International Journal of Trend in Scientific Research and Development (IJTSRD)
International Open Access Journal
ISSN No: 2456 - 6470 | www.ijtsrd.com | Volume - 2 | Issue – 1

Location Based Nearest Keyword Search

Miss Rachana V. Kurhekar
ME (CSE) , HVPM College of Engineering And Technology, Amravati, India

Prof. R. R. Shelke
Asst. Professor, HVPM College of Engineering And Technology, Amravati, India

ABSTRACT

It is common that the objects in a spatial database are associated with keyword to indicate their businesses/services/features. An interesting problem known as Closest Keywords search is to query objects, called nearest keyword search, which together cover a set of query keywords and have the minimum inter-objects distance. In recent years, I observe the increasing availability and importance of keyword rating in object evaluation for the better decision making. This motivates us to investigate a generic version of Closest Keywords search called Best Keyword Cover which considers inter-objects distance as well as the keyword rating of objects. The baseline algorithm is inspired by the methods of Closest Keywords search which is based on exhaustively combining objects from different query keywords to generate candidate keyword covers. When the number of query keywords increases, the performance of the baseline algorithm drops dramatically as a result of massive candidate keyword covers generated. To recover this drawback, this work proposes a much more scalable algorithm called keyword nearest neighbor expansion (keyword-NNE). Compared to the baseline algorithm, keyword-NNE algorithm significantly reduces the number of candidate keyword covers generated. The in-depth analysis and extensive experiments on real data sets have justified the superiority of our keyword-NNE algorithm.

Keyword: *Spatial database, Point of Interests, Keywords, Keyword Rating, Keyword Cover*

INTRODUCTION

An increasing number of applications require the efficient execution of nearest neighbor (NN) queries constrained by the properties of the spatial objects. Due to the popularity of keyword search, particularly on the Internet, many of these applications allow the user to provide a list of keywords that the spatial objects (henceforth referred to simply as objects) should contain, in their description or other attribute. For example, online yellow pages allow users to specify an address and a set of keywords, and return businesses whose description contains these keywords, ordered by their distance to the specified address location. As another example, real estate web sites allow users to search for properties with specific keywords in their description and rank them according to their distance from a specified location. We call such queries spatial keyword queries. A spatial keyword query consists of a query area and a set of keywords. The answer is a list of objects ranked according to a combination of their distance to the query area and the relevance of their text description to the query keywords. A simple yet popular variant, which is used in our running example, is the distance-first spatial keyword query, where objects are ranked by distance and keywords are applied as a conjunctive filter to eliminate objects that do not contain them. Which is our running example, displays a dataset of fictitious hotels with their spatial coordinates and a set of descriptive attributes (name, amenities)? An example of a spatial keyword query is “find the nearest hotels to point that contain keywords internet and pool”. The top result of this query is the hotel object. Unfortunately there is no efficient support for top-k spatial keyword queries, where a prefix of the results list is required. Instead, current systems use ad-hoc combinations of nearest neighbor (NN) and keyword search techniques to tackle the problem. For instance, an R-Tree is used to find the nearest neighbors and for each neighbor an inverted index is used to check if the query keywords are contained. We show that such two-phase approaches are inefficient.

Driven by mobile computing, location-based services and wide availability of extensive digital maps and satellite imagery (e.g., Google Maps and Microsoft Virtual Earth services), the spatial keywords search problem has attracted much attention recently in a spatial database, each tuple represents a spatial object which is associated with keywords to indicate the information such as its businesses/services/features. Given a set of query keywords, an essential task of spatial keywords search is to identify spatial objects which are associated with keywords relevant to a set of query keywords, and have desirable spatial relationships (e.g., close to each other and/or close to a query location). This problem has unique value in various applications because users' requirements are often expressed as multiple keywords. For example, a tourist who plans to visit a city may have particular shopping, dining and accommodation needs. It is desirable that all the needs can be satisfied without long distance traveling. Due to the remarkable value in practice, several variants of spatial keyword search problem have been studied. The works aim to find a number of individual objects, each of which is close to a query location and the associated keywords (or called document) are very relevant to a set of query keywords (or called query document).

