© 2015 IJSRSET | Volume 1 | Issue 2 | Print ISSN : 2395-1990 | Online ISSN : 2394-4099
Themed Section: Engineering and Technology

Fast Searching of Nearest Neighbor using Key Values in Data Mining

Sri Vidhya. A, Prof. Ashwin. M
Dept of CSE, Adhiyamaan College of engineering, Hosur, Tamilnadu, India

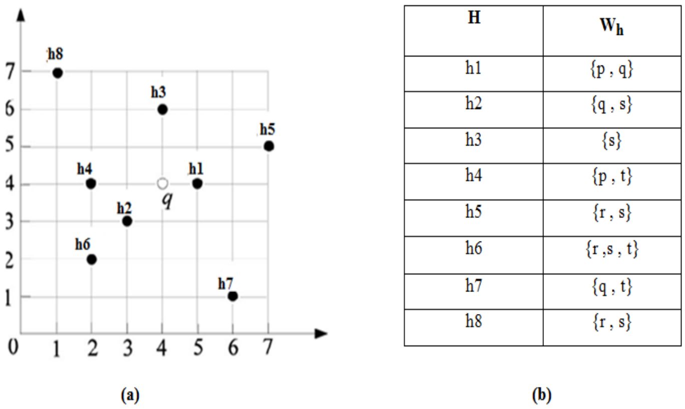
ABSTRACT

Spatial query which focus only on the geometrics properties of an object like points, rectangle etc. Now a day's many new applications which involve the queries that completely aim to return an object which satisfy equally on spatial predicate and their associated text. Spatial query takes the given location and a keyword as the input and finds the object that matches the both spatial predicate and the text related to the given query. Some of the spatial queries are range search and nearest neighbor retrieval which includes only geometric properties of an object. For example, In case of considering all the hotels, a nearest neighbor query would find for the hotel which is near, along with menu that user required to have in hotel among all the hotels in particular location simultaneously. At present the better solution is based on IR2-Tree which as few drawbacks that affect the efficiency in query retrieval. So we develop a new method Spatial inverted index that cope with 3D data to answer the nearest neighbor query using keyword along with key values in real time. Searching nearest neighbor query using key values will result in quick response of query when compared to keyword in real time.

Keywords: Spatial Query, Nearest Neighbour Search, IR^2-Tree, Key Value and Spatial inverted index

I. INTRODUCTION

Knowledge and Data Engineering focus mainly on extraction of data from database. i.e. Data mining .Data mining helps in extraction of most important data from the huge collection of data that is stored i.e. data warehouse. Data warehouse is a subject-oriented, integrated, time-variant and non-volatile collection of data in support of management's decision making process. A spatial database is used to store huge amount of space related data such as maps, medical imaging data etc. and manages multidimensional objects (such as points, rectangles, etc.), and provides quick access to those objects based on different choice criteria. The importance of spatial databases [1] is it provides a convenient path to model the entities of reality in a geometric manner. As an example, locations of restaurants, hotels, hospitals and then on are typically shown as points in an exceedingly map, whereas larger extents like parks, lakes, and landscapes typically as a mixture of rectangles. Several functionalities of a spatial information are helpful in numerous ways that in specific contexts. For example, in an exceedingly geographic data system, vary search may be deployed to search out all restaurants in a particular space, whereas nearest neighbour retrieval will discover the restaurant nearest to a given address.



H	W$_h$
h1	{p , q}
h2	{q , s}
h3	{s}
h4	{p , t}
h5	{r , s}
h6	{r ,s , t}
h7	{q , t}
h8	{r , s}

Figure 1: (a) shows the location of hotels and (b) given their associated text i.e menu's.

Let H be a set of multidimensional points in Figure 1.(a) which indicates the set of hotels around the query point 'q' i.e. static location. As our aim is to combine keyword search with the existing location-searching services on facilities such as hospitals, restaurants, hotels, etc., we will focus on 3D data's, but our technique can be extended to arbitrary dimensionalities with no technical obstacle. We will assume that the points in H have

IJSRSET151180 | Received: 24 March 2015 | Accepted: 11 March 2015 | March-April 2015 [(1)2: 81-85]

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 integer with proper scaling.

