrow$	$t_1\rightarrow$	$t_2\rightarrow$
a	1	1	0
		...	
e	1	1	0
f	3	5	4
		...	
l	1	0	1

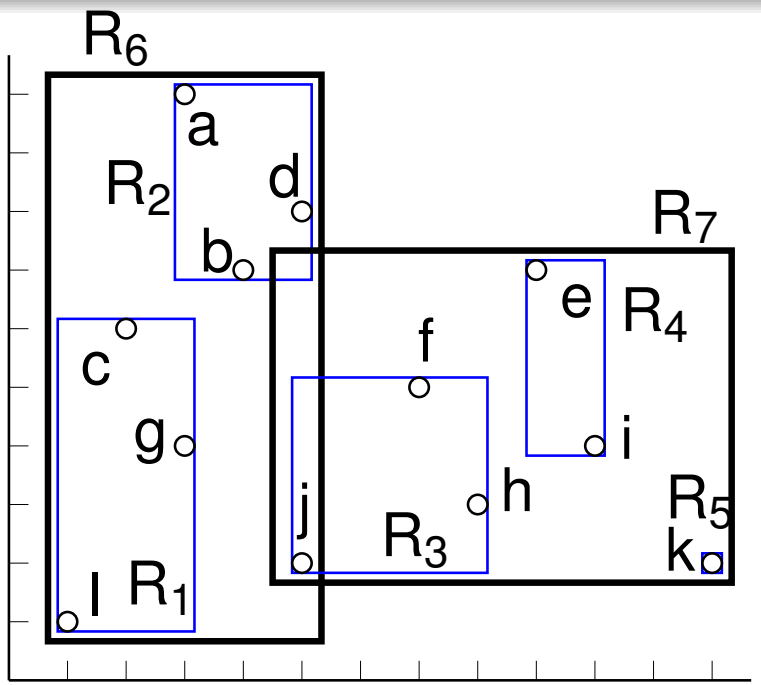
- Query: q, $[t_0, \text{now}]$, $\alpha = 0.3$, $k = 1$

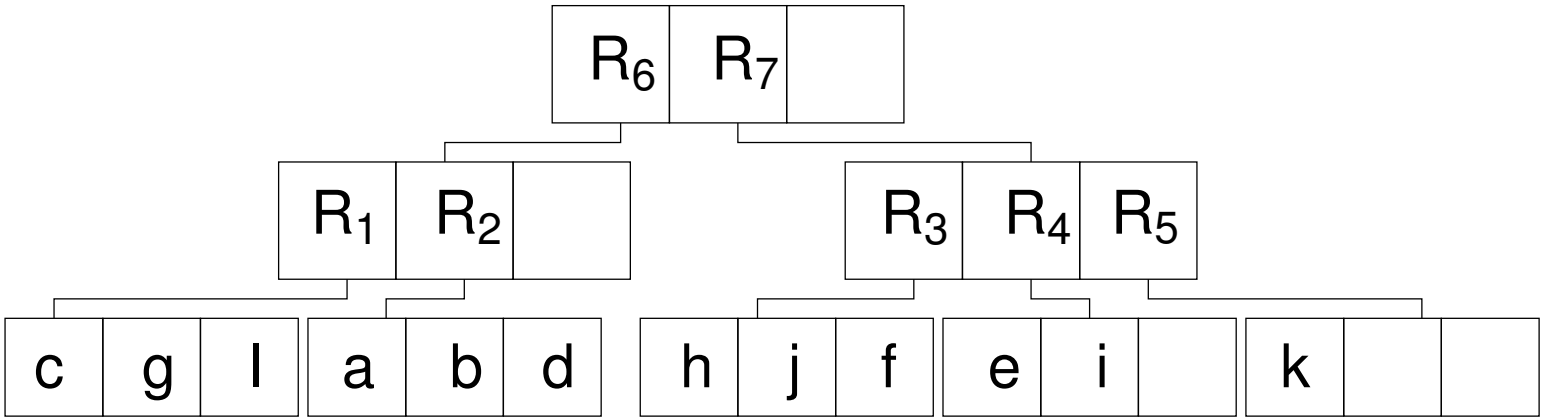
- $f(\text{e}) = 0.3 \cdot \frac{2.24}{15.6} + (1 - 0.3) \cdot (1 - \frac{2}{12}) = 0.626$
- $f(\text{f}) = 0.3 \cdot \frac{3}{15.6} + (1 - 0.3) \cdot (1 - \frac{12}{12}) = 0.058$

A Straightforward Approach May Encounter Very High Cost

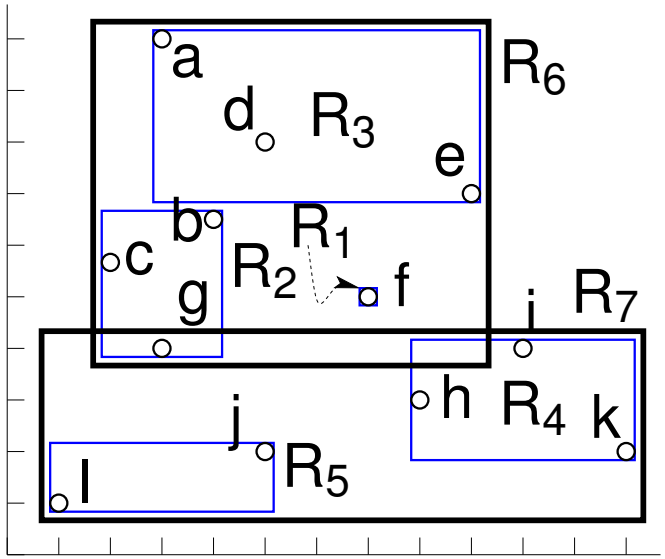
- Number of locations in Foursquare is 60 million
- Number of records for each location is $525,600$
- Much more for applications like Instagram or Twitter