The document similarity is applied to measure the relevance between two sets of keywords. Since it is likely none of individual objects is associated with all query keywords, this motivates the studies to retrieve multiple objects, called keyword cover, which together cover (i.e., associated with) all query keywords and are close to each other. This problem is known as m Closest Keywords (mCK) query in. The problem studied in additionally requires the retrieved objects close to a query location. a generic version of mCK query, called Best Keyword Cover (BKC) query, which considers inter-objects distance as well as keyword rating. It is motivated by the observation of increasing availability and importance of keyword rating in decision making. Millions of businesses/services/features around the world have been rated by customers through online business review sites such as Yelp, City search, ZAGAT and Dianping, etc. For example, a restaurant is rated 65 out of 100 (ZAGAT.com) and a hotel is rated 3.9 out of 5 (hotels.com). According to a survey in 2013 conducted by Dimensional Research (dimensionalresearch.com), an overwhelming 90 percent of respondents claimed that buying decisions are influenced by online business review/rating. Due to the consideration of keyword rating, the solution of BKC query can be very different from that of mCK query. Fig. 1 shows an example. Suppose the query keywords are “Hotel”, “Restaurant” and “Bar”. mCK query returns ft2;s2;c2g since it considers the distance between the returned objects only. BKC query returnsft1;s1;c1g since the keyword ratings of object are considered in addition to the inter-objects distance. Compared to mCK query, BKC query supports more robust object evaluation and thus underpins the better decision making. This work develops two BKC query processing algorithms, baseline and keyword-NNE. The baseline algorithm is inspired by the mCK query processing methods. Both the baseline algorithm and keyword-NNE algorithm are supported by indexing the objects with an R*-tree like index, called KRR*-tree. . In the baseline algorithm, the idea is to combine nodes in higher hierarchical levels of KRR*-trees to generate candidate keyword covers. Then, the most promising candidate is assessed in priority by combining their child nodes to generate new candidates. Even though BKC query can be effectively resolved, when the number of query keywords increases, the performance drops dramatically as a result of massive candidate keyword covers generated. To overcome this critical drawback, we developed much scalable keyword nearest neighbor expansion (keyword-NNE) algorithm which applies a different strategy.

KeywordNNE selects one query keyword as principal query keyword. The objects associated with the principal query keyword are principal objects. For each principal object, the local best solution (known as local best keyword cover) is computed. Among them, the lbkc with the highest evaluation is the solution of BKC query. Given a principal object, its lbkc can be identified by simply retrieving a few nearby and highly rated objects in each non-principal query keyword (two-four objects in average as illustrated in experiments). Compared to the baseline algorithm, the number of candidate keyword covers generated in keyword-NNE algorithm is significantly reduced. The in-depth analysis reveals that the number of candidate keyword covers further processed in keyword-NNE algorithm is optimal, and each keyword candidate cover processing generates much less new candidate keyword covers than that in the baseline algorithm.

LITERATURE REVIEW

This issue has remarkable esteem in different applications since clients' prerequisites are often communicated as various keywords For instance, a traveller who arrangements to visit a city may have specific shopping, feasting and convenience needs. It is attractive that every one of these necessities can be achieved without long separation travelling. Because of the amazing quality practically speaking, a few variations of spatial keyword search issue have been examined. The works mean to detect various individual protests, each of which is close a query location and the related keywords (or called document) are very important to a set of query keywords.

1.IR Tree : An efficient index for geographic document search [1] From This Paper we Discussed- In this paper, we propose an actual record, called IR-tree, that composed with a top-k document search algorithm inspires four of note tasks in file searches, to be detailed, 1) spatial filtering, 2) textual filtering, 3) relevance computation, and 4) document ranking in a entirely coordinated mode. What's more, IR-tree permits searches to hold diverse weights on textual and spatial relevance of documents at the runtime and in this way cooks for a wide variety of utilizations. An arrangement of full examinations over an wide variety of situations has been focussed and the experimental comes about show that IR-tree beats the cutting edge line of attack for geographic file searches.

