International Journal of Computer Applications Technology and Research
Volume 4– Issue 1, 53 - 57, 2015, ISSN:- 2319–8656

Designing of Semantic Nearest Neighbor Search: Survey

Pawar Anita R.
Student BE Computer
S.B. Patil College of Engineering
Indapur, Pune,Maharashtra
India

Pansare Rajashree B.
Student BE Computer
S.B. Patil College of Engineering
Indapur, Pune, Maharashtra
India.

Mulani Tabssum H.
Student BE Computer
S.B. Patil College of Engineering
Indapur, Pune, Maharashtra
India

Bandgar Shrimant B.
Asst. Professor
S.B. Patil College of Engineering
Indapur, Pune, Maharashtra
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, and comes with algorithms that can answer nearest neighbor queries with keywords in real time. As verified by experiments, the proposed techniques outperform the IR2-tree in query response time significantly, often by a factor of orders of magnitude.

Keywords: spatial Index, K-mean, Merge multiple, keyword-based Apriori item-set, neighbour search

1. INTRODUCTION

1.1 Concept of spatial index

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 modelling entities of reality in a geometric manner[5][6][7][8]. 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 neighbour 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 neighbour query), but the nearest restaurant among only those providing all the demanded foods and drinks. 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 neighbour lies quite far away from the query point, while all the closer neighbours are missing at least one of the query keywords.

1.2. Concept of IR2-tree.

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[1]. It is until recently that attention was diverted to multidimensional data. The best method to date for nearest neighbour 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[2]. 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. The IR2-tree, however, also inherits a drawback of signature files: false hits[2]. That is, a signature file, due to its conservative nature, may still direct the search to some objects, even though they do not have all the keywords. The penalty thus caused is the need to verify an object whose satisfying a query or not cannot be resolved using only its signature, but requires loading its full text description, which is expensive due to the resulting random accesses. It is noteworthy that the false hit problem is not specific only to signature files, but also exists in other methods for approximate set membership tests with compact storage. Therefore, the problem cannot be remedied by simply replacing signature file with any of those methods.

1.3 Concept of merge multiple

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. Mean while, 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 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[2].

1.4 Concept of keyword-based nearest neighbour search

There are many process mining algorithms and representations, making it difficult to choose which algorithm to use or compare results. Process mining is essentially a machine learning task, but little work has been done on systematically analyze in algorithms to understand their fundamental properties, such a show much data are needed for confidence in mining. We propose a framework for analyzing process mining algorithms. Processes are viewed as distributions over traces of activities and mining algorithms as learning these distributions. The access to a large quantity of textual documents turns out to be effectual because of the growth of the digital libraries, web, technical documentation, medical data and more. These textual data comprise of resources which can be utilized in a better way. Text mining is major research field due to the need of acquiring knowledge from the large number of available text documents, particularly on the web. Both text mining and data mining are part of information mining and identical in some perspective. Text mining can be described as a knowledge intensive process in which a user communicates with a collection of documents. In order to mine large document collections, it is require pre-processing the text documents and saving the data in the data structure, which is suitable for processing it further than a plain text file. Information Extraction is defined as the mapping of natural language texts like text database, WWW pages, electronic mail etc. into predefined structured representation, or templates which, when filled, represent an extract of key information from the original text.

2. Problem Definition

Let P be a set of multidimensional points. As our goal is to combine keyword search with the existing location-finding services on facilities such as hospitals, restaurants, hotels, etc., we will focus on dimensionality, but our technique can be extended to arbitrary dimensionalities with no technical obstacle .We will assume that the points in P have integer coordinates, such that each coordinate ranges in [0, t], where t is a large integer. This is not as restrictive as it may seem, because even if one would like to insist on real-valued coordinates, the set of different coordinates represent able under a space limit is still finite and enumerable; therefore, we could as well convert everything to integers with proper scaling. As with, each point p 2 P is associated with a set of words, which is denoted as Wp and termed the document of p. For example, if p stands for a restaurant, Wp can be its menu, or if p is a hotel, Wp can be the description of its services and facilities, or if p is a hospital, Wp can be the list of its out-patient specialities. It is clear that Wp may potentially contain numerous words. Traditional nearest neighbour search returns the data point closest to a query point. Following, we extend the problem to include predicates on objects' texts. Formally, in our context, a nearest neighbour (NN) query specifies a point q and a set Wq of keywords (we refer to Wq as the document of the query). It returns the point in Pq that is the nearest to q, where Pq is defined as,

