SAJET

South Asian Journal of Engineering and Technology Vol.2, No.17 (2016) 308–313

ISSN No: 2454-9614

Inverted Indexing in Spatial Database

S.S. Saranya, N. Abinaya*

Department Of Computer Science and Engineering, Nandha Engineering College
Erode, Tamilnadu, India.

*Corresponding Author: S.S. Saranya

E-mail: saranyass@nandhaengg.org

Received: 16/11/2015, Revised: 18/12/2015 and Accepted: 11/03/2016

Abstract

The spatial database allows users to search queries in geographical locations. Most recently, the users search for objects that satisfy a particular constraint. For example, consider a hotel database which contains geographical information about the hotels and also the list of items (consider as keywords) in which it is famous for. The spatial inverted indexing technique creates an inverted list for each of the keyword and merges the list, thus satisfying the constraint. It then finds the nearest location from the query point. The SI indexing technique is efficient only for small set of keywords. Moreover, returning a nearest spatial web object that best matches all the keywords is typically hard. The proposed approach retrieves a nearest group of spatial web objects to the query point 'q', where the keywords of the objects in the group are aggregated to satisfy the constraint. The approach is efficient if the number of keyword is large.

Reviewed by* **ICETSET'16 *organizing committee*

Keywords: Nearest objects, grouping, decomposing keywords.

1. Introduction

The spatial indexing is needed to find the objects in the given space. The entries in this index depend on the location of the objects and the values of these objects are the x and y coordinates. Spatial indexing uses R-trees or Quad-trees or both for indexing purpose. The R-trees are most widely used when compared to Quad-trees. The reason is the performance of the Quad-tree decreases when the spatial query processing is performed based on the features of the spatial objects. An R-tree index first creates a rectangular geometry and the objects are enclosed in the geometry. This rectangular geometry is called Minimum Bounding Rectangle (MBR).

There are easy ways to support queries that combine spatial and text features. For example, for the above query, it could first fetch all the restaurants whose menus contain the set of keywords, and then from the retrieved restaurants, find the nearest one. Similarly, one could also do it reversely by targeting first the spatial conditions – browse all the restaurants in ascending order of their distances to the query point until encountering one whose menu

has all the keywords. The major drawback of these straightforward approaches is that they will fail to provide real time answers on difficult inputs. A typical example is that the real nearest neighbour lies quite far away from the query point, while all the closer neighbours are missing at least one of the query keywords. Inverted indices are optimized solutions for multidimensional points, and are thus named the spatial inverted index (SI-index). This access method successfully incorporates point coordinates into a conventional inverted index with small extra space; meanwhile, an SI-index preserves the spatial locality of data points, and comes with an R-tree built on every inverted list at little space overhead. Nearest Neighbour search problem is an optimization problem for finding closest points in metric spaces. Given a set S of points in a metric space M and a query point $q \in M$, find the closest point in S to q. M is taken to be d-dimensional Euclidean space and distance is measured by Euclidean distance or Manhattan distance. There is various kind of the NNS problem. K-nearest neighbour search and the ε-approximate nearest neighbour search are the well-known algorithms that are being widely used.

2. Methodology

In this approach a group of spatial web objects are retrieved such that the objects keywords are aggregated together to form the group keyword.

The group keyword matches with the queries keywords. Then the groups which are nearest to the query location are extracted. The point which has to be noted is that the group object should have lowest inter-object distances. The existing solutions to these kinds of queries either may bring upon excessive space consumption or are unable to give real time answers. The objects those contain user keywords are grouped. The grouped objects distance is calculated from given query point and ranked based on distance. The top ranked object group is returned as an end result.

To identify the need for such spatial keyword queries, we assume a database of spatial web objects which consist of spatial and textual information. Then the next step is retrieving a group of spatial objects that collectively meet the user's needs. The conditions that are to be satisfied with the group keywords are: 1) the textual information of the group of objects must satisfy the query keywords, 2) the identified group must be closer to the query point, and 3) the objects in the group are close to each other.

Consider a set of spatial web objects D, and a query $q = (\lambda, \gamma)$, where λ is a location and is a set of keywords. The aggregated object approach finds the group of objects X, such that X is a subset of D, where $\cup_{r \in X} r.\gamma = q.\gamma$ and the distance is minimized. The group is obtained by considering the textual information of the points. If any of the query keyword is matched in the point p1 then this point is considered. Finally the points p1, p2… pn are grouped such that the union of all the textual information of the points must match the query keyword. Here the point to be considered is the number of objects that are grouped.

