ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 7, July 2015

An Novel Location Aware Spatial Data Recommender Using MobiFeeds

P.Alekya, Bullarao Domathoti, P.Nageswara Rao

M.Tech Student, Dept. of CSE., SITS, JNT University, Anantapuram, AP, India

Assistant Professor, Dept. of CSE., SITS, JNT University, Anantapuram, AP, India

Associate Professor, Dept. of CSE., SITS, JNT University, Anantapuram, AP, India

ABSTRACT: A location-aware news feed system enables mobile users to share geo-tagged user-generated messages, Traditional recommender systems do not consider spatial properties of users nor items; LARS*, on the other hand, 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.We present a framework designed for scheduling news feeds for mobile users. Mobi Feed consists of three key functions, location prediction, relevance measure, and news feed scheduler. The location prediction function is designed to predict a mobile user's locations based on an existing path prediction algorithm User rating locations are exploited by user partitioning technique which influences recommendations with ratings close to the querying user spatially, without diminishing the system scalability and recommendation quality. Item locations are exploited by using travel penalty technique which favors recommendations which are close in travel distance to the querying user by avoiding exhaustive access to all items.

KEYWORDS: Spatial Databases and GIS;LARS; Location; Scalable**;**

I. INTRODUCTION

A news feed is a common functionality of existing location aware social network systems. It enables mobile users to post geo-tagged messages and receive nearby user-generated messages, e.g., "Alice can receive 4 messages that are the most relevant to her among the messages within 1 km from her location every 10 seconds". Since a location-aware social network system usually possesses a huge number of messages, there are many messages in a querying user's vicinity. Coupled with user mobility, a key challenge for the location aware news feed system is how to efficiently schedule the kmost relevant messages for a user and display them on the user's mobile device. Although location-aware news feed and social network systems have attracted a lot of attention from different research communities, none of these applications has focused on how to schedule news feeds for mobile users. The state-of-the-art research prototype of a location-aware news feed system is GeoFeed [2]. In contrast to GeoFeed, Mobi Feed focuses on challenges in providing location-aware news feeds for mobile users. We design a location-aware news feed scheduler that works with our location prediction and message relevance measure functions to provide news feeds for mobile users.

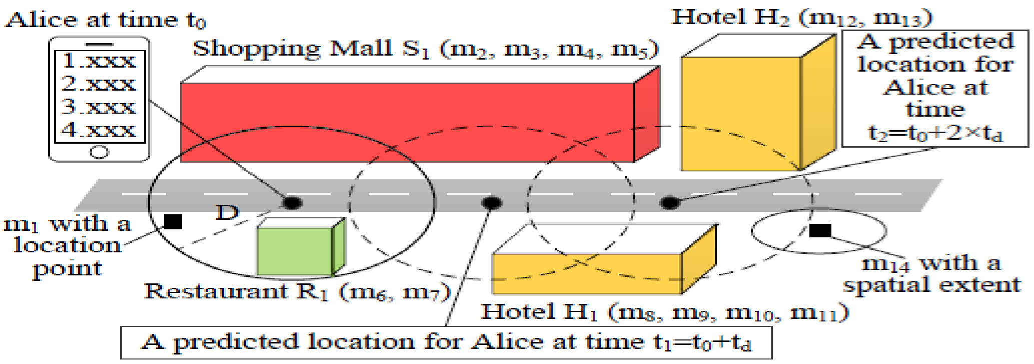
Alice at time t_0
1.xxx
2.xxx
3.xxx
4.xxx
Shopping Mall S_1 (m_2, m_3, m_4, m_5)
Hotel H_2 (m_{12}, m_{13})
A predicted location for Alice at time t_2=t_0+2×t_d
D
m_1 with a location point
Restaurant R_1 (m_6, m_7)
Hotel H_1 (m_8, m_9, m_{10}, m_{11})
m_{14} with a spatial extent
A predicted location for Alice at time t_1=t_0+t_d

Figure 1: Location-aware news feed scheduling

In this paper, we present MobiFeed that is a locationaware ews feed framework designed for social network systems to schedule news feeds for mobile users. Figure 1 depicts an application scenario. A MobiFeed user, Alice, can generate a message and tag a point (e.g., m1), a spatial extent (e.g., m14 is associated with a circular spatial area), or a venue (e.g., m6 and m7 are spatially associated with restaurant R1) as its geo-location. Alice can also issue a location-aware news feed query to retrieve the k most relevant messages within her specified range distance D from her location. MobiFeed consists of three key functions: location prediction, relevance measure, and news feed scheduler. Given a user u's location u:location at the current time t0, u's required minimum message display time td, u's specified range distance D, u's requested number of messages per news feed, and a look-ahead steps n, the location prediction function estimates n future locations for u at times t1 = t0 + td, t2 = t0 + 2 × td, . . . , and tn = t0 + n × td, the relevance measure function calculates the relevance score of each candidate message with a geo-location intersecting any u's query region (i.e., a circular area centered at u:location or a predicted location with a radius D), and the news feed scheduler generates news feeds from the candidate messages for u's query regions at t0, t1….. tn with the best total relevance score.

