LARS*: Location-Aware Recommendation System

Swati S. Avhad
Department of Computer Engineering
SND COE & RC, Yeola, India

Santosh R. Durugkar
Department of Computer Engineering
SND COE & RC, Yeola, India

Abstract

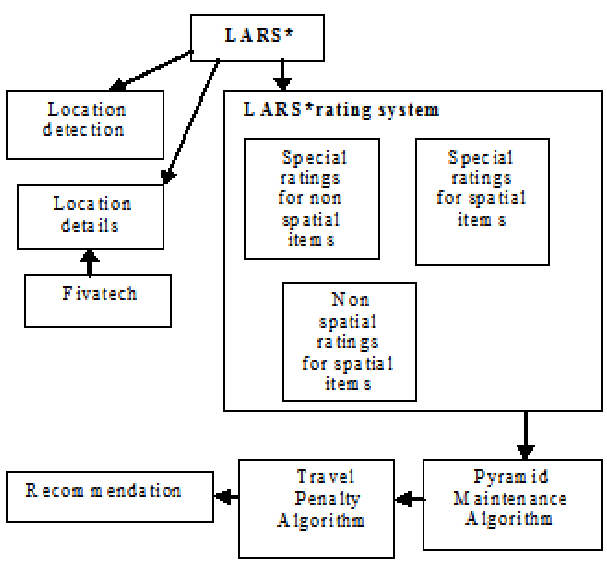
LARS* is a location aware recommendation system that detects the location of user from IP address of system, gives details of that location and uses location based rating to produced recommendations. Recommender systems which are traditional do not consider spatial properties of users nor items; LARS*, on the other hand, supports a technique of three classes of location-based ratings, i.e. the spatial ratings for non-spatial items, the non-spatial ratings for the spatial items, and the spatial ratings for the spatial items. Details about user's location are provided by using FivaTech. It propose an unsupervised, page-level data extraction to deduce the schema and templates for each individual huge Websites, which contains either singleton or multiple data records in one Webpage. FiVaTech applies tree matching, tree alignment, and mining techniques to achieve the challenging task. FivaTech is going to take place for web mining purpose.

Keywords: Recommendation system, spatial location performance, FivaTech efficiency

---

I. INTRODUCTION

This system is going to detect the location of user from the IP address of user's system. After detection of location of user, the details about that particular location will be given to user such as picnic spots, clinics, theators, restaurants etc. present at that location. After that, rating system is going toprovided as per following discussion. RECOMMENDATION systems make use of community opinions to help users identify useful items from a considerably large search space (e.g., Amazon inventory,Netflix movies1). The technique used by many of these systems is collaborative filtering. Which determines past community suggestions to find correlation of same users and items to suggest k personalized items to a querying user u. community suggestions are collected by user, rating, item that represented 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., Facebook Places) 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 recommenation system exploits the spatial aspect of ratings when produce location aware recommendations. Also FivaTech is going to be used for giving details of detected location.



\

Fig. 1.1: LARS* structure

II. RELATED WORK

The inspiration for work comes from analysis of two real-world location-based rating datasets:

1) a subset of the well-known MovieLens dataset containing approximately 87K movie ratings associated with user zip codes (i.e., spatial ratings for non-spatial item
2) data from the location-based social network containing user visit data for 1M users to 643K venues across the United States (i.e., spatial ratings for spatial items).

In analysis, continuously observed two interesting properties that inspire the need for location-aware recommendation techniques. Preference locality suggestions from the special region prefer items that are different than items preferred by users from other, even adjacent, regions. As shown in an example in results,lists the top-4 movie genres using average MovieLens ratings of users from different U.S. states. While each list is different, the top genres from Florida differ vastly from the others. Florida's list contains three genres ("Fantasy", "Animation", "Musical") not in the other lists. This difference implies movie preferences are unique to specific spatial regions, and confirms previous work from the New York Times that analyzed Netflix user queues across U.S. zip codes and found similar differences. Preference locality suggests that recommendations should be influenced by locationbased ratings spatially close to the user. The intuition is that localization influences recommendation using the unique preferences found within the spatial region containing the user. Travel locality. Second observation is that, when recommended items are spatial, users tend to travel a limited distance when visiting these venues. We refer to this property as "travel locality." In our analysis of Foursquare data, it is observed that 45% of users travel 10 miles or less, while 75% travel 50 miles or less. This observation suggests that spatial items closer in travel distance to a user should be given precedence as recommendation candidates. In other words, a recommendation loses efficacy the further a querying user must travel to visit the destination. Existing recommendation techniques do not consider travel locality, thus may recommend users destinations with burdensome travel distances (e.g., a user in Chicago receiving restaurant recommendations /in Seattle). All the above system will work on rating system. Then Fivatech will work on giving details of detected location.

