

Closest Keywords Search on Spatial Databases

¹A. YOJANA, ²Dr. A. SHARADA

¹ M. Tech Student, Department of CSE, G.Narayanamma Institute of Technology & Science, Telangana, India.

² Associate Professor, Department of CSE, G.Narayanamma Institute of Technology & Science, Telangana, India.

ABSTRACT- *The objects commonly used in spatial databases are associated with keywords to indicate their businesses/services/features. The work focus on retrieving individual objects by specifying a query consisting of a query location and a set of keywords and each retrieved object is associated with keywords relevant to the given query keywords and close to the query location. There is an attentive problem known as Closest Keywords search called keyword cover, which together cover a set of query keywords i.e. whichever have the minimum inter-objects distance. Now-a-days there is a more availability and priority is given for keyword rating in evaluating the objects for making the better choice. This is the reason we are developing algorithms for best keyword cover which includes inter-object distance as well as rating provided by the customers through online business sites. There is an algorithm called baseline algorithm which combines the objects from various keywords to generate more number of candidate keyword covers. Due to the fast increasing of the candidate keyword covers the performance of an algorithm gets degraded. To run over this issue there is a scalable algorithm called keyword nearest neighbour expansion (K-NNE) algorithm which symbolically reduces the generated candidate keyword covers.*

Keywords: Spatial keyword search, Spatial databases, Candidate Keyword covers, keyword rating.

1. INTRODUCTION

Now-a-days the purpose of the data is playing a major role in the case of searching a keyword. This searching has become a frequent thing in data mining. For the last few years there is a much use of mobile computing services, location-based services (e.g., Google maps, Microsoft Virtual Earth services) and satellite imagery. Due to this, spatial keyword search has got more attention recently. So this made researchers enthusiastic for searching the spatial objects from spatial databases. This leads to develop the methods for retrieving the spatial objects. In spatial database, each tuple acts as a spatial objects which mainly assists with keyword(s), indicate the information such as businesses/services/features and also involves spatial data along with latitude and longitude values of the location. Performing Querying on such kind of data is called Best Keyword Cover querying. This type of search is called Best Keyword Cover search.

Many research works have focused on the spatial keyword search problem. This paper aims to find the number of individual objects, each of the object close to the query location and the keywords which are relevant to the set of keywords given. This paper investigates the generic form of mCK query called BKC, which mainly provides inter-object distance as well as keyword rating. This is done because of the observation, as there is an increasing availability and importance of keyword rating in finding the better decision. For processing the BKC query we develop two algorithms, Baseline algorithm and K-NNE algorithm. These algorithms use R-trees for indexing the objects. In the Baseline algorithm the main concept is to combine the child nodes in hierarchical levels of R-trees and the priority is taken, by combining their child nodes for generating the new candidates. Here for finding BKC Query, the number of Candidate keywords increases, hence the performance of the algorithm drops dramatically.

To overcome this drawback, we develop a scalable algorithm called (k-NNE) which applies a different strategy, where it retrieves keyword nearest neighbor by combining both keyword search and nearest neighbor search. It introduces the concept of keyword rating, which helps in better decision making. The one with least distance with respect to the query location and rating, it has been considered as best keyword covers. As the algorithm is optimal it generates

less number of candidate keyword covers, and the performance will be improved than the baseline algorithm.

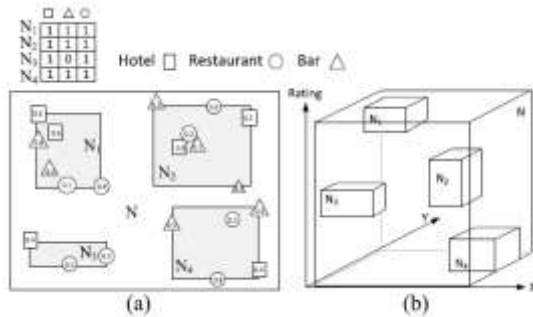
2. RELATED WORK

Many research community works have investigated the problem of spatial keyword search. Most of the works focus on retrieving individual objects by providing a query which consists of a query location and set of keywords. Now each retrieved object assists with keyword relevant to the given query and close to the query location. The similarity measures are taken between the documents to find the relevance between the set of keywords. Potentially large number of object combinations will be retrieved. But the research problem is that there should be a desirable spatial relationship. X.Cao puts the problem to retrieve the objects in such a way that they should

- 1) cover all the keywords
- 2) Should have minimum inter-object distance and
- 3) Close to the query location

In this paper we have focused on the problem which is a generic version of mCK query and also rating for the objects. The methods proposed by Cong et al. and Li et al. provides the indexing, which will be useful for verifying whether the node is relevant to the set of query keywords. For better understanding they have proposed R-

trees, where a bitmap is taken for each node. Each bit corresponds to a keyword. If a bit is “1” it indicates that there is a presence of object under that node which means it associates with keyword; whereas if it is “0” then no object is present.