a) Existing system:

Recommender systems make use of community opinions to help users identify useful items from a considerably large search space. The technique used by many of these systems is collaborative filtering (CF), 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. Myriad applications can produce location-based ratings that embed user and/or item locations. Existing recommendation techniques assume ratings are represented by the (user, rating, item) triple.

- The existing systems are ill-equipped to produce location aware recommendations.
- The existing system provides more expensive operations to maintain the user partitioning structure.
- The existing system does not provide spatial ratings.
-

b) *Proposed system*:

We have proposed LARS*, a location-aware recommender system that uses location-based ratings to produce recommendations. LARS*, 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. Within LARS*, we propose:

(i) A user partitioning technique that exploits user locations in a way that maximizes system scalability while not sacrificing recommendation locality

(ii) A travel penalty technique that exploits item locations and avoids exhaustively processing all spatial recommendation candidates.

- LARS*, 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* achieves higher locality gain using a better user partitioning data structure and algorithm.
- LARS* exhibits a more flexible tradeoff between locality and scalability.
- LARS* provides a more efficient way to maintain the user partitioning structure

II. LITERATURE SURVEY

a. *Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions*:

G. Adomavicius and A. Tuzhilin presents an overview of the field of recommender systems and describes the current generation of recommendation methods that are usually classified into the following three main categories: content-based, collaborative, and hybrid recommendation approaches. This paper also describes various limitations of current recommendation methods and discusses possible extensions that can improve recommendation capabilities and make recommender systems applicable to an even broader range of applications. These extensions include, among others, an improvement of understanding of users and items, incorporation of the contextual information into the recommendation process, support for multcriteria ratings, and a provision of more flexible and less intrusive types of recommendations.

b. LARS: A location-aware recommender system

J. J. Levandoski, M. Sarwat, A. Eldawy, and M. F. Mokbel authors proposes LARS, a location-aware recommender system that uses location-based ratings to produce recommendations. Traditional recommender systems do not consider spatial properties of users nor items, LARS, on the other hand, supports 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 in concert, 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.

c. Empirical analysis of predictive algorithms for collaborative filtering:

J. S. Breese, D. Heckerman, and C. Kadie Collaborative filtering or recommender systems use a database about user preferences to predict additional topics or products a new user might like. In this paper we describe several algorithms designed for this task, including techniques based on correlation coefficients, vector-based similarity calculations, and statistical Bayesian methods. We compare the predictive accuracy of the various methods in a set of representative problem domains. We use two basic classes of evaluation metrics. The first characterizes accuracy over a set of individual predictions in terms of average absolute deviation. The second estimates the utility of a ranked list of suggested items. This metric uses an estimate of the probability that a user will see a recommendation in an ordered list. Experiments were run for datasets associated with 3 application areas, 4 experimental protocols, and the 2 evaluation metrics for the various algorithms. Results indicate that for a wide range of conditions, Bayesian networks with decision trees at each node and correlation methods outperform Bayesian-clustering and vector-similarity methods. Between correlation and Bayesian networks, the preferred method depends on the nature of the dataset, nature of the application (ranked versus one-by-one presentation), and the availability of votes with which to make predictions. Other considerations include the size of database, speed of predictions, and learning time.

III. IMPLEMENTATION:

Spatial ratings for non-spatial items:

This section describes how LARS* produces recommendations using spatial ratings for non-spatial items represented by the tuple (user, ulocation, rating, item). The idea is to exploit preference locality, i.e., the observation that user opin-ions 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 spatially 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 influences their recommendations.

Non-spatial ratings for spatial items:

This section describes 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. 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 with-out calculating the travel distance to all items.

Spatial ratings for spatial items:

This section describes how LARS* produces recommendations using spatial ratings for spatial items represented by the tuple (user, ulocation, rating, item, ilocation). A salient feature of LARS* is that both the user partitioning and travel penalty techniques can be used together with very little change to produce recommendations using spatial user ratings for spatial items. The data structures and maintenance techniques remain exactly the same as discussed in Sections 4and5; only the query processing frame-work requires a slight modification. Query processing uses Algorithm2 to produce recommendations. However, the only difference is that the item-based collaborative filtering prediction score P(u,i) used in the recommendation score calculation (Line16in Algorithm 2) is generated using the (localized) collaborative filtering model from the partial pyramid cell that contains the querying user, instead of the system-wide collaborative filtering model as was used.

