International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 10, October 2015
Available at http://internationaljournalofresearch.org

Nearest Neighbor Search with Keywords Using Spatial Inverted Index

Melam Sreedevi [1]& K.S. Kiran Kumar [2]

[1]PG Scholar,Dept of CSE, Rao & Naidu Engineering College, Ongole,Prakasam Dist,Andhra Pradesh
[2]Assistant Professor,Dept of CSE, Rao & Naidu Engineering College, Ongole,Prakasam Dist,Andhra Pradesh

Abstract—

Many applications require finding the nearest objects closest to a specified location that contains a keywords. Nearest neighbor queries that aims to find objects both a spatial predicate and predicate on their associated texts. For example a nearest neighbor query search restaurant that is the closest among many restaurants within a particular area, whose menu contain required food keyword with respect to query. The problems of the nearest neighbor search on spatial data and keyword search on text data have been studied separately. Existing solution to queries is based on IR2 -Tree (Information Retrieval R-Tree), but it has a few deficiencies that it requires more time to process the query and fails to give real time answers. To overcome these problems, we present an efficient access method to answer spatial keyword queries. we introduce an indexing structure called spatial inverted index that extends the inverted index.

Keywords: Nearest Neighbor; Keyword; SI-Index; R-Tree

I. INTRODUCTION

An increasing number of applications require an efficient execution of nearest neighbor (NN) queries constrained by the properties of the spatial objects. Due to the popularity of keyword search, particularly on the Internet, many of these applications allow the users to provide keywords that the spatial objects should contain description of spatial keyword query. A spatial database manages multidimensional objects (such as points, rectangles, etc.) and provides fast access to those objects based on different selection criteria. The importance of spatial databases is, the real entities are represented in a geometric manner. For example, locations of restaurants, hotels, hospitals and so on are represented as points in a map, while larger areas such as parks, lakes and landscapes as a combination of rectangles. Queries focus on objects' geometric properties only, such as whether a point is in a rectangle or how close two points are from each other. Some modern applications that call for the ability to select objects based on both of their geometric coordinates and their associated texts.

For example, it would be useful if a search engine can be used to find the nearest restaurant that offers "steak, spaghetti and brandy" all at the same time. Note that this is not the "globally" nearest restaurant, but the nearest restaurant among only those providing all the demanded foods. In this paper, we design another form of inverted index that is optimized for multidimensional points and it is named as spatial inverted index (SI-index). This access method incorporates point coordinates into a conventional inverted index with small extra space. 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. As a result, it offers two competing ways for query processing. We

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 10, October 2015
Available at http://internationaljournalofresearch.org

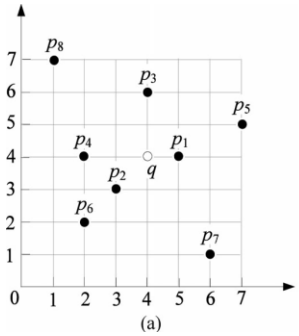
can (sequentially) merge multiple lists very much like merging traditional inverted lists by ids. Alternatively, we can also influence the R-trees to browse the points of all relevant lists in ascending order of their distances to the query point.

We present a method to efficiently answer top-k spatial Keyword queries, which is based on the tight integration of data structures and algorithms used in spatial database search and Information Retrieval (IR).In particular, our method consists of building an Information Retrieval R-Tree (IR2-Tree), which is a structure based on the R-Tree . At query time an incremental algorithm is employed that uses the IR2-Tree to efficiently produce the top results of the query. The IR2-Tree is an R-Tree where a signature (Fallouts and Christodoulakis) is added to each node v of the IR2-Tree to denote the textual content of all spatial objects in the sub tree rooted at v. Our top-k spatial keyword search algorithm, which is inspired by the work of Hjaltason and Samet , exploits this information to locate the top query results by accessing a minimal portion of the IR2-Tree.

This work has the following contributions:

• The problem of top-k spatial keyword search is defined.

• The IR2-Tree is proposed as an efficient indexing Structure to store spatial and textual information for a set of objects. Efficient algorithms are also presented to maintain the IR2-Tree, that is, insert and delete objects. • An efficient incremental algorithm is presented to answer Top-k spatial keyword queries using the IR2 -Tree. Its performance is evaluated and compared to current approaches. Real datasets are used in our experiments that show the significant improvement in execution times. Note that our method can be applied to arbitrarily-shaped and multi-dimensional objects and not just points on the two dimensions, which are used in our running examples for clarity.



p	W_p
p_1	{a, b}
p_2	{b, d}
p_3	{d}
p_4	{a, e}
p_5	{c, e}
p_6	{c, d, e}
p_7	{b, e}
p_8	{c, d}

(b)

Fig. 1. (a) Shows the locations of points and (b) gives their associated texts.