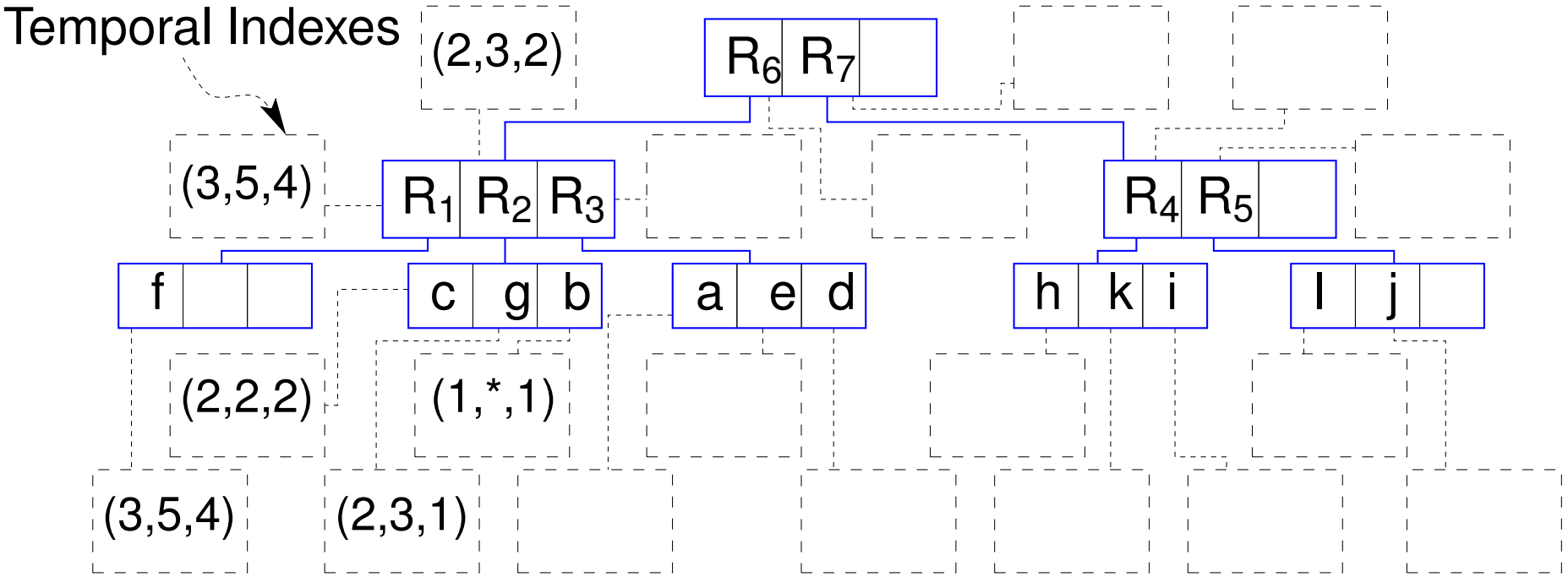
A Brief Reminder of R-tree





Basic Structure of TAR-tree





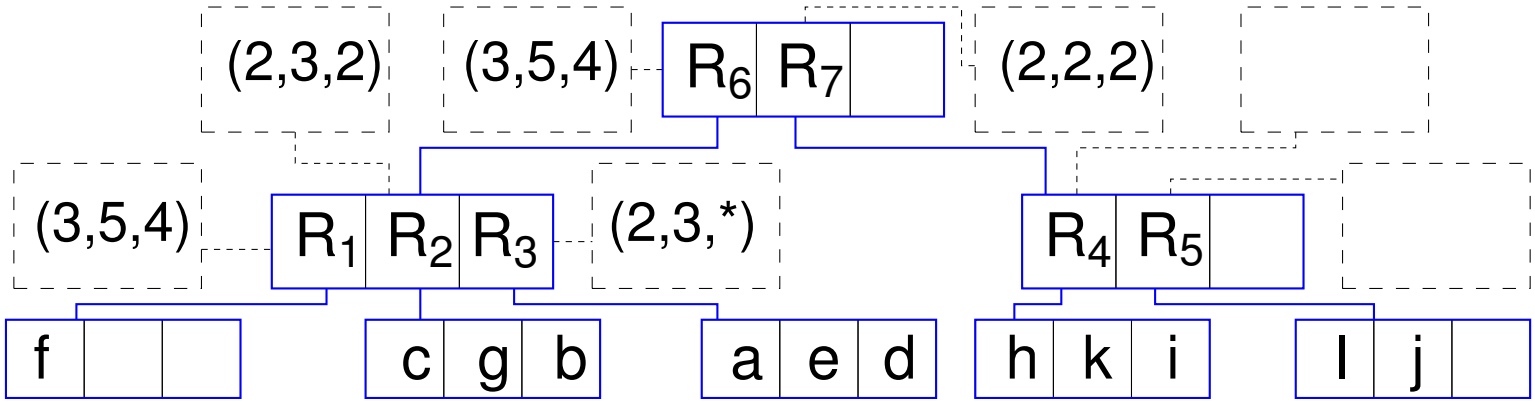
kNNTA Query Processing using TAR-tree

Best-First Search:

R_6 (0.000)	R_7 (0.427)