2. Retrieving top-k prestige-based relevant spatial web objects [2] From This Paper we Discussed- The location-aware keyword query returns ranked objects that are almost a query location and that have printed portrayals that match query keywords. This query occurs certainly in many sorts of useful and conventional web administrations and applications, e.g., Maps administrations. Previous work considers the possible significances of such a query as being independent when ranking them. All the same, a relevant outcome question with adjacent objects that are similarly applicable to the query is likely to be perfect over an significant protest short of important close-by objects. The paper suggests the idea of prestige-based significance to catch both the printed significance of a question a query and the effects of close-by objects. Established on this, additional sort of query, the Location-aware top- k Prestige-based Text recovery (LkPT) query, is not compulsory that recovers the top-k spatial web objects categorized by prestige-based significance and location closeness. We suggest two calculations that process LkPT questions. Exact analyses with open spatial information display that LkPT inquiries are more exciting in recovering web objects than a previous approach that does not consider the effects of adjacent objects; and they prove that the proposed calculations are adjustable and out Performa standard approach necessarily.

3. Efficient retrieval of the top-k most relevant spatial web objects [3] From This Paper we Discussed- The customary Internet is make safe a geo-spatial dimension. Web information are being geo-labeled, and geo referenced protests, for case in point, purposes of intrigue are being associated with attractive content records. The following grouping of geo-location and reports allows additional kind of top-k query that takes into record both location vicinity and content implication. To our information, just local systems occur that is fit for recording a general web information recovery query while as well taking location into record. This paper put forward another collection framework for location aware top-k content recovery. The framework impacts the disappointed document for content recovery and the R-tree for spatial nearness querying. Rare collation methodologies are studied inside the framework. The framework encloses calculations that use the future records for imagining the top-k query, therefore taking into record both content reputation and location nearness to crop the inquiry space. Significances of experimental analyses with an performance of the framework display that the paper's proposal offers flexibility and is equipped for excellent performance.

4. Keyword search on spatial databases[4] In this paper, mostly attention on finding top-k Nearest Neighbors, in this way each node has to match the entire querying keywords. As this way cup tie the entire query to every node, it does not reflect the density of data objects in the spatial space. When no of queries rises then it hints to minor the efficiency and quickness. They present an efficient way to response top-k spatial keyword queries. This work has the next contributions: 1) the problematic of top-k spatial keyword search is defined. 2) The IR2-Tree is projected as an efficient indexing structure to collection spatial and textual data for a set of objects. There are efficient algorithms are used to keep the IR2-tree, that is, insertion and remove objects. 3) An efficient incremental algorithm is existing to response top-k spatial keyword queries by means of the IR2-Tree. Its presentation is projected and likened to the current methods. Actual datasets are used in our trials that display the significant enhancement in performance times.

EXISTING SYSTEM

3.1 Related Works and Disadvantages

Existing system focus on baseline algorithm and Indexing Keyword Ratings, The baseline algorithm is inspired by the mCK query processing methods [5], [4]. For mCK query processing, the method in [4] browses index in top-down manner while the method in [5] does bottom-up. Given the same hierarchical index structure, the top-down browsing manner typically performs better than the bottom-up since the search in lower hierarchical levels is always guided by the search result in the higher hierarchical levels. However, the significant advantage of the method in [5] over the method in [4] has been reported. This is because of the different index structures applied. Both of them use a single tree structure to index data objects of different keywords. But the number of nodes of the index in [5] has been greatly reduced to save I/O cost by keeping keyword information with inverted index separately. Since only leaf nodes and their keyword information are maintained in the inverted index, the bottom-up index browsing manner is used. When designing the baseline algorithm for BKC query processing, we take the advantages of both methods [5], [4].

Indexing Keyword Ratings : To process BKC query, we augment R*-tree with one additional dimension to index keyword ratings. Keyword rating dimension and spatial dimension are inherently different measures with different ranges. It is necessary to make adjustment. In this work, a three-dimensional R*-tree called keyword rating R*-tree (KRR*-tree) is used. The ranges of both spatial and keyword rating dimensions are normalized into [0, 1].

