NOVEL CACHE SEARCH TO SEARCH THE KEYWORD COVERS FROM SPATIAL DATABASE

[1]Asma Akbar, [2]Mohammed Naqueeb Ahmad

[1]M.Tech Student, Department of CSE, Deccan College of Engineering and Technology, Darussalam Road, Mandal Nampally, Hyderabad, Telangana, India

[2]Assistant Professor, Department of CSE, Deccan College of Engineering and Technology, Darussalam Road, Mandal Nampally, Hyderabad, Telangana, India

ABSTRACT– ***Recently, to search spatial objects from spatial dataset, we are using rating of the spatial objects along with distance. At present, everyone wants to get searched results as soon as possible it means users expecting less time to get searched spatial object results. To search best keyword cover from spatial database, we have a solution i.e., keyword nearest neighbor expansion (keyword-NNE) algorithm. By using this algorithm, users can get the optimal results but, it may take some time to search the results. For that reason, in this paper we propose a novel cache search method along with Keyword-NNE algorithm. Through this new method we can get the optimal keyword cover results within the less time.***

1. INTRODUCTION

Spatial data mining is a special sort of information mining. The most important difference between records mining and spatial records mining is that during spatial information mining duties we use not most effective non-spatial attributes (as it's far standard in records mining in non-spatial information), but additionally spatial attributes.

Spatial statistics mining is the utility of information mining methods to spatial records. The goal of spatial information mining is to discover patterns in information with respect to geography. The monstrous explosion in geographically referenced data occasioned through trends in IT, digital mapping, far off sensing, and the worldwide diffusion of Geographic Information Systems (GIS) emphasize the importance of growing data-pushed inductive strategies to geographical analysis and modeling. Today, widely used of engines like Google has made it practical to jot down spatial queries in a new manner. Traditionally, queries focus on objects handiest geometric residences, for example whether a point is in rectangle or how two factors are near from every other. Few novel applications let in users to search items based totally on both of their geometric coordinates and their related texts. Such sort of queries called as spatial keyword query. For example, if a search engine can be used to discover nearest resort that provide facilities such as pool and internet at the same time. From this query, we may want to first achieve the entire inn whose offerings contain the set of keywords, after which find the nearest one from the retrieved eating place. The main downside of this technique is that, on the hard input they do not provide real time solution. For example, from the query factor the real neighbor lies pretty a ways away, while all of the closer friends are lacking as a minimum one of the query keywords. Spatial key-word queries have not been widely explored. In the past years, the group of people has confirmed interest in reading keyword search in to multidimensional records. The satisfactory technique for nearest neighbor seek with keywords is because of Felipe et al. They combine the spatial index R-tree and signature record. So they developed a shape referred to as IR^2-tree. This tree has the potential of each R-tree and signature documents. Like R-tree it shops the spatial proximity of object and like signature record it filters the ones items that do not include all query keywords.

