LARS*: An Efficient and Scalable Location-Aware Recommender System

Prof. Rathod R. B.[1], Dongare Deepak M.[2], Nagargoje Nilesh R.[3], Karale Mahendra A.[4]

[1] *Professor, Computer Engineering, P.D.E.A.'s COE Manjari(Bk), Maharastra, India*
[2] *Student, Computer Engineering, P.D.E.A.'s COE Manjari(Bk), Maharastra, India*
[3] *Student, Computer Engineering, P.D.E.A.'s COE Manjari(Bk), Maharastra, India*
[4] *Student, Computer Engineering, P.D.E.A.'s COE Manjari(Bk), Maharastra, India*

ABSTRACT

The problem of hyper-local place ranking. Given a user location and query string (e.g., "Indian restaurant"), hyper-local ranking provides a list of top-k points of interest influenced by previously logged directional queries (e.g., map direction searches from point A to point B).This paper proposes LARS, a location-aware recommender system that uses their location-based ratings to show recommendations. Traditional recommender systems do not have spatial properties of users nor items; LARS*, next, supports a taxonomy of three novel classes of location-based ratings, namely, spatial ratings for non-spatial items, non-spatial ratings for spatial items, and spatial ratings for spatial items. LARS* exploits user rating locations through user partitioning, a technique that influences recommendations with ratings spatially close to querying users in a manner that maximizes system scalability while not sacrificing recommendation quality. LARS* exploits item locations using travel penalty, a technique that favors recommendation candidates closer in travel distance to querying users in a way that avoids exhaustive access to all spatial items. LARS* can apply these techniques separately, or together, depending on the type of location-based rating available. Experimental evidence using large-scale real-world data from both the Foursquare location-based social network and the Movie Lens movie recommendation system reveals that LARS* is efficient, scalable, and capable of producing recommendations twice as accurate compared to existing recommendation approaches. Our proposed location-aware recommender system, tackles a problem untouched by traditional recommender systems by dealing with three types of location-based ratings: spatial ratings for non-spatial items, non-spatial ratings for spatial items, and spatial ratings for spatial items. LARS* employs user partitioning and travel penalty techniques to support spatial ratings and spatial items, respectively.*

Keyword: - • *Recommender system,* • *Spatial location* • *Social,*

1. INTRODUCTION

RECOMMENDER systems make use of community opinions to help users identify useful items from a considerably large search space (e.g., Amazon inventory [1], Netflix movies1). The technique used by many of these systems is collaborative filtering (CF) [2], which analyzes past community opinions to find correlations of similar users and items to suggest k personalized items (e.g., movies) to a querying user u. Community opinions are expressed through explicit ratings represented by the triple (user, rating, item) that represents a user providing a numeric rating for an item. Currently, myriad applications can produce location-based ratings that embed user and/or item locations. For example, location-based social networks (e.g., Foursquare2 and Facebook Places [3]) allow users to "check-in" at spatial destinations (e.g., restaurants) and rate their visit, thus are capable of associating both user and item locations with ratings. Such ratings motivate an interesting new paradigm of location-aware recommendations, whereby the recommender system exploits the spatial aspect of ratings when producing recommendations. Existing recommendation techniques [4] assume ratings are represented by the (user, rating, item) triple, thus are ill-equipped to produce locationaware recommendations. In this paper, we propose LARS*, a novel locationaware recommender system built specifically to produce high-quality location-based recommendations in an efficient manner. LARS* produces recommendations using a taxonomy of three types of location-based ratings within a single framework: (1) Spatial ratings for non-spatial items, represented as a four-tuple (user, ulocation, rating, item), where ulocation represents a user location, for example, a user located at home rating a book; (2) non-spatial ratings for spatial items, represented as a four-tuple (user, rating, item, ilocation), where ilocation represents an item location, for example, a user with unknown location rating a restaurant; (3) spatial ratings for spatial items, represented as a five-tuple (user, ulocation, rating, item, ilocation), for example, a user at his/her office rating a restaurant visited for lunch. Traditional rating triples can be classified as non-spatial ratings for non-spatial items and do not fit this taxonomy.

1.1 LARS* - A Location-Aware Recommender System

LARS* produces recommendations using spatial ratings for non-spatial items, i.e., the tuple (user, ulocation, rating, item), by employing a user partitioning technique that exploits preference locality. This technique uses an adaptive pyramid structure to partition ratings by their user location attribute into spatial regions of varying sizes at different hierarchies. For a querying user located in a region R, we apply an existing collaborative filtering technique that utilizes only the ratings located in R. The challenge, however, is to determine whether all regions in the pyramid must be maintained in order to balance two contradicting factors: scalability and locality.

a novel classification of three types of location-based ratings not supported by existing recommender systems: spatial ratings for non-spatial items, non-spatial ratings for spatial items, and spatial ratings for spatial items.