There are n numbers of points that collectively form a group but in order to consider the efficiency n should

be minimized so that maximum performance can be obtained. This approach uses IR tree (Range tree with inverted files) as a data structure. An inverted file contains two main components: objects description and the pointer of the object containing the description.

3. Related Work

3.1 IR^2 tree

An IR^2 tree is variant of the R-trees which is used to index the spatial objects. Here the nodes of the tree contains values as spatial information and also a signature file, which contains set of keywords that are related to that object. The signatures may be a bitmap or superimposed code. The search is based on depth first manner and it adopts branch and bound technique to traverse the nodes. In IR^2 tree, given a keyword query Q={k1,k2,…,kl} where KI is the list of keywords, the answer is the list of objects { O1,O2,..,On} where each objects contains textual information that matches the query keyword.

3.2 Spatial Inverted Index

Spatial inverted indices are used by spatial databases to cope with the multi-dimensional functionality of the data. This indexing technique is implemented to answer the nearest neighbour with key words. The data structure that has been implemented here is Inverted File-R-tree. In this structure, first an inverted index file is built. Then, the file is modified by building a R-tree to the set of objects MBR pointed by each keyword in the file. The leaf node of R-tree points to a page list of object ids whose entry contains the keyword and the MBR. When a query is issued, the query keywords are filtered using the inverted index. Later, the R-tree corresponding to each query keyword is used to filter the spatial part of the query. The intersection of object ids from the R-trees produces the final answer set. It enforces R-trees to browse the points of all relevant lists in ascending order of their distances to the query point.

It offers two competing ways for query processing. It can merge multiple lists very much like merging traditional inverted lists by ids. SI-index preserves the spatial locality of data points. The performance of this approach depends on objects selection satisfying the text or spatial part of a query. If the number of objects in the spatial region of the query is small, then spatial filtering is done first, then the keyword extraction and vice versa.

3.3 Drawbacks

Since the keyword combines more than two keywords, objects cannot be identified by satisfying all the constraints. The result may be simply empty. It Need to load the documents of many objects, incurring expensive overhead as each loading necessitates a random access. And also for retrieving each object id, the entire tree has to be searched in worst case resulting in time complexity. Since this approach concentrates on generating the inverted search list first, the resultant object obtained from the search may satisfy all the keyword constraints but, may not be a nearest neighbour. The efficiency of this technique gets lowered and may not produce real time results when number of keywords in a query gets increased. Space consumption is high because the tree data structure stores the

object id and also the pointers to retrieve the spatial and textual information of the object IDs. It is basically an exhaustive search on the object space.

4. Solution Based on Inverted Index

4.1 Grouping of Objects

Object aggregation is the method to group the objects of similar types such that, the objects in the group must have relationship among themselves. The assembling process combines the individual objects in a cooperative manner thus forming a single composite group. The group characteristics can be inherited from the properties of its component members. It has a "is-a" relationship. Since the composed of object, it contains "has" relationship.

We propose a new indexing strategy called Aggregation of objects, in order to remove the unwanted textual and spatial information and to handle queries with multiple keywords at different spatial locations. The following steps represent the procedure to process the group of keywords

Step 1: Decompose the query „q into sequence of partial queries which contain different set of keywords.

Step 2: Consider the given user query as the first partial query and find the objects that satisfy part or all the keywords in the query q.

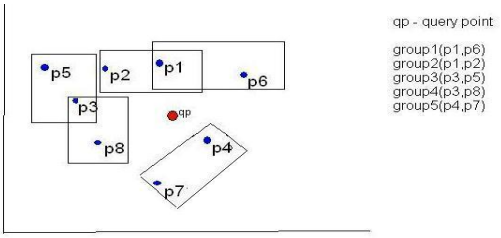
Step 3: The object matched for respective keyword is added to the list.

Step 4: Repeat the above two steps until all the keywords are covered or some keywords that cannot be grouped at any cost. The resultant is the number of result sets say G1,G2..Gn, where Gi is the group of objects that satisfies the user query q. For example, consider a sample spatial location which contains set of point's p1...p8. The following table contains points with their textual information.