Some existing works focus on retrieving individual objects by specifying a query consisting of a query location and a set of query keywords (or known as document in some context). Each retrieved object is associated with keywords relevant to the query keywords and is close to the query location. The approaches proposed by Cong et al. and Li etal. employ a hybrid index that augments nodes in non-leaf nodes of an R/R*-tree with inverted indexes. In virtual bR*-tree based method, an R*-tree issued to index locations of objects and an inverted index is used to label the leaf nodes in the R*-tree associated with each keyword. Since only leaf nodes have keyword information the mCK query is processed by browsing index bottom-up.

Disadvantages Of Existing System:

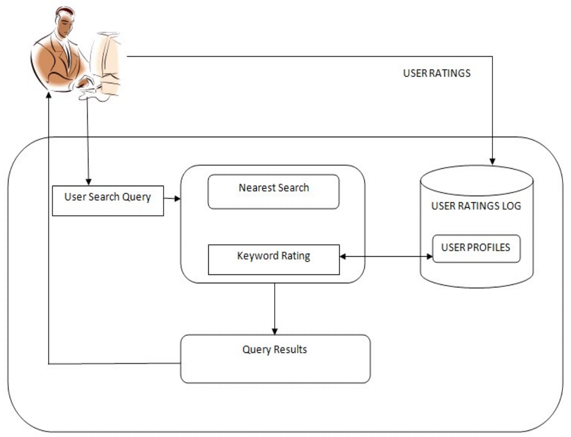
- When the number of query keywords increases, the performance drops dramatically as a result of massive candidate keyword covers generated.
- The inverted index at each node refers to a pseudo document that represents the keywords under the node. Therefore, in order to verify if a node is relevant to a set of query keywords, the inverted index is accessed at each node to evaluate the matching between the query keywords and the pseudo-document associated with the node.

3.2 Analysis Of Problem.

This test shows the impact of the performance. Is an application specific parameter to balance the weight of keyword rating and the diameter in the score function. Compared to m, the impact of the performance is limited. When _ = 1, BKC query is degraded to mKC query where the distance between objects is the sole factor and keyword rating is ignored. When _ changes from 1 to 0, more weight is assigned to keyword rating. An interesting observation is that with the decrease of _ the number of keyword covers generated in both the baseline algorithm and keyword-NNE algorithm shows a constant trend of slight decrease. The reason behind is that KRR*-tree has a keyword rating dimension. Objects close to each other geographically may have very different ratings and thus they are in different nodes of KRR*tree. If more weight is assigned to keyword ratings, KRR*-tree tends to have more pruning power by distinguishing the objects close to each other but with different keyword ratings. As a result, less candidate keyword covers are generated.

SYSTEM ARCHITECTURE

The figure gives idea about system architecture. A query including a query region and a course of action of query catchphrases. Each recovered thing is connected with watchwords basic to the query catchphrases and is close to the query region. The identicalness between reports is connected with assess the criticalness between two arrangements of watchwords. Since it is likely no individual article is related with all query watchwords, some particular works mean to recover diverse things which together cover all query catchphrases. Framework finds main problems like: 1)cover all query watchwords, 2) have slightest between things partition and 3) are close to a query territory. The objective of the interface is to give purpose of interest data (static and segment ones) with, no not precisely, a domain, a few necessaries qualities and open slight segments (depiction). In requesting to give those data, the segment that executes the interface utilizes the associate database data to find and exhibit purpose of interest (POI) or to pick a POI as course way point and top pick. This part not just gives seek usefulness to the area database also a way to deal with partner outside web record to this section and overhaul the chase criteria and the once-over of results.



PROPOSED SYSTEM

In keyword-NNE algorithm one query keyword is selected as principal query keyword, and the objects retrieved are nearer to this principal query keyword. So the query point will be the principal object. And the inter object distance from this point to other points of interests should be minimum. The result places are closer to the principal object. Principal query keyword is selected as the one in which number of objects will be minimum. Although the method keyword-NNE outperforms, it faces the following limitations.

