ILARS: An Improved Empirical Analysis for Lars* Using Partitioning and Travel Penalty

Mrs. Ashwini.K.Bhavsar
[1]PG Student ,Department of Computer Engineering,
Alard College of engineering and management
Savitribai Phule Pune University, India
ashwini.k.bhavsar@gmail.com

Mrs. Sonali.Patil
[2]Professor, Department of Computer Engineering
Alard College of engineering and management
Savitribai Phule Pune University, India
sonalin69@gmail.com

Abstract:- In this paper we develop an improved web based location-aware recommender software system, ILARS, that uses location-based ratings to provide proper advice and counseling. Present recommender systems don't consider about spatial attributes of users and also of items; But, ILARS*considers major classes regarding location such as spatial scores rate for the non-spatial things, non-spatial score rate for the spatial things, and spatial score rate for the spatial things. ILARS* deals with recommendation points for accomplishing user ranking locations with help of user partitioning methods, which that are spatially near querying users in an effective way that maximizes system computability by not reducing the systems quality. A style that supports recommendation successors nearer in travel distance to querying users is used by ILARS* to exploits item locations using travel penalty. For avoiding thorough access to any or all spatial things. ILARS* will apply these art singly, or based on the rating that is obtained. The experimental results show information from various location based social networks. Various social network tells that LARS* is magnified , most expanded ,inexpensive ,reasonable ,capable of showing recommendations which are accurate as compared to existing recommendation software systems.

Keywords- *Recommendation system rule, spatial location, highlighted execution, ability, high availability, community*.

*****

I. Introduction

What is a recommendation system?

A software which web based that predicts user responses depended upon their past history of likes and dislikes related to information or products etc.

Examples of recommendation systems are:

1. Online recommendations for movies based on reviews and interests.

2. Online customer search for home made appliances based on previous likes and dislikes.

Different types of recommendation system are present, but the main two categories are:

1. **Content-based filtering systems.**
2. **Collaborative filtering systems(CF)**
   a. Memory Based
   b. Model Based
   c. Hybrid
3. **Hybrid Recommender Systems**

A method that filters and gives result upon evaluation and past history of agreed similarities of others to recommend the future results. The recommendations of some people who have same interests are trusted much more than the recommendations from other people based on which decision is made.

We see an example utility matrix, representing users' ratings of movies on a 1–5 scale, with 5 the highest rating. Null represent the situation where the user has not given any rate to the movie. For eg. Little Krishna part 1,2,3 movie as LK1, LK2, LK3 for Jungle Book part1, part2 and part3 episodes as JB1,JB2 and JB3. The users are denoted in the first column. [Figure-1] is an example of a utility matrix representing ratings of movies on a 1–5 scale.

	LK1	LK2	LK3	JB1	JB2	JB3
ASHA	4			5	1	
BEENA	5	5	4			3
CRISTINA				2	4	5
DYNA		3				

Figure: 1. A multi-attribute search form interface.

Existing recommendation techniques assume ratings are delineate by the (user, rating, item) triple, so are unequipped to supply location aware recommendations. In this paper, we have a tendency to propose LARS*, a unique location aware recommender system designed specifically to supply high-quality location-based recommendations in associate economical manner. LARS* produces recommendations employing a taxonomy of 3 varieties of location-based ratings at intervals a single framework: Traditional rating triples are often classified as non-spatial ratings for non-spatial things and don´t work this taxonomy. This project includes three types of location-based ratings within single framework.

1.1 Spatial Ratings for Non-Spatial Items

Consists of four fields i.e. user, uloc, item, rating, where uloc is denoting the user location from he is rating the item. For example, a person at office rating a electronic device.

1.2 Non-Spatial Ratings for Spatial Items

Consists of four fields i.e. user, item, iloc, rating, where iloc is denoting an item location. Item is a device or a landmark which is not movable. For example ,a user with location hidden is rating a building construction.

1.3 Spatial Ratings for Spatial Items

The combination of above both i.e. the five fields such as user, uloc, rating, item, iloc. A technique that is partitioning is used to divide user locations for increasing scalability. And a technique which is recommending travel distance is travel penalty which processes all spatial recommendation candidates. Thus, ILARS, Improved Location aware recommender system achieves higher improvement in locality gain than previous location systems. It is more flexible and efficient for large systems.