III. PROBLEM STATEMENT

First of all, this system will identify location of user from an IP address of user's system. After location detection, system is going to provide us 2 facilities i.e. providing location details and providing Rating system.

LARS* produces recommendations using special ratings for non special items, i.e, the rows by employing a user partitioning technique that exploits preference locality. This technique uses an adaptive pyramid structure to partition ratings by their user location parameters into spatial areas of different sizes at different levels. For a querying user located in a areas 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. Maintaining a maximum number of areas increases locality (i.e., recommendations unique to smaller spatial regions),till now adversely affects system scalability because each area requires storage and maintenance of a collaborative filtering data structure necessary to produce recommendations (i.e., the recommender model). The LARS* pyramid dynamically adapts to find the right pyramid shape that balances scalability and recommendation locality.

There are a novel classification of 3 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.

IV. FIVATECH

Very Huge Web, as is known to everyone, contains magnitudes more and valuable information than the surface Web. However, making use of such consolidated information requires substantial efforts since the pages are generated for visualization not for data exchange. Thus, extracting information from Webpages for searchable Websites has been a key step for Web information integration. Generating an extraction program for a given search form is equivalent to wrapping a data source such that all extractor or wrapper programs return data of the same format for information integration.An important characteristic of pages belonging to the same Website is that such pages share the same template since they are encoded in a consistent manner across all the pages. In other words, these pages are generated with a predefined template by plugging data values. In practice, template pages can also occur in surface Web (with static hyperlinks). For example, commercial Websites often have a template for displaying company logos, browsing menus, and copyright announcements, such that all pages of the same Website look consistent and designed. In addition, templates can also be used to render a list of records to show objects of the same kind. Thus, information extraction from template pages can be applied in many situations. What's so special with template pages is that the extraction targets for template Webpages are almost equal to the data values embedded during page generation.

Thus, there is no need to annotate the Webpages for extraction targets as in non template page information extraction.

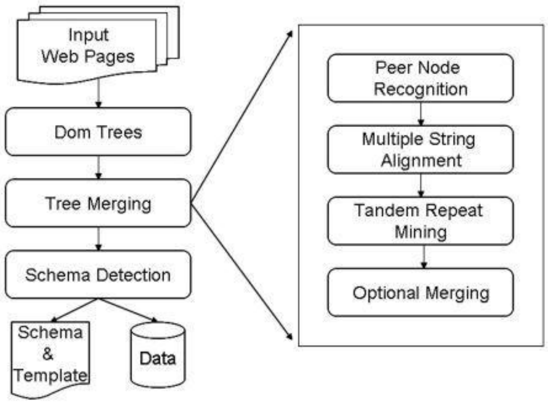


Fig. 1.2: FivaTech structure

The above figure shows the structure of FivaTech.

V. ALGORITHMS USED FOR IMPLEMENTATION

A. *Pyramid Maintenance Algorithm:*

1) /* Called after cell *C* receives *N*% new ratings */
2) Function Pyramid Maintenance(Cell *C*, Level *h*)
3) /* Step I: Statistics Maintenance*/
4) Maintain cell *C* statistics
5) /*Step II: Model Rebuild */
6) if (Cell *C* is an *α*-Cell) then
7) Rebuild item-based collaborative filtering model for cell *C*
8) end if
9) /*Step III: Cell Child Quadrant Maintenance */
10) if (*C* children quadrant *q* cells are *α*-Cells) then
11) CheckDownGradeToSCells(*q*,*C*)
12) else if (*C* children quadrant *q* cells are *γ* -Cells) then
13) CheckUpGradeToSCells(*q*,*C*)
14) else
15) isSwitchedToMcells ← CheckUpGradeToMCells(*q*,*C*) /* covered in Section 4.5.3 */
16) if (isSwitchedToMcells is False) then
17) CheckDownGradeToECells(*q*,*C*)
18) end if
19) end if
20) return