Each point in H in Figure 1. (b) is associated with a set of words (h1,h2,h3,h4,h5,h6,h7,h8), and termed the document of H which is denoted as W_h. For example, if H stands for a restaurant, W_h can be its menu, or if H is a hotel, W_h can be the description of its services and facilities, or if H is a hospital, W_h can be the list of its out-patient specialties. It is clear that W_h may potentially contain numerous words. Traditional nearest neighbor search [3] returns the data point closest to a query point. Following [12], we extend the problem to include predicates on objects' texts. Formally, in our context, a nearest neighbor (NN) query specifies a point q and a set W_q of keywords (we refer to W_q as the document of the query). It returns the point in H_q that is the nearest to q, Where H_q is defined as

$$H_q = \{h \epsilon H \mid W_q \dot{\subseteq} W_h\} \qquad (1)$$

II. LITERATURE REVIEW

TABLE 1: Content of Previous Work in Table

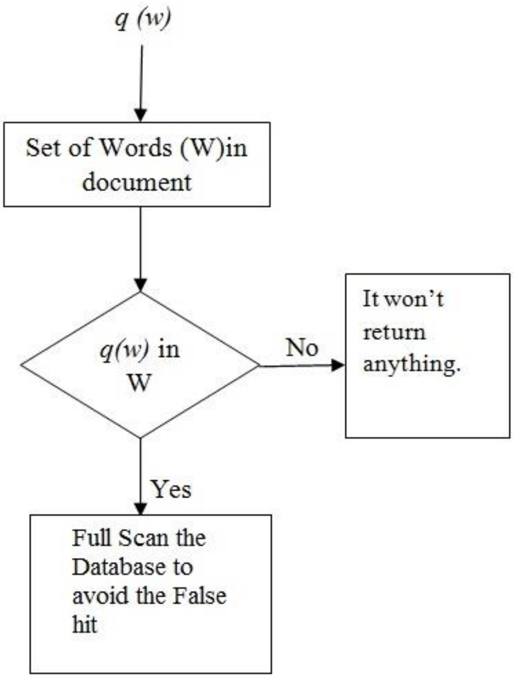
AUTHOR	PAPER TITLE	CONCEPT	APPROACHES
G. Cong, C.S. Jensen, and D. Wu[7]	Efficient Retrieval of the Top-k Most Relevant Spatial Web Objects	Keyword based nearest neighbor Query	Computes the relevance between the object p and query q.It returns Partial satisfaction.
Y.-Y. Chen, T. Suel, and A. Markowetz[8] R. Hariharan, B. Hore, C. Li, and S. Mehrotra, [13]	Efficient Query Processing in Geographic Web Search Engines.[8] Processing Spatial-Keyword (SK) Queries in Geographic Information Retrieval (GIR) Systems[13]	Combines keyword search and range queries	Geographic web search returns based on ranking
I.D. Felipe, V. Hristidis, and N. Rishe,[9]	Keyword Search on Spatial Databases[9]	Nearest neighbor search using keywords by IR^2-Tree	IR^2-Tree preserves object's spatial proximity, which solve the spatial query efficiently. Here there is no partial satisfaction in returning the output.
D. Zhang, Y.M. Chee, A. Mondal, A.K.H. Tung, and M. Kitsuregawa,[14]	Keyword Search in Spatial Databases: Towards Searching by Document	Search for m-closest keyword	Collaborative in nature, resulting m points should cover the query keyword together.
X. Cao, G. Cong, C.S. Jensen, and B.C. Ooi,[5]	Collective Spatial Keyword Querying	Collective Spatial keyword querying	Similar idea as the above but it mainly focuses at optimizing different objective function.
X. Cao, G. Cong, and C.S. Jensen, [4]	Retrieving Top-k Prestige-Based Relevant Spatial Web Objects	Prestige based spatial keyword	Evaluates the similarity of an object p to a query q by considering the objects p neighbours.

