Performing Keyword Cover Search using Keyword Nearest Neighbor Expansion Algorithm

K. Yamini*1 and Padmini. Chattu*2

* M.Tech (CSE), Dhanekula Institute of Engineering & Technology, A.P., India.
* Assistant Professor, Dept. of Computer Science & Engineering, Dhanekula Institute of Engineering & Technology, A.P., India.

Abstract— Searching is a common activity happening in data mining. Searching for spatial objects from spatial database has recently sparked enthusiasm among researchers. This motivated to develop methods to retrieve spatial objects. Best keyword cover query aims to find objects associated with keywords. The method proposed considers keyword rating, keyword relevance and spatial relevance. It also helps to retrieve data based on Boolean range query. This is the main reason for developing this new algorithm called Best keyword cover which is considers inter-distance as well as the rating provided by the customers through the online business review sites. Closest keyword search algorithm combines the objects from different query keywords to generate candidate keyword covers. Baseline algorithm and keyword nearest neighbor expansion algorithms are used to find the best keyword cover. The performance of the closest keyword algorithm drops dramatically, when the number of query keyword increases. To solve this problem of the existing algorithm, this work proposes generic version called keyword nearest neighbour expansion which reduces the resulted candidate keyword covers. As a result new methods for best keyword cover search were developed. Traditional nearest neighbour search compute nearest neighbour by considering distance as feature. In this context, nearest neighbour search focus on finding nearest neighbours where keywords and spatial data plays a major impact.

Index Terms— Best keyword cover query, spatial database, keyword cover, Nearest neighbour search.

I. INTRODUCTION

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.

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

Fig. 1: Best Keyword Search
from the query point, while all the closer neighbors are missing at least one of the query keywords. For better decision making, concept of keyword rating was introduced along with its features other than distance. For such search, query will take form of feature of objects. It search for nearest neighbour based on a new similarity measure, named weighted average of index rating which combine keyword rating, keyword search and nearest neighbour search.

II. PROBLEM STATEMENT

Best keyword cover query takes form of keywords or objects. For example, college. Given a spatial database P, which consist of set of points. For a query q, where q belong to set of objects, it search for nearest neighbour within the object by searching its importance in that data and then perform nearest neighbour search to obtain the answer to the query. For better decision making, concept of keyword rating was introduced along with its features other than distance. For such search , query will take form of feature of objects. It search for nearest neighbour based on a new similarity measure, named weighted average of index rating which combine keyword rating, keyword search and nearest neighbour search.

A. Keyword nearest neighbour expansion variant algorithm :

Baseline algorithm only retrieve answer to single query. It does not respond to multiple query keywords. It does not respond to spatial features. To overcome the drawback, the author proposed a variant of nearest neighbour algorithm called keyword nearest neighbour expansion variant. This algorithm focuses on retrieving nearest neighbour by combining both keyword search and nearest neighbour search. The new algorithm introduces the concept of keyword rating, spatial relevance and keyword relevance. Keyword rating helps in decision making. For better decision making keyword rating play a significant role. Since all performance operations depend on objects, there exist a problem of choosing which objects first for querying when given multiple features of different objects. For this purpose keyword rating has been associated with objects. Rating is based day to day importance of object in daily life. Rating takes value of integer ranging from 1 to 5. This algorithm not only involves keyword rating but also involve features of objects as well. Objects must be selected to add features. Input to keyword nearest neighbour expansion variant algorithm is a set of query keywords in the form of features associated with objects. The first step is to select principle query keyword to perform search. In other words, to identify the first object in which feature has been associated for searching. Objects linked with principle query keyword are called principle objects. Indexing has been used to find required object associated with keyword. After identifying the object, it search for objects having highest keyword rating. The one with highest keyword rating are usually set as the first object in which search has to be carried out.

B. Baseline Algorithm :

Keyword search and nearest neighbour search, there is a need to perform query on such data. Baseline algorithm focus on retrieving data with respect to query keywords. Best keyword cover can be obtained by baseline algorithm. Baseline algorithm requires spatial objects in the form of files which include fields like spatial location and its document identifier and its address. Spatial objects are objects obtained from spatial data. All operations revolve around spatial objects. Input to baseline algorithm requires single query keyword in the form of objects. The first step in baseline algorithm is to set a variable bkc as zero. The next step is to generate candidate keyword cover. Candidate keyword cover generates spatial objects that contain those query keywords. Keyword significance has been calculated using term frequency inverse document frequency as similarity measure. Term frequency inverse document frequency is a combination of term frequency and inverse document frequency. Term frequency is the number of times a term occurs in a single document divided by total number of terms in a document. Inverse document frequency is the inverse of number of times a term occurs in a document divided by total number of documents. Term frequency inverse document frequency is the product of term frequency and inverse document frequency.

III. RELATED WORK

A. Indexing Keyword Ratings:

A single tree structure is used to index objects of different keywords. The single tree can be extended with an additional dimension to index keyword rating. A single tree structure suits the situation that most keywords are query keywords. For the above mentioned example, all keywords, i.e., “hotel”, “restaurant” and “bar”, are query keywords. However, it is more frequent that only a small fraction of keywords are query keywords. For example in the experiments, only less than 5 percent keywords are query keywords. In this situation, a single tree is poor to approximate the spatial relationship between objects of few specific keywords. Therefore, multiple KRR*- trees are used in this work, each for one keyword. I The KRR*-tree for keyword ki is denoted as KRR*kitree. Given an object, the rating of an associated keyword is typically the mean of ratings given by a number of customers for a period of time. The change does happen but slowly. Even though dramatic change occurs, the KRR*-tree is updated in the standard way of R*-tree update.

