Cognizable Recommendation System using Spatial Ratings with Collaborative Filter

Shilpa Mathapati[1], Bhuvaneswari Raju[2]

[1] M.tech , Dept. of CSE, Atria Institute of Technology , Bengaluru, Karnataka, India
[2] Assistant Professor Dept. of CSE, Atria Institute of Technology , Bengaluru, Karnataka, India

---***---

Abstract - Traditional recommender systems do not consider spatial properties of users nor items; proposed system cognizable recommender system is based on locations of item and user also referred as location aware recommender system that exploits community opinions or ratings to produce suggestions for other items also referred as recommendations. CRS (Cognizable Recommender System) uses recent transactions to avoid excessive processing of data which is old and not interested by user. It also supports taxonomy of three novel classes of location-based ratings, namely, non-spatial ratings for spatial items, spatial ratings for non-spatial items, and spatial ratings for spatial items. The technique that is user partitioning exploits user rating locations, influences recommendations with ratings spatially close to querying users. This will help to achieve maximum system scalability without giving-up quality of recommendations. CRS exploits location of item using a technique referred as travel penalty that helps recommendation candidates closer in travel distance to querying users in a way that helps to avoid exhaustive access to all spatial items. These three techniques can be applied separately, or together, depending on the type of location-based rating available. CRS works to provide recommendations which are more accurate than existing approaches and also proven to be efficient and scalable.

Key Words: Spatial ratings, recommender system, performance, efficiency, scalability, collaborative filter

1 INTRODUCTION

Depending upon community opinion in terms of ratings, recommender system predict useful product from a considerably huge space. In traditional recommender system the technique used is collaborative filtering [1], which works based on previous transactions and opinions in term of ratings to find similarity between users and items to suggest k number of items to a user u who is interested in purchasing an item (e.g., Toy, Dress, Movie, Hotel) Product reviews or opinions are given through explicit ratings represented by the triple (*user*, *rating*, *item*) that represents a *user* who purchase an item providing rating for it. At present, myriad applications can produce *location-based ratings* that exploit user and/or item locations. For example, social networks (e.g., Facebook Places [2]) are which allow users to "check-in" at spatial destinations (e.g., restaurants) and rate their visit. These sites are thus able to associate both user and item locations with ratings. Such ratings are used in interesting new concept of *location-aware recommendations*, in which the recommender system uses the spatial aspect of ratings to produce new recommendations. Existing systems assume ratings which are represented by the (*user*, *rating*, *item*) triple, thus are not enough to produce location aware recommendations. In cognizable recommendation system, location aware recommender system built to produce more efficient location-based recommendations.

Cognizable Recommendation System 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 (*user*, *ulocation*, *rating*, *item*), where *ulocation* represents a user location, for example, user who buys an electronic product from home ; (2) non-spatial ratings for spatial items, represented as (*user*, *rating*, *item*, *ilocation*), where *ilocation* represents an item location, for example, a user whose location in unkown rating a movie; (3) spatial ratings for spatial items, represented as a five-tuple (*user*, *ulocation*, *rating*, *item*, *ilocation*), for example, a user rating a restaurant he visited recently from his home.

2. BACKGROUND

Traditional recommender systems produce *K* items personalized for user u whereas CRS also produce location aware recommendations by exploiting three classes of location based ratings in a framework.

Like traditional recommender systems, LARS [3] and LARS*[4] suggests *k* items personalized for a querying user *u*. However, LARS* is distinct in its ability to produce location-aware recommendations using *each* of the three types of location-based rating within a *single* framework. CRS adds one more feature to existing system to limit the excess processing by accessing recent *n* number of transactions. Depending on density of spatial ratings available in particular region number of transactions is to be used to calculate the new recommendations after a purchasing a product and registering as a new user.

CRS employs a *user partitioning* technique that uses preference locality to calculate recommendations using *spatial ratings for non-spatial items*, i.e., the tuple (*user*, *ulocation*, *rating*, *item*). In order to save storage space, proposed system adapt pyramid data structure [5] similar to quad tree. To partition ratings given by user by their location information in terms of different sizes and hierarchies, existing collaborative filtering techniques when user's location in in region R only uses ratings located in R. which regions of the pyramid has to be maintained is thus tricky while tow important factors has to be considered: *scalability* and *locality*. Locality is gained when large number of regions is maintained and system scalability decreases to maintain right pyramid shape balance between scalability and locality has to be maintained.