III. METHODS AND MATERIAL

A. Ir^2-Tree

Information Retrieval R-tree which is the state of answering the NN Queries.IR2-tree is the combination of the R-tree[2] with signature files. First we will see about Signature file. Signature file refers to a hash-based framework, whose instantiation in [9] is called as superimposed coding (SC), which is shown to be more effective than other instantiation [5]. It is charted to perform membership tests that is to determine whether a query word q exists in a set of Words W[11].If it returns “no”, then q is surely not in W. If SC returns “yes”, then q is in W, to avoid a false hit W is scanned as a whole if it returns “yes”. The IR2-tree is an R-tree where each leaf or non-leaf entry E is augmented with a signature which summarizes the union of the texts in the sub tree. On traditional R-trees, the best-first algorithm [12] is a well-known solution to NN search. It is now directly to adapt it to IR2-trees.

B. Working of Signature file

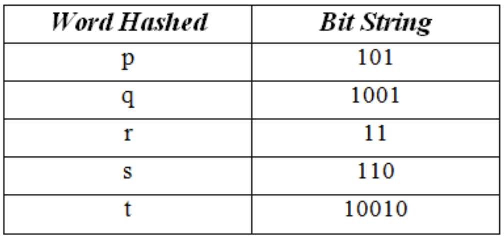


q (w)
Set of Words (W)in document
q(w) in W
No
It won’t return anything.
Yes
Full Scan the Database to avoid the False hit

Figure 2: Working of SC

For example, Consider the Table.2 which describes the bit string computation with l=5 and m=2.In that the bit string h(p) of p, the third and fifth (counting from left) bits are set to 1. As mentioned earlier, the bit signature of a set W of words simply ORs the bit strings of all the members of W. For instance, the signature of a set f(p,q) equals 01101, while that of f(q, s) equals 01111.Given a question keyword w, SC performs the membership ,take a glance at in W by checking whether or not or not all the 1‟s of h(w) appear at constant positions at intervals the signature of W.

TABLE 2:- Example of bit string computation with l =5 and m = 2.



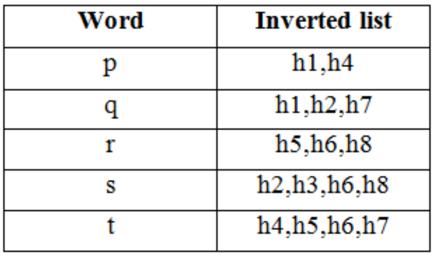
Word Hashed	*Bit String*
p	101
q	1001
r	11
s	110
t	10010

If not, it’s secured that w cannot belong to W. Otherwise; the take a glance at can’t be resolved victimization alone the signature, and a scan of W follows. [10]A false hit happens if the scan reveals that W really doesn’t contain w. As associate example, assume that we tend to would like to envision whether or not or not word c may be a member of set mistreatment the set’s signature 01101. Since the fourth bit of h(r) = zero11 is one but that of 01101 is 0, SC in real time reports “no”. As another example, take under consideration the membership take a glance at of r in whose signature is 01111. This time, SC returns “yes” as a results of 01111 has 1‟s within the least the bits where h(r) is on the brink of 1; as a result, a full scan of the set is required to verify that this will be a false hit.

C. Inverted Indexes

The inverted index system is also a central part of a typical software system categorization formula. A goal of a research engine performance is to optimize the speed of the query: Understand the documents where word happens. Once associate index is developed, that provisions lists of words per document; it’s next inverted to develop associate inverted index [10]. Querying the index would wish serial iteration through each document and to each word to verify the same document. The time, memory and method property to execute such a question don’t appear to be forever in theory realistic. Instead of listing the words per article inside the index, the inverted index system is developed that lists the documents per word. The inverted index created, the question can presently be determined by jumping to the word id inside the inverted index. Table.3. describes the inverted index of above example i.e for the associated text (p,q,r,s,t).

Table.3. Example of an inverted index



Word	Inverted list
p	h1,h4
q	h1,h2,h7
r	h5,h6,h8
s	h2,h3,h6,h8
t	h4,h5,h6,h7

D. Spatial inverted index (SI-index).

Spatial inverted index [6] will merge multiple list by their Id's instead we will additionally we provide the R-trees [2] to browse the purposes of all relevant hotel lists that near to the static point i.e. particular location. This access methodology will incorporates the purpose of coordinates into a traditional inverted index.

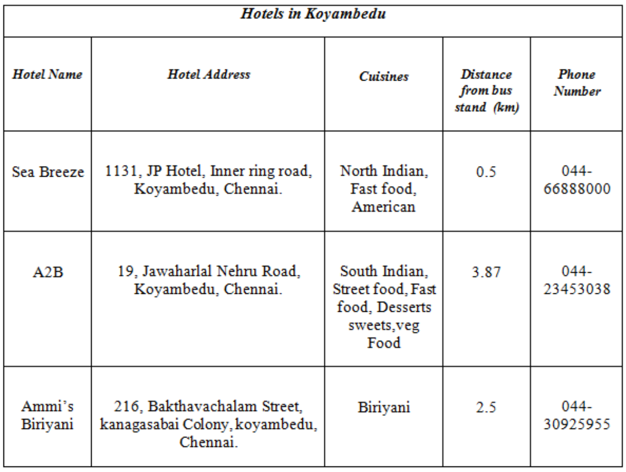
I. RESULTS AND DISCUSSION

The Modules are implemented by using HTML, JAVA, JAVA Script, JSP, and Database. In the implementation part, a web application is created with the page where the user can post their query and retrieve their response which satisfies the user requirements. For example, it would be fairly useful if a search engine can find the nearest hotel that offers user menu all at same time but this cannot be done in case of global. This can achieved locally within a particular zone with N location by making one location as static location among the N location.

We create an application with different web pages like Admin login, Admin data entry, User Registration, User Login, User Keyword Search etc. This application takes particular ten location of Chennai City like Koyambedu, T-Nagar, Avadi, Adyar, Guindy, Numgambakkam, Ambattur, Poonamalee, Anna Nagar, and Ashok Nagar.

Data Set of Each location are collected and stored in database which includes the Hotel And its details like name, address ,Specialize , phone number and distance of the particular hotel from Koyambedu by making that particular location as the Static location.Table.4. Describes the data set of location koyambedu. By this similar way N data set of each location is collected and stored in database.

Table 4:.Example for data set in location Koyambedu.

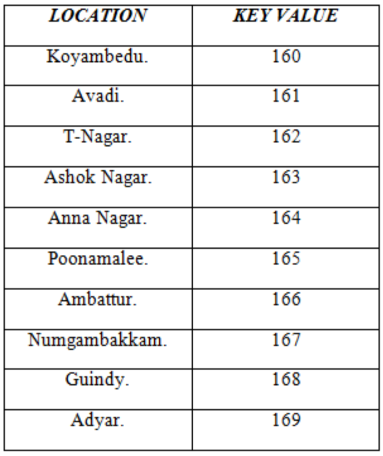


Hotels in Koyambedu				
Hotel Name	***Hotel Address***	***Cuisines***	***Distance from bus stand (km)***	***Phone Number***
Sea Breeze	1131, JP Hotel, Inner ring road, Koyambedu, Chennai.	North Indian, Fast food, American	0.5	044-66888000
A2B	19, Jawaharlal Nehru Road, Koyambedu, Chennai.	South Indian, Street food, Fast food, Desserts sweets,veg Food	3.87	044-23453038
Ammi's Biriyani	216, Bakthavachalam Street, kanagasabai Colony, koyambedu, Chennai.	Biriyani	2.5	044-30925955

A. Key Value Implementation

In our application, we deal with 10 locations which is specified above. Admin has the full rights to edit, update the data set in database. While adding the location to the database, a key value is generated for each location in sequence manner. For example, Table.5. Describes the key value for each location. This key value act as the primary key in the database for a particular location.

Table.5.Examples for key value assigned to locations.



LOCATION	*KEY VALUE*
Koyambedu.	160
Avadi.	161
T-Nagar.	162
Ashok Nagar.	163
Anna Nagar.	164
Poonamalee.	165
Ambattur.	166
Numgambakkam.	167
Guindy.	168
Adyar.	169

B. Building R-Tree

R-Tree is a real tree which is the graphical representation of the data set that is stored in the particular location. R-Tree is built for each location for user convenient. Figure 3 describes the R-Tree for dataset in Table.4.