(a) A R-tree. (b) The R-tree for Keyword “Restaurant”

For example in the above fig R-tree is taken, where non leaf node N has 4-child nodes N1, N2, N3, and N4. The bitmaps shows for {N1} {N2} {N4} are 111 and N3 is 101. The bitmap 101 indicates that object under N3 is not present the associated keywords are “hotels” and “restaurants” respectively. The object which is not associated is “bar”. This bitmap provides us to combine nodes for generating candidate keyword covers. If a node consists of all query keywords then it is said to be known as candidate keyword cover. If multiple nodes at a time cover all the query keywords, they constitute a candidate keyword cover. When a node ‘N’ is visited its child nodes N1, N2, N3, N4 are processed. N1, N2, N4 assists with only 2 keywords. The candidate keywords covers

generated are {N1}, {N2}, {N4}, {N1, N2}, {N1, N3}, {N1, N4}, {N2, N3}, {N2, N4}, {N3, N4}, {N1, N2, N3}, {N1, N3, N4} and {N2, N3, N4}. As more number of candidate keyword covers are generated, among these the one with best evaluation should be taken by combining their child nodes to generate more candidates. The number of the candidates which generated are very large in the existing approach. Thus, in order to obtain the best solution from this number of candidate keyword covers, a depth-first tree browsing strategy is used. Hence after pruning all these candidates, the current best solution is obtained which is called as best keyword covers. One of the drawbacks is that it generates more number of candidate keyword covers, so the performance of existing algorithm gets dropped as there is a fast increasing of candidate keyword covers. This motivates us to develop an algorithm called *Keyword Nearest Neighbor Expansion*.

3. PROPOSED APPROACH

This paper proposes on an algorithm called *Keyword Nearest Neighbor Expansion*. When we provide set of keywords as an input, then the main focus will be on a particular keyword, called principal query keyword to perform search. The objects which are associated with this principal query keyword are called principal objects. Indexing is done for the required principal object which assists with keyword. After identifying the principal object it searches for the objects having highest keyword rating.

One with highest keyword rating is considered as first object and further the search has been carried out. For each sub principal objects the score has been calculated called *Local Best Keyword Cover*. With the help of similarity measures and weighted average of index rating the LBKC is obtained. This weighted average of index rating includes both nearest neighbor search and keyword search. Now for each principal objects the LBKC is computed. After the computation of lbkc the one with the highest score results as a best keyword cover, so that minimum number of candidates is selected in order to achieve high performance.

Local Best Keyword Cover: For the given a set of keywords T and the principal query keyword $k \in T$, the objects associated with k are called as principal objects o_k a lbkc is computed. During the computation of $lbkc_{o_k}$ it incrementally retrieves the keyword nearest neighbors of the principal objects. This lbkc is processed in k-NNE algorithm only when the score of the lbkc is greater than existing BKC score. So those numbers of candidate keyword covers generated are organized in a group. In this group if one of the keyword cover has a score greater than BKC score, then we can say that there is a possibility of getting the answer for BKC query in this group.

For principal query keywords R-trees are used for indexing principal objects. Principal objects are processed in the blocks instead of doing individually. For each principal object the lbkc

is computed in such a way that the resultant will be close to the query location and with high keyword rating is considered. By this nearest neighbors are retrieved. Significantly the candidate keyword covers are reduced.

4. CONCLUSION

The baseline algorithm generates a large number of candidate keyword covers which leads to dramatic performance drop when more query keywords are given. In this paper, we have proposed a new mechanism called k-NNE, it uses a strategy called LBKC where it finds nearest distance and better rating. Due to that it reduces the more number of candidate keywords covers and improves the performance. While searching for best key word cover, every time our application will search in the entire data set. In the future work we will maintain a local cache (at client side), so all the searched history will be saved in this local cache so whenever the user search for the same it will search from the local cache instead of the whole data set which results in improving the efficiency.

REFERENCES

- [1] I. D. Felipe, V. Hristidis, and N. Rische, "Keyword search on spatial databases," in Proc. IEEE 24th Int. Conf. Data Eng., 2008, pp. 656–665.
- [2] Ke Deng, Xin Li, Jiaheng Lu, and Xiaofang Zhou, "Best Keyword Cover Search", IEEE

transactions on knowledge and data engineering, vol. 27, no. 1, january 2015.

[3] 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.

[4] 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.

[5] 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.

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

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

[8] 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.

[9] J. Rocha-Junior, O. Gkorgkas, S. Jonassen, and K. Nørva g, “Efficient processing of top-k

spatial keyword queries,” in Proc. 12th Int. Conf. Adv. Spatial Temporal Databases, 2011, pp. 205–222.

[10] S. B. Roy and K. Chakrabarti, “Location-aware type ahead search on spatial databases: Semantics and efficiency,” in Proc. ACM SIGMOD Int. Conf. Manage. Data, 2011, pp. 361–372.

[11] 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.