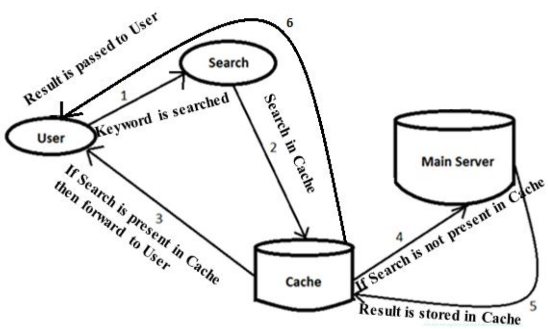


Figure 1: System Architecture

In spatial database, every tuple acts as a spatial objects which particularly assists with key-word (s), indicate the data including functions and also includes spatial information in conjunction with range and longitude values of the region. Performing Querying on such form of statistics is known as Best Keyword Cover querying. This sort of search is referred to as Best Keyword Cover search. Several research have centered on the spatial key-word seek issue. This paper ambitions to locate the quantity of individual items, each of the items near the query vicinity and the key phrases which are relevant to the set of key phrases given. This paper investigates the general shape of mCK query known as BKC, which particularly affords inter-object distance as well as key-word score. This is accomplished because of the commentary, as there is a growing availability and significance of key-word rating in finding the better decision. For processing the BKC query we expand algorithms, Baseline algorithm and K-NNE algorithm. These algorithms use R-trees for indexing the objects. The Baseline algorithm principal goal is to combine the child nodes in hierarchical tiers of R-bushes and the concern is taken, by using combining their baby nodes for producing the new candidates. Here for locating BKC Query, the wide variety of Candidate keywords increases, for this reason the performance of the set of rules drops dramatically. To triumph over this drawback, we increase a scalable set of rules known as (K-NNE) which applies an exclusive method, where it retrieves keyword nearest neighbor via combining both key-word search and nearest neighbor search. It introduces the idea of key-word score, which allows in higher selection making.

2. RELATED WORK

In a spatial database system, spatial joins are one of the most important operations for combining spatial objects of several relations. Contrary to window queries, execution time is super linear in the number of objects and therefore, spatial joins may require more than one access to each of the objects. Moreover, approaches suggested for performing traditional joins, e.g. natural-joins, are not applicable to spatial joins. Thomas Brinkhoff, Hans-Peter Kriegel and Bernhard Seeger presented the first detailed study of spatial join processing using R-trees, particularly R*-trees. The R*-tree is one of the most efficient members of the R-tree family. Several database and geographic information systems already use R-trees as their basic spatial access method. Therefore, it is of considerable interest to design efficient Join algorithms particularly for R*-trees.

A distinctive assessment of processes to browsing spatial items in an R-tree on the premise of their distances from an arbitrary spatial query object changed into provided. It turned into shown that an Incremental Algorithm (INN) drastically outperforms (in terms of execution time, R-tree node disk I/O, and item distance calculations) an answer based on a k-nearest neighbor algorithm (k-NN). This turned into proper even if the k-NN technique turned into optimized for this utility by means of carefully choosing the increments for k and the usage of previous search effects for pruning when the k-NN set of rules must be re-invoked. The incremental method was also found to have superior performance whilst applied to the trouble of computing the k nearest acquaintances of a given query object. The experiments verify that the INN algorithm achieves a better stage of pruning than the k-NN set of rules. This is crucial because it reduces the amount of R-tree node disk I/O as well as the range of distance calculations, which, when blended, account for a main portion of the execution time. Moreover, because the statistics sets became larger, the superiority of INN algorithm have become greater suggested.

Some preliminary comparisons with techniques based totally on integration of (pairwise) spatial join algorithms suggest that constraint-primarily based techniques carry out higher for dense queries and datasets, due to the fact they take benefit of more than one joins to restriction the hunt space. An thrilling route for future work is the combination of our strategies with pairwise be part of algorithms; as an instance, we may want to cut up a query graph in two (or greater) subgraphs to be processed with the aid of ST (or any other method) and then integrate the intermediate results using spatial be part of algorithms for non-indexed inputs. Such a methodology might efficaciously support parallel processing of multiway spatial joins.

3. FRAMEWORK

A. Proposed System Overview

For searching keyword covers (i.e., more than two keywords) from spatial datasets, we have several searching schemes are available at present like, single keyword search, multi-keyword search and fuzzy keyword search. In these searching schemes we have limit to search the keywords. In addition, we have a traditional searching scheme, i.e., Closest Keyword search and in this searching scheme we used baseline algorithm to search the keywords from spatial databases. When the number of searching query keywords increases then automatically decrease the performance of the baseline algorithm. This is the main drawback of the baseline algorithm. To overcome that drawback, we proposed an algorithm K-Nearest Neighbor Expansion. But users cannot satisfy with this algorithm to searching spatial database keywords. For that reason, in this project a novel cache search method along with Keyword-Nearest Neighbor algorithm is proposed. Through this new method we can get the optimal keyword cover within less time. In experiments it can be observed that cache search scheme increases accuracy and importance.

Candidate Generation:

In Generate Candidate function, it is needless to actually generate all possible keyword covers of input nodes (or objects). In practice, the keyword covers are generated by incrementally combining individual nodes (or objects). The figure shows all possible combinations of input nodes incrementally generated bottom up. There are three keywords k_1, k_2 and k3 and each keyword has two nodes N_1 and N_2.