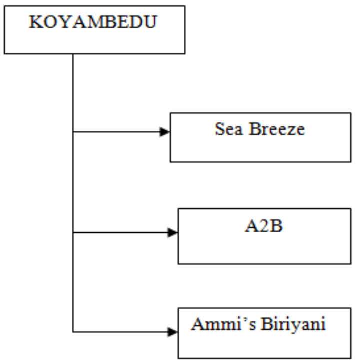


Figure.3: R-Tree for Koyambedu

Figure.3. is the R-Tree for location koyambedu i.e. here the root of the tree is location which act as a static point location and the hotels in that particular location are the child of the tree.

C. Result

New User registers them by giving their details in the user registration page. After registration user gets its own ID and password to access the application to search the nearest hotel form his location with his required food to have. For example, User searches with the associated text i.e. location as “KOYAMBEDU” and its food as “NORTH INDIAN” in the web page.

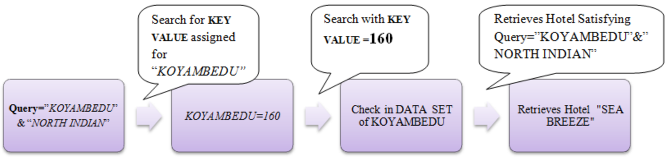

Query="KOYAMBEDU" &"NORTH INDIAN"
Search for KEY VALUE assigned for "KOYAMBEDU"
KOYAMBEDU=160
Search with KEY VALUE =160
Check in DATA SET of KOYAMBEDU
Retrieves Hotel Satisfying Query="KOYAMBEDU"&" NORTH INDIAN"
Retrieves Hotel "SEA BREEZE"

Figure.4: Query Processing Using Key Value

Now it retrieves both the hotels which are near to koyambedu bus stand along with user required food “NORTH INDIAN” at the same time. This is achieved by the key value which is assigned to each location. When the user gives the location as keyword to search for hotels which in turn call the key value that is assigned to that particular location and searches the database through the key value and retrieve the data that satisfy the user requirement from the data set in the database.

II. CONCLUSION

Now days, Google search engine plays a vital role in case of searching the particular thing that the user want with user query. Conventionally, queries solely focus on object’s geometric properties. We have seen some fashionable applications that have an ability to pick objects supported each of their geometric coordinates and their associated texts. In this paper, we use a style variant of inverted index which is optimized for third-dimensional points, and is thus named the spatial inverted index (SI index). This access technique incorporates point coordinates into a standard inverted index with little additional space, attributable to a delicate compact storage theme. Meanwhile, an SI-index preserves the spatial locality of information points, and comes with an R-tree designed on each inverted list at space overhead. Moreover, as a result of the SI-index depends on the quality technology of inverted index, which satisfy the user query with quick response using key values along with keyword search in the database.

III. 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 Data Engineering (ICDE), pages 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.” In Proc. of ACM Management of Data (SIGMOD), pages 322–331, 1990.

[3] G. Bhalotia, A. Hulgeri, C. Nakhe, S. Chakrabarti, and S. Sudarshan. “Keyword searching and browsing in databases using banks.” In Proc. of International Conference on Data Engineering (ICDE), pages 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.” In ER, pages 16–29, 2012.

[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] X. Cao, G. Cong, C. S. Jensen, and B. C. Ooi. “Collective spatial keyword querying.” In Proc. of ACM Management of Data (SIG- MOD), pages 373–384, 2011.

[7] 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 30–39, 2004.

[8] 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 277–288, 2006.

[9] E. Chu, A. Baid, X. Chai, A. Doan, and J. Naughton. “Combining keyword search and forms for ad hoc querying of databases.” In Proc. of ACM Management of Data (SIGMOD), 2009.

[10] 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.

[11] C. Faloutsos and S. Christodoulakis. “Signature files: An access method for documents and its analytical performance evaluation.” ACM Trans- actions on Information Systems (TOIS), 2(4):267–288, 1984.

[12] I. D. Felipe, V. Hristidis, and N. Rishe. “Keyword search on spatial databases.” In Proc. of International Conference on Data Engineering (ICDE), pages 656–665, 2008.

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

[14] D. Zhang, Y.M. Chee, A. Mondal, A.K.H. Tung, and M. Kitsuregawa, “Keyword Search in Spatial Databases: Towards Searching by Document,” Proc. Int’l Conf. Data Eng. (ICDE), pp. 688-699, 2009.