Movie Recommendations:

Friends do suggest for movies and based on their reviews and score rating recommendation is made. Similarly Netflix, is one of the most popular system that even won prize for its top and popular suggestion. It is a system that gives customers likes and dislikes which help the other remaining customers. Bellkor's Pragmatic Chaos team won prize in dollars after long waiting period of three years.

What is a cluster centroid?

Cluster is a group and its center or middle part is said to be the a centroid .Cluster values are the mean related to the centroid, which is a vector as one number assigned to each variable value. As a part of measurement the centroid can be used with its minimum distance as minimum and maximum distance as maximum.

What is a spatial data?

The exact longitude and latitude location that is used to identify the location coordinates is the spatial data /information.For example the location of a tower, building, river, mountain etc. This can also be called as geospatial data or geographic information which is analyzed as Geographic Information Systems (GIS).

What is a data?

Data is raw material which when grouped together gives meaningful information .Data can be images, shapes, text, audio, video etc.

II. RELATED WORK

J. J. Levandoski, M. Sarwat, A. Eldawy, and M. F. Mokbel. This paper 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 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.

B. Sarwar, G. Karypis, J. Konstan, and J. Riedl Recommender systems apply knowledge discovery techniques to the problem of making personalized recommendations for information, products or services during a live interaction. These systems, especially the k-nearest neighbor collaborative filtering based ones, are achieving widespread success on the Web. The tremendous growth in the amount of available information and the number of visitors to Web sites in recent years poses some key challenges for recommender systems. These are: producing high quality recommendations, performing many recommendations per second for millions of users and items and achieving high coverage in the face of data sparsity. In traditional collaborative filtering systems the amount of work increases with the number of participants in the system. New recommender system technologies are needed that can quickly produce high quality recommendations, even for very large-scale problems. To address these issues we have explored item-based collaborative filtering techniques. Item-based techniques first analyze the user-item matrix to identify relationships between different items, and then use these relationships to indirectly compute recommendations for users.

In this paper we analyze different item-based recommendation generation algorithms. We look into different techniques for computing item- item similarities (e.g., item- item correlation vs. cosine similarities between item vectors) and different techniques for obtaining recommendations from them (e.g.,

 ISSN: 2321-8169
4535 - 4541

weighted sum vs. regression model). Finally, we experimentally evaluate our results and compare them to the basic k- nearest neighbor approach. Our experiments suggest that item- based algorithms provide dramatically better performance than user-based algorithms, while at the same time providing better quality than the best available user-based algorithms.

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 a teach node and correlation methods out perform 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.

W. G. Aref and H. Samet Window operations serve as the basis of a number of queries that can be posed in a spatial database. Examples of these window-based queries include the exist query (i.e., determining whether or not a spatial feature exists inside a window) and the report query, (i.e., reporting the identity of all the features that exist inside a window). Algorithms are described for answering window queries in O(gr);(n log T) time for a window of size n x n in a feature space (e.g., an image) of size (T x T)(e.g., pixel elements). The significance of this result is that even though the window contains n2 pixel elements, the worst-case time complexity of the algorithms is almost linearly proportional (and not quadratic) to the window diameter, and does not depend on other factors. The above complexity bounds are achieved via the introduction of the incomplete pyramid data structure (a variant of the pyramid data structure) as the underlying representation to store spatial features and to answer queries on them.

R. A. Finkel and J. L. Bentley The quad tree is a data structure appropriate for storing information to be retrieved on composite keys. We discuss the specific case of two-dimensional retrieval, although the structure is easily generalized to arbitrary dimensions. Algorithms are given both for straightforward insertion and for a type of balanced insertion into quad trees. Empirical analyses show that the average time for insertion is logarithmic with the tree size. An algorithm for retrieval within regions is presented along with data from empirical studies which imply that searching is reasonably efficient. We define an optimized tree and present an algorithm to accomplish optimization in n log n time. Searching is guaranteed to be fast in optimized trees. Remaining problems include those of deletion from quad trees and merging of quad trees, which seem to be inherently difficult operations.