II.LITERATURE SURVEY

2.1 DBXplorer: A System for Keyword-Based Search over Relational Databases

Internet search engines have popularized the keyword based search paradigm. While traditional database management systems offer powerful query languages, they do not allow keyword-based search. In this paper, we discuss DBXplorer, a system that enables keyword based search in relational databases. DBXplorer has been implemented using a commercial relational database and web server and allows users to interact via a browser front-end. We outline the challenges and discuss the implementation of our system including results of extensive experimental evaluation

2.2 Efficient Query Processing in Geographic Web Search Engines

Geographic web search engines allow users to constrain and order search results in an intuitive manner by focusing a query on a particular geographic region. Geographic search technology, also called local search, has recently received significant interest from major search engine companies. Academic research in this area has focused primarily on techniques for extracting geographic knowledge from the web. In this

paper, we study the problem of efficient query processing in scalable geographic search engines. Query processing is a major bottleneck in standard web search engines, and the main reason for the thousands of machines used by the major engines

2.3 Retrieving Top k Prestige Based Relevant Spatial Web Objects

The location-aware keyword query returns ranked objects that are near a query location and that have textual descriptions that match query keywords. This query occurs inherently in many types of mobile and traditional web services and applications, e.g., Yellow Pages and Maps services. Previous work considers the potential results of such a query as being independent when ranking them. However, a relevant result object with nearby objects that are also relevant to the query is likely to be preferable over a
relevant object without relevant nearby objects.

2.4 The R*-tree: An Efficient and Robust Access Method for Points and Rectangles

The R-tree, one of the most popular access methods for rectangles, IS based on the heuristic optimization of the area of the enclosing rectangle in each inner node By running numerous experiments m a standardized test bed under highly varying data, queries and operations, we were able to design the R*-tree which incorporates a combined optimization of area, margin and overlap of each enclosing rectangle m the directory Using our standardized test bed m an exhaustive performance comparison, It turned out that the R*-tree clearly outperforms the existing R-tree variants Guttman's linear and quadratic R-tree and Greene's variant of the R-tree This superiority of the R*-tree holds for different types of queries and operations, such as map overlay.

III. NEAREST NEIGHBOR SEARCH TECHNIQUE

A. IR-Tree, Approximation algorithm and Exact algorithm:

This method is used to retrieve a group of spatial web objects such that the query's keywords are cover by group's keywords and objects are near to the query location and have the lowest inter object distances. This method addresses the two instantiation of the group keyword query. First is to find the group of objects that cover the keywords such that the sum of their distances to the query is minimized. Second is to find a group of objects that cover the keywords such that sum of the maximum distance among an object in group of objects and query and maximum distance among two objects in group of objects is minimized. Both of these sub problems are NP-complete. Greedy algorithm is used to provide an approximation solution to the problem that utilizes the spatial keyword index IR-tree to reduce the search space. But in some application query does not contain a large number of keywords, for this exact algorithm is used that uses the dynamic programming. [1]

B. IUR-tree (Intersection union R-tree)

Geographic objects associated with descriptive texts are becoming common. This gives importance to spatial keyword queries that take both the location and text description of content. This technique is used to analyze the problem of reverse spatial and textual k nearest neighbor search i.e finding objects that takes the query object as one of their spatial textual similar objects. For this type of search hybrid index structure is used that successfully merge the location proximity with textual similarity. For searching, branch and bound algorithm is used. In addition to increase the speed of query processing a variant of IURtree and two optimization algorithm is used. To enhance the IUR-tree text clustering is used, in this objects of all the data base is group into clusters according to their text similarity. Each node of the tree is extended by the cluster information to create a hybrid tree

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 10, October 2015
Available at http://internationaljournalofresearch.org

which is called as cluster IUR-tree. To enhance the search performance of this tree two optimization methods is used, first is based on outlier detection and extraction and second method is based on text entropy. [2]

C. BR*-tree :

This hybrid index structure is used to search m-closest keywords. This technique finds the closest tuples that matches the keywords provided by the user. This structure combines the R*-tree and bitmap indexing to process the closest keyword query that returns the spatially closest objects matching m keywords To reduce the search space a priori based search strategy is used. Two monotone constraints is used as a priori properties to facilitates efficient pruning which is called as distance mutex and keyword mutex. But this approach is not suitable for handling ranking queries and in this number of false hits is large.[3].

