International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

Survey of Fast Nearest Neighbor Search

Rutuja Panjabrao Desai[1], S. R. Patil [2]

[1]ME Student, Department of Computer Engineering, Sinhagad Institute of Technology, University of Pune, Maharashtra, India

[2]Department of Computer Engineering, Sinhagad Institute of Technology, University of Pune, Maharashtra, India

Abstract: ***A spatial query takes a location and given keywords as arguments and returns objects that are ranked according to both spatial proximity and text relevance relative to the query. Spatial queries like nearest neighbor retrieval and range search, occupy only conditions on geometric properties of object. For finding objects which are satisfying a spatial predicate and predicate on their associated texts, novel form of queries are called by many applications. Consider situation of retrieving a nearest neighbor query will call for all nearest restaurant whose menu list contains "butter chicken, biryani, pulav", without calling all the restaurants nearest to it. At present, IR2-tree is the best suited solution for such queries. 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 access method which will be solution for this problem. Spatial inverted index extends the conventional inverted index to handle the multidimensional data. To deal with nearest neighbor queries with keywords, it has algorithms which will handle with those queries. Spatial inverted index do better than the IR2-tree by using a feature of orders of magnitude, in query response time appreciably.***

Keywords: Spatial query, IR2-tree, Nearest Neighbor Retrieval, Range search, Spatial inverted index

1. Introduction

In spatial database, multidimensional databases are managed. It offers fast access to objects which are based on different selection criteria. There are many proposed present for spatial objects. The easiness of modeling entities of reality in a geometric manner reflects importance of spatial databases. For examples places like malls, motels, temple, church, schools, shops, stores etc are represented as point inn maps while lakes, parks, a particular region is displayed as combinations of rectangles.

Spatial databases can be used in various ways. Range search can be used to search all hotels in certain area; while nearest neighbor search can find out the restaurant nearest to given location. By using geo-positioning mechanism, accurate user location is increasing available. Also there is increase in objects available on the web which has associated with geographical location. These spatial web objects normally include businesses, tourist attractions, hotels, and stores.

This development gives importance spatial queries with keywords [5] [6] [9] [10]. Spatial queries with keywords take arguments like location and specified keywords and provides web objects that are arranged depending upon spatial proximity and text relevancy. Some other approaches take keywords as Boolean predicates [1] [2], finding out web objects that contain keywords and rearranging objects based on their spatial proximity. Some approaches use a linear ranking function [7] [8] to combine spatial proximity and textual relevance. In last few years, study of keyword search in relational databases is gaining importance. Recently this attention is diverted to multidimensional data [3] [4] [11]. N. Rishe, V. Hristidis and D. Felipe [12] has proposed best method to develop neighbor search with keywords. For keyword-based retrieval, they have integrated R-tree [14] with spatial index and signature file [13]. By combining these two methods they have developed a structure called the IR2-tree [12]. 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 efficiently. IR-2 reduces objects to be examined by filtering a considerable portion of the objects which do not contain all keywords specified in the query. The IR2-tree also inherits a drawback of signature files. Because of conservative nature of signature files, it may direct search to objects which do not contain all keywords. It creates the need of examining of an object whose satisfying a query or not. The results needed be resolved by using not only its signature, but also requires full text description. Random accesses are reasons behind expensiveness of it. This disadvantage is not limited for signature files but also present in other methods for approximate set membership tests with compact storage. Hence, the problem does not get solved by simply replacing signature file with any of those methods.

2. Literature Review

Literature review is classified into The IR2 - Tree, Drawbacks of the IR2-tree, Solutions based on inverted indexes, Spatial Keyword Search, Merging and Distance browsing.

1. IR2 – Tree

The IR2 – Tree [12] combines the R-Tree and signature file. First we will review Signature files. Then IR2-trees are discussed. Consider the knowledge of R-trees and the best-first algorithm [15] for Near Neighbor Search. Signature file is known as a hashing-based framework and hashing -based framework is which is known as superimposed coding (SC) [12]. Other instantiations are less useful than SC [13]. It performs membership test of determining whether a query word w exists in a set W of words. Conservative nature is followed by superimposed coding. It means if w is definitely not in W, it will return "NO". If it returns "YES", true can be in either way in this case whole W must be scanned to prevent false hit.

SC works same as the classic technique of Bloom Filter does. It generates a bit signature of length l from W by hashing each word in W to a string of l bits. Then it generates disjunction of all bit strings. To compute, indicate

Licensed Under Creative Commons Attribution CC BY

by *h (w)* the bit string of word w. It will set all the l bits of *h (w)* to 0. Then steps mention below are repeated: randomly choose a bit and set it to 1. To confirm that the same w always ends up with a matching *h (w)*, randomization must use w as its seed. The m choices may happen to be the same bit and are mutually independent. The concrete values of l and m affect the space cost and false hit probability.

