LARS: A Location-Aware Recommender System

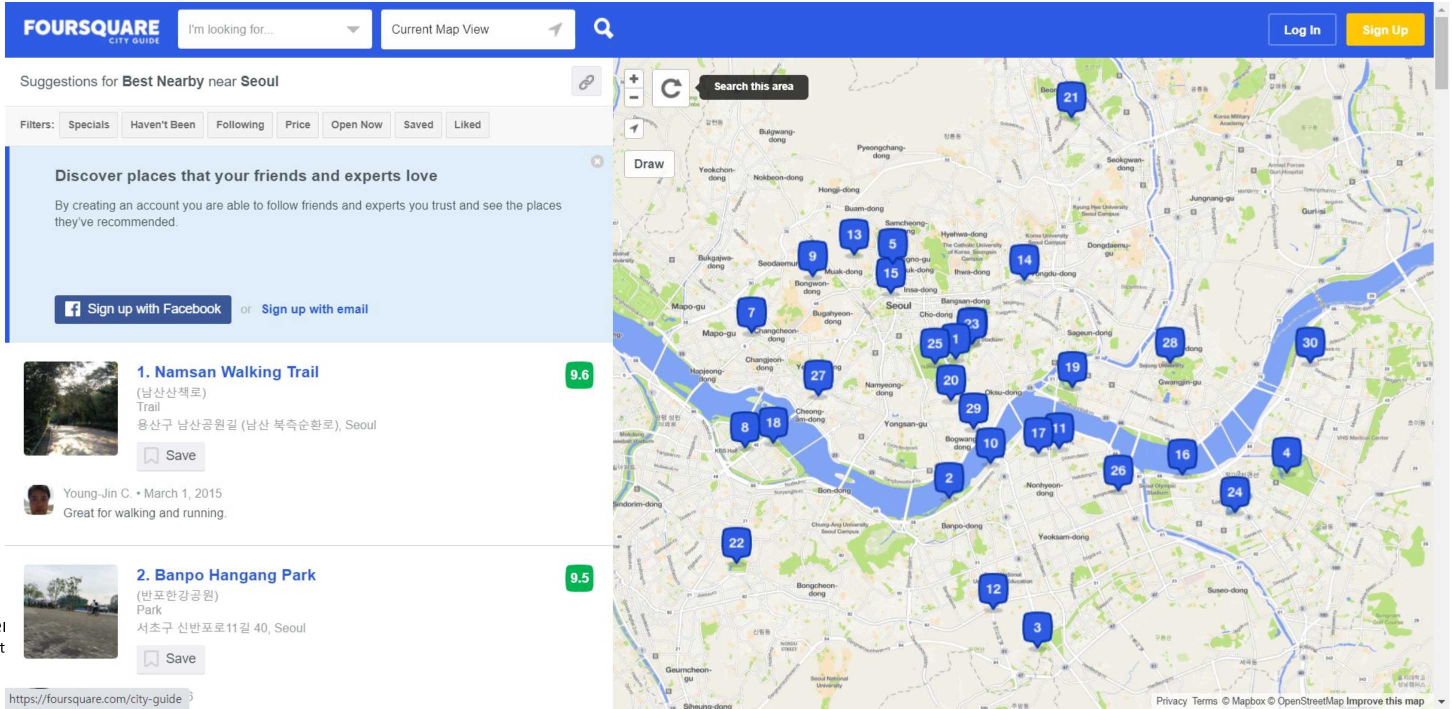
Authors: Justin J. Levandoski, Mohamed Sarwat, Ahmed Eldawy, Mohamed F. Mokbel

Presenter: Hyunwoo Jung

SEOUL NATIONAL UNIVERSITY

Motivation

- How to build an efficient location-aware recommendation system?



Related Works

- Location-based services
  - Ex1) "local favorites" of Netflix
  - Ex2) Hyper-local place ranking

    (user location, location-related-query) ⇒ top points of interest

  ⇒ Doesn't provide personalized recommendations.

- Geo-measured friend-based collaborative filtering

  ⇒ Large-scale real-world deployment is not considered.

Types of Location-based Ratings

- Spatial ratings for non-spatial items
  (user, **user location**, rating, item)
  ex) A user located at home rating a book.
- Non-spatial ratings for spatial items
  (user, rating, item, **item location**)
  ex) A user with unknown location rating a restaurant.
- Spatial ragins for spatial items
  (user, user location, rating, item, item location)
  ex) A user at his/her office rating a restaurant visited for lunch.