D. IR^2-tree :

The growing number of applications requires the efficient execution of nearest neighbor queries which is constrained by the properties of spatial objects. Keyword search is very popular on the internet so these applications allow users to give list of keywords that spatial objects should contain. Such queries called as a spatial keyword query. This is consisted of query area and set of keywords. The IR2-tree is developed by the combination of R-tree and signature files, where each node of tree has spatial and keyword information. This method is efficiently answering the top-k spatial keyword queries. In this signature is added to the every node of the tree. An able algorithm is used to answer the queries using the tree. Incremental nearest algorithm is used for the tree traversal and if root node signature does not match the query signature then it prunes the whole subtrees. But IR2-tree has some drawbacks such as false hits where the object of final result is far away from the query or this is not suitable for handling ranking queries.[4]

E. Spatial inverted index and Minimum bounding method:

So, new access method spatial inverted access method is used to remove the drawbacks of previous methods such as false hits. This method is the variant of inverted index using for multidimensional points. This index stores the spatial region of data points and on every inverted list Rtree is built. Minimum bounding method is used for traversing the tree to prune the search space.

IV.SPATIAL INVERTED INDEX ALGORITHM

The kNN spatial keyword query process is shown in Algorithm. The inputs to the algorithm are the query point q, the boundary object BO , the parameter k and keyword Kw. The kNN result returned by the server is retrieved by calling BO.result () on line 2. H is a min-heap which sorts points according to their distances to query q. First (lines 3-12), the algorithm constructs the boundary cell (BC) of the first object p1 and checks whether q falls inside BC (p1). If not, p1 is not the first NN and the verification process fails. Otherwise, p1 is verified as the first NN and is added to the Visited set. The subsequent for loop (lines 13-22) iterates through all objects in L (kNNs from the BO) and performs the following operations: 1) if the neighbor of the last verified object (L[i]) has not been visited yet, it is inserted into the min-heap H and the Visited set (lines 14-18) and 2) it compares the next object in the result set (L[i+1]) with the top of H (lines 19-21). If they are identical, L[i+1] is verified as the next NN. Otherwise, verification fails and the program returns false.

Algorithm 1: KNN (q, BO, K, Kw)

1: H ← ∅; visited ← ∅
2: L ← BO.result ( ); p1=L[1];
3: BCP ← compute BC(p1);

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 10, October 2015
Available at http://internationaljournalofresearch.org

4: if (q ∉ BCP) then
5: return false; {the first NN fails}
6: else
7: if (Kw ϵ p1) then
8: visited.add (p1);
9: else
10: return false;
11: end if
12: end if
13: for i=1 to k-1 do
14: for all (n ϵ L[i]. Neighbors) do
15: if (n ∉ visited) then
16: visited.add (n);
17: end if
18: end for
19: if (L [i+1].location ≠ H.pop( )) then
20: return false; {the (i+1) th NN fails }
21: end if
22: end for
23: return true;

V.Processing Spatial-Keyword (SK) Queries in Geographic Information Retrieval (GIR) Systems.

Location based information stored in GIS database. These information entities of such databases have both spatial and textual descriptions. This paper proposes a framework for GIR system and focus on indexing strategies that can process spatial keyword query. The following contributions in this paper: 1) It gives framework for query processing in Geo- graphic Information Retrieval (GIR) Systems. 2) Develop a novel indexing structure called KR*-tree that captures the joint distribution of keywords in space and significantly improves performance over existing index structures. 3) This method have conducted experiments on real GIS datasets showing the effectiveness of our techniques compared to the existing solutions. It introduces two index structures to store spatial and textual information.

A) Separate index for spatial and text attributes:

Advantages: -

1. Easy of maintaining two separate indices.
2. Performance bottleneck lies in the number of candidate object generated during the filtering stage.

Disadvantages: -

1. If spatial filtering is done first, many objects may lie within a query is spatial extent, but very few of them are relevant to query keywords. This increases the disk access cost by generating a large number of candidate objects. The subsequent stage of keyword filtering becomes expensive.

B) Hybrid index

Advantages and limitations: -

1. When query contains keywords that closely correlated in space, this approach suffer from paying extra disk cost accessing R*-tree and high overhead in subsequent merging process.

Hybrid Index Structures for Location-based Web Search.