Points	**Textual information**
P1	Hospital, Shopping mall
P2	Theatre
P3	Hospital
P4	Shopping mall
P5	Theatre, Shopping mall
P6	Theatre
P7	Hospital, Theatre
P8	Theatre, Shopping mall

Table.1 Points with textual information

Consider the query keyword as qp(hospital, theatre, shopping mall). Now the query is decomposed partially to form set of queries as {hospital, theatre, shopping mall, hospital & theatre, hospital & shopping mall, theatre & shopping mall}. The points are searched such that it satisfies any of the queries set. The obtained points are grouped together with respect to the textual match. It is graphically illustrated below.



Given a group G1,G2,..Gn; this procedure will find the nearest neighbour group from the query point q. In order to compute the shortest distance, two factors are considered: 1.Compute the distance between the group G and the query point ,,q., 2. Compute the distance among the objects within the group.

Consider the fig.2; the average group distance from the query point is to be calculated. Assume the distance of group G1 is minimum from query point when compared to other groups in the graph. Therefore the solution is obtained from group G1 which nearer to qp and also satisfies all the keywords.

6. Conclusion

There are plenty of applications calling for a search engine that is able to efficiently support novel forms of spatial queries that are integrated with keyword search. The existing solutions to such queries either incur prohibitive space consumption or are unable to give real time answers. The proposed system has remedied the situation by developing an access method called aggregated objects. It will provide a nearest neighbour group which is closest to a given query point when two situations occurs. At first when a number of keywords in a user query get increased, the resultant object may not satisfy all constraints. Second a real nearest neighbour lies quite far away from the query point.

References

[1] X. Cao, L. Chen, G. Cong, C.S. Jensen,Q. Qu, A.Skovsgaard, D. Wu, and M. L. Yiu. “Spatial keyword querying”. In ER, pp 16–29, 2012.
[2] X.Cao, G. Cong, and C. S. Jensen.”Retrieving top-k prestige-based relevant spatial web objects”. PVLDB, 3(1):373–384, 2010.
[3] G. Cong, C.S. Jensen, B.C. Ooi. “Collective spatial keyword querying”. In Proc. of ACM Mangt of Data (SIGMOD), pp 373–384, 2011.
[4] Y. Chen, T. Suel, and A. Markowetz. “Efficient query processing in geographic web search engines”. In Proc. of ACM Management of Data (SIGMOD), pages 277–288, 2006.
[5] E. Chu, A. Baid, X. Chai, A. Doan, and J. Naughton, “Combining keyword search and forms for ad hoc querying of databases”. In Proc. of ACM Management of Data (SIGMOD), 2009.
[6] G. Cong, C. S. Jensen, and D. Wu. “Efficient retrieval of the top-k most relevant spatial web objects”. PVLDB,
[7] 2(1):337–348, 2009.
[8] D. Felipe, V. Hristidis, and N. Rishe. “Keyword search on spatial databases”. In Proc. of International Conference on Data Engineering (ICDE), pages 656–665, 2008.

[9] J. Lu, Y. Lu, and G. Cong. “Reverse spatial and textual k nearest neighbor search”. In Proc. of ACM Management of Data (SIGMOD), pages 349–360, 2011.

[10] D. Zhang, Y. M. Chee, A. Mondal, A. K. H. Tung, and M. Kitsuregawa. "Keyword search in spatial databases: Towards searching by document". In Proc. of International Conference on Data Engineering (ICDE), pages 688–699, 2009.
[11] Amruta Joshi, Prof. U. M. Patil Indexing techniques for Geospatial Searching: A survey International Journal of Scientific & Engineering Research, Volume 4, Issue 8, August-2013.
[12] Nitin Bhatia, Vandana Survey of Nearest Neighbours Techniques (IJCSIS) International Journal of Computer Science and Information Security, Vol. 8, No. 2, 2010.
[13] Yufei Tao Cheng Sheng Fast Nearest Neighbor Search with Keywords IEEE transactions on knowledge and data engineering, 2013.
[14] R. Hariharan, B. Hore, C. Li, and S. Mehrotra. Processing spatial keyword (SK) queries in geographic information retrieval (GIR) systems. In Proc. Of Scientific and Statistical Database Management(SSDBM), 2007.