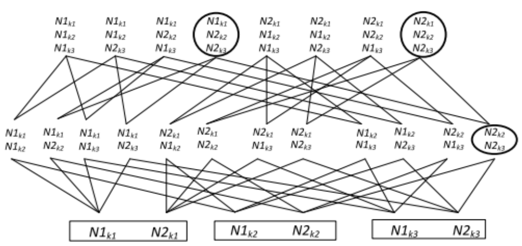


Figure 2: Candidate Generation

B. K-NNE algorithm

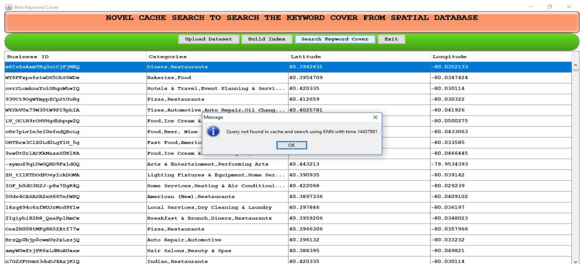
The K-Nearest Neighbor Expansion (K-NNE) algorithm is used to search the best keyword cover. It will give the results based on rating and the distance of the spatial query object. This algorithm is not time efficient algorithm to search the spatial query keywords. By using this K-NNE, if we want search any query keywords it will search from local server for every time. If we need to search same keyword cover more than two times, then the algorithm also search more than two times in local server. It means for every time it should read the all dataset on local server and finally, it will give the result.

C. Proposed Cache Search

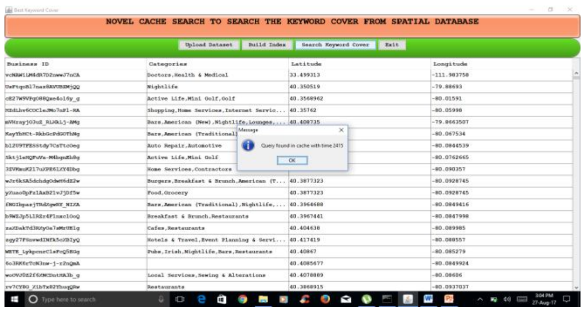
We propose a cache search scheme to search spatial database query keywords. This proposed method using the K-NNE algorithm only. When the user searches a query keyword it will select a principal query keyword and for every principal query keyword we get some principal objects. To those objects we must calculate the local best keyword cover (lbkc computation). From lbkc results it will store the optimal keyword, means it displays the results based keyword rating and distance. In spatial database, if existing keyword has low rating and large distance than the new keyword cover then the proposed scheme store the new keyword cover as best keyword cover. This best keyword covers will be stored in the cache memory. Similarly, number of searched keyword query's optimal results will be stored cache only. If anyone can search the keyword cover, it first searches the result in cache memory and if cache has that keyword cover then it displays the results from cache memory only. If the cache not found that keyword cover, searching is done on local server and cache will store that new keyword cover results.

4. EXPERIMENTAL RESULTS

In this experiment, first we need to upload the spatial dataset. After uploading the dataset we have to build the index for the uploaded dataset by using Build Index, we can search the spatial keywords by using the Keyword-Nearest Neighbor Expansion. Enter a keyword cover to search and the keyword is first searched in the cache, if in cache the results will be available then query results comes to the user from the cache only. Here, if the keyword cover results not in cache so that it display the query found or not found in cache and also display the processing time for query search results.



If we search the same query then the results will come from cache only.



The proposed cache search scheme can provide the processing time of the searched results. From the experimental results, we can say, the proposed cache search scheme is the time efficient approach than k-NNE algorithm.

5. CONCLUSION

In this paper we introduced a novel cache search scheme to search spatial objects from spatial databases. Before cache search scheme, we used the Keyword- Nearest Neighbor Expansion (K-NNE) algorithm to get optimal results. Even though Keyword-Nearest Neighbor Expansion provides optimal results, users cannot satisfy with this algorithm because, the Keyword-Nearest Neighbor Expansion algorithm is not time efficient to provide optimal results. From the proposed cache search, we can get the optimal searched results within the limited time.