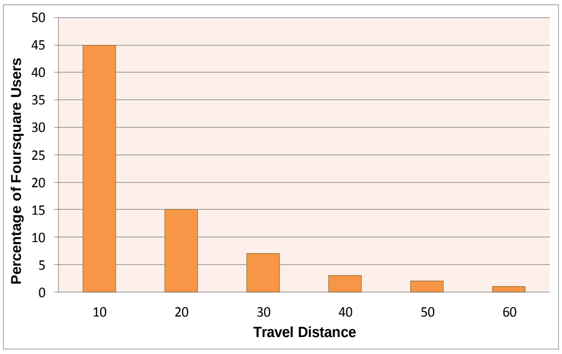

Chart 1: Travel locality: Travel Distance in Foursquare

As shown in figure 1 the graph of the foursquare [10] users, 75% of users travel less than 50 meters. This observation motivates to think about the spatial location of an item. This lead to introduce a technique called *Travel penalty*. CRS thus uses *travel penalty* to produce recommendations using *non-spatial ratings for spatial items*, i.e., the tuple (*user*, *rating*, *item*, *ilocation*). This technique further uses travel distance between user and items to produce recommendation. However system resources are consumed to compute travel distance. This challenge is addressed by employing an efficient query processing framework capable of early termination once it discovers that the list of *k* answers cannot be altered by processing more recommendation candidates. CRS employs both the techniques *user partitioning* and *travel penalty* to produce recommendations using *spatial ratings for spatial items*, i.e., the tuple (*user*, *ulocation*, *rating*, *item*, *ilocation*). All three cased mentioned above are thus used separately or combined together based on types of ratings available

3. PROBLEM STATEMENT

Existing system which are based on collaborative filtering and spatial information of item and user .To gain more accuracy in recommendations, locations of both item and user is subjected to play an important role. Spatial ratings are exploited to provide more accurate recommendations in addition with existing system where collaborative filter is used .Existing system consider all previous transactions to calculate the similarity between items. Instead of considering all transaction if recent transactions are used for the calculations for recommendations, both time and processing power can be saved to make more efficient system. After combination of collaborative filter and spatial ratings of items which users location also known, if enough number of recent transactions are used the recommendations provided to querying user are more accurate and with more efficient manner.

4. PROPOSED SOLUTION

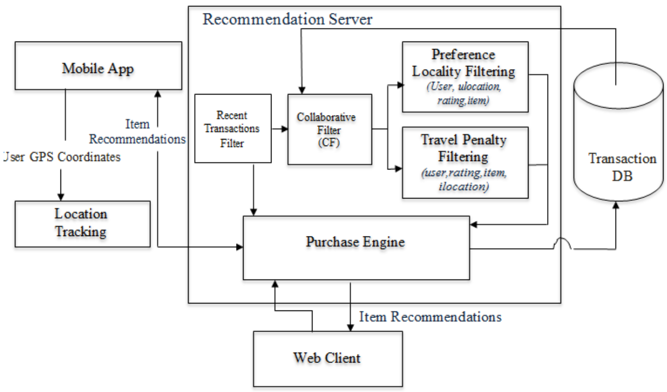

Fig 1: Architectural Design

As shown in Fig 1, there are four main modules of the architecture. Purchase engine is a shopping portal where user can login and buy a product and give his/her opinion in terms of rating value that is numeric value ranges from 1 to 5.The functionalities of these modules are as explained below.

Recommendation process: will run periodically. It involves following modules

Recent Transaction Filter: This is the module where only recent transactions are considered among huge amount of data.

Collaboration filtering: This module will prepare to identify the similar user based on the transaction they have made.

Preference Locality Filtering: This will group the users based on the user locality and based on it recommend the products.

International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 02 Issue: 03 | June-2015 www.irjet.net p-ISSN: 2395-0072

Travel Penalty Filtering: This will associate the products by penalizing the purchase based on the distance between user location and item location and use this information for product recommendation.

All the transaction will go through Preference locality based recommendation for rating and location of the user. All the transaction will go through Traversal penalty based recommendation for rating and location of the item. These will send filtered transaction from each of recommendation module. These will feed as input to the collaboration filtering, here score will be calculated for each product and ranking will be performed based on score and the K value of the product. Then *K* – recommendation will be obtained. Query processing is of two types 1) snap shot queries 2) continuous queries. Snapshot queries are given buy user who is located at place with fixed spatial coordinates; whereas continuous queries are entered by users with mobile device from which he is browsing the web portal. Continuous tracing of x and y coordinates of the location is transferred through location tracking method.

Traditional approach of recommender system by using item based collaborative filtering is not replaced with proposed system instead, added feature where spatial ratings are also considered while calculating the new highest scored recommended items. Thus the cost of installing proposed recommendation system is negligible.

The goal is to design a recommender system on product domain. The system will be composed of server side components and client side components. The server-side component will manage the database operations and algorithms that produce recommendation results. The client-side components will be graphical interfaces that are integrated into corresponding larger systems.

5. SYSTEM OVERVIEW

This section provides an overview of system by discussing the query model, collaborative filtering method and recent transaction filter.