Most of the geographic studies use distance as a simple measure of accessibility. Straight-line (Euclidean) distance is most often used in spatial databases because of the ease of its calculation. Actual travel distance over a road network is a better alternative, although historically an expensive and labour intensive task. This is not true always, because using commercial website one can directly compute time and distance, without the need to own or purchase specialized GIS software or street files. Taking advantage of this feature, compare straight-line and travel distance and travel time to calculate distance between query point and other nearby locations.

A major limitation of keyword-NNE is that user cannot specify his current location. So that the query does not retrieve distance of the path from user's current location to principal object in GBKC. Instead of taking euclidean distance from user's current location to the query point, travelling distance and time is calculated. Because euclidean distance may not always give an accurate result as user expected.

Let Ok be the set of principal objects under principal query keyword k. ok ∈ Ok be the principal object in GBKCk. Distance of ok to the user's current location L is not specified in this method. Shortest travelling distance of the path taken by user from location L to the principal object in Global Best Keyword Cover can be obtained using Google API [14]. Adding this feature can make the searching more user friendly and give more support for a traveller in good decision making.

Another problem with the keyword-NNE method is that algorithm set one query keyword with minimum number of objects as principal keyword. So that the retrieved results are surrounded by this keyword. User cannot give principal query keyword according to his own choice. Suppose a user wants to know locations nearer to non principal object, such provision is not provided in this algorithm. In current location based closest keyword search user can set any keyword as principal query keyword according to his choice. Instead of selecting the one with minimum number of objects, user can set principal keyword as the first entered keyword. The method can retrieve the same result (GBKC) as keyword-NNE. Along with that result user can select an object in GBKC and can search user's interested keyword nearer to that selected object.

Location Aware Closest Keyword Search In Spatial Data :

The method is based on current location of user. User specify his points of interest and current location. After calculating GBKC, the system returns an itinerary (a planned route) covering user's current location and POIs (Points of Interest) Initially specifying current location of user. Using Geocoding API, corresponding address is converted to its latitude and longitude. From the current location nearest object in GBKC is calculated, and the process continues upto the last object. All these Points of Interests are represented as waypoints in map. Waypoints specifies an array of points. It can alter a route by routing it through the specified location(s). A waypoint is specified as a latitude/longitude coordinate, an encoded polyline, a place ID, or an address which will be geocoded. A path covering all these waypoints are created. So the method creates an itinerary (a planned route) covering users current location and all objects in GBKC.

- our paper investigates a generic version of mCK query, called Best Keyword Cover (BKC) query, which considers inter-objects distance as well as keyword rating. It is motivated by the observation of increasing availability and importance of keyword rating in decision making. Millions of businesses/services/features around the world have been rated by customers through online business review sites such as Yelp, City search, ZAGAT and Dianping, etc.
- This work develops two BKC query processing algorithms, baseline and keyword-NNE. The baseline algorithm is inspired by the mCK query processing methods. Both the baseline algorithm and keyword-NNE algorithm are supported by indexing the objects with an R*-tree like index, called KRR*-tree.
- We developed much scalable keyword nearest neighbor expansion (keyword-NNE) algorithm which applies a different strategy. Keyword-NNE selects one query keyword as principal query keyword. The objects associated with the principal query keyword are principal objects. For each principal object, the local best solution (known as local best keyword cover lbkc) is computed .Among them, the lbkc with the highest evaluation is the solution of BKC query. Given a principal object, its lbkc can be identified by simply retrieving a few nearby and highly rated objects in each non-principal query keyword (two-four objects in average as illustrated in experiments).