$$Pq=\{p2P|Wq\_Wp\} \quad (1)$$

In other words, Pq is the set of objects in P whose documents contain all the keywords in Wq. In the case where Pq is empty, the query returns nothing. The problem definition can be generalized to k nearest neighbour (kNN) search, which finds the k points in Pq closest to q; if Pq has less than k points, the entire Pq should be returned. For example, assume that P consists of 8 points whose locations are as shown in Figure 1a (the black dots), and their documents are given in Figure 1b. Consider a query point q at the white dot of Figure 1a with the set of keywords

$$Wq=\{c,d\} \quad (2)$$

Nearest neighbour search finds p6, noticing that all points closer to q than p6 are missing either the query keyword c or d. If k = 2 nearest neighbours are wanted, p8 is also returned in addition. The result is still {p6, p8} even if k increases to 3 or higher, because only 2 objects have the keywords c and d at the same time. We consider that the dataset does not fit in memory, and needs to be indexed by efficient access methods in order to minimize the number of I/Os in answering a 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 reflected by the convenience of modeling entities of reality in a geometric manner[5][6][7][8]. 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 all the demanded foods and drinks.

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 far away 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[1]. 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-[2], 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 considerable portion of the objects that do not contain all the query keywords, thus significantly reducing the number of objects to be examined.

The IR2-tree, however, also inherits a drawback of signature files: false hits. That is, a signature file, due to its conservative nature, may still direct the search to some objects, even though they do not have all the keywords. The penalty thus caused is the need to verify an object whose satisfying a query or not cannot be resolved using only its signature, but requires loading its full text description, which is expensive due to the resulting random accesses. It is noteworthy that the false hit problem is not specific only to signature files, but also exists in other methods for approximate set membership tests with compact storage. Therefore, the problem cannot be remedied by simply replacing signature file with any of those methods.

Data fusion and multicue data matching are fundamental tasks of high-dimensional data analysis. In this paper, we apply the recently introduced diffusion framework to address these tasks. Our contribution is three-fold: First, we present the Laplace- Beltrami approach for computing density invariant embeddings which are essential for integrating different sources of data. Second, we describe a refinement of the Nystro¨m extension algorithm called "geometric harmonics." We also explain how to use this tool for data assimilation. Finally, we introduce a multicue data matching scheme based on nonlinear spectral graphs alignment. The effectiveness of the presented schemes is validated by applying it to the problems of lip-reading and image sequence alignment..

3. Code Review Technique

There are so many techniques to apply the nearest elementas well as locations**.**

3.1 The k-means algorithm

The k-means algorithm is a simple iterative method to partition a given dataset into a user specified number of clusters, *k*. This algorithm has been discovered by several researchers across different disciplines, most notably Lloyd.

3.2 Support vector machines

In today's machine learning applications, support vector machines (SVM) [83] are considered must try—it offers one of the most robust and accurate methods among all well-known algorithms. It has a sound theoretical foundation, requires only a dozen examples for training, and is insensitive to the number of dimensions. In addition, efficient methods for training SVM are also being developed at a fast pace.

3.3 The Apriori algorithm

One of the most popular data mining approaches is to find frequent item sets from a transaction dataset and derive association rules. Finding frequent itemsets (itemsets with frequency larger than or equal to a user specified minimum support) is not trivial because of its combinatorial explosion. Once frequent itemsets are obtained, it is straightforward to generate association rules with confidence larger than or equal to a user specified minimum confidence.