B. Keyword nearest Neighbor Expansion :

Using the baseline algorithm, BKC query can be effectively resolved. However, it is based on exhaustively combining objects (or their MBRs). Even though pruning techniques have been explored, it has been observed that the performance drops dramatically, when the number of query keywords increases, because of the fast increase of candidate keyword covers generated. This motivates us to develop a different algorithm called keyword nearest neighbor expansion. We
focus on a particular query keyword, called principal query keyword. The objects associated with the principal query keyword are called principal objects. Page 403 The goal of the interface is to provide point of interest information (static and dynamic ones) with, at least, a location, some mandatory’s attributes and optional details (description,…). In order to provide that information, the component that implements the interface uses the map database information to locate and display point of interest (POI) or to select POI as route waypoint and favorite. This component not only provides search functionalities for the local database but also a way to connect external search engine to this component and enhance the search criteria and the list of results. It also proposes a solution to get custom POIs (not part of the local map database) or to dynamically update content and description of local POI. This is achieved by specifying and providing interfaces to: Select POIs from one of their attributes (e.g., Category, Name,….) Retrieve POI attributes (e.g., Location and Description) Get dynamic content for a given POI. Add custom POI to the map display Import new POIs and POIs categories from local file.

C. LBKC Computation:
Given a spatial database, each object may be associated with one or multiple keywords. Without loss of generality, the object with multiple keywords is transformed to multiple objects located at the same location, each with a distinct single keyword. When further processing a candidate keyword cover, keyword-NNE algorithm typically generates much less new candidate keyword covers compared to BFbaseline algorithm. Since the number of candidate keyword covers further processed in keyword-NNE algorithm is optimal the number of keyword covers generated in BF-base algorithm is much more than that in keyword-NNE algorithm. In turn, we conclude that the number of keyword covers generated in baseline algorithm is much more than that in keyword-NNE algorithm. This conclusion is independent of the principal query keyword since the analysis does not apply any constraint on the selection strategy of principal query keyword.

IV. PROPOSED METHOD
In spatial database, each object present in database may be associated with either one or multiple keywords. In this object with multiple keywords are directly transformed to multiple objects located at the same location without loss of generality. These objects are in the form of where location of the objects in two dimensional geographical space represented by x and y. Definition 1 (Diameter): Let O be a set of objects {o1,…….,on}. For oi; oj € O, dist(oi, oj) is the euclidean distance between oi, oj in the two-dimensional geographical space. The diameter of O is Diom(O)=max dist(oi, oj), eq.(1)

Each objects has its score with respect to diameter of object and keyword rating of objects in O. Interest of the user may be different in keyword ratings of the objects. Definition 2 (keyword Cover): Let T be a set of keywords {k1, . . . , kn} and O a set of objects {o1, . . . , on} O is a keyword cover of T if one object in O is associated with one and only one keyword in T. Definition 3 (Best Keyword Cover Query): Given a spatial database D and a set of query keywords T, BKC query returns a keyword cover O of T (O subset D) such that O.score ≥ O*.score for any keyword cover O* of T (O* subset D). In keyword-NNE algorithm, instead of individually processing principal objects are processed in blocks. Suppose k be the principal query keyword. KRR*k-tree used for indexing principal objects. Given principal node Nk in KRR*k-tree, and lbkcNk consider as local keyword cover of Nk, that consists of Nk and other corresponding nodes of Nk in each non-principal query keyword.

A. Keyword-NNE:
In previous work, BKC algorithm drops its performance when the number of query keywords is increases. To solve this problem, here developed a more efficient keyword nearest neighbour expansion (keyword-NNE) which uses the different strategy. In this algorithm, one query is considered as a principal query keyword. Those objects are associated with principal query keyword are considered as principal objects. Keyword-NNE computes local best solution for each principal object. BKC algorithm returns the lbkc with having highest evaluation. For each principal object, its lbkc can be simply selects few nearby and highly rated objects by the viewer/customer. Compared with the baseline algorithm, the keyword covers significantly reduced. These keyword covers further processed in keyword-NNE algorithm that will be optimal, and each keyword candidate cover processing generates very less new candidate keyword covers.

V. CONCLUSION & FUTURE ENHANCEMENT
Spatial database consist of large spatial objects, the time required for searching objects is more. By combining R-tree technique with minimum bounding method the performance of system for retrieving a data from database is improved, also access time is minimized. 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
Transactions on Knowledge and Data Engineering vol 26 Issue No.03 March 2014


[6] Feifei Li, Member, IEEE, Bin Yao, Student Member, IEEE, and Piyush Kumar, "Group Enclosing Queries", Transactions on Knowledge and Data Engineering, VOL. 23, NO. 10, October 2011.


Mrs. Padmini. Chattu is presently working as Assistant professor in CSE department, Dhanekula Institute of Engineering and Technology, Ganguru, VIJAYAWADA. She has obtained B.Tech, degree from JNTU, Hyderabad in 2004 and M.Tech, degree from Andhra University in 2010. She has published several research papers in various national and international Journals.

Ms. K. Yamini is a student of Dhanekula Institute of Engineering and Technology, Ganguru, VIJAYAWADA. She is presently pursuing her M.Tech degree from JNTU, Kakinada. She has obtained B.Tech, degree from JNTU, Kakinada.