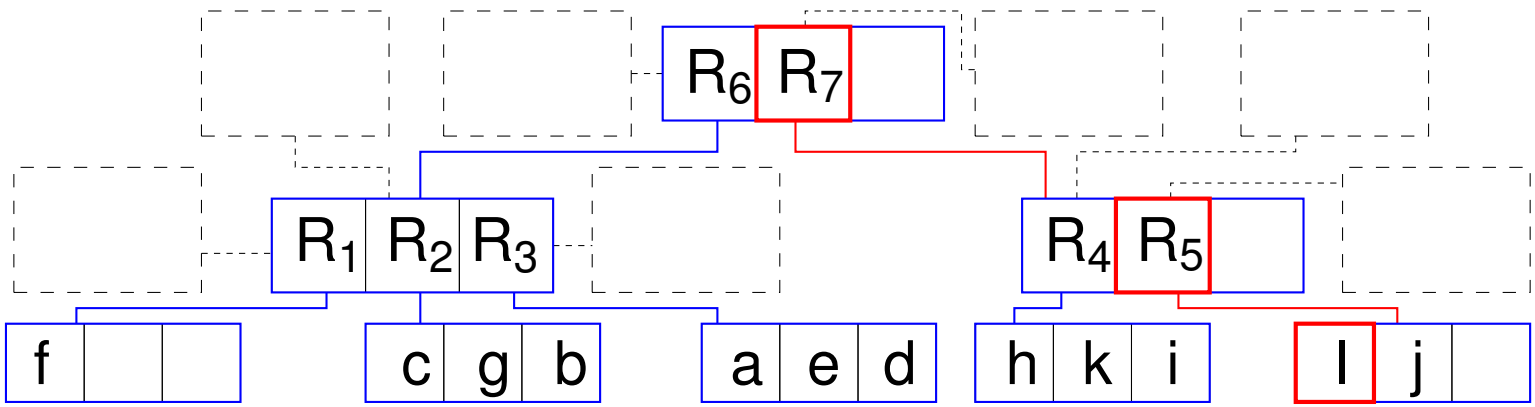
R_1 (0.058)	R_2 (0.352)	R_3 (0.408)	R_7 (0.427)

f (0.058)	R_2 (0.352)	R_3 (0.408)	R_7 (0.427)