Scalable Incremental Processing of continuous Queries in Spatiotemporal Databases:

The Scalable Incremental Algorithm (SINA) which is hash based algorithm for evaluating a set of continuous concurrent spatiotemporal queries. SINA is designed with two goals i) Scalability with respect to number of continuous concurrent spatiotemporal queries. ii) The other is incremental evaluation of three queries. This algorithm achieves scalability by using shared execution paradigm where executions of spatiotemporal queries which are continuous are abstracted as spatial join with set of objects which are moving. We get incremental evaluation by computing the updates of the answers which were previously reported. Computation of different updates like positive and negative updates were introduced in the paper, when certain object should be added or removed from the answers which were reported previously indicates the positive or negative updates respectively. updates are managed in the algorithm in three phases, the hashing phase, invalidation phase and joining phase. We use an in memory hash-based join algorithm which gives the set of positive updates. The second phase is involved when memory is fully occupied to give set of negative updates or it is triggered every t seconds. At the end of this phase the joining phase is triggered which produces set of both positive as well as negative updates. Comparing with previous results experimentally proves that this algorithm is scalable and efficient as compared to other spatiotemporal algorithm.

IV. RESEARCH WORK OF DESIGN

The multi-language capability of the .NET Framework and Visual Studio .NET enables developers to use their existing programming skills to build all types of applications and XML Web services. The .NET framework supports new versions of Microsoft's old favorites Visual Basic and C++ (as VB.NET and Managed C++), but there are also a number of new additions to the family.The link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

i. Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

ii. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

iii. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

DOI: 10.15680/ijircce.2015. 0307022

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

i. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.
ii. Select methods for presenting information.
iii. Create document, report, or other formats that contain information produced by the system.The output form of an information system should accomplish one or more of the following objectives.Convey information about past activities, current status or projections of theFuture.
iv. Signal important events, opportunities, problems, or warnings.
v. Trigger an action.
vi. Confirm an action.

V. Conclusion

The proposed system We presented MobiFeed that is a location-aware news feed framework designed for scheduling news feeds for mobile users. We described the three key functions of MobiFeed, namely, *location prediction*, *relevance measure*, and *news feed scheduler*. The *location prediction* function is designed to estimate a user's location based on the path prediction algorithm. LARS uses two techniques one is user partitioning and the other is travel penalty which supports spatial ratings and spatial items respectively. Experimenting with real and synthetic datasets show that this recommender system is efficient and scalable which gives the better quality of recommendation as compared to the approaches which were used by traditional recommender systems.

REFERENCES

1. G. Linden et al, "Amazon.com Recommendations: Item-to-Item Collaborative Filtering," *IEEE Internet Computing*, vol. 7, no. 1, pp. 76–80, 2003.
2. P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: An Open Architecture for Collaborative Filtering of Netnews," In *CSWC,* 1994.
3. "The Facebook Blog,"Facebook Places" 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 Transactions on Knowledge and Data Engineering, TKDE,* vol. 17, no. 6, pp. 734–749, 2005
5. "MovieLens *Available*: http://www.movielens.org/."
6. "Foursquare *Available*: http://foursquare.com."
7. New York Times – A Peek into Netflix Available: http://www.nytimes.com/interactive/2010/01/10/ nyregion/20100110- netflix-map.html."
8. J. J. Levandoski, M. Sarwat, A. Eldawy, and M. F. Mokbel, "LARS: A Location-Aware Recommender System," in *Proceedings of the International Conference on Data Engineering, ICDE,* 2012.
9. B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-Based Collaborative Filtering Recommendation Algorithms," in *Proceedings of the International World Wide Web Conference, WWW,* 2001.
10. J. S. Breese, D. Heckerman, and C. Kadie, "Empirical Analysis of Predictive Algorithms for Collaborative Filtering," in *Proceedings of the Conference on Uncertainty in Artificial Intelligence, UAI,* 1998
11. M. F. Mokbel, X. Xiong, and W. G. Aref, "SINA: Scalable incremental processing of continuous queries in spatiotemporal databases," in *Proc. SIGMOD*, Paris, France, 2004.
12. G. K. Gupta, *Introduction to Data Mining with Case Studies,* 2nd ed, PHI Learning Pvt Ltd 2013.