A. Guttman In order to handle spatial data efficiently, as required in computer aided design and geo-data applications, a database system needs an index mechanism that will help it retrieve data items quickly according to their spatial locations However, traditional indexing methods are not well suited to data objects of non-zero size located m multi-dimensional spaces In this paper we describe a dynamic index structure called an R-tree which meets this need, and give algorithms for searching and updating it. We present the results of a series of tests which indicate that the structure performs well, and conclude that it is useful for current database systems in spatial applications.

K. Mouratidis, S. Bakiras, and D. Papadias Wireless data broadcast is a promising technique for information dissemination that leverages the computational capabilities of the mobile devices in order to enhance the scalability of the system. Under this environment, the data are continuously broadcast by the server, interleaved with some indexing information for query processing. Clients may then tune in the broadcast channel and process their queries locally without contacting the server. Previous work on spatial query processing for wireless broadcast systems has only considered snapshot queries over static data. In this paper, we propose an air indexing framework that 1) outperforms the existing (i.e., snapshot) techniques in terms of energy consumption while achieving low access latency and 2) constitutes the first method supporting efficient processing of continuous spatial queries over moving objects.

III. PROPOSED APPROACH FRAMEWORK AND DESIGN

A. Problem Defination

Existing recommendation techniques assume ratings are delineate by the (user, rating, item) triple, so are unequipped to supply location aware recommendations. In this paper, we have

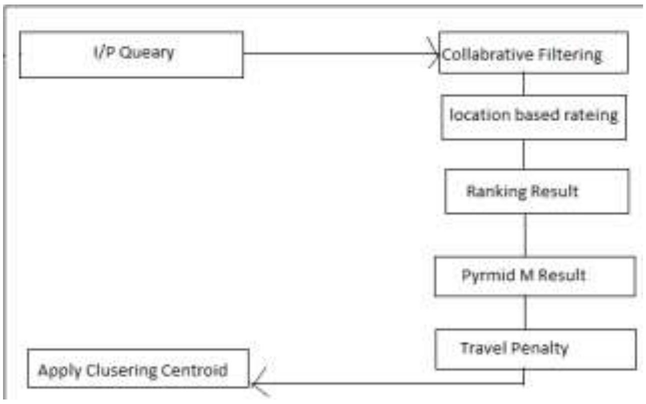
 ISSN: 2321-8169
4535 - 4541

a tendency to propose LARS*, a unique location aware recommender system designed specifically to supply high-quality location-based recommendations in associate economical manner. LARS* produces recommendations employing a taxonomy of 3 varieties of location-based ratings at intervals a single framework: Traditional rating triples are often classified as non-spatial ratings for non-spatial things and don´t work this taxonomy. This project includes three types of location-based ratings within single framework.

Goals and objectives

To design a web based a location-aware recommender system that uses location-based ratings to produce recommendations exploiting user rating locations through user partitioning and using travel penalty.

B. System Architechture



C. Mathematical Equations

Our Mathematical Model contribution can be summarized as follows:

Model Based Collaborative Filtering

The first step is model building based on the past history of user predictions and ratings on item. Model is build from the data which is present for the purpose to provide recommendation based on similarity.

Item-Based Collaborative Filtering

A: Model Building.

Every phase calculates the similarity ,sim(ip,iq) for each set of objects, ip and iq that have got at least one common rating by the same user.
Where R ,U are the number of ratings and users respectively. List L , only the most similar items with the highest similarity score.

B: Recommendation Generation.

For the given ,User = u, Items = i,then predicted rating = P(u,i) is calculated as

$$\mathbf{P(u,i)= \sum l\ \varepsilon\ L\ sim(i,l) * ru,l\ \div \sum l\ \varepsilon\ L\ |sim(i,l)|.}$$

similarity list = L ,the summation of all similarities from list L is gained from ratings from user for the item i ε L. Then top-k items ranked are recommended by P(u,i) to the user.

Computing Similarity.