B. *Travel Penalty Algorithm:*

1) Function LARS*_SpatialItems(User *U*, Location *L*, Limit *K*)
2) /* Populate a list *R* with a set of *K* items*/
3) *R* ← *φ*
4) for (*K* iterations) do
5) *i* ← Retrieve the item with the next lowest travel penalty (Section 5.2)
6) Insert *i* into *R* ordered by *RecScore(U, i)* computed by Equation 7
7) end for
8) *LowestRecScore*← *RecScore*of the *kth*object in *R*
9) /*Retrieve items one by one in order of their penalty value */
10) while there are more items to process do
11) *i* ← Retrieve the next item in order of penalty score (Section 5.2)
12) *MaxPossibleScore*← *MAX_RATING - i.penalty*
13) if *MaxPossibleScore*≤ *LowestRecScore*then
14) return *R* /* early termination - end query processing */
15) end if
16) *RecScore(U, i)* ← *P(U, i) - i.penalty*/* Equation 7 */

17) if *RecScore(U, i) >LowestRecScore*then
18) Insert *i* into *R* ordered by *RecScore(U, i)*
19) *LowestRecScore*← *RecScore*of the *kth*object in *R*
20) end if
21) end while
22) return *R*

C. Multiple Tree Merge Algorithms

D. Fivamatchscore Algorithm

E. Matrix Alignment

F. Pattern mining

The above stated 3,4,5,6 algorithms are used in FivaTech which are based on concept of tree.

VI. IMPLEMENTATION MODULES

A. FivaTech:

FivaTech technology is used for Web mining for providing location details to users.
By using various algorithms stated above, FivaTech is implemented.

1) Spatial Ratings for Non Spatial Ratings:

Spatial ratings for non-spatial items stated by the tuple (*user*, *ulocation*, *rating*, *item*). The idea is to states *preference locality*, i.e., the observation that user opinions are spatially unique. We identify three requirements for producing recommendations using spatial ratings for non-spatial items:

1) Locality: recommendations should be influenced by those ratings with user locations patially close to the querying user location (i.e., in a spatial neighborhood);
2) Scalability: the recommendation procedure and data structure should scale up to large number of users.
3) Influence: system users should have the ability to control the size of the spatial neighborhood (e.g., city block, zip code, or county) that affects their recommendations.

2) Non Spatial Ratings for Non Spatial Items:

The traditional item-based collaborative filtering (CF) method is a special case of LARS*. CF takes as input the classical rating triplet (*user*, *rating*, *item*) such that neither the user location nor the item location are specified. In such case, LARS* directly employs the traditional model building phase (Phase-I in section 2) to calculate the similarity scores between all items. Moreover, recommendations are produced to the users using the recommendation generation phase (Phase-II in section 2). During the rest of the paper, we explain how LARS* incorporates either the user spatial location or the item spatial location to serve location-aware recommendations to the system users.

3) Non Spatial Ratings for Spatial Items:

This section tells that how LARS* produces recommendations using non-spatial ratings for spatial items represented by the tuple (user, rating, item, ilocation). The idea is to exploit travel locality, i.e., the observation that users limit their choice of spatial venues based on travel distance based on analysis in Traditional (non-spatial) recommendation techniques may produce recommendations with burdensome travel distances (e.g., hundreds of miles away). LARS* produces recommendations within reasonable travel distances by using travel penalty, a technique that penalizes the recommendation rank of items the further in travel distance they are from a querying user. Travel penalty may incur expensive computational overhead by calculating travel distance to each item. Thus, LARS* employs an efficient query processing technique capable of early termination to produce the recommendations without calculating the travel distance to all items.