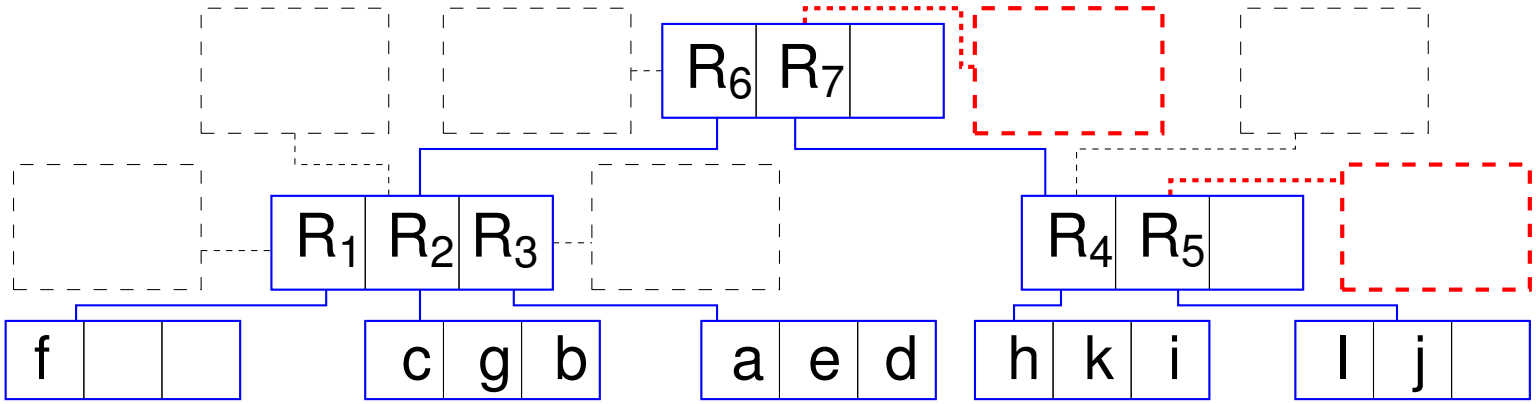
Maintenance of TAR-tree

- Insert *Visits* or *Likes*
- Insert location
- Re-Insert
- Note split

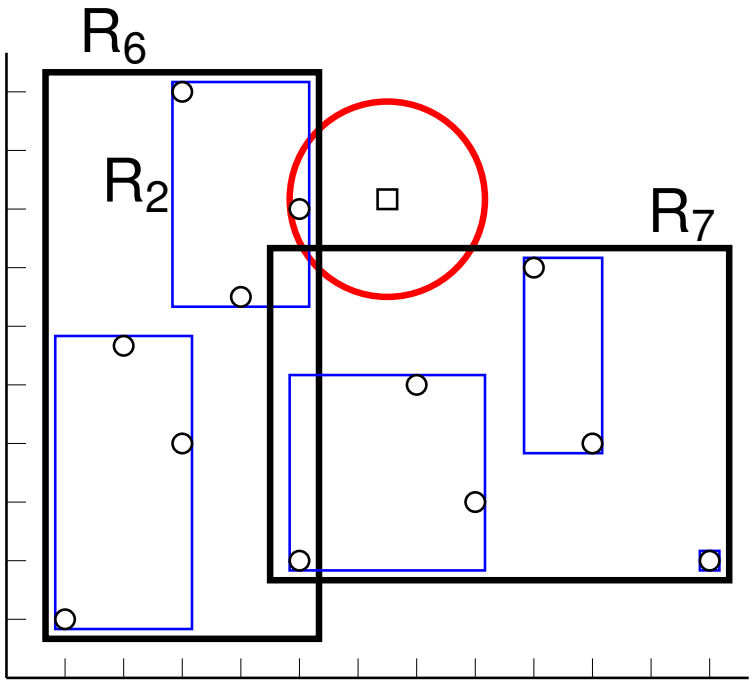


Maintenance of TAR-tree

- Insert *Visits* or *Likes*
- Insert location
- Re-Insert
- Note split



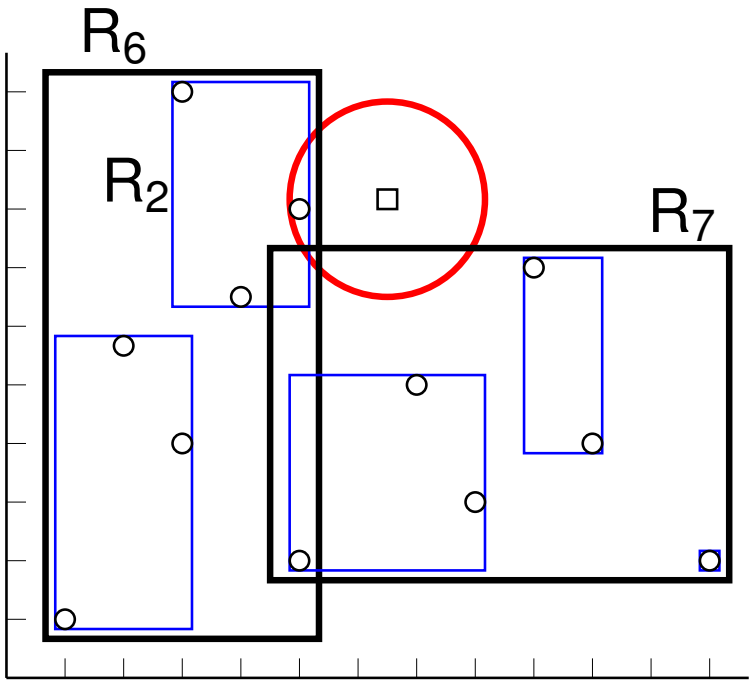
The Importance of Entry Grouping Strategy



Root, R_6, R_7, R_2

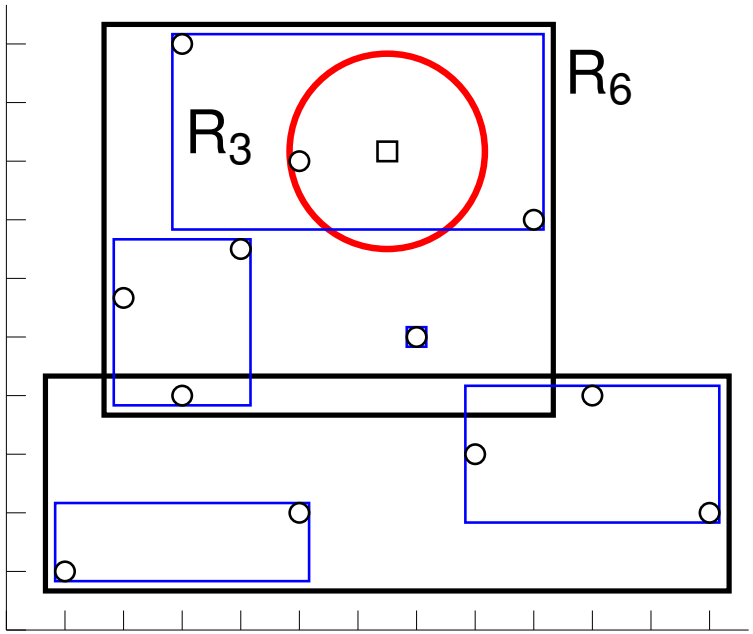
Four node accesses

The Importance of Entry Grouping Strategy



Root, R_6, R_7, R_2

Four node accesses



Root, R_6, R_3

Three node accesses

Properly Integrating the Spatial Distance and Temporal Aggregate

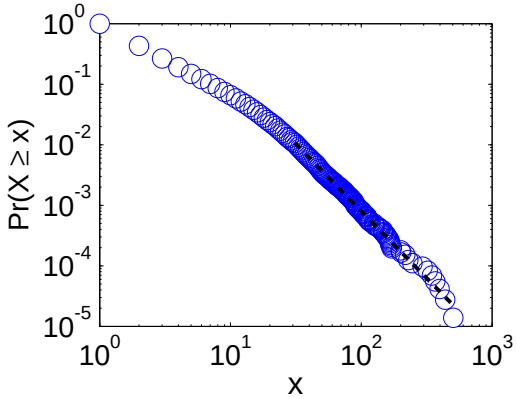
- Straightforward one: Use the spatial information
- Straightforward two: Use the aggregate distribution
  - $(1, 0, 1)$ with $(1, 1, 0)$
  - $(1, 0, 1)$ not with $(10, 8, 9)$
- Propose: 3D MBR in R*-tree
  - two spatial dimensions
  - third is the aggregate dimension
- Coordinate of the aggregate dimension

$$\widehat{\lambda}_p = \frac{1}{m} \sum_{i=1}^{m} v_i$$

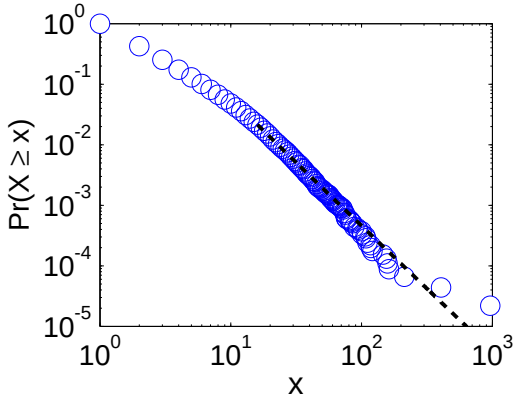
- Example: $(1, 0, 1) \rightarrow 0.67$ and $(10, 2, 9, 8) \rightarrow 7.25$

Power-law Distribution of the Aggregate Data

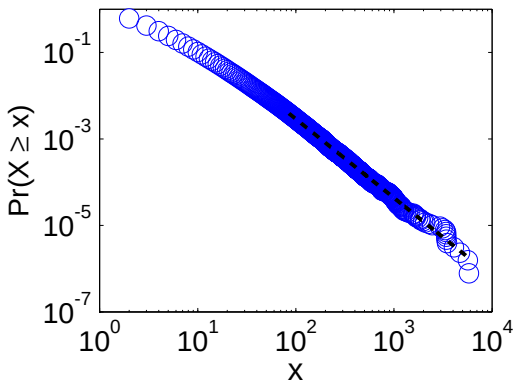
Roughly 80% of the visits are at 20% of the locations



