ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

Analysis on Fast Nearest Neighbor Search with Keywords

Pramod Khandare, Dr. Nilesh Uke

Department of Information Technology, SCOE Vadgaon, Pune, India

ABSTRACT: Many search engines are used to search anything from anywhere; this system is used to fast nearest neighbor search using keyword. 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 incremental query. But in intended system, for example when there is search for nearest restaurant, instead of considering all the restaurants, a nearest neighbor query would ask for the restaurant that is, closest among those whose menus contain spicy, brandy all at the same time, solution to such queries is based on the IR2-tree, but IR2-tree having some drawbacks. Efficiency of IR2-tree badly is impacted because of some drawbacks in it. The solution for overcoming this problem should be searched. The spatial inverted index is the technique which will be the solution for this problem.

KEYWORDS: Nearest Neighbor Search, IR2-tree, Nearest, Range search, Spatial inverted index.

I. INTRODUCTION

Nearest neighbor search (NNS), also known as closest point search, similarity search. It is an optimization problem for finding closest (or most similar) points. Nearest neighbor search which returns the nearest neighbor of a query point in a set of points, is an important and widely studied problem in many fields, and it has wide range of applications. We can search closest point by giving keywords as input; it can be spatial or textual. A spatial database use to manage multidimensional objects i.e. points, rectangles, etc. Some spatial databases handle more complex structures such as 3D objects, topological coverage's, linear networks. While typical databases are designed to manage various NUMERIC'S and character types of data, additional functionality needs to be added for databases to process spatial data type's efficiently and it provides fast access to those objects based on different selection criteria. Keyword search is the most popular information discovery method because the user does not need to know either a query language or the underlying structure of the data. The search engines available today provide keyword search on top of sets of documents. When a set of query keywords is provided by the user, the search engine returns all documents that are associated with these query keywords. Solution to such queries is based on the IR2-tree, but IR2- tree having some drawbacks. Efficiency of IR2-tree badly is impacted because of some drawbacks in it. The solution for overcoming this problem should be searched. Spatial inverted index is the technique which will be the solution for this problem. Spatial database manages multidimensional data that is points, rectangles. This paper gives importance spatial queries with keywords. Spatial queries with keywords take arguments like location and specified keywords and provide web objects that are arranged depending upon spatial proximity and text relevancy. Some other approaches take keywords as Boolean predicates, searching out web objects that contain keywords and rearranging objects based on their spatial proximity. Some approaches use a linear ranking function to combine spatial proximity and textual relevance. Earlier study of keyword search in relational databases is gaining importance. Recently this attention is diverted to multidimensional data . N. Rishe, V. Hristidis and D. Felipe has proposed best method to develop neighbor search with keywords. For keyword-based retrieval, they have integrated R-tree with spatial index and signature file. By combining R-tree and signature they have developed a structure called the IR2-tree. IR2-tree has merits of both R-trees and signature files. The IR2-tree preserves object's spatial proximity which important for solving spatial queries.

II. PROBLEM DEFINATION

Implement k nearest neighbor search algorithm using mat lab for given data set and to find out closest point from give query also analyze the result fetch time and accuracy. Implement the Inverted index algorithm by extending point the k

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

nearest neighbor and forming R tree to find closest point from given set of query and also analyze the result against time and result accuracy.

III. LITERATURE SURVEY

In the paper 'fast nearest neighbor search with keywords', there are methods like spatial index, inverted index, nearest neighbor search. The first method 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. If the data storage created in the form of indices then space required is less also time needed for searching the keyword is less.

Second method is inverted index. The inverted index data structure is a central component of a typical search engine indexing algorithm. A goal of a search engine performance is to optimize the speed of the query: find the documents where word occurs. Once an index is developed, which provisions lists of words per document; it is next inverted to develop an inverted index. Querying the index would require sequential iteration through each document and to each word to verify a matching document. The time memory and processing property to execute such a query are not always theoretically realistic. Instead of listing the words per article in the index, the inverted index data structure is developed which lists the documents per word. The inverted index produced, the query can now be determined by jumping to the word id in the inverted index. These were effectively inverted indexes with a small amount of supplementary explanation that required a implausible amount of attempt to produce.
Third method is nearest neighbor search. Nearest neighbor search (NNS), also identified as closeness search, parallel search is an optimization problem for finding closest points in metric spaces. In the paper 'Efficient Keyword-Based Search for Top-K Cells in Text Cube' 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. Method like inverted index used for giving index rather than providing whole data which can be space consuming.

Existing System:
Spatial queries with keywords have not been extensively explored. In the past years, the community has sparked enthusiasm in studying keyword search in relational databases. It is until recently that attention was diverted to multidimensional data. The best method to date for nearest neighbor search with keywords is due to Felipe et al.. They nicely integrate two well-known concepts: R-tree, a popular spatial index, and signature file, an effective method for keyword-based document retrieval. By doing so they develop a structure called the IR2 -tree, which has the strengths of both R-trees and signature files. Like R-trees, the IR2 - tree preserves objects' spatial proximity, which is the key to solving spatial queries efficiently. On the other hand, like signature files, the IR2 -tree is able to filter a considerable portion of the objects that do not contain all the query keywords, thus significantly reducing the number of objects to be examined.