CONCLUSION

Compared to the most relevant mCK query, BKC query provides an additional dimension to support more sensible decision making. The introduced baseline algorithm is inspired by the methods for processing mCK query. The baseline algorithm generates a large number of candidate keyword covers which leads to dramatic performance drop when more query keywords are given. The proposed keyword-NNE algorithm applies a different processing strategy, i.e., searching local best solution for each object in a certain query keyword. As a consequence, the number of candidate keyword covers generated is significantly reduced. The analysis reveals that the number of candidate keyword covers which need to be further processed in keyword-NNE algorithm is optimal and processing each keyword candidate cover typically generates much less new candidate keyword covers in keyword-NNE algorithm than in the baseline algorithm.

REFERENCES

1) Z. Li, K. C. Lee, B. Zheng, W.-C. Lee, D. Lee, and X. Wang, “IR Tree: An efficient index for geographic document search,” IEEE Trans. Knowl. Data Eng., vol. 99, no. 4, pp.585–599, Apr. 2010.

2) Jo˜ao B. Rocha-Junior et al. ―Efficient processing of top-k spatial keyword queries‖. In: Proceedings of the 12th international conference on Advances in spatial and temporal databases. 2011, pp. 205–222.

3) G. Cong, C. Jensen, and D. Wu. ―Efficient retrieval of the top-k most relevant spatial web objects‖. In: Proc. VLDB Endow. 2.1 (2009), pp. 337–348.

4) Ian De Felipe, Vagelis Hristidis, and Naphtali Rishe. ―Keyword Search on Spatial Databases‖. In: ICDE. 2008, pp. 656–665.

5) Dongxiang Zhang, Beng Chin Ooi, and Anthony K. H. Tung. ―Locating mapped resources in web 2.0‖. In: ICDE (2010).

6) Ramaswamy Hariharan et al. ―Processing Spatial-Keyword (SK) Queries in Geographic Information Retrieval (GIR) Systems‖. In: Proceedings of the 19th International Conference on Scientific and Statistical Database Management. 2007, pp. 16–23.

7) Dongxiang Zhang, Beng Chin Ooi, and Anthony K. H. Tung. ―Locating mapped resources in web 2.0‖. In: ICDE (2010).

8) Dongxiang Zhang et al. ―Keyword Search in Spatial Databases: Towards Searching by Document‖. In: ICDE. 2009, pp. 688–699.

9) Ke Deng, Xin Li, Jiaheng Lu, and Xiaofang Zhou.‖ Best Keyword Cover Search‖ TKDE.2014

10) S. B. Roy and K. Chakrabarti. ―Location-Aware Type Ahead Search on Spatial Databases: Semantics and Efficiency‖. In: SIGMOD (2011)

11) 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.

12) T. Brinkhoff, H. Kriegel, and B. Seeger, “Efficient processing of spatial joins using r-trees,” in Proc. ACM SIGMOD Int. Conf. Manage. Data, 1993, pp. 237–246.

13) 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.

14) 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.

15) R. Fagin, A. Lotem, and M. Naor, “Optimal aggregation algorithms for middleware,” J. Comput. Syst. Sci., vol. 66, pp. 614–656, 2003.

16) 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.

17) G. R. Hjaltason and H. Samet, “Distance browsing in spatial databases,” ACM Trans. Database Syst., vol. 24, no. 2, pp. 256–318, 1999.

18) N. Mamoulis and D. Papadias, “Multiway spatial joins,” ACM Trans. Database Syst., vol. 26, no. 4, pp. 424–475, 2001.

19) D. Papadias, N. Mamoulis, and B. Delis, “Algorithms for querying by spatial structure,” in Proc. Int. Conf. Very Large Data Bases, 1998, pp. 546–557.

20) 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.

21) J. M. Ponte and W. B. Croft, “A language modeling approach to information retrieval,” in Proc. 21st Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval, 1998, pp. 275–281.

22) 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.

23) 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, pp. 688–699.

24) Zhicheng Li, Hu Xu, Yansheng Lu, Ailing Qian, ”Aggregate Nearest Keyword Search in Spatial Databases,” IEEE 12th International Asia-Pacific Web Conference, 2010.

25) Gnttman A, ”R-trees: A Dynamic Index Structure for Spatial Searching,” Proc. ACM SIG MOD Int. Coaf. on Management, pp. 47-57, 1984.