Computing similarity, sim(ip, iq),for various items, we represent it in form of rating matrix. Many similarity functions have been proposed (e.g., Pearson Correlation, Cosine); we use the Cosine similarity in LARS* due to its popularity.

Data Structure Maintenance

ILARS* is more improved as compared to previous LARS because of data structure to maintain the shape of the adaptive pyramid with three goals of locality, scalability, and influence. Initially, to build the pyramid, all location-based ratings present currently build a complete pyramid of height H, such that all cells in all H levels are α-Cells and contain ratings statistics and a collaborative filtering model.
The algorithm takes as input a pyramid cell C and level h, and includes three main steps:
1.Maintaining the Items Ratings Statistics Table for making decision when new location are rated.
2.The second step is to rebuild the item-based collaborative filtering (CF) model for a cell C.
3.For calculating the tradeoff a maintenance step may be carried out on cell C child quadrant as it is needed to be switched to a different cell type.

Major constraints:
Preference policy :-
Authorization(with one of two modes: either positive or negative) is determined by the system installer at configuration time. This policy determines which authorization wins when both positive and negative authorization (or neither negative nor positive authorization) can be derived for a particular subject. Negative authorization is preferred(known as closed policy) in more restricted systems such as military; positive authorization may be preferred in more open application such as public information system. For example,(User at Delhi receiving recommendation with best result for Mumbai) thus do not consider travel locality.

Locality policy :-
The common mode of this distance – based policy states that most specific authorization takes preference. It applies to distributed organizations whose local branches may recognize

ISSN: 2321-8169
4535 - 4541

an exception to a general rule. For instance, a department in university may admit an outstanding applicant although a general admission requirement is not completely met. Thus,for a given subject , when both positive and negative authorizations can be derived from different ancestors, the one that is closer to the subject wins.

Pyramid data structure

LARS* employs a partial in memory pyramid structure. To provide a tradeoff between recommendation locality and system scalability, the pyramid data structure maintains three types of cells:

Recommendation Model Cell (α-Cell).

Each α-Cell the root cell (level 0) of the pyramid is an α-Cell and represents a “traditional" (i.e., non-spatial) item-based collaborative filtering model which stores an item-based collaborative filtering model built using only the spatial ratings with user locations contained in the cell’s spatial region.

Statistics Cell (β-Cell).

β-Cell maintains statistics (i.e., items ratings Statistics Table) about the user/item ratings that are located within the spatial range.

Empty Cell (γ -Cell).

A γ -Cell is a cell that maintains neither the statistics nor the recommendation model for the ratings lying within its boundaries. a γ -Cell cannot have any children.

Top recommendations by ranking each spatial item i for a querying user u based on RecScore(u, i), is computed as:

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

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

D. Algorithms

In this paper we present a framework and an efficient Predicting Future Locations Using Clusters' Centroids.

Function for LARS*_Spatial Items clustering(User U, Location L, Limit K):

```
{
for all items
do
{
SELECT all list in DESC order;
Get Result
Based on user ratings and threshold
If {
Get uirating between >10 &&  <= 14) ,then
set as  0.8
Else if Get uirating between >15 ,then
set as  1.0
Else if Get uirating between >10 && < 24,then
set as  0.9
}
```

Function for Pyramid Maintenance (Cell C, Level h) contains following steps:-

Step I: Maintaining the statistics
Step II: Rebuilding of the model based on similarity
Step III: Cell Child Quadrant Maintenance

Algorithm for Improved LARS*:

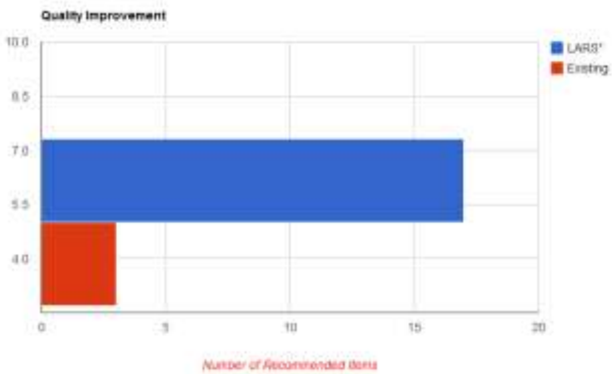
```
/* Called after cell C receives N% new ratings */
Function Pyramid Maintenance (Cell C, Level h)
 Maintain cell C statistics
if (Cell C is an α-Cell) then
Rebuild item-based collaborative filtering model for cell C
end if
if (C children quadrant q cells are α-Cells) then
CheckDownGradeToSCells(q,C)
else if (C children quadrant q cells are γ -Cells) then
CheckUpGradeToSCells(q,C)
else
isSwitchedToMcells ← CheckUpGradeToMCells(q,C)
if (isSwitchedToMcells is False) then
CheckDownGradeToECells(q,C)
end if
end if
return
Function LARS*_SpatialItems(User U, Location L, Limit K)
/* Construct a list R with a set of K items*/
R ← φ
for (K iterations) do
i ← Retrieve the item with the next lowest travel penalty
Insert i into R ordered by RecScore(U, i) computed
end for
LowestRecScore ← RecScore of the kth object in R
/*Retrieve items one by one in order of their penalty value */
while there are more items to process do
i ← Retrieve the next item in order of penalty score
MaxPossibleScore ← MAX_RATING - i.penalty
if MaxPossibleScore ≤ LowestRecScore then
return R /* early termination - end query processing */
```

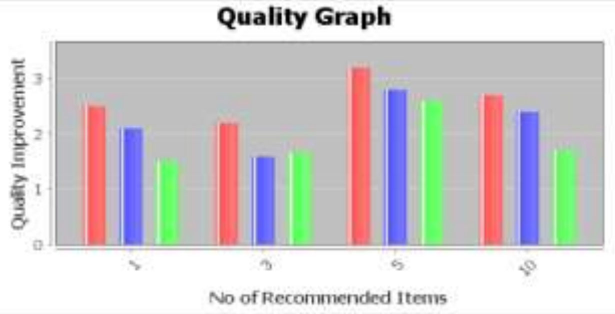
end if
RecScore(U, i) ← P(U, i) - i.penalty
if RecScore(U, i) > LowestRecScore then
Insert i into R ordered by RecScore(U, i)
LowestRecScore ← RecScore of the kth object in R
end if
end while
return R

IV. EXPERIMENTAL RESULT

A. Snapshots

B. Graphs





V. CONCLUSION

LARS*, our projected location-aware recommender sys- tem, tackles a retardant untouched by ancient recommender systemsbyaddressing3kindsoflocation-basedratings:spatial ratings for non-spatial things, non-spatial ratings for spatial things, and spatial ratings for spatial things. LARS* employs user partitioning and travel penalty techniques to support spatial ratings and spatial things, severally. Each technique is applied individually or in concert to support the varied kinds of location-based ratings. Experimental analysis victimization real and artificial knowledge sets show that LARS* is economical, scalable, and provides better quality recommendations than techniques employed in traditional recommender systems.

ACKNOWLEDGMENT

It is with the greatest pleasure and pride that I present this paper. At this moment, I cannot neglect all those who helped me in the successful completion of this paper. I am very thankful to my respected project guide Prof. Sonali Patil, Associate Professor, for her ideas and help proved to be valuable and helpful during the creation of this paper and guide me in the right path. I would also like to thank all the faculties who have cleared all the major concepts that were involved in the understanding of techniques behind this paper. Lastly, I am thankful to my friends who shared their knowledge in this field with me.

REFERENCES

[1] Michael Bergman, “The deep Web: surfacing hidden value”. In the Journal Of Electronic Publishing 7(1) (2001).

[2] 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.

[3] P.Resnick,N.Iacovou,M.Suchak,P.Bergstrom,andJ.Riedl, “GroupLens: An open architecture for collaborative filtering of netnews,” inProc. CSWC, Chapel Hill, NC, USA, 1994.

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

[5] 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.

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

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

[8] New York Times - A Peek into Netflix Queues[Online]. Available:http://www.nytimes.com/interactive/2010/01/10/nyregion/20100110-netflix-map.html

[9] J. J. Levandoski, M. Sarwat, A. Eldawy, and M. F. Mokbel,“LARS: A location-aware recommender system,” inProc. ICDE,Washington, DC, USA, 2012.