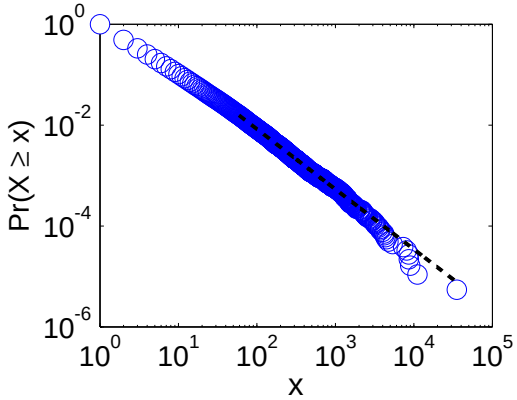
(a) NYC



(b) LA

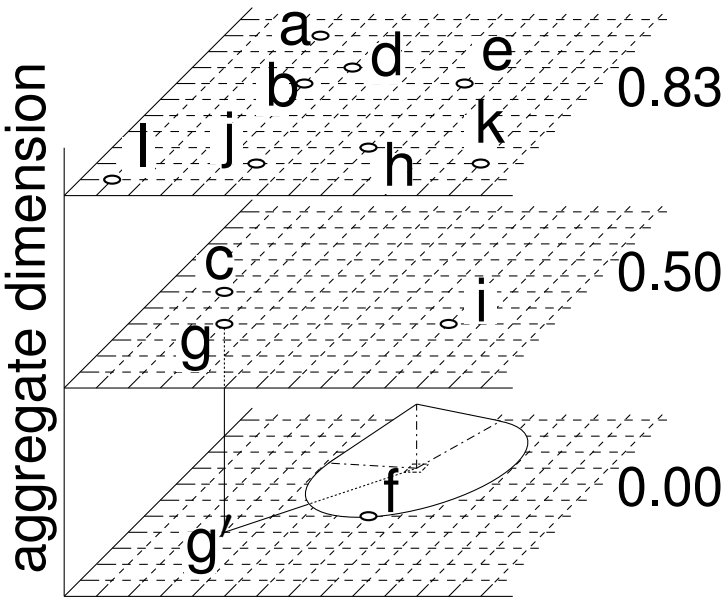


(c) GW



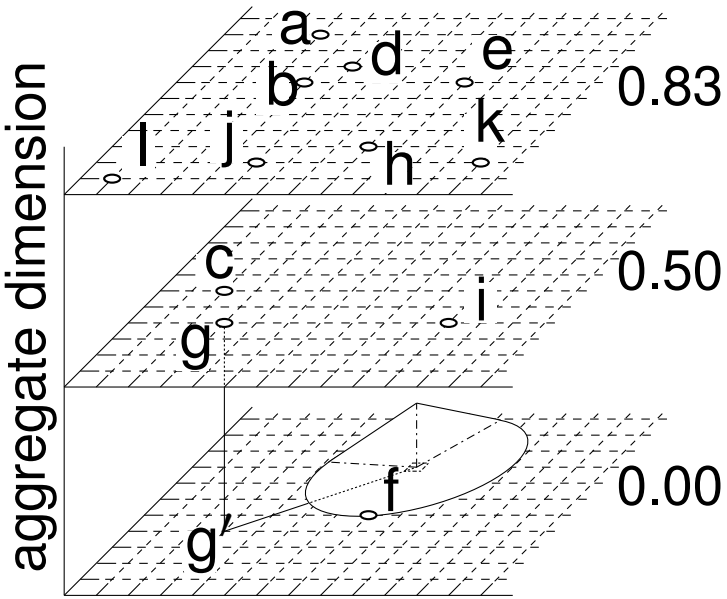
(d) GS

Estimation of the Query Search Region and Number of Node Accesses

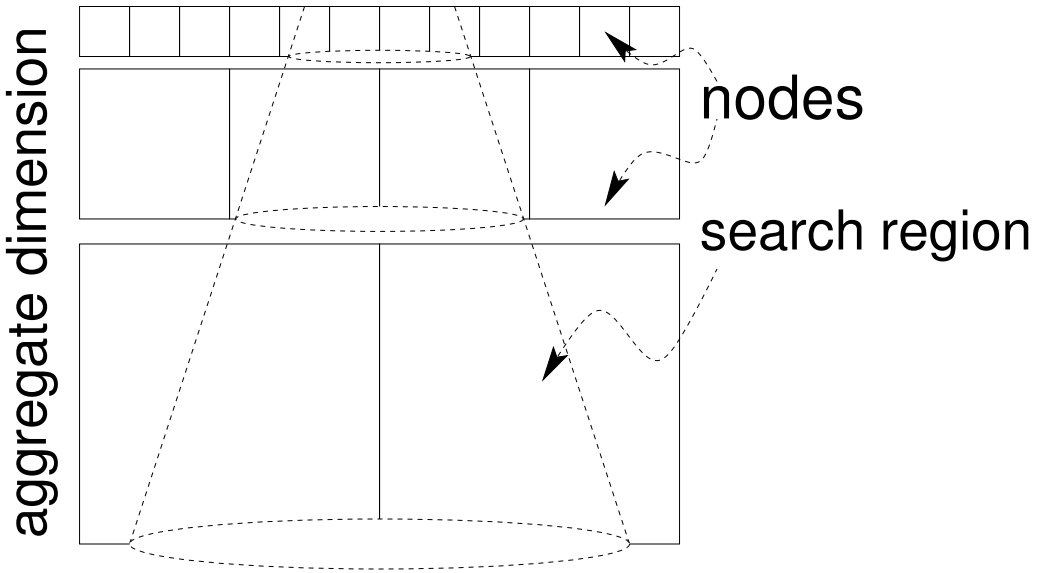


Conic shape search region

Estimation of the Query Search Region and Number of Node Accesses

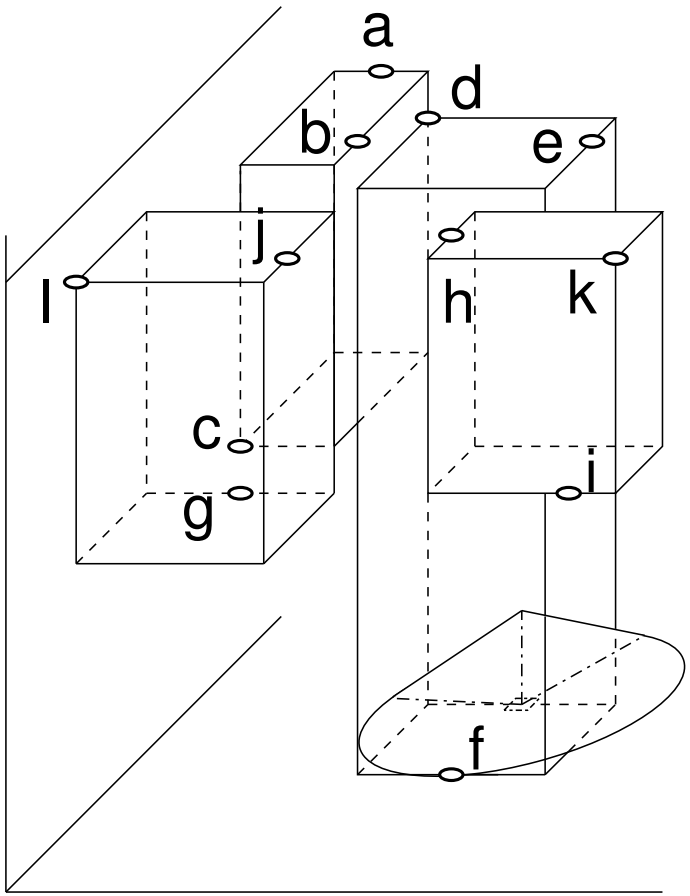


Conic shape search region



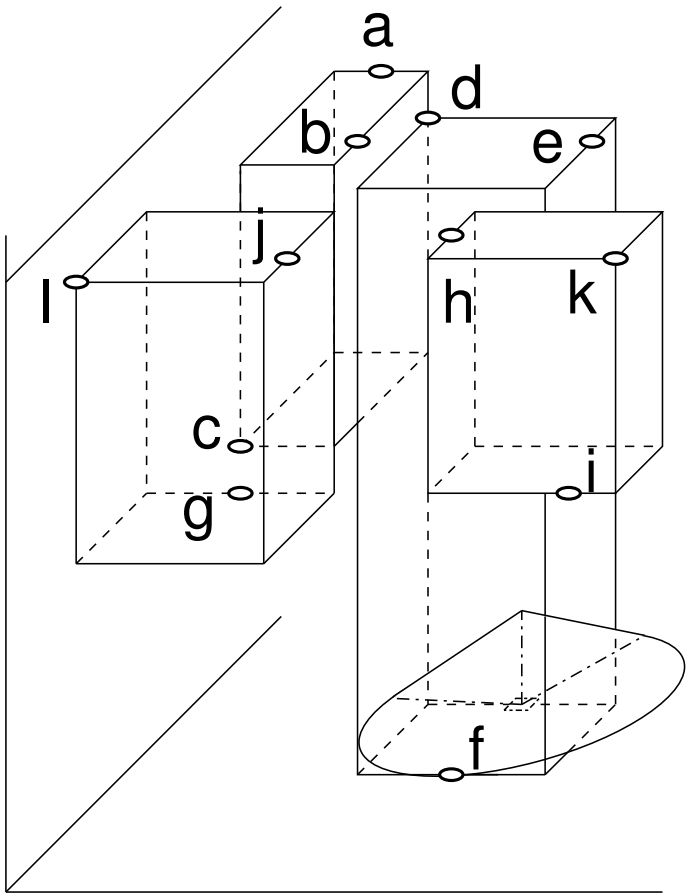
Power-law like node extents

Bad Behavior of the Two Straightforward Strategies

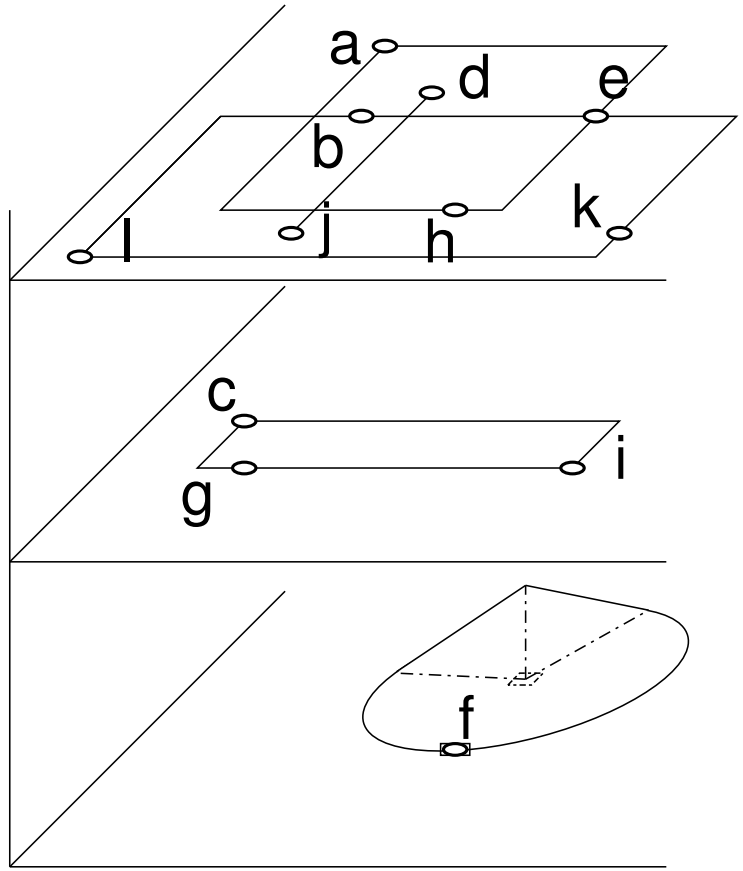


Spatial grouping

Bad Behavior of the Two Straightforward Strategies



Spatial grouping



Aggregate grouping

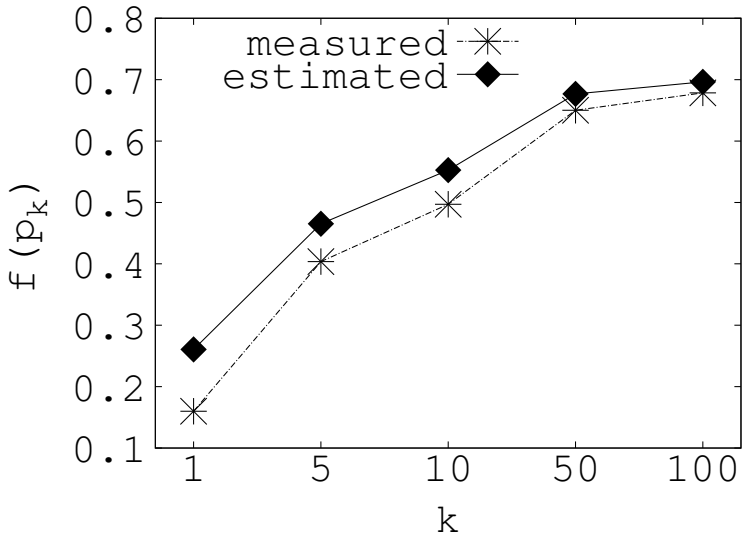
Experiment Set Up

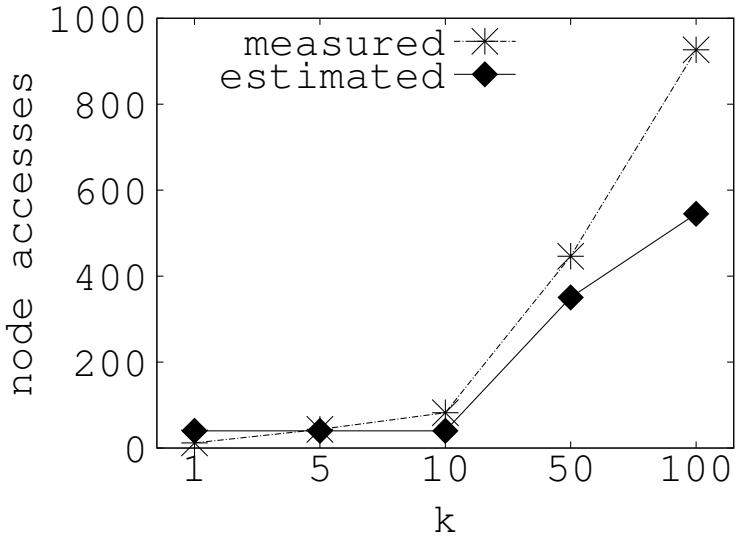
Table: Data Set