3.4 The EM algorithm

Finite mixture distributions provide a flexible and mathematical-based approach to the modelling and clustering of data observed on random phenomena. We focus here on the use of normal mixture models, which can be used to cluster continuous data and to estimate the underlying density function. These mixture models can be fitted by maximum likelihood via the EM (Expectation–Maximization) algorithm.

3.5 CART

The 1984 monograph, "CART: Classification and Regression Trees," co-authored by Leo Breiman, Jerome Friedman, Richard Olshen, and Charles Stone, [9] represents a major milestone in the evolution of Artificial Intelligence, Machine Learning, non-parametric statistics, and data mining. The work is important for the comprehensiveness of its study of decision trees, the technical innovations it introduces, its sophisticated discussion of tree structured data analysis, and its authoritative treatment of large sample theory for trees[2]. While CART citations can be found in almost any domain, far more appear in fields such as electrical engineering, biology, medical research and financial topics than, for example, in

marketing research or sociology where other tree methods are more popular. This section is intended to highlight key themes treated in the CART monograph so as to encourage readers to return to the original source for more detail.

4. R-trees System

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 i.e., by merging several inverted lists. In the sequel, we will explore the option of indexing each inverted list with an R-tree[2]. As explained in these trees allow us to process a query by distance browsing, which is efficient when the query keyword set we is 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 point in the inverted list. Furthermore, our goal is to offer two search strategies simultaneously merging and distance browsing .As before, merging demands those points of all lists should be ordered following the same principle. This is not a problem because our design in the previous subsection has laid down such a principle: ascending order of Z-values. Moreover, this ordering has a crucial property that conventional id-based ordering lacks: preservation of spatial proximity. 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[2]. The proximity preserving nature of the Z-curve will ensure that the MBRs are reasonably small when the dimensionality is low.

5. Proposed System

Our treatment of nearest neighbor search falls in the general topic of spatial keyword search, which has also given rise to several alternative problems. A complete survey of all those problems goes beyond the scope of this project. 1. Strictly speaking, this is not precisely true because merging may need to jump across different lists; however, random I/Os will account for only a small fraction of the total overhead as long as a proper perfecting strategy is employed, e.g., reading 10 sequential pages at a time. Considered a form of keyword-based nearest neighbor queries that is similar to our formulation, but differs in how objects' texts play a role in determining the query result. Specifically, aiming at an IR flavor, the approach of computes the relevance between the documents of an object p and a query q. This relevance score is then integrated with the Euclidean distance between p and q to calculate an overall similarity of p to q. The few objects with the highest similarity are returned. In this way, an object may still be in the query result, even though its document does not contain all the query keywords. In our method, same as, object texts are utilized in evaluating a Boolean predicate, i.e., if any query keyword is missing in an object's document, it must not be returned. Neither approach subsumes the other, nor do both make sense in different applications

4.1 System Architecture

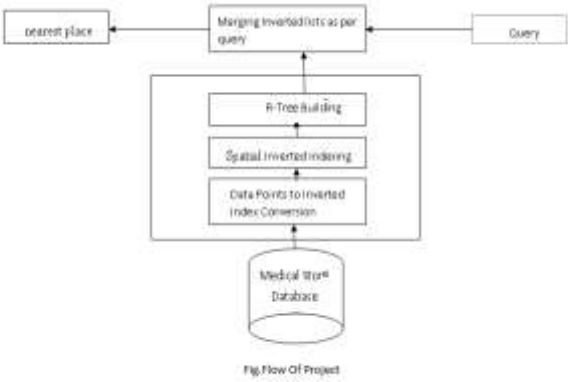

Fig Flow Of Project

As an application in our favor, consider the scenario where we want to find a close restaurant serving "steak, spaghetti and brandy", and do not accept any restaurant that does not serve any of these three items. In this case, a restaurant's document either fully satisfies our requirement, or does not satisfy at all. There is no "partial satisfaction", as is the rationale behind the approach of, In geographic web search, each webpage is assigned a geographic region that is pertinent to the webpage's contents. In web search, such regions are taken into account so that higher rankings are given to the pages in the same area as the location of the computer issuing the query (as can be inferred from the computer's IP address) . The underpinning problem that needs to be solved is different from keyword-based nearest neighbor search, but can be regarded as the combination of keyword search and range queries. Specifically, let P be a set of points each of which carries a single keyword. Given a set Wq of query keywords (note: no query point q is needed), the goal is to find m = |Wq| points from P such that (i) each point has a distinct keyword in Wq, and (ii) the maximum mutual distance of these points is minimized (among all subsets of m points in P fulfilling the previous condition).

