IPHV2I10019X
International Journal Of Advanced Research and Innovation -Vol.II, Issue .1
ISSN Online: 2319 – 9253
Print: 2319 – 9245

SPATIAL DATA RANKING WITH SPATIAL PREFERENCE AND QUERIES

[1]G. RENUKA, [2] MOHD FAYAZ, [3]B.VEERA PRATHAP
1. M.Tech-(SE) Pursuing, 2. M.Tech-(SE) Pursuing ,
3. HOD, Dept of CSE, MOTHER THERESSA COLLEGE OF ENGINEERING & TECHNOLOGY

Abstract

Spatial data are specialsed data than common alpha numerical based data. These data items are critical to handle and more sensitive to process, in that apply ranking on these type of data requires spatial preference. A spatial preference query ranks objects based on the qualities of features in their spatial neighborhood concept can be specified by the user via different functions. It can be an explicit circular region within a given distance from the flat. Another intuitive definition is to assign higher weights to the features based on their proximity to the flat. In this paper, we formally define spatial preference queries and propose appropriate indexing techniques and search algorithms for them.

INTRODUCTION

SPATIAL database systems manage large collections of geographic entities, which apart from spatial attributes contain nonspatial information (e.g., name, size, type, price, etc.). In this paper, we study an interesting type of preference queries, which select the best spatial location with respect to the quality of facilities in its spatial neighborhood.

Given a set D of interesting objects (e.g., candidate locations), a top-k spatial preference query retrieves the k objects in D with the highest scores. The score of an object is defined by the quality of features (e.g., facilities or services) in its spatial neighborhood. As a motivating example, consider a real estate agency office that holds a database with available flats for lease. Here “feature” refers to a class of objects in a spatial map such as specific facilities or services. A customer may want to rank the contents of this database with respect to the quality of their locations, quantified by aggregating nonspatial characteristics of other features (e.g., restaurants, cafes, hospital, market, etc.) in the spatial neighborhood of the flat (defined by a spatial range around it). Quality may be subjective and query-parametric. For example, a user may define quality with respect to nonspatial attributes of restaurants around it (e.g., whether they serve seafood, price range, etc.). Traditionally, there are two basic ways for ranking objects: 1) spatial ranking, which orders the objects according to their distance from a reference point, and 2) nonspatial ranking, which orders the objects by an aggregate function on their nonspatial values. Our top-k spatial preference query integrates these two types of ranking in an intuitive way. As indicated by our examples, this new query has a wide range of applications in service recommendation and decision support systems. To our knowledge, there is no existing efficient solution for processing the top-k spatial preference query. A brute force approach for evaluating it is to compute the scores of all objects in D and select the top-k ones. This method, however, is expected to be very expensive for large input data sets.

. SYSTEM ANALYSI

Existing Work

To our knowledge, there is no existing efficient solution for processing the top-k spatial preference query.

Object ranking is a popular retrieval task in various applications. In relational databases, we rank tuples using an aggregate score function on their attribute values.

For example, a real estate agency maintains a database that contains information of flats available for rent. A potential customer wishes to view the top-10 flats with the largest sizes and lowest prices. In this case, the score of each flat is expressed by the sum of two qualities: size and price, after normalization to the domain (e.g., 1 means the largest size and the lowest price).

In spatial databases, ranking is often associated to nearest neighbor (NN) retrieval. Given a query location, we are interested in retrieving the set of nearest objects to it that qualify a condition (e.g., restaurants). Assuming that the set of interesting objects is indexed by an R-tree, we can apply distance bounds and traverse

IPHV2I10019X
International Journal Of Advanced Research and Innovation -Vol.II, Issue .1
ISSN Online: 2319 – 9253
Print: 2319 – 9245

the index in a branch-and-bound fashion to obtain the answer.

Proposed Work:

Ranking

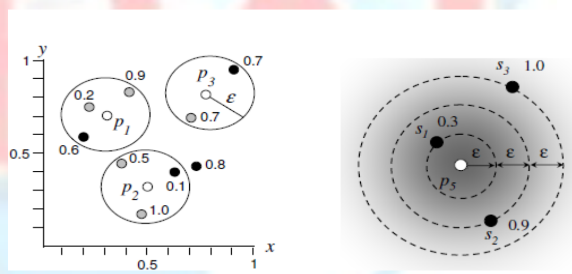
In general scenarios Ranking items according to the rating variance of neighbors of a particular user for a particular item. There exist a number of different ranking approaches that can improve recommendation diversity by recommending items other than the ones with topmost predicted rating values to a user. A comprehensive set of experiments was performed using every rating prediction technique in conjunction with every recommendation ranking function on every dataset for different number of top-N recommendations

We Propose this ranking approaches to spatial data

(i) Spatial ranking, which orders the objects according to their distance from a reference point, and

(ii) Non-spatial ranking, which orders the objects by an aggregate function on their non-spatial values. Our top- k spatial preference query integrates these two types of ranking in an intuitive way. As indicated by our examples, this new query has a wide range of applications in service recommendation and decision support systems. To our knowledge, there is no existing efficient solution for processing the top-k spatial preference query. A brute-force approach for evaluating it is to compute the scores of all objects in D and select the top-k ones. This method, however, is expected to be very expensive for large input datasets.

System Architecture



Approaches

1. Spatial Ranking
2. Non-Spatial ranking
3. Neighbor (NN) Retrieval
4. Spatial Query Evaluation on R-trees

Spatial Ranking

Spatial ranking, which orders the objects according to their distance from a reference point.