Disadvantages of Existing System:

1. Fail to provide real time answers on difficult inputs.
2. 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 keyword

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

No.	Technique	Key Idea	Advantages	Disadvantages	Target Data
1	k Nearest Neighbor (kNN)	Uses nearest neighbor rule	Training very fast, simple &easy to learn, robust to noisy training data and effective if mining data large.	Biased for value k, computational complexity, memory limitation, runs slowly, easily fooled by irrelevant attributes.	Large data sample.
2	Weighted k Nearest Neighbor (kNN)	Assign weights to neighbors as per distance calculated.	Overcomes limitations of kNN of assigning equal weight to k neighbors implicitly, use all training samples not just k, algorithm global one.	Computation complexity increases in calculating weights, algorithm runs slow.	Large sample data
3	Condensed Nearest Neighbor (CNN)	Eliminate data sets that show similarity and do not add any extra information.	Reduce size of training data, improves query time& memory requirements, reduce the recognition rate.	CNN order dependent, unlikely to pick up points on boundary and computational complexity.	Data set where memory requirement is main criteria.
4.	Model based k Nearest neighbor (MkNN)	Model constructed from data & classifies new data, using model.	More classification accuracy, value of k selected automatically, high efficiency as reduces number of data points.	Do not consider marginal data outside the region.	Dynamic web mining for large repository.
5	Pseudo/Generalized Nearest Neighbor (GNN)	Utilizes information of (n-1) neighbors also instead of that of the nearest neighbor only.	Uses (n-1) classes that consider the whole training data set.	Does not hold good for small data, computational complexity.	Large data set.
6	Orthogonal Search Tree Nearest Neighbor	Uses orthogonal trees.	Less computation time, effective for large data sets.	Query time more.	Pattern recognition

Table No.1 Comparison of Nearest Neighbor Techniques.

In this paper, we design a variant of inverted index that is optimized for multidimensional points, and is thus named the spatial inverted index (SI-index). This access method successfully incorporates point coordinates into a conventional inverted index with small extra space, owing to a delicate compact storage scheme. 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. As a result, it offers two competing ways for query processing. We can (sequentially) merge multiple lists very much like merging traditional inverted lists by ids. Alternatively, we can also leverage the R-trees to browse the points of all

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

relevant lists in ascending order of their distances to the query point. As demonstrated by experiments, the SI-index significantly outperforms the IR2 -tree in query efficiency, often by a factor of orders of magnitude.

Advantages of Proposed System:

1. Distance browsing is easy with R-trees. In fact, the best-first algorithm is exactly designed to output data points in ascending order of their distances
2. It is straight forward to extend our compression scheme to any dimensional space

:

System Architecture:

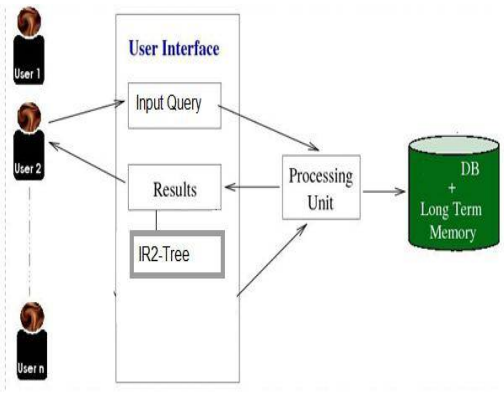

Figure 1: Proposed Structure Interface

Solution is based on two Method:

1) Solutions based on Inverted Indexes
2) Merging and distance browsing

1. Solutions based on Inverted Indexes

For keyword based document retrieval, Inverted Indexes
have proven to be an effective access method. We can
consider the text description Wp of a point p as a document, and then we can build an I-index. Each word in the vocabulary has an inverted list which enumerates the ids of the points that have the word in their documents. To provide significant ease in query processing by allowing an efficient combine step, a sorted order of point ids is maintained the list of each word. For example, suppose that we want to find the points which have words c and d. Then the intersection of the two words' inverted lists is essential to calculate. It will be done by merging them, as both lists are sorted in the same order, whose Input / Output and Processing times are both linear to the total length of the lists.

In Near Neighbor Search with IR2-Tree, point retrieved from the index must be verified which means load and check its text description. For Inverted Index technique, verification is also necessary but for exact opposite reason. In IR2-Tree, verification is required because we do not have the detailed texts of a point. But in I-index, it is done because we do not coordinate. In particular, a given Near Neighbor query q with keyword set Wq, the query algorithm of I-index first by merging generates the set Pq of all points that have all the keywords of Wq, and then, performs |Pq| random I/Os to get the coordinates of each point in Pq in order to evaluate its distance to q.

2.Merging and distance browsing

As verification is affecting performance, we should try to evade it. The simplest way to avoid it mentioned in Inverted Index is that one needs to store the coordinates of each point together with each of its appearances in the inverted lists. The formation of an IR-tree on each list indexing the points is motivated by the presence of coordinates in the inverted lists. With such a combined structure, we will how to execute keyword-based nearest neighbor search. In the RTree, we

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

are allowed to solve uneasiness in the way in which Near Neighbor queries are processed with an I-Index. At present, first we have to obtain all the points carrying all the query words in Wq by merging several lists, to answer a query. It is not fair, if the point p of final results, present literally close to the query point q. The algorithm can stop its execution right away if we could find p very early in all the related lists which will be great. This can be true, but for that if we can search in the list simultaneously by distances as opposed to by ids. A point p would be easily discovered if we can process the points of all lists in ascending order of their distances to q and also its copies in lists can appear in sequence in our process order. For that we have to go on counting number of copies of same point that has came across continuously. Then by reporting, we can terminate when count reaches at |Wq|. Remembering only one count at any instant is sufficient, as it is secure to forget the preceding count when new point occurs.

V. CONCLUSION

In this review Paper, we have study a Searching Nearest Neighbor based on Keywords using Spatial Inverted Index and evaluate the needs and challenges present in Nearest Neighbor Search. This report covers existing techniques for that and also covers upon new improvements in current technique. In this paper, we have surveyed topics like IR2 – Tree, Drawbacks of the IR2-Tree, Spatial keyword search, Solutions based on Inverted Indexes. 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.

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.