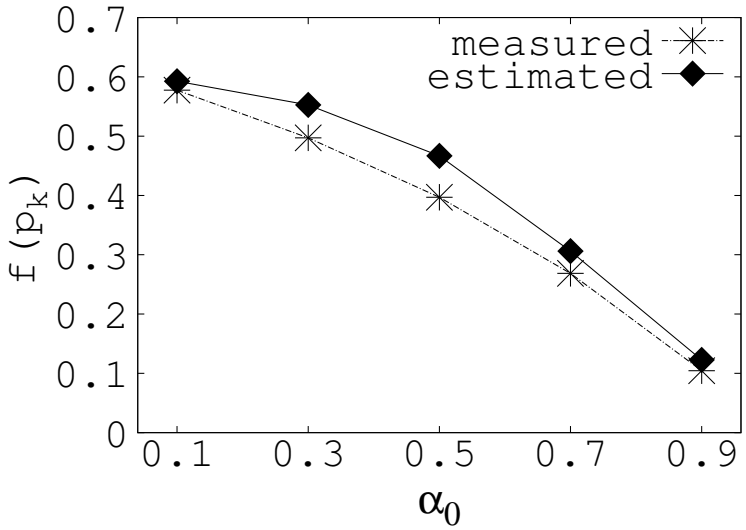
Name	Time	Locations	Check-ins
NYC	05/2008-06/2011	72,626	237,784
LA	02/2009-07/2011	45,591	127,924
GW	02/2009-10/2010	1,280,969	6,442,803
GS	01/2011-07/2011	182,968	1,385,223

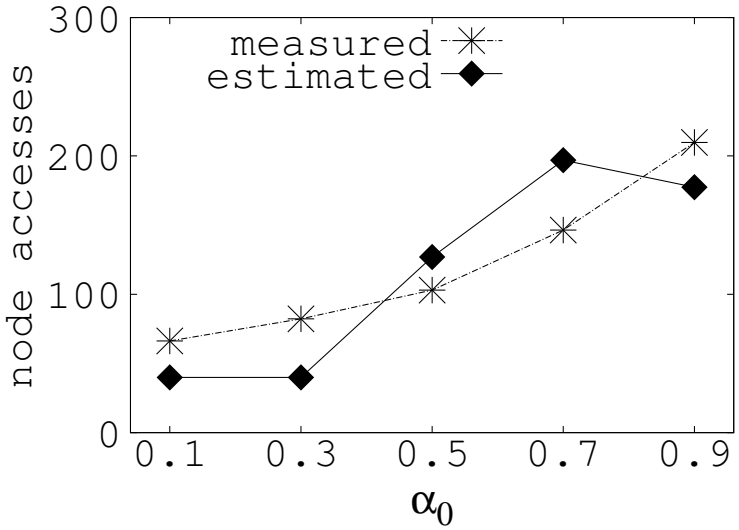
- Temporal index: Multi-version B-tree
- Desktop with 3.40GHz CPU and 16GB RAM
- Results are averaged over $1,000$ queries
- By default $k = 10$ and $\alpha = 0.3$.

Validation of The Analysis

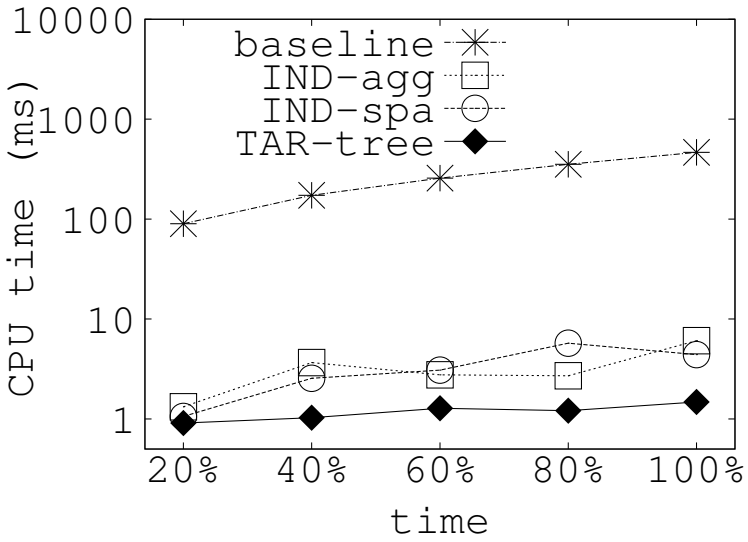


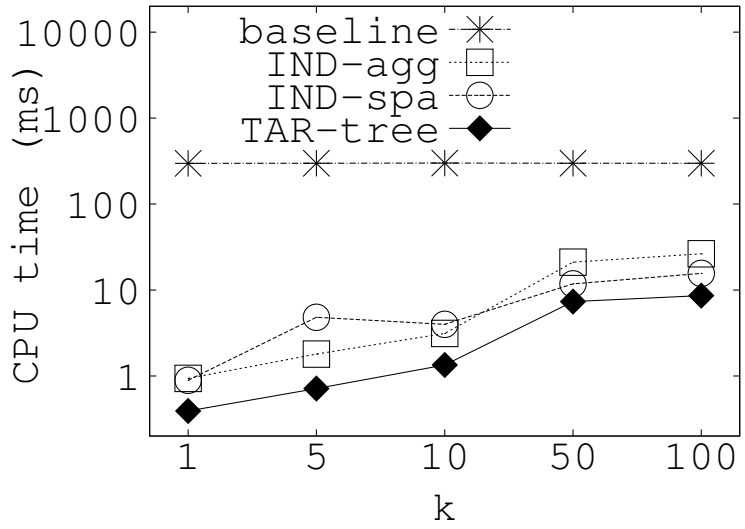


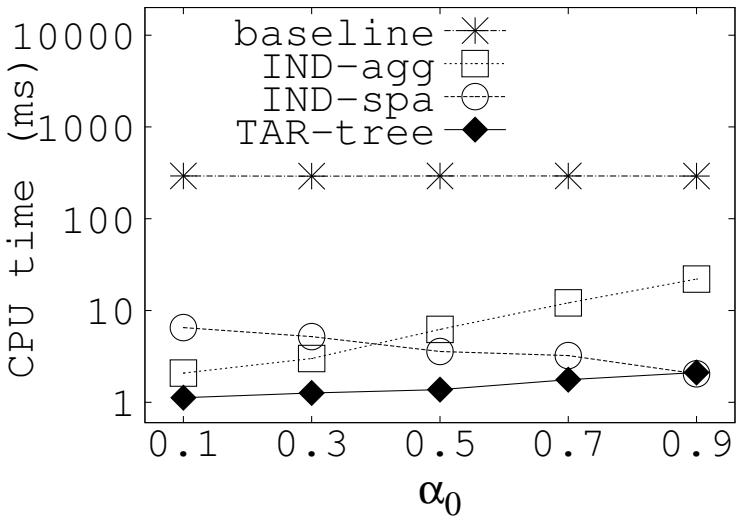


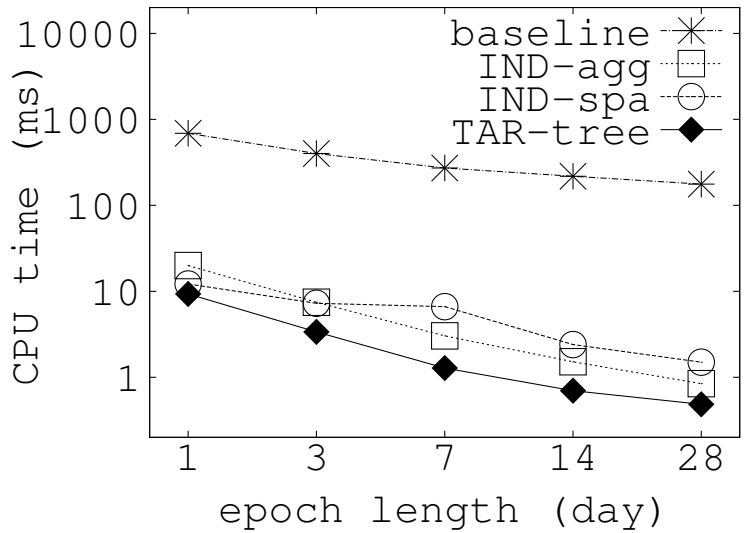


Performance of the TAR-tree









Conclusion

- The kNNTA query can provide highly customized location retrieval and has wide applications.
- The TAR-tree index efficiently processes the kNNTA query.

Questions?