There is more and more research interest in location-based web search, i.e. searching web content whose topic is related to a particular place or region. This type of search contains location information; it should be indexed as well as text information. text search engine is set-oriented where as location information is two-dimensional and in Euclidean space. In previous paper we see same two indexes for spatial as well as text information. This creates new problem, i.e. how to combine two types of indexes. This paper uses hybrid index structure, to handle textual and location based queries, with help of inverted files and R*-trees. It considered three strategies to combine these indexes namely: 1) inverted file and R*-tree double index.2) first inverted file then R*-tree.3) first R*-tree then inverted file. It implements search engine to check performance of hybrid structure, that contains four parts:(1) an extractor which detects geographical scopes of web pages and represents geographical scopes as multiple MBRs based on geographical

coordinates. (2) The work of indexer is use to build hybrid index structures integrate text and location information. (3) The work of ranker is to ranks

the results by geographical relevance as well as non-geographical relevance. (4) an interface which is friendly for users to input location-based search queries and to obtain geographical and textual relevant results.

Advantages: -

1. Instead of using two indexes for textual and spatial information. this paper gives hybrid index structures that integrate text indexes and spatial indexes for location based web search.

Disadvantages: -

1. Indexer wants to build hybrid index structures to integrate text and location information of web pages. To textually index web pages, inverted files are a good. To spatially index web pages, two-dimensional spatial indexes are used, both include different approaches, this cause to degrading performance of indexer.

2. In ranking phase, it combine geographical ranking and non-geographical ranking, combination of two rankings and the computation of geographical relevance may affects on performance of ranking

VI. CONCLUSION

We have seen 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. In this paper, we have remedied the situation by developing an access method called the spatial inverted index (SIindex). Not only that the SI-index is fairly space economical, but also it has the ability to perform keyword augmented nearest neighbour search in time that is at the order of dozens of milli-seconds. Furthermore, as the SIindex is based on the conventional technology of inverted index, it is readily incorporable in a commercial search engine that applies massive parallelism, implying its immediate industrial merits.

VII. REFERENCES

[1] X. Cao, G. Cong, C.S. Jensen, and B.C. Ooi, "Collective Spatial Keyword Querying," Proc. ACM SIGMOD Int'l Conf. Management of Data, pp. 373-384, 2011.

[2] J. Lu, Y. Lu, and G. Cong, "Reverse Spatial and Textual k Nearest Neighbor Search," Proc. ACM SIGMOD Int'l Conf. Management of Data, pp. 349-360, 2011.

[3] D. Zhang, Y.M. Chee, A. Mondal, A.K.H. Tung, and M. Kitsuregawa, "Keyword Search in Spatial Databases: Towards Searching by Document," Proc. Int'l Conf. Data Eng. (ICDE), pp. 688-699, 2009.

[4] G. Cong, C.S. Jensen, and D. Wu, "Efficient Retrieval of the Top-k Most Relevant Spatial Web Objects," PVLDB, vol. 2, no. 1, pp. 337- 348, 2009.

[5] X. Cao, G. Cong, and C.S. Jensen, "Retrieving Top-k Prestige- Based Relevant Spatial Web Objects," Proc. VLDB Endowment, vol. 3, no. 1, pp. 373-384, 2010.

[6] X. Cao, G. Cong, C.S. Jensen, and B.C. Ooi, "Collective Spatial Keyword Querying," Proc. ACM SIGMOD Int'l Conf. Management of Data, pp. 373-384, 2011.

[7] B. Chazelle, J. Kilian, R. Rubinfeld, and A. Tal, "The Bloomier Filter: An Efficient Data Structure for Static Support Lookup Tables," Proc. Ann. ACM-SIAM Symp. Discrete Algorithms (SODA), pp. 30 39, 2004.

[8] Y.-Y. Chen, T. Suel, and A. Markowetz, "Efficient Query Processing in Geographic Web Search Engines," Proc. ACM SIGMOD Int'l Conf. Management of Data, pp. 277-288, 2006.

[9] E. Chu, A. Baid, X. Chai, A. Doan, and J. Naughton, "Combining Keyword Search and Forms for Ad Hoc Querying of Databases," Proc. ACM SIGMOD Int'l Conf. Management of Data, 2009.

[10] G. Cong, C.S. Jensen, and D. Wu, "Efficient Retrieval of the Top-k Most Relevant Spatial Web Objects," PVLDB, vol. 2, no. 1, pp. 337- 348, 2009.

[11] C. Faloutsos and S. Christodoulakis, "Signature Files: An Access Method for Documents and Its Analytical Performance Evaluation," ACM Trans. Information Systems, vol. 2, no. 4, pp. 267-288,