International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 04 Issue 06
Dec 2016

Fast Nearest Neighbor Search With Keywords

[1]U Jabeen, [2]Mrs.T.Neetha

[1] M.Tech Student, Dept of CSE, Brilliant grammar school educational institutions group of institutions integrated campus, T.S, India

[2]Associate Professor, Dept of CSE, Brilliant grammar school educational institutions group of institutions integrated campus, T.S, India

Abstract

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 "steak, spaghetti, brandy" all at the same time. Currently, the best solution to such queries is based on the IR2-tree, which, as shown in this paper, has a few deficiencies that seriously impact its efficiency. Motivated by this, we develop a new access method called the spatial inverted index that extends the conventional inverted index to cope with multidimensional data.

INTRODUCTION

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.

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 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 restaurantsin 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 one of the query keywords. 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.

LITERATURE SURVEY

Literature survey is the most important step in software development process. Before developing the tool it is necessary to determine the time factor, economy n company strength. Once these things r satisfied, ten next steps are to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need lot of external support. This support can be obtained from senior programmers, from book or from websites. Before building the system the above consideration r taken into account for developing the proposed system.

IR2-Tree

As mentioned before, the IR2-tree [12] combines the R-treewith signature files. Next, we will review what is a signature file before explaining the details of IR2-trees. Our discussion assumes the knowledge of R-trees and the best-first algorithm for NN search, both of which are well-known techniques in spatial databases. Signature file in general refers to a hashing-based framework, whose instantiation is known as superimposed coding (SC), which is shown to be more effective than other]. It is designed to perform membership tests: determine whether a query word w exists in a set W of words. SC is conservative, in the sense that if it says “no”, then w is definitely not in W. If, on the other hand, SC returns “yes”, the true answer can be either way, in which case the whole W must be scanned to avoid a false hit. In the context of SC works

in the same way as the classic technique of bloom filter.

Theconcrete values of l and m affect the space cost and false hit probability, as will be discussed later.Given a query keyword w, SC performs the membership test in W by checking whether all the 1s of hðwÞappear at the same positions in the signature of W. If not, it is guaranteed that w cannot belong to W. Otherwise, the test cannot be resolved using only the signature, and a scan of W follows. A false hit occurs if the scan reveals that W actually does not contain w

The IR2-tree is an R-tree where each (leaf or nonleaf) entry E is augmented with a signature that summarizes the union of the texts of the objects in the subtree of E.

demonstrates an example based on the data set of and the hash values in Fig. 2. The string 01111 in the leaf entry p2, for example, is the signature of Wp2 ¼ fb; dg The string 11111 in the non leafentry E3 is the signature of Wp2 [Wp6 , namely, the setof all words describing p2 and p6.

It continues until no more entry remains to be processed. In Fig. 3, assume that the query point q hasa keyword set Wq ¼ fc; dg. It can be verified that the algorithm must read all the nodes of the tree, and fetch the documents of p2, p4, and p6 (in this order). The final answer is p6, while p2 and p4 are false hits. Based on Inverted IndexesInverted indexes (I-index) have proved to be an effective access method for keyword-based document retrieval. In the spatial context, nothing prevents us from treating the

Related Work

The R-trees allow us to remedy an awkwardness in the way NN queries are processed with an I-index. Recall that, to answer a query, currently we have to first get all the points carrying all the query words in Wq by merging several lists (one for each word in Wq). This appears to be unreasonable if the point, say p, of the final result lies fairly close to the query point q. It would be great if we could discover p very soon in all the relevant lists so that the algorithm can terminate right away.

This would become a reality if we could browse the lists synchronously by distances as opposed to by ids. In particular, as long as we could access the points of all lists in ascending order of their distances to

q (breaking ties by ids), such a p would be easily discovered as its copies in all the lists would definitely emerge consecutively in our access order. So all we have to do is to keep counting how many copies of the same point have popped up continuously.

Reporting the point once the count reaches jWqj. At any moment, it is enough to remember only one count, because whenever a new point emerges, it is safe to forget about the previous one.As an example, assume that we want to perform NN search whose query point, and whose Wq equals fc; dg.

However, we must coordinate the execution of best-first on jWqj R-trees to obtain a global access order. This can be easily achieved by, for example, at each step taking a "peek" at the next point to be returned from each tree, and output the one that should come next globally. This algorithm is expected to work well if the query keyword nsetWq is small.

For sizableWq, the large number of random accesses it performs may overwhelm all the gains over the sequential algorithm with merging. A serious drawback of the R-tree approach is its space cost. Notice that a point needs to be duplicated once for every word in its text description, resulting in very expensive space consumption. In the next section, we will overcome the problem by designing a variant of the inverted index that supports compressed coordinate embedding.

Distance browsing is easy with R-trees. In fact, thebest-first algorithm is exactly designed to output data points in ascending order of their distances to q. However ,we must coordinate the execution of best-first on jWqj R-trees to obtain a global access order. This can be easily achieved by, for example, at each step taking a"peek" at the next point to be returned from each tree, and output the one that should come next globally. This algorithm is expected to work well if the query keyword set Wq is small. For sizable Wq, the large number of random accesses it performs may overwhelm all the gains over the sequential algorithm with merging.

Building R-Trees:

Remember that an SI-index is no more than a compressed version of an ordinary inverted index with coordinates embedded, and hence, can be queried in the same way

as described in Section by merging several inverted lists. In the sequel, we will explore the option of indexing each inverted list with an R-tree. As explained in, these trees allow us to process a query by distance browsing, which is efficient when the query keyword set Wqis small. Our goal is to let each block of an inverted list be directly a leaf node in the R-tree.