4) Pyramid Maintenance Algorithm:

In this algorithm 1, A α-Cell requires the highest storage and maintenance overhead because it maintains a CF model as well as the user or item ratings statistics. On the other hand, an α-Cell (as opposed to β-Cell and γ -Cell) is the only cell that can be leveraged to answer recommendation queries. A pyramid structure that only contains α-Cells achieves the highest recommendation locality, and this is why an α-Cell is considered the highly ranked cell type in LARS*. a β-Cell is the secondly ranked cell type as it only maintains statistics about the user/item ratings. The storage and maintenance overhead incurred by a β-Cell is less expensive than an α-Cell. The statistics maintained at a β-Cell determines whether the children of that cell need to be maintained as α-Cells to serve more localized recommendation. Finally, a γ -Cell (lowest ranked cell type) has the least maintenance cost, as neither a CF model nor statistics are maintained for that cell. Moreover, a γ -Cell is a leaf cell in the pyramid. LARS* upgrades (downgrades) a cell to a higher (lower) cell rank, based on trade-offs between recommendation locality and system scalability. If recommendation locality is preferred over scalability, more α-Cells are maintained in the pyramid. On the other hand, if scalability is favored over locality, more γ -Cells exist in the pyramid. β-Cells comes as an intermediary stage between α-Cells and γ -Cells to further increase the recommendation locality whereas the system scalability is not quite affected.

5) Travel Penalty Algorithm:

An Algorithm 2 provides the pseudo code of our query processing algorithm that takes a querying user id U, a location L, and a limit K as input, and returns the list R of top-k recommended items. The algorithm starts by running a k-nearest-neighbor algorithm to populate the list R with k items with lowest travel penalty; R is sorted by the recommendation score. This initial part is concluded by setting the lowest recommendation score value (LowestRecScore) as the RecScore of the kth item in R . Then, the algorithm starts to retrieve items one by one in the order of their penalty score. This can be done using an incremental k-nearest-neighbor algorithm. Travel penalty requires very little maintenance. The only maintenance necessary is to occasionally rebuild the single system-wide item-based collaborative filtering model in order to account for new location-based ratings that enter the system.

6) Recommendation:

A large number of techniques are capable of producing recommendations using non spatial ratings for non-spatial items represented as the triple (user, rating, item). We prefer to those techniques 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).All the algorithms stated in algorithms section are used for giving recommendation to users.

Recommendation is the most important module which is actually providing rating system to user. And the drawbacks in traditional systems are overcame in this recommendation system.

VII. RESULTS

The proposed system will give us following results:

- First of all location of user is going to be detected from an IP address of system.
- Then one whether module is also provided which will give temperature of detected location.
- Details of detected location will be provided such as restaurants, hospitals, picnic spots, temples etc available on thet location.
- In short, system is going to provide location detection, detailed information of detected location and rating system for movies in detected location.

Ex:-Suppose Nasik city is detected, then it will provide all information regarding picnic spots, hospitals, theaters etc available in nasik. Then system will provide rating system for movies in detected city Nasik.

Table – 1
Example of rating system of movie

Maharashtra State	*Top movies*	*Avg. Rating*
Mumbai	*Duniyadari*	*3.8*
	Roy	*3.7*
	Baby	*3.6*
	Lokmanya	*3.6*
Pune	*Lokmanya*	*4.0*
	Baby	*4.0*
	Mitawa	*3.9*
	Duniyadari	*3.8*
Nasik	*Baby*	*4.3*
	Roy	*4.2*
	Duniyadari	*4.1*
	Lokmanya	*3.8*

VIII. CONCLUSION

This system is going to work on two things i.e. after location detection, it will provide details of detected location and also going to provide rating system.

LARS* faces a problems which are unsolved 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.

ACKNOWLEDGEMENT

This project is done by me during my study of post-graduation. I am very glad for having such good platform for developing such a good work & this project development is one of very good experience during my studies.

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] Mohammed Kayed and Chia-Hui Chang, Member ,”FivaTech: Page-Level Web Data Extraction from Template Pages”,IEEE transactions on knowledge and data engineering,vol. 22, no. 2, Feb 2010
[3] 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.
[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/map.html
[6] J. J. Levandoski, M. Sarwat, A. Eldawy, and M. F. Mokbel, “LARS: A location-aware recommender system,” in Proc. ICDE, Washington, DC, USA, 2012.

