International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 4 Issue: 6 364 - 368

Fast nearest Neighbor Search with Keywords Using IR2-Tree

Mr. Pramod Khandare
Department of Information Technology
SCOE Vadgaon, Pune
Pune, India
pramod9611@gmail.com

Dr. Nilesh Uke
Department of Information Technology
SCOE Vadgaon, Pune
Pune, India
nilesh.uke@gmail.com

Abstract: Conventional abstraction queries, like vary search and nearest neighbor retrieval, involve alone conditions on objects geometric properties. Today, many trendy applications concern novel varieties of queries that aim to go looking out objects satisfying every a abstraction predicate, and a predicate on their associated texts. As associate example, instead of considering all the restaurants, a nearest neighbor question would instead provoke the eating place that is the nearest among those whose menus contain asteak, ˆ spaghetti, brandyaˆ all at identical time. Currently, the best answer to such queries depends on the IR2-tree, which, as shown throughout this paper, contains many deficiencies that seriously impact its efficiency. motivated by this, It tend to develop a latest access methodology called the abstraction inverted index that extends the traditional inverted index to subsume f-dimensional info, and comes with algorithms that will answer nearest neighbor queries with keywords in real time. As verified by experiments, the projected techniques trounce the IR2-tree in question latent amount considerably, generally by a part of orders of magnitude. –

Keywords— *NEAREST NEIGHBOR SEARCH, IR2-TREE, NEAREST, RANGE SEARCH, SPATIAL INVERTED INDEX*

*****

I. Introduction

A Spatial Information that manages multidimensional objects and provides fast Objects on different selection criteria based on access. The importance of spatial databases feature is reflected by a geometric reality of entities modeled for. For example, the location of restaurants, hotels, hospitals And so on a map are represented as points, while Parks, lakes, and the landscape as often as large extents a combination of rectangles. Many functionality of a spatial there are various ways in which specific references useful database. For example, in geography information system Division Search all restaurants can be deployed to search for a particular nearest neighbor retrieval search whiles the restaurant area, Closest to a given address.

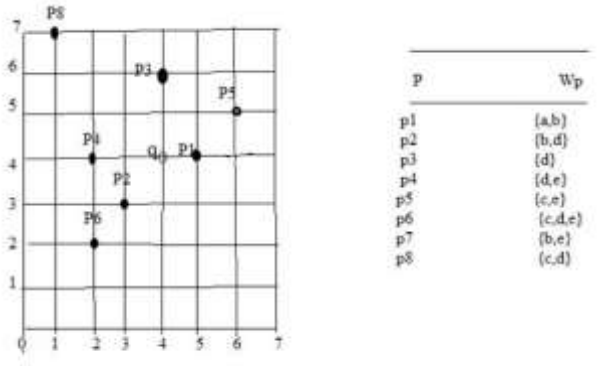
A spatial database manages multidimensional objects (such as points, rectangles, etc.), and provides fast access to those objects based on different selection criteria. The importance of spatial databases is reflected by the convenience of modeling entities of reality in a geometric manner. For example, locations of restaurants, hotels, hospitals and so on are often represented as points in a map, while larger extents such as parks, lakes, and landscapes often as a combination of rectangles. Many functionalities of a spatial database are useful in various ways in specific contexts. For instance, in a geography information system, range search can be deployed to find all restaurants in a certain area, while nearest neighbor retrieval can discover the restaurant closest to a given address. Today, the widespread use of search engines has made it realistic to write spatial queries in a brand new way. Conventionally, 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. We have seen 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 fairly 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 (which would have been returned by a traditional nearest neighbor query), but the nearest restaurant among only those providing [19].

There are easy ways to support queries that combine spatial and text features. For example, for the above query, we could first fetch all the restaurants whose menus contain the set of keywords {steak, spaghetti, brandy}, 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 neighbor lies quite faraway from the query point, while all the closer neighbors are missing at least [19].

A. One of the query keywords.Maintaining the Integrity of the Specifications:

A signature file, due to its conservative nature, still looking for something to do can direct objects, even if they don't have all the keywords. Queries are the easiest ways to support coalition Spatial and text features. For example, the above Query It can bring before the restaurant whoasˆ Menus steak, spaghetti, set of keywords Brandy, and then accessing the restaurant, PA Nearest one. Similarly, one can also do this by reversely Browse all restaurant first targeting spatial conditions Query in ascending order of their distances is a point whose menu of all keywords These direct the big drawback Approach that they will fail to provide real time Difficult to answer on inputs A typical example is that Real lies quite far from the nearest neighbor query Closer neighbors are all missing the point, while at least One of the query keywords.IR2-tree, however, also inherits a vulnerability signature Files: false hits. Thus the penalty due to the need to verify an object No query or not satisfying cannot be resolved using only requires y signature but its full text description, which as a result of random accesses expensive loading. The problem of false its, but not specific to the signature files are set to test with an estimated membership of other methods is