In other words, the problem has a "collaborative" nature in that the resulting m points should cover the query keywords together. This is fundamentally different from our work where there is no sense of collaboration at all, and instead the quality of each individual point with respect to a query can be quantified into a concrete value. Proposed collective spatial keyword querying, which is based on similar ideas, but aims at optimizing different objective functions[4][12].

6. Conclusion

This paper describe, A user-set minimum support decides about which rules have high support .Once the rules are selected, they are all treated the same, irrespective of how high or how low their support. Their locations are uniformly distributed in Uniform, whereas in Skew, they follow the Zip f distribution. For both datasets, the vocabulary has 200 words, and each word appears in the text documents of 50kpoints. The difference is 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.

7. REFERENCES

[1] S. Agrawal , S.Chaudhuri,and G.Das.Dbxplorer:A system for keyword-based search over relational databases. In Proc.Of International Conference on DataEngin-eering (ICDE), pages 516, 2002.Ding, W. and Marchionini, G. 1997 A Study on Video Browsing

Strategies. Technical Report. University of Maryland at College Park.

[2] N.Beckmann, H.Kriegel,R.Schneider, and B.Seeger.The R*-tree:Ane_cient and robust access method for points and rectangles.In Procof ACM Management of Data(SIGMOD), pages 322331, 1990.Tavel, P. 2007 Modeling and Simulation Design. AK Peters Ltd.

[3] G. Bhalotia, A. Hulgeri, C. Nakhe, S. Chakrabarti, and S. Sudarshan. Keyword searching and browsing in databases using banksInProc.of International Conf- erence on Data Engineering (ICDE),pages 431440, 2002.

[4] 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 1629, 2012.Brown, L. D., Hua, H., and Gao, C. 2003. A widget framework for augmented interaction in SCAPE.

[5] X.Cao, G.Cong, and C.S.Jensen.Retrieving top-k prestige-based relevant spatial web objects. PVLDB, (1):373384, 2010

[6] Y .-Y .Chen,T.Suel, and A. Markowetz.E_cient query processing geographic web search engines. In Proc.of ACM Management of Data(SIGMOD),pages277288, 2006

[7] G .Cong, C. S .Jensen, and D. Wu. E_cient retrieval of the top-k most relevant spatial web objects. PVLDB, (1):337348, 2009.

[8] C .Faloutsos and S .Christodoulakis Signature _les:An access method for docu-6mentsand its analytical performance evaluation.ACM Trans-actions on Infor- mation Systems (TOIS),2(4):267288, 1984.

[9] I.D .Felipe, V.Hristidis, and N.Rishe. Keyword search on spatial databases.In Proc.of International Conference on Data Engineering(ICDE) pages 656665, 2008

[10] R. Hariharan,B. Hore, C. Li, and S.Mehrotra.Processing spatial keyword(SK) in geographic information retrieval (GIR) systems. In Proc. of Scienti_c and Statistical Database Management (SSDBM), 2007.

[11] X. Cao, G. Cong, C. S. Jensen, and B. C. Ooi .Collective spatial keyword quer-ying .In Proc. of ACM Management of Data (SIGMOD),pages 373384, 2011.

[12] I.Kamel and C.Faloutsos. Hilbert R-tree : An improved r-tree using fractals. In Proc. of Very Large Data Bases(VLDB), pages 500509, 1994.

[13] B .Chazelle, J .Kilian, R .Rubinfeld, and A.Tal. The bloomier _lter:ane_cient data structure for static support lookup tables. In Proc.of the Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pages 3039, 2004.