This is in contrast to the alternative approach of building an R-tree that shares nothing with the inverted list, which wastes space by duplicating each pointin the inverted list. Furthermore, our goal is to offer twosearch strategies simultaneously:

The property makes it possible to build good R-trees without destroying the Z-value ordering of any list. Specifically, we can (carefully) group consecutive points of a list into MBRs, and incorporate all MBRs into an R-tree. The proximity-preserving nature of the Z-curve will ensure that the MBRs are reasonably small when the dimensionality is low. For example, assume that an inverted list includes all the points, sorted in the order shown.

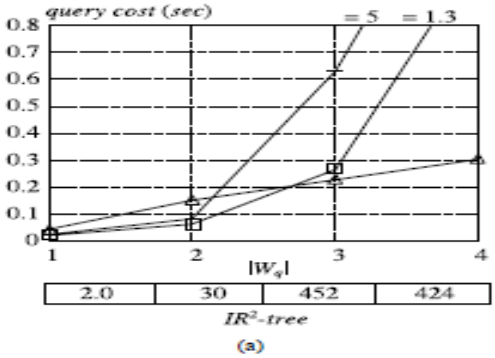
EXPERIMENTS

In the sequel, we will experimentally evaluate the practical efficiency of our solutions to NN search with keywords, and compare them against the existing methods.Competitors. The proposed SI-index comes with two query algorithms based on merging and distance browsing respectively. We will refer to the former as SI-m and the other as SI-b. Our evaluation also covers the state-of-the-art IR2-tree; in particular, our IR2-tree implementation is the fast variant developed in which uses longer signatures for higher levels of tree. Furthermore, we also include the method, named index file R-tree (IFR) henceforth, which, as discussed in indexes each inverted list (with coordinates embedded) using an R-tree, and applies distance browsing for query processing. IFR can be regarded as an uncompressed version of SI-b.

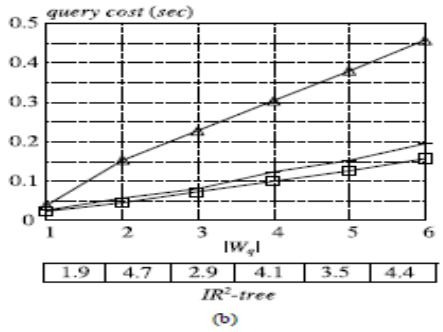
Data. Our experiments are based on both synthetic and real data. The dimensionality is always 2, with each axis consisting of integers from 0 to 16; 383. The synthetic category has two data sets: Uniform and Skew, which differ in the

International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 04 Issue 06
Dec 2016

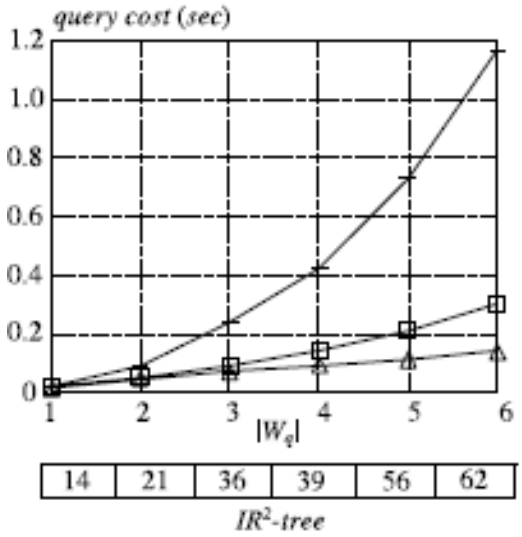
distribution of data points, and in whether there is a correlation between the spatial distribution and objects' text documents. Specifically, each data set has 1 million points. Their locations are uniformly distributed in Uniform, whereas in Skew, they follow the Zipf distribution. For bot data sets, the vocabulary has 200 words, and each word.



(a)



(b)



(c)

appears in the text documents of 50k points. The differenceis that the association of words with points is completely random in Uniform, while in Skew, there is a pattern of "word-locality": points that are spatially close have almost identical text documents. Our real data set, referred to as Census below, is a combination of a spatial data set published by the US Census Bureau,4 and the web pages from Wikipedia.5 The spatial data set contains 20;847 points, each of which represents a

county subdivision. We use the name of the subdivision to search for its page at Wikipedia, and collect the words there as the text description of the corresponding data point. All the points, as well as their text documents, constitute the data set Census.

p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 04 Issue 06
Dec 2016

CONCLUSION

We have seen plenty of applications calling for a searchengine 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 (SI-index). Not only that the SI-index is fairly space economical, but also it has the ability to perform keyword-augmented nearest neighbor search in time that is at the order of dozens of milli-seconds. Furthermore, as the SI-index 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.

Reference

[1] S. Agrawal, S. Chaudhuri, and G. Das, "Dbxplorer: A System for Keyword-Based Search over Relational Databases," Proc. Int'l Conf. Data Eng. (ICDE), pp. 5-16, 2002.

[2] 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.

[3] G. Bhalotia, A. Hulgeri, C. Nakhe, S. Chakrabarti, and S. Sudarshan, "Keyword Searching and Browsing in Databases Using Banks," Proc. Int'l Conf. Data Eng. (ICDE), pp. 431-440, 2002.

[4] X. Cao, L. Chen, G. Cong, C.S. Jensen, Q. Qu, A. Skovsgaard, D.Wu, and M.L. Yiu, "Spatial Keyword Querying," Proc. 31st Int'l Conf. Conceptual Modeling (ER), pp. 16-29, 2012.

[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