2. LITERATURE SURVEY

Some previous systems shows different studies on Solid Waste Collection System In Ipoh City.One such system which Amirhossein Malakahmad and Noor Diana Khalil.[1] proposed shows the concept of collection system.

Similarly, Patrick Brézillon [4] proposed an Modeling Temporal Effectiveness for Context-aware Web Services Recommendation approach from the security. Based on the concept of “social circles” and places

.

This was done on the basis of time of the day and location. The study of whether the keywords and captions used for A key step for temporal-based CARS methods is to explore the time decay process of past invocation records to make the Quality of Services (QoS) prediction.

3. RELATED WORK

Location-based services. Current location-based services employ two main methods to provide interesting destinations to users. (1) KNN techniques [22] and variants (e.g., aggregate KNN [24]) simply retrieve the k objects nearest to a user and are completely removed from any notion of user personalization. (2) Preference methods such as skylines [25] (and spatial variants [26]) and location-based top-k methods [27] require users to express explicit preference constraints. Conversely, LARS* is the first locationbased service to consider implicit preferences by using location-based ratings to help users discover new items.

Recent research has proposed the problem of hyper-local place ranking [28]. Given a user location and query string (e.g., “French restaurant"), hyper-local ranking provides a list of top-k points of interest influenced by previously logged directional queries (e.g., map direction searches from point A to point B). While similar in spirit to LARS*, hyper-local ranking is fundamentally different from our work as it does not personalize answers to the querying user, i.e., two users issuing the same search term from the same location will receive exactly the same ranked answer. Traditional recommenders. A wide array of techniques are capable of producing recommendations using nonspatial ratings for non-spatial items represented as the triple (user, rating, item) (see [4] for a comprehensive survey). We refer to these as “traditional" recommendation techniques. The closest these approaches come to considering location is by incorporating contextual attributes into statistical recommendation models (e.g., weather, traffic to a destination) [29]. However, no traditional approach has studied explicit location-based ratings as done in LARS*. Some existing commercial applications make cursory use of location when proposing interesting items to users. For instance, Netflix displays a “local favorites” list containing popular movies for a user’s given city.

However, these movies are not personalized to each user (e.g., using recommendation techniques); rather, this list is built using aggregate rental data for a particular city [30]. LARS*, on the other hand, produces personalized recommendations influenced by location-based ratings and a query location.

4. LARS* OVERVIEW:

provides an overview of LARS* by discussing the query model and the collaborative filtering method

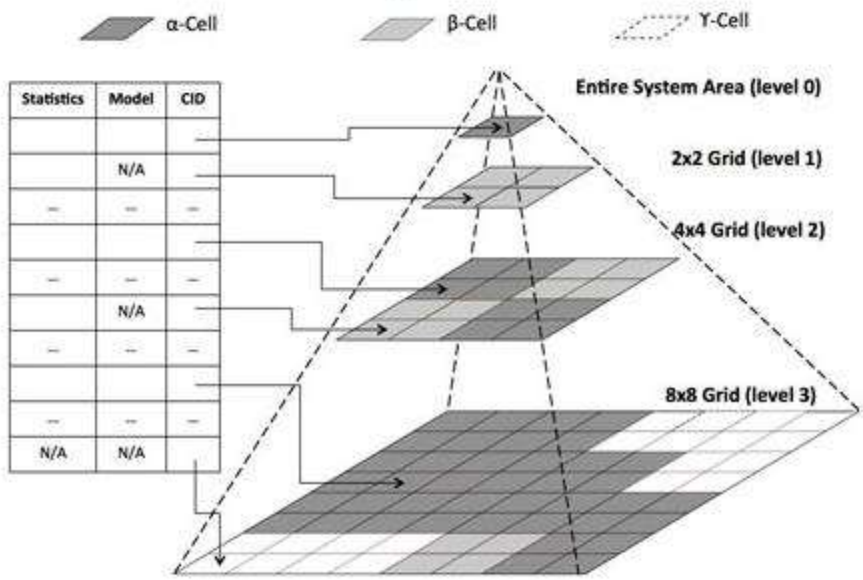

Fig 1. LARS* Framework

2.1 LARS* Query Model

Users (or applications) provide LARS* with a user id U,numeric limit K, and location L; LARS* then returns K recommended items to the user. LARS* supports both snapshot (i.e., one-time) queries and continuous queries, whereby a user subscribes to LARS* and receives recommendation updates as her location changes. The technique LARS* uses to produce recommendations depends on the type of location-based rating available in the system.