5.1 Query Model

CRS supports two types of queries. Snap shot queries where user one time query is sent by user *U* and continuous queries [8] [9] entered by user with mobile device where continuous tracking of user's location is required. Users provide system with a user id *U*, numeric limit *K*, and location *L*; *K* recommended items are returned to the user. The technique system uses to produce recommendations depends on the type of location-based rating available in the system.

5.2 Item-Based Collaborative Filtering

Item-based collaborative filtering (abbr. CF[6]) is used as its primary recommendation technique. Reason to choose this technique is its popularity. Collaborative filter is adapted widely by commercial systems (e.g. Amazon [7]).

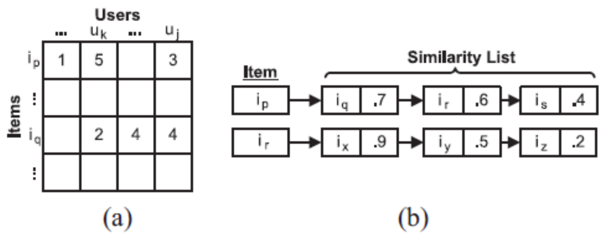

Fig. 2 Item-based CF model generation. (a) Ratings matrix. (b) Item-based CF model

The cosine method is used to calculate the similarity between items and it is given as follow:

$$sim(ip, iq) = \frac{\vec{ip}.\vec{iq}}{\|\vec{ip}\| \, \|\vec{iq}\|}$$

- This score is calculated using the vectors' co-rated dimensions, e.g., the Cosine similarity between *ip* and *iq*.
- To predict rating for not purchased product from similarity of the product purchased can be calculated by using

$$P(u, i) = \frac{\sum_{l \in L} sim(i, l) * ru, l}{\sum_{l \in L} |sim(i, l)|}$$

Given a querying user *u*, recommendations are produced by computing *u*'s predicted rating $P_{(u,i)}$ for each item *i* not rated by *u*.

Before this computation, we reduce each similarity list *L* to contain only items *rated* by user *u*. The prediction is the sum of $r_{u,l}$, a user *u*'s rating for a related item $l \in L$ weighted by *sim(i,l)*, the similarity of *l* to candidate item *i*, then normalized by the sum of similarity scores between *i* and *l*. The user receives as recommendations the top-*k* items ranked by $P_{(u,i)}$.

5.3 Query Processing

For spatial items using traveling distance i.e. travel penalty technique, query processing includes a single item based collaborative filtering model to generate top-*k* recommendations with the help of giving rank to each spatial item *I* for user *u* based on following formula

RecScore(u, i) = P(u, i) – TravelPenalty(u, i).

P(u, i) is the standard item-based CF predicted rating of item *i* for user *u*. *TravelPenalty(u, i)* is the road network travel distance between *u* and *i* normalized to the same value range as the rating scale

To calculate above equation for all candidate items to find top k recommendations can become too expensive while processing recommendations. Such computations can be avoided by evaluating items in increasing order of travel penalty in terms of Euclidian distance that helps to use early termination from top k query processing. Thus travel penalty doesn't need much processing to maintain as it only rebuild the item based collaborative filtering model whenever new ratings enter the system. For lower maintenance model can be rebuilt only after N% of new ratings are entered.

5.4 Recent Transaction Filter:

There are different types of user who do internet shopping. Depending how frequently an user is buying items and rating them, After N% of transactions time period considered for particular user varies.is user is frequent buyer, number of transactions within a month will be enough for him to get recommendations. However if user doesn't have much transaction history, an year old transactions are to be considered to fulfill the requirement of N% of transaction.
The influence level of time factor can also be given by user depending on his search criteria.

6. ADVANTAGES OF OUR SOLUTION

CRS addresses the main flaw of LARS by improving the pyramid data structure by adding one more type of cell that help to maintain balance between locality and scalability. Added feature of considering recent transactions for calculation, CRS produces more accurate recommendations by limiting the processing power. To gain more accurate recommendations with less processing of recent transactions, CRS is cognizable system to provide more accurate recommendations in efficient and scalable manner.
Scalability: The maintenance algorithm in CRS avoids the expensive speculative splitting .This feature helps to increase scalability of the system. Accessing only recent transactions for processing data also helps to gain scalability.
Efficiency: CRS adapts efficient way to calculate recommendations by accessing recent transactions and pyramid data structure where maintenance algorithm is used to avoid excessive amount of processing which is economically better than the existing system.

7. RELATED WORK

Location-based services. Current location-based services employ two main methods to provide interesting destinations to users. (1) KNN techniques [11] and variants simply retrieve the *k* objects nearest to a user and are completely removed from any notion of user *personalization*. (2) Preference methods such as sky-lines [12]. Conversely, *CRS* is the first location-based service to consider *implicit* preferences by using location-based ratings to help users discover new items.