[10] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, “Item-based collaborative filtering recommendation algorithms,” inProc. Int. Conf.WWW, Hong Kong, China, 2001.

[11] J. S. Breese, D. Heckerman, and C. Kadie, “Empirical analysisof predictive algorithms for collaborative filtering,” inProc. Conf.UAI, San Francisco, CA, USA, 1998.

[12] W. G. Aref and H. Samet, “Efficient processing of window queriesin the pyramid data structure,” in Proc. ACM Symp. PODS, New York, NY, USA, 1990.

[13] R. A. Finkel and J. L. Bentley, “Quad trees: A data structure for retrieval on composite keys,”Acta Inf., vol. 4, no. 1, pp. 1–9, 1974.

[14] A. Guttman, “R-trees: A dynamic index structure for spatial searching,” inProc. SIGMOD, New York, NY, USA, 1984.

[15] K. Mouratidis, S. Bakiras, and D. Papadias, “Continuous monitoring of spatial queries in wireless broadcast environments,” IEEE Trans. Mobile Comput., vol. 8, no. 10, pp. 1297–1311, Oct. 2009.

[16] K. Mouratidis and D. Papadias, “Continuous nearest neighborqueries over sliding windows,”IEEE Trans. Knowl. Data Eng.,vol. 19, no. 6, pp. 789–803, Jun. 2007.

[17] M. F. Mokbel, X. Xiong, and W. G. Aref, “SINA: Scalable incremental processing of continuous queries in spatiotemporal databases,” inProc. SIGMOD, Paris, France, 2004.

[18] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, “Evaluating collaborative filtering recommender systems,”ACM TOIS, vol. 22, no. 1, pp. 5–53, 2004.

[19] M. J. Carey and D. Kossmann, “On saying "Enough Already!" in SQL,” inProc. SIGMOD, New York, NY, USA, 1997.

[20] S. Chaudhuri and L. Gravano, “Evaluating top-k selection queries,” inProc. Int. Conf. VLDB, Edinburgh, U.K., 1999.

[21] [R. Fagin, A. Lotem, and M. Naor, “Optimal aggregation algorithms for middleware,” inProc. ACM Symp. PODS,NewYork, NY, USA, 2001.

[22] J. Bao, C.-Y. Chow, M. F. Mokbel, and W.-S. Ku, “Efficient evaluation of k-range nearest neighbor queries in road networks,” in Proc. Int. Conf. MDM, Kansas City, MO, USA, 2010.

[23] G. R. Hjaltason and H. Samet, “Distance browsing in spatial databases,”ACM TODS, vol. 24, no. 2, pp. 265–318, 1999.

[24] K. Mouratidis, M. L. Yiu, D. Papadias, and N. Mamoulis, “Continuous nearest neighbor monitoring in road networks,” in Proc. Int. Conf. VLDB, Seoul, Korea, 2006.

[25] D. Papadias, Y. Tao, K. Mouratidis, and C. K. Hui, “Aggregate nearest neighbor queries in spatial databases,” ACM TODS, vol. 30, no. 2, pp. 529–576, 2005.

[26] M. Sharifzadeh and C. Shahabi, “The spatial skyline queries,” in Proc. Int. Conf. VLDB, Seoul, Korea, 2006.

[27] N. Bruno, L. Gravano, and A. Marian, “Evaluating top-k queries over web-accessible databases,” inProc. ICDE, San Jose, CA, USA, 2002.

[28] P. Venetis, H. Gonzalez, C. S. Jensen, and A. Y. Halevy, “Hyperlocal, directions-based ranking of places,” PVLDB,vol.4,no.5, pp. 290–301, 2011.

[29] M.-H. Park, J.-H. Hong, and S.-B. Cho, “Location-based recommendation system using Bayesian user’s preference model inmobile devices,” inProc. Int. Conf. UIC, Hong Kong, China, 2007.

[30] Y. Takeuchi and M. Sugimoto, “An outdoor recommendation system based on user location history,” inProc. Int. Conf. UIC, Berlin, Germany, 2006.