REFERENCES

[1] Ke Deng, Xin Li, Jiaheng Lu, and Xiaofang Zhou, “Best Keyword cover Search”, IEEE January 2015.

[2] R. Agrawal and R. Srikant, “Fast algorithms for mining association rules in large databases,” in Proc. 20th Int. Conf. Very Large Data Bases, 1994, pp. 487–499.

[3] D. Papadias, N. Mamoulis, and Y. Theodoridis, “Processing and optimization of multiway spatial joins using r-trees,” in Proc. 18th ACM SIGMOD-SIGACT-SIGART Symp. Principles Database Syst., 1999, pp. 44–55.

[4] X. Cao, G. Cong, and C. Jensen, “Retrieving top-k prestige-based relevant spatial web objects,” Proc. VLDB Endowment, vol. 3, nos. 1/2, pp. 373–384, Sep. 2010.

[5] X. Cao, G. Cong, C. Jensen, and B. Ooi, “Collective spatial keyword querying,” in Proc. ACM SIGMOD Int. Conf. Manage. Data, 2011, pp. 373–384.

[6] G. Cong, C. Jensen, and D. Wu, “Efficient retrieval of the top-k most relevant spatial web objects,” Proc. VLDB Endowment, vol. 2, no. 1, pp. 337–348, Aug. 2009.

[7] I. D. Felipe, V. Hristidis, and N. Rishe, “Keyword search on spatial databases,” in Proc. IEEE 24th Int. Conf. Data Eng., 2008, pp. 656–665.

[8] R. Hariharan, B. Hore, C. Li, and S. Mehrotra, “Processing spatial keyword (SK) queries in geographic information retrieval (GIR) systems,” in Proc. 19th Int. Conf. Sci. Statist. Database Manage, 2007, pp. 16–23.

[9] D. Papadias, N. Mamoulis, and Y. Theodoridis, "Processing and optimization of multiway spatial joins using r-trees," in Proc. 18th ACM SIGMOD-SIGACT-SIGART Symp. Principles Database Syst., 1999, pp. 44–55.

[10] N. Mamoulis and D. Papadias, "Multiway spatial joins," ACM Trans. Database Syst., vol. 26, no. 4, pp. 424–475, 2001.

[11] D. Zhang, Y. Chee, A. Mondal, A. Tung, and M. Kitsuregawa, "Keyword search in spatial databases: Towards searching by document," in Proc. IEEE Int. Conf. Data Eng., 2009.

[12] J. Rocha-Junior, O. Gkorgkas, S. Jonassen, and K. Nørvag, "Efficient processing of top-k spatial keyword queries," in Proc. 12th Int. Conf. Adv. Spatial Temporal Databases, 2011, pp. 205–222.

[13] Dimitris Papadias, Nikos Mamoulis, Yannis Theodoridis. "Processing and Optimization of Multi-way Spatial Joins Using R- trees", 18th ACM SIGMOD-SIGACT-SIGART Symp. Principles of Database Systems.

[14] J. Lu, Y. Lu, and G. Cong, "Reverse Spatial and Textual k Nearest Neighbor Search", Proc. ACM SIGMOD Int'l Conf. Management of Data, 2011.

[15] G. R. Hjaltason and H. Samet, "Distance browsing in spatial databases," ACM Trans. Database Syst., vol. 24, no. 2, pp. 256–318, 1999.

[16] Z. Li, K. C. Lee, B. Zheng, W.-C. Lee, D. Lee, and X. Wang, "IRTree: An efficient index for geographic document search," IEEE Trans. Knowl. Data Eng., vol. 99, no. 4, pp. 585–599, Apr. 2010.

[17] www.wikipedia.com/memoryorganisation

[18] D. Zhang, B. Ooi, and A. Tung, "Locating mapped resources in web 2.0," in Proc. IEEE 26th Int. Conf. Data Eng., 2010, pp. 521–532.

[19] https://en.wikipedia.org/wiki/cache_computing

[20] Priyanka Yadav, Vishal Sharma, Priti Yadav "Cache Memory", International Journal of Computer Science and Mobile Computing, September- 2014, pg. 838-840.