Non-Spatial Ranking

Non-spatial ranking, which orders the objects by an aggregate function on their non-spatial values. Our top- k spatial preference query integrates these two types of ranking in an intuitive way. As indicated by our examples, this new query has a wide range of applications in service recommendation and decision support systems. To our knowledge, there is no existing efficient solution for processing the top-k spatial preference query.

Neighbor (NN) Retrieval

Object ranking is a popular retrieval task in various applications. In relational databases, we rank tuples using an aggregate score function on their attribute values. For example, a real estate agency maintains a database that contains information of flats available for rent. A potential customer wishes to view the top-10 flats with the largest sizes and lowest prices. In this case, the score of each flat is expressed by the sum of two qualities: size and price, after normalization to the domain [0,1] (e.g., 1 means the largest size and the lowest price). In spatial databases, ranking is often associated to nearest neighbor (NN) retrieval. Given a query location, we are interested in retrieving the set of nearest objects to it that qualify a condition (e.g., restaurants). Assuming that the set of interesting objects is indexed by an R-tree [3], we can apply distance bounds and traverse the index in a branch-and-bound fashion to obtain the answer.

Nevertheless, it is not always possible to use multidimensional indexes for top-k retrieval. First, such indexes break-down in high dimensional spaces. Second, top-k queries may involve an arbitrary set of user-specified attributes (e.g., size and price) from possible ones (e.g., size, price, distance to the beach, number of bedrooms, floor, etc.) and indexes may not be available for all possible attribute combinations (i.e., they are too expensive to create and maintain). Third, information for different rankings to be combined (i.e., for different attributes) could appear in different databases (in a distributed database scenario) and unified indexes may not exist for them. Solutions for top-k queries focus on the efficient merging of object rankings that may arrive from different (distributed) sources. Their motivation is to minimize the number of accesses to the input rankings until the objects with the top-k aggregate scores have been identified. To achieve this, upper and lower bounds for the objects seen so far are maintained while scanning the sorted lists.

Spatial Query Evaluation:

The most popular spatial access method is the R-tree, which indexes minimum bounding rectangles (MBRs) of objects. Figure 2 shows a set D = fp1;: : : ; p8g of spatial objects (e.g., points) and an R-tree that indexes them. R-trees can efficiently process main spatial query types, including spatial range queries, nearest neighbor queries, and spatial joins. Given a spatial region W, a spatial range query retrieves from D the objects that intersect W. For instance, consider a range query that asks for all objects within the shaded area in Figure 2. Starting from the root of the tree, the query is processed by recursively following entries, having MBRs that intersect the query region. For instance, e1 does not intersect the query region, thus the sub-tree pointed by e1 cannot contain any query result. In contrast, e2 is followed by the algorithm and the points in the corresponding node are examined recursively to find the query result.

CONCLUSION

we formally define spatial preference queries and propose appropriate indexing techniques and search algorithms for them. Extensive evaluation of our methods on both real and synthetic data reveals that an optimized branch-and-bound solution is efficient and robust with respect to different parameters.

REFERENCE:

[1] Man Lung Yiu, Hua Lu, Nikos Mamoulis, and Michail Vaitis, "Ranking Spatial Data by Quality Preferences", **IEEE Transactions on Knowledge and Data Engineering, Vol. 23, No.3, March 2011.**

[2] P. Desnoyers, D. Ganesan, and P. Shenoy, "TSAR: a two tier sensor storage architecture using interval skip graphs," in Proc. ACM SenSys'05, San Diego, California, USA, Nov. 2005, pp. 39–50.

ISSN Online: 2319 – 9253
Print: 2319 – 9245

[3] M. Shao, S. Zhu, W. Zhang, and G. Cao, "pDCS: security and privacy support for data-centric sensor networks," in Proc. IEEE INFOCOM'07, Anchorage, Alaska, USA, May 2007, pp. 1298–1306.

[4] N. Subramanian, C. Yang, and W. Zhang, "Securing distributed data storage and retrieval in sensor networks," in IEEE PerCom'07, White Plains, NY, Mar. 2007.

[5] B. Sheng and Q. Li, "Verifiable privacy-preserving range query in sensor networks," in Proc. IEEE INFOCOM'08, Phoenix, AZ, Apr. 2008, pp. 46–50.

[6] Q. Wang, K. Ren, W. Lou, and Y. Zhang, "Dependable and secure sensor data storage with dynamic integrity assurance," in IEEE INFOCOM'09, Rio de Janeiro, Brazil, Apr. 2009.

[7] O. Gnawali, K.-Y. Jang, J. Paek, M. Vieira, R. Govindan, B. Greenstein, A. Joki, D. Estrin, and E. Kohler, "The tenet architecture for tiered sensor networks," in Proc. ACM SenSys'06, Boulder, Colorado, USA, Oct. 2006, pp. 153–166.

[8] X. Li, Y. J. Kim, R. Govindan, and W. Hong, "Multi-dimensional range queries in sensor networks," in Proc. ACM SenSys'03, Los Angeles,California, USA, Nov. 2003, pp. 63–75.

[9] H. Hacig¨um¨us¸, B. Iyer, C. Li, and S. Mehrotra, "Executing SQL over encrypted data in the database-service-provider model," in Proc. ACM SIGMOD'02, Madison, Wisconsin, June 2002, pp. 216–227.

[10] B. Hore, S. Mehrotra, and G. Tsudik, "A privacy-preserving index for range queries," in Proc. VLDB'04, Toronto, Canada, Aug. 2004, pp. 720–731