2. Drawbacks of the IR2-Tree

The first access method to answer Near Neighbor queries with keywords is the IR2-Tree. As many popular techniques IR-2 Tree also has few drawbacks which in turn affect its efficiency. The disadvantage called as False hit affecting it seriously. The number of false hit is really large when the object of the final result is far away from the query point and also when the result is simply empty. In these cases, the query algorithm will load the documents of many objects; as each loading necessitates a random access, it acquires costly overhead

3. Spatial keyword search

Because creation of online objects by using an associated geo-location and a text description, a spatial dimension is getting by the web. Web users and content are more and more being geo-positioned and geo coded. Simultaneously, textual descriptions of points of interest are more and more becoming available on the web. This technique leads to a approach which will enable the indexing of data that contains both text descriptions and geo-location and which will also supports the efficient processing of spatial keyword queries which will take a geo-location and a set of keywords as arguments and will return related content that is matching the arguments. In real life applications, spatial keyword queries are being supported such as Google Maps where points of interest can be extracted, Twitter where tweets can be extracted and Foursquare where geo-tagged documents can be extracted. The researchers are giving importance spatial keyword querying. In which number if approaches have been proposed for proficiently processing spatial keyword queries. Some spatial keyword queries are getting great attention. These types are:

1. Boolean *k*NN query
2. Top-*k* *k*NN query
3. Boolean range query

4. 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.

5. 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 R-Tree, we 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.

3. Conclusion

In this report, we have surveyed a Fast Nearest Neighbor Search to search web objects 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 and Merging and distance browsing.

Reference

[1] I. De Felipe, V. Hristidis, and N. Rishe. Keyword search on spatial databases. In *ICDE*, pp. 656–665, 2008.
[2] D. Zhang, Y. M. Chee, A. Mondal, A. K. H. Tung, and M. Kitsuregawa. Keyword search in spatial databases: Towards searching by document. In *ICDE*, pp. 688–699, 2009.
[3] I.D. Felipe, V. Hristidis, and N. Rishe, "Keyword Search on Spatial Databases," Proc. Int'l Conf. Data Eng. (ICDE), pp. 656-665, 2008.

Licensed Under Creative Commons Attribution CC BY

[4] R. Hariharan, B. Hore, C. Li, and S. Mehrotra, "Processing Spatial- Keyword (SK) Queries in Geographic Information Retrieval (GIR) Systems," Proc. Scientific and Statistical Database Management (SSDBM), 2007.
[5] X. Cao, G. Cong, and C. S. Jensen. Retrieving top-k prestige-based relevant spatial web objects. *PVLDB*, 3(1):373–384, 2010.
[6] Y.-Y. Chen, T. Suel, and A. Markowetz. Efficient query processing in geographic web search engines. In *SIGMOD*, pp. 277–288, 2006.
[7] Y.-Y. Chen, T. Suel, and A. Markowetz. Efficient query processing in geographic web search engines. In *SIGMOD*, pp. 277–288, 2006.
[8] G. Cong, C. S. Jensen, and D. Wu. Efficient retrieval of the top-k most relevant spatial web objects. *PVLDB*, 2(1):337–348, 2009.
[9] G. Cong, C. S. Jensen, and D. Wu. Efficient retrieval of the top-k most relevant spatial web objects. *PVLDB*, 2(1):337–348, 2009.
[10] I. De Felipe, V. Hristidis, and N. Rishe. Keyword search on spatial databases. In *ICDE*, pp. 656–665, 2008.
[11] Y. Zhou, X. Xie, C. Wang, Y. Gong, and W.-Y. Ma, "Hybrid Index Structures for Location-Based Web Search," Proc. Conf. Information and Knowledge Management (CIKM), pp. 155-162, 2005.
[12] I.D. Felipe, V. Hristidis, and N. Rishe, "Keyword Search on Spatial Databases," Proc. Int'l Conf. Data Eng. (ICDE), pp. 656-665, 2008.
[13] 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.
[14] N. Beckmann, H. Kriegel, R. Schneider, and B. Seeger, "The R- tree: An Efficient and Robust Access Method for Points and Rectangles," Proc. ACM SIGMOD Int'l Conf. Management of Data, pp. 322-331, 1990.
[15] G.R. Hjaltason and H. Samet, "Distance Browsing in Spatial Databases," ACM Trans. Database Systems, vol. 24, no. 2, pp. 265-318, 1999.

Licensed Under Creative Commons Attribution CC BY