also noted in Compact storage. Therefore, this problem cannot be remedied imply by Signature file replace with any of those methods



p	Wp
p1	{a,b}
p2	{b,d}
p3	{d}
p4	{d,e}
p5	{e,e}
p6	{c,d,e}
p7	{b,e}
p8	{c,d}

Figure 1: Architecture Diagram

In designed a version of the inverted index that is optimized for multidimensional points and thus Local inverted index is named [9]. This access Law was involved in a successful point coordinates Traditional small inverted index with additional space, A delicate compact storage plan. Meanwhile, No SI-index preserves spatial locality of data points, And an R-tree made on each comes with inverted list little space overhead. As a result, it provides two competition Methods for query processing. It can merge much like conventional inverted merging multiple lists ID lists [4]. Alternatively, they can take advantage of r-trees in ascending order, browse all relevant lists of points Query to the point of their distance. As performance Experiments by SI-index significantly outperforms IR2 tree query efficiency, often by a factor Orders of There are easy ways to support queries that combine spatial and text features. For example, it will could first fetch all the restaurants whose menus contain the set of keywords steak, spaghetti, brandy, and then from the retrieved restaurants, find the nearest one. Similarly, one could also do it reversely by targeting first the spatial conditions abrowseˆ 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 neighbor lies quite far away from the query point, while all the closer neighbors are missing at least one of the query keywords.

1.1.1 Spatial Inverted index

In query processing with an SI-index can be done either by merging or together with R-trees in a distance browsing manner. spatial index is used for creating indices because there is huge amount of data need to be stored for searching that data stored in the form of xml documents[16].Compression is widely used to reduce the size of an inverted index in the conventional context where each inverted list contains only ids. If the data storage created in the form of indices then space required is less also time needed for searching the keyword is less. The reversed index data structured in a central module of a usual search engine indexing algorithms. A goal of a search engine presentation is optimize the speed of the query. Find the documents where word occurs.

1.1.2 Nearest neighbor search (NNS)

It also identified as closeness search. Parallel search is an optimization problem for finding closest points in metric spaces. It also search Keyword-Based Search for Top-K Cells in Text Cubeaˆ methods used are inverted-index one-scan, document sorted-scan, bottom-up dynamic programming, and search-space ordering. In the top k cells, there is a searching of nearest key to the query. Cubes forms clusters of single unique group which shows its identity [19].

II. LITERATURE SURVEY

Conventional spatial queries, such as range search and nearest neighbor retrieval, involve only conditions on objects' geometric properties. Today, many modern applications call for novel forms of queries that aim to find objects satisfying both a spatial predicate, and a predicate on their associated texts. For example, instead of considering all the restaurants, a nearest neighbor query would instead ask for the restaurant that is the closest among those whose menus contain asteak, spaghetti, brandya all at the same time. Currently, the best solution to such queries is based on the IR 2-tree, which, has a few deficiencies that seriously impact its efficiency. Motivated by this, they develop a new access method called the spatial inverted index that extends the conventional inverted index to cope with multidimensional data, and comes with algorithms that can answer nearest neighbor queries with keywords in real time. As verified by experiments, the proposed techniques outperform the IR 2-tree in query response time significantly, often by a factor of orders of magnitude.Authors and Affiliations [20].

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 in a standardized tested under highly varying data, queries and operations, they were able to design the R*-tree which incorporates a combined optimization of area, margin and overlap of each enclosing rectangle in the directory. Using standardized tested in an exhaustive performance comparison, it turned out that the R*-tree clearly outperforms the existing R-tree variants. Guttmann'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, for both rectangles and multidimensional points in all experiments. From a practical point of view the R*-tree is very attractive because of the following two reasons 1 it efficiently supports point and spatial data at the same time and 2 its implementation cost is only slightly higher than that of other R-trees.

The web is increasingly being used by mobile users. In addition, it is increasingly becoming possible to accurately geo-position mobile users and web content. This development gives prominence to spatial web data management. Specifically, a spatial keyword query takes a user location and user-supplied keywords as arguments and re-turns web objects that are spatially and textually relevant to these arguments. This paper reviews recent results by the authors that aim to

achieve spatial keyword querying functionality that is easy to use, relevant to users, and can be supported efficiently. The paper covers different kinds of functionality as well as the ideas underlying their definition.

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, 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. Geographic search engine query processing is different in that it requires a combination of text and spatial data processing techniques.