Observation: Preference Locality

- Users in a region share interests.

U.S. State	Top Movie Genres	Avg. Rating
Minnesota	Film-Noir	3.8
	War	3.7
	Drama	3.6
	Documentary	3.6
Wisconsin	War	4.0
	Film-Noir	4.0
	Mystery	3.9
	Romance	3.8
Florida	Fantansy	4.3
	Animation	4.1
	War	4.0
	Musical	4.0

(a) Movielens preference locality

Users from:	Visited venues in:	% Visits
Edina, MN	Minneapolis , MN	37 %
	Edina , MN	59 %
	Eden Prarie , MN	5 %
Robbinsdale, MN	Brooklyn Park, MN	32 %
	Robbinsdale, MN	20 %
	Minneapolis, MN	15 %
Falcon Heights, MN	St. Paul, MN	17 %
	Minneapolis, MN	13 %
	Roseville, MN	10 %

(b) Foursquare preference locality

Observation: Travel Locality

- Users prefer to travel a limited distance.
  - From Foursquare data analysis:
    - 45% of users travel 10 miles or less
    - 75% of users travel 50 miles or less

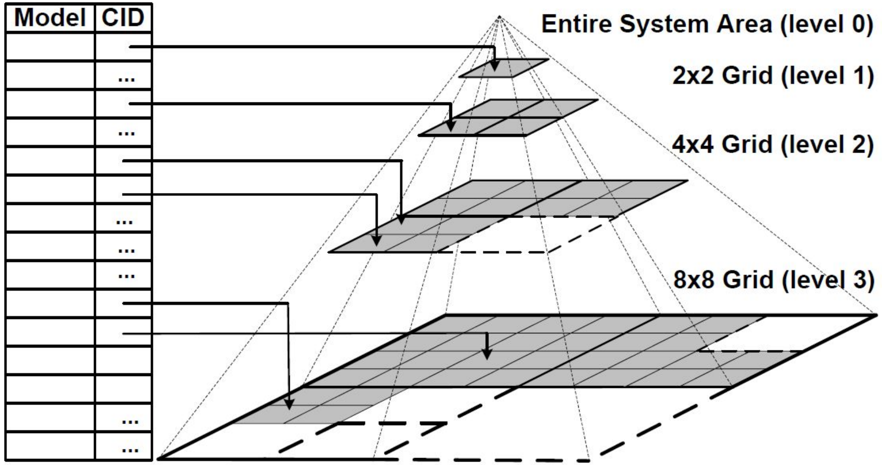
LARS: A Location-Aware Recommender

- LARS: Efficient and scalable Location-aware recommender system that uses location-based ratings.

- Two main considerations/components:
  - Preference locality ⇐ ***user partitioning***
    - Collaborative filter utilizing ratings only located in the querying user's region.
  - Travel locality ⇐ ***travel penalty***

Partial Pyramid for User Partitioning

- LARS employs a partial pyramid for user partitioning.
- For a given level *h*, the space is partitioned into 4^h equal area.
- Each cell contains an item-based collaborative filtering model for corresponding region.



Balancing Scalability/Locality

- Challenge: How to balance scalability and locality of partial pyramid?
  - Maintains a large number of regions increases both ***locality (higher the better)*** and ***scalability (lower the better).***

- Solution: Merging/Split maintenance algorithm
  - scalability_gain < locality_loss ⇒ split the pyramid cell
  - scalability_gain > locality_loss ⇒ merge the pyramid cells