Traditional recommenders: A wide array of techniques is capable of producing recommendations using non-spatial ratings for non-spatial items represented as the triple (*user*, *rating*, *item*) We refer to these as "traditional" recommendation techniques. The locations in considered in the closest approach incorporating contextual attributes into statistical recommendation models 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 [13] displays a "local favorites" list containing popular movies for a user's given city. However, these movies are *not* personalized to each user; rather, this list is built using aggregate rental data for a particular city. CRS on the other hand, produces personalized recommendations influenced by location-based ratings and a query location.

Location-aware recommenders: The CityVoyager system [14] mines a user's personal GPS trajectory data to determine her preferred shopping sites, and provides recommendation based on where the system predicts the user is likely to go in the future. *CRS*, conversely, does not attempt to predict future user movement, as it produces recommendations influenced by user and/or item locations embedded in community ratings

8. CONCLUSION AND FUTURE WORK

Cognizable Recommender, is location-aware recommender system, covers a problem which is never considered by traditional recommender systems by working 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*. Cognizable Recommender employs tow important techniques *user partitioning* and *travel penalty* to support spatial ratings and spatial items, respectively. These two techniques can be applied separately or in combination to support the various types of location-based ratings. Experimental analysis using real and synthetic data sets show that Cognizable Recommender is scalable, efficient and provides better quality recommendations with highest recommendation score than just an item based collaborative filtering technique used in traditional recommender systems.

Future work may include the use recent transactions for furthermore simplification in calculations to get even better results. This can be achieved by using one more attributes i.e. time in the form of temporal database. Influence level can be edited by admin depending on the fact that how frequent the user is purchasing and rating the product.

ACKNOWLEDGEMENT

The authors are grateful to the editor and reviewers for their thoughtful comments that played important role to improve the quality of this paper.

REFERENCES

[1] 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
[2] The facebook blog. *Facebook Places* [Online]. Available: http://tinyurl.com/3aetfs3
[3] J. J. Levandoski, M. Sarwat, A. Eldawy, and M. F. Mokbel, "LARS: A location-aware recommender system," in *Proc. ICDE*, Washington, DC, USA, 2012.
[4] Mohamed Sarwat, Justin J. Levandoski, Ahmed Eldawy, and Mohamed F. Mokbel "LARS*: An Efficient and Scalable Location-Aware Recommender System"IEEE Trans. Volume: 26, Issue: 6
[5] W. G. Aref and H. Samet, "Efficient processing of window queries in the pyramid data structure," in *Proc. ACM Symp. PODS*, New York, NY, USA, 1990.
[6] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collab-orative filtering recommendation algorithms," in *Proc. Int. Conf. WWW*, Hong Kong, China, 2001.
[7] G. Linden, B. Smith, and J. York, "Amazon.com recommenda-tions: Item-to-item collaborative filtering," *IEEE Internet Comput.*, vol. 7, no. 1, pp. 76–80, Jan./Feb. 2003
[8] K. Mouratidis, S. Bakiras, and D. Papadias, "Continuous monitor-ing of spatial queries in wireless broadcast environments," *IEEE Trans. Mobile Comput.*, vol. 8, no. 10, pp. 1297–1311, Oct. 2009
[9] K. Mouratidis, S. Bakiras, and D. Papadias, "Continuous monitor-ing of spatial queries in wireless broadcast environments," *IEEE Trans. Mobile Comput.*, vol. 8, no. 10, pp. 1297–1311, Oct. 2009.
[10] *Foursquare* [Online]. Available: http://foursquare.com
[11] J. Bao, C.-Y. Chow, M. F. Mokbel, and W.-S. Ku, "Efficient evalu-ation of k-range nearest neighbor queries in road networks," in *Proc. Int. Conf. MDM*, Kansas City, MO, USA, 2010
[12] M. Sharifzadeh and C. Shahabi, "The spatial skyline queries," in *Proc. Int. Conf. VLDB*, Seoul, Korea, 2006.
[13] *New York Times - A Peek into Netflix Queues* [Online]. Available: http://www.nytimes.com/interactive/2010/01/10/nyregion/ 20100110-netflix-map.html
Y. Takeuchi and M. Sugimoto, "An outdoor recommendation system based on user location history," in *Proc. Int. Conf. UIC*, Berlin,Germany, 2006

BIOGRAPHIES

Mrs. Shilpa Mathapati has completed B.E in Computer Science at Dr BAMU Maharashtra, India in 2003. She is currently pursuing M.Tech in Computer Science from Visvesvaraya Technological University, India. She is interested with areas of research related to Data mining and Mobile computing.

Mrs. Bhuvaneswari Raju has completed B.E (CSE) from Bharathidasan University and M.Tech (CSE) Visvesvaraya Technological University. She is currently working with Atria Institute of Technology, Bangalore as Asst Professor. She is interested in Database Management System, Computer Graphics and Computer Networks.