III. STRUCTURE DESIGN

Existing works mainly focus on finding top-k nearest Neighbors, where each node has to match the whole Querying keywords .It does not consider the density of data objects in the spatial space. Also these methods are low efficient for processing query. The existing data structure called the IR2-tree was used for processing the query. But IR2-tree has a drawback of signature files: false hits. That is, a signature file, due to its conservative nature, it searches to some objects, even though they do not have the keywords. The penalty thus caused is the need to verify an object whose satisfying a query or cannot be resolved using only its signature, but requires loading its full text description, which is expensive due to the resulting random accessed.

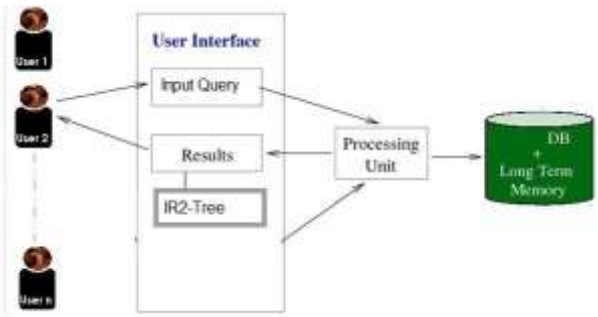


Figure 2: Proposed Structure Interface

.

Many solutions have been developed to evaluate spatial keyword queries. Location-based web search is studied by Zhou et al. to find web pages related to a spatial region. They described three different hybrid indexing structures of integrating inverted files and R2-trees together. According to their experiments, the best scheme is to build an inverted index on the top of R2-trees. In other words, the algorithm first sets up an inverted index for all keywords, and then creates an R2-tree for each keyword. This method performs well in spatial keyword queries in their experiments, but its maintenance cost is high. When an object insertion or deletion occurs, the solution has to update the R2-trees of all the keywords of the object. Cong et al. illustrated a hybrid index structure, the IR-tree, which is a combination of an R-tree and inverted files to process location-aware text retrieval and provide k best candidates according to a rank system. It minimizes areas of enclosing rectangles and maximizing text into account during construction procedures. Felipe et al. developed a novel index, IR2-Tree which integrates an R-tree and signature files together, to answer top- k spatial keyword queries. They record signature information in each node of R-trees in order to decide whether there is any object which satisfies both spatial and keyword constraints simultaneously. However, the size of space for storing signatures in each node is decided before IR2-Tree construction. Once the IR2-Tree has been built, it is impossible to enlarge the space unless the tree is reconstructed. If the number of keywords grows quickly, a system will spend a lot of time repeatedly rebuilding the IR2-Tree. Hariharan et al. proposed an indexing mechanism, IR2-tree, which combines an R2-tree and an inverted index. The difference between their solution and is that they only store related keywords in each node of an R*-tree in order to avoid merging operations to find candidates containing all keywords. However, such a complicated indexing technique has a high maintenance cost as well. Although there are a number of previous studies on spatial keyword queries, most of their solutions can only evaluate queries in Euclidean spaces. This limitation is due to the adoption of the R-tree (or its variants), which cannot index spatial objects based on network distances, into their hybrid index structures. In this paper now spatial database manages dimensional objects (such as points, rectangles, etc.) and provides quick access to those objects. The importance of spatial databases is, it represents entities of reality in geometric manner. For example, Locations of restaurants, hotels, hospitals are described as points in map, whereas larger extents like parks, lakes and landscapes as a mix of rectangles. In this system they have design a proposed system called spatial inverted index (SI-index).SI-preserves the spatial location of data points and builds R-tree on every inverted list at little space overhead.

IV. MODULE AND EXPERIMENTAL RESULT

Module 1: Admin Activity

In this module, the Admin will have to login first. Once the admin does not the registration then he/she can access the number of activity. Admin has number of activity like add college, update college, delete college, add course, delete course, add user, activate user, delete user etc.

Fig 3: Admin Login

Fig 4: Admin Activity

Module 2: College Information:
In this module, the Admin will be update all the college information like College name, College contact number, Address of college, latitude and longitude of college and also update which course available in the college. The admin can also update the college information using searching the college id and college name and also delete college record from the database.

Fig 5: College information

Fig 6: Update college information

Module 3: User Activity
In this module, the user will have to register first. Once the user does the registration then he/she can access the application. For registration user have to enter the basic information about him. User also has to set the username and password. This all registration information is get stored into database. The IMEI number is automatically get stored into database once user do the registration. In this module, after the registration customer can login through mentioned username and password. User can also do update, delete, change password activity are done by user. Admin can delete the user account.