Travel Penalty for Travel Locality

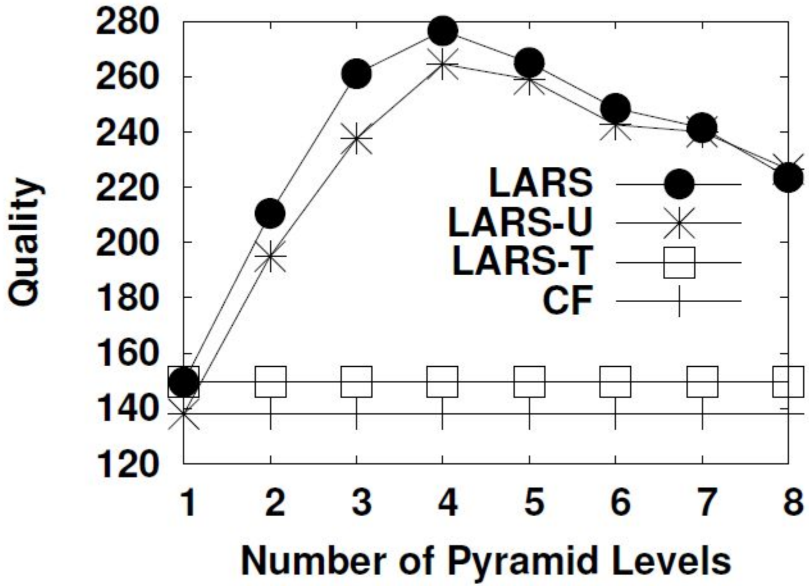
- Approach: $RecScore(u,i) = P(u,i) - TravelPenalty(u,i)$
- Challenge: Computational complexity of calculating *TravelPenalty(u, i)* for all items online is too high. $O(k + logN)$
- Solution: Partition space into grids, and compute the *TravelPenelty* of each grid and items offline.

Datasets for Experiments

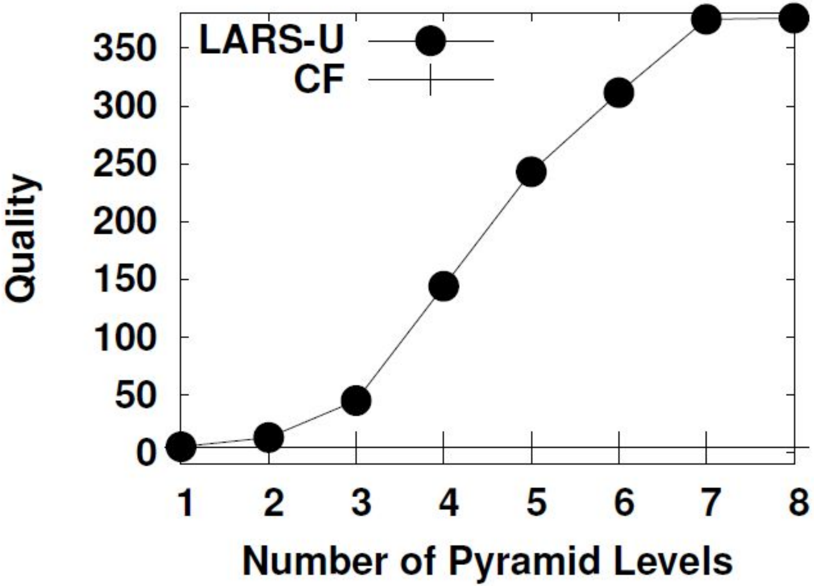
- Foursquare data
  - Data crawled from Foursquare application, a location-based SNS.
  - Contains user notes for venues.
- MovieLens data
  - Movie rating data taken from MovieLens.
- Synthetic data
  - Random data for testing scalability and query efficiency.

Evaluation

- LARS surpasses collaborative filter.



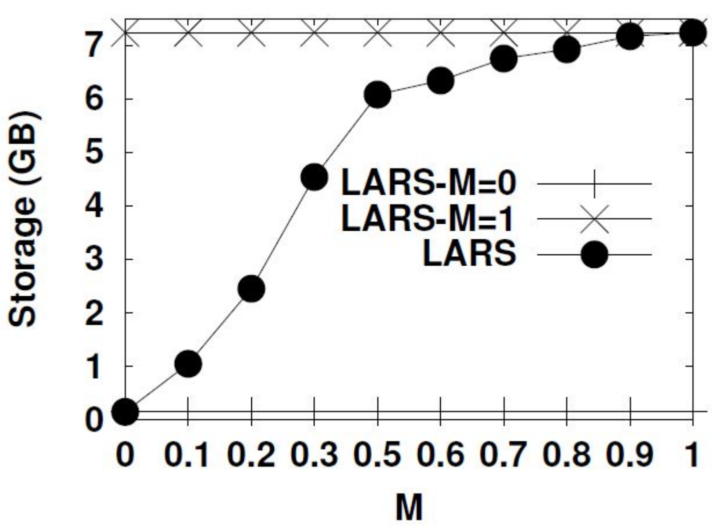
(a) Foursquare data



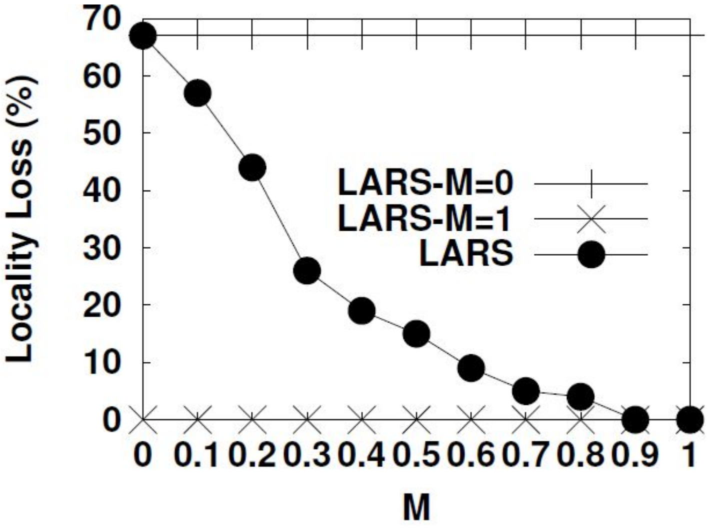
(b) MovieLens data

Scalability-Locality Tradeoff

- Scalability-locality tradeoff of *partial pyramid*



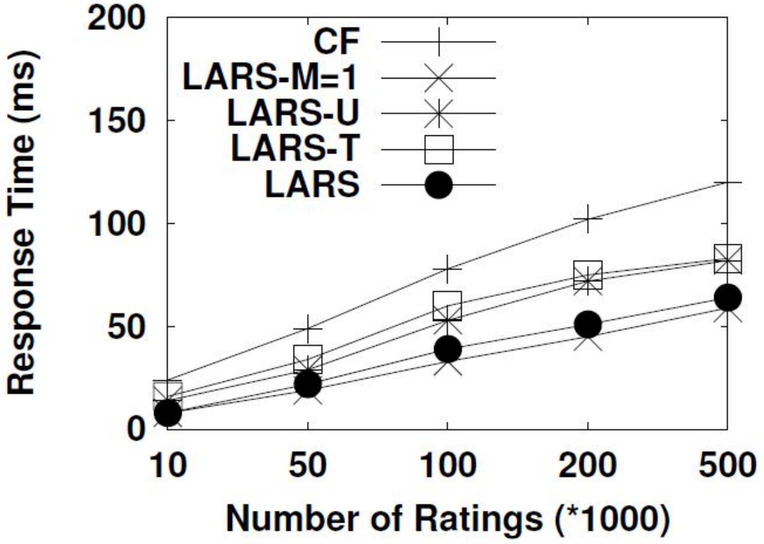
(a) Storage



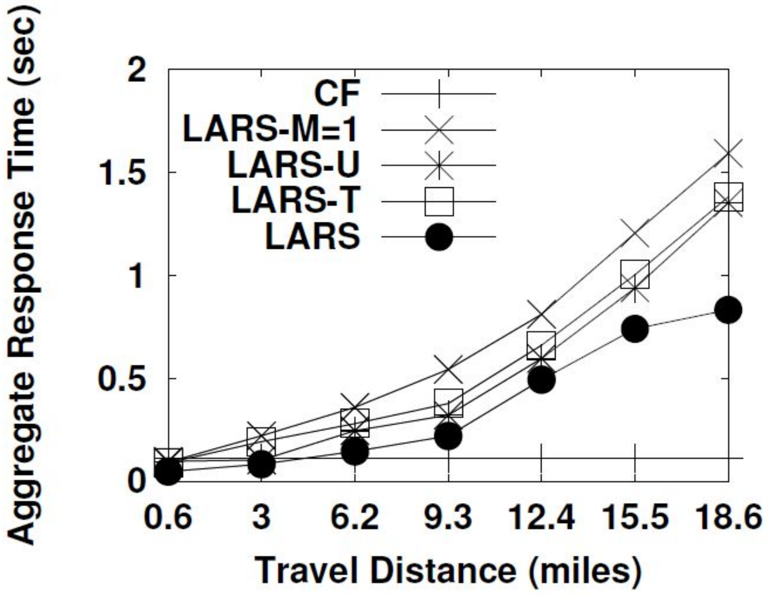
(b) Locality

Snapshot vs Continuous Queries

- LARS reduces the response time of naive approaches.



(a) Snapshot Queries



(b) Continuous Queries

Summary

- LARS is the first location-aware recommender system to consider implicit preferences considering user/travel locality.

- LARS effectively leverages computational resources which enables real-world deployment.

Thank You
Q & A