2.2 Item-Based Collaborative Filtering

LARS* uses item-based collaborative filtering (abbr. CF) as its primary recommendation technique, chosen due to its popularity and widespread adoption in commercial systems (e.g., Amazon [1]).

5 Relevant Mathematics Associated with the Project

Module 1) Statistics Maintenance.

The first step is to maintain the Items Ratings Statistics Table. The maintained statistics are necessary for cell type switching decision, especially when new location-based ratings enter the system. As the items ratings statistics table is implemented using a hash table, then it can be queried and maintained in O(1) time, requiring O(|IC|) space such that IC is the set of all items rated at cell C and |IC| is the total number of items in IC.

Module 2) Model Rebuild.

The second step is to rebuild the item-based collaborative filtering (CF) model for a cell C, as described in Section 2.2 (line 7). The model is rebuilt at cell C only if cell C is an α-Cell, otherwise (β-Cell or γ -Cell) no CF recommendation model is maintained, and hence the model rebuild step does not apply. Rebuilding the CF model is necessary to allow the model to "evolve" as new location-based ratings enter the system (e.g., accounting for new items, ratings, or users). Given the cost of building the item-based CF model is O(R2 U) (per Section 2.2), the cost of the model rebuild for a cell C at level h is (R/4h)2 (U/4h)= R2 4hU, assuming ratings and users are uniformly distributed.

Module 3) Cell Child Quadrant Maintenance.

LARS* invokes a maintenance step that may decide whether cell C child quadrant need to be switched to a different cell type based on trade-offs between scalability and locality. The algorithm first checks if cell C child quadrant q at level h + 1 is of type α-Cell . If that case holds, LARS* considers quadrant q cells as candidates to be downgraded to β-Cells (calling function Check Down Grade ToS Cells We provide details of the Downgrade α-Cells to β-Cells. On the other hand, if C have a child quadrant of type γ -Cells at level h+1,LARS* considers upgrading cell C four children cells at level h+1 to β-Cells (calling function Check Up Grade ToS Cells. The Upgrade to β-Cells operation However, if C has a child quadrant of type β-Cells at level h+1, LARS* first considers upgrading cell C four children cells at level h + 1 from β-Cells to α-Cells (calling function CheckUpGradeToMCells). If the children cells are not switched to α-Cells, LARS* then considers downgrading them to γ -Cells (calling function CheckDownGradeToECells). Cell Type switching operations are performed completely in quadrants (i.e., four equi-area cells with the same parent).We made this decision for simplicity in maintaining the partial pyramid.

6. CONCLUSIONS

LARS*, our proposed location-aware recommender system, tackles a problem untouched by traditional recommender systems by dealing with three types of location-based ratings: spatial ratings for non-spatial items, non-spatial ratings for spatial items, and spatial ratings for spatial items. LARS* employs user partitioning and travel penalty techniques to support spatial ratings and spatial items, respectively. Both techniques can be applied separately or in concert to support the various types of location-based ratings. Experimental analysis using real and synthetic data sets show that LARS* is efficient, scalable, and provides better quality recommendations than techniques used in traditional recommender systems.

7. ACKNOWLEDGEMENT

This work was supported in part by the US National Science Foundation under Grants IIS-0811998, IIS-0811935, CNS-0708604, IIS-0952977 and in part by a Microsoft Research Gift.

The authors would like to thank University Grants Commission (UGC) of India for supporting this research work as part of the Major Research Project "Trustworthy Proactive Recommender System" to Dr. Punam Bedi as PI.

8. REFERENCES

[1] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: Item-to-item collaborative filtering," IEEE Internet Comput., vol. 7, no. 1, pp. 76–80, Jan./Feb. 2003.

[2] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: An open architecture for collaborative filtering of netnews," in Proc. CSWC, Chapel Hill, NC, USA, 1994.

[3] The facebook blog. Facebook Places [Online]. Available: http://tinyurl.com/3aetfs3

[4] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," IEEE Trans. Knowl. Data Eng., vol. 17, no. 6, pp. 734–749, Jun. 2005.

[5] MovieLens [Online]. Available: http://www.movielens.org/

[6] Foursquare [Online]. Available: http://foursquare.com