Fig 7: User Login

Fig 8: Activate user by Admin

Fig 9: Update Profile

Fig 10: Delete User by Admin

Module 4: Searching Keyword
In this module, the user will enter the keyword searching for menus available in restaurant which will nearer from its position. Whenever user will enter keyword (menu name) it will match data with the hotel database server and find the nearest restaurant with the available entered menu by customer. For nearest restaurant we are using IR2tree & compression. The IR2-Tree is a combination of an R-Tree and signature files. In particular, each node of an IR2-Tree contains both spatial and keyword information; the former in the form of a minimum bounding area and the latter in the form of a signature. An IR2-Tree facilitates both top-k spatial

queries and top-k spatial keyword queries as we explain below. More formally, an IR2-Tree R is a height-balanced tree data structure, where each leaf node has entries of the form (Obj Ptr, A, S). Obj Ptr and A are defined as in the R-Tree while S is the signature of the object referred by Obj Ptr. Anon-leaf node has entries of the form (Node Ptr, A, S). Node Ptr and A are defined as in the R-Tree while S is the signature of the node. The signature of a node is the superimposition (OR-ing) of all the signatures of its entries. Thus a signature of a node is equivalent to a signature for all the documents in its sub tree.

Fig 11: Keyword Searching

V. CONCLUSION

Here is this applications calling for a search engine that is cable to 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. Future scope In the future it will like to suggest deploying this proposed online work for testing purpose in real time environments like education systems, medical systems, banking where users can provide their feedbacks as well as system itself can provide their better feedbacks and check its real time performances.

VI. REFERENCES

[1] Yufei Tao and Cheng Sheng: ”Fast Nearest Neighbor Search with Keywords”. Na-tional Research Foundation of Korea, GRF 4166/10, 4165/11, and 4164/12 from HKRGC,.

[2] S. Agrawal, S. Chaudhuri, and G. Das.Dbxplorer: ”A system for keyword-based search over relational databases”. In Proc. Of International Conference on Data Engineering (ICDE) , pages 5 aˆ 16, 2002.

[3] N. Beckmann, H. Kriegel, R. Schneider, and B. Seeger. ”The R*-tree: An efficient and robust access method for points and rectangles”. In Proc. of ACM Management of Data (SIGMOD), pages 322 aˆ 331, 1990.

[4] G. Bhalotia, A. Hulgeri, C. Nakhe, S. Chakrabarti, and S. Sudarshan. ”Keyword searching and browsing in databases using banks”. In Proc. of International Con-ference on Data Engineering (ICDE) , pages 431 aˆ 440, 2002.

[5] X. Cao, L. Chen, G. Cong, C. S. Jensen, Q. Qu, A. Skovsgaard, D. Wu, and M. L. Yiu. ”Spatial keyword querying”.In ER , pages 16 aˆ 29, 2012.

[6] X. Cao, G. Cong, and C. S. Jensen.”Retrieving top-k prestige-based relevant spatial web objects”. PVLDB , 3(1):373 aˆ 384, 2010.

[7] X. Cao, G. Cong, C. S. Jensen, and B. C. Ooi.”Collective spatial keyword query-ing”.In Proc. of ACM Management of Data (SIGMOD) , pages 373 aˆ 384, 2011.

[8] B. Chazelle, J. Kilian, R. Rubinfeld, and A. Tal. ”The bloomier filter: an efficient data structure for static support lookup tables”. In Proc. of the Annual ACM-SIAM Symposium on Discrete Algorithms (SODA) , pages 30 aˆ 39, 2004

[9] Y.-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 aˆ 288, 2006.

[10] 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

[11] 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

[12] 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, 1984.

[13] I.D. Felipe, V. Hristidis, and N. Rishe, “Keyword Search on Spatial Databases,” Proc. Int’l Conf. Data Eng. (ICDE), pp. 656-665, 2008.

[14] G.R. Hjaltason and H. Samet, “Distance Browsing in Spatial Databases,” ACM Trans. Database Systems, vol. 24, no. 2, pp. 265-318, 1999.

[15] V. Hristidis and Y. Papakonstantinou, “Discover: Keyword Search in Relational Databases,” Proc. Very Large Data Bases (VLDB), pp. 670-681, 2002.

[16] I. Kamel and C. Faloutsos, “Hilbert R-Tree: An Improved R-Tree Using Fractals,” Proc. Very Large Data Bases (VLDB), pp. 500-509, 1994.

[17] 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

[18] S. Stiassny, “Mathematical Analysis of Various Superimposed Coding Methods,” Am. Doc., vol. 11, no. 2, pp. 155-169, 1960.

[19] Tejaswini Channe, Gurudev Sawarkar, "Review on NLP based Technique to Improve the Performance of knn", IJCTT volume 11 number 2 – May 2014.

[20] Yufei Tao, Cheng Sheng, "Fast Nearest Neighbor Search with Keywords", IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING.