International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 04
February 2018

An Adaptive Nearest Keyword Search Using Multi-Scale Hashing and Projection Technique in Spatial Databases

D. Jyothirmai & Somasekhar . T

[1]M.Tech Student, Dept. of CSE, BIT Institute Of Technology, Affiliated to JNTUA , Andhra Pradesh, India
[2]Associate Professor & HOD in Dept. of CSE, BIT Institute Of Technology, Affiliated to JNTUA, Andhra Pradesh, India

ABSTRACT----- *The Nearest keyword set (referred to as NKS) queries on text-rich multi-dimensional datasets. An NKS query is a set of user-provided keywords, and the result of the query may include k sets of data points each of which contains all the query NKS queries are useful for many applications, such as photo-sharing in social networks, graph pattern search, geo location search in GIS systems1 and so on.NKS queries are useful for graph pattern search, where labeled graphs are embedded in a high dimensional for scalability. In this case, a search for a sub graph with a set of specified labels can be answered by an NKS query in the embedded space. NKS queries can also reveal geographic patterns. GIS can characterize a region by a high-dimensional set of attributes, such as pressure, humidity, and soil types. Meanwhile, these regions can also be tagged with information such as diseases. An epidemiologist can formulate NKS queries to discover patterns by finding a set of similar regions with all the diseases of her interest. we develop an exact ProMiSH (referred to as ProMiSH-E) that always retrieves the optimal top-k results, and an approximate ProMiSH (referred to as ProMiSH-A) that is more efficient in terms of time and space, and is able to obtain near-optimal results in practice.. Based on this index, ProMiSH-A which searches near-optimal results with better efficiency. ProMiSH is faster than state-of-the-art tree-based techniques, with multiple orders of magnitude performance improvement.*

Index Term: Querying, multi-dimensional data, indexing, hashing

I.INTRODUCTION

In today's digital world the amount of data which is developed is increasing day by day. There is different multimedia in which data is saved. It's very difficult to search the large dataset for a given query as well to archive more accuracy on user query. In the same time query will search on dataset for exact keyword match and it will not find the nearest keyword for accuracy. Ex: Flickr.

A variety of queries, semantically dissimilar from our NKS queries, have been studied in literature on text rich spatial datasets. Location-specific keyword queries on the web and in the GIS systems were earlier answered using a combination of R-Tree and inverted index. The NKS query is similar to the m-closest keywords query. They designed bR-Tree based on a R*-tree that also stores bitmaps and minimum bounding rectangles (MBRs) of keywords in every node along with points MBRs.
ProMiSH (short for Projection and Multi-Scale Hashing) to enable fast processing for NKS queries. In particular, we develop an exact ProMiSH (referred to as ProMiSH-E) that always retrieves the optimal top-k results, and an approximate ProMiSH (referred to as ProMiSH-A) that is more efficient in terms of time and space, and is able to obtain near-optimal results in practice.. Based on this index, ProMiSH-A which searches near-optimal results with better efficiency. ProMiSH is faster than state-of-the-art tree-based techniques, with multiple orders of magnitude performance improvement.

The problem of top-k nearest keyword set search in multi-dimensional datasets. The novel index called ProMiSH based on random projections and hashing. Based on this index, a ProMiSH-E is developed that finds an optimal subset of points and ProMiSH-A which searches near-optimal results with better efficiency.

II. LITERATURE SURVEY

1) Locating mapped resources in Web 2.0

AUTHORS: D. Zhang, B. C. Ooi, and A. K. H. Tung Mapping mashups are emerging Web 2.0 applications in which data objects such as blogs, photos and videos from different sources are combined and marked in a map using APIs that are released by online mapping solutions such as Google and Yahoo Maps. These objects are typically associated with a set of tags capturing the embedded semantic and a set of coordinates indicating their geographical locations. Traditional web resource searching strategies are not effective in such an environment due to the lack of the gazetteer context in the tags. Instead, a better alternative approach is to locate an object by tag matching. However, the number of tags associated with each object is typically small, making it difficult for an object to capture the complete semantics in the query objects. In this paper, we focus on the fundamental application of locating geographical resources and propose an efficient tag-centric query processing strategy. In particular, we aim to find a set of nearest co-located objects which together match the query tags. Given the fact that there could be large number of data objects and tags, we develop an efficient search algorithm that can scale up in terms of the number of objects and tags. Further, to ensure that the results are relevant, we also propose a geographical context sensitive geo-tf-idf ranking mechanism. Our experiments on synthetic data sets demonstrate its scalability while the experiments using the real life data set confirm its practicality.

2) Geo-clustering of Images with Missing GeoTags

AUTHORS: V. Singh, S. Venkatesha, and A. K. Singh

Images with GPS coordinates are a rich source of information about a geographic location. Innovative user services and applications are being built using geo tagged images taken from community contributed repositories like Flickr. Only a small subset of the images in these repositories is geo tagged, limiting their exploration and effective utilization. We propose to use optional meta-data along with image content to geo-cluster all the images in a partly geo tagged dataset. We formulate the problem as a graph clustering problem where edge weights are vectors of incomparable components. We develop probabilistic approaches to fuse the components into a single measure and then, discover clusters using an existing random walk method. Our empirical results strongly show that meta-data can be successfully exploited and merged together to achieve geo clustering of images missing geo tags.

3) Keyword Search in Spatial Databases: Towards Searching by Document

AUTHORS: D.Zhang,Y.M.Chee,A. Mondal, A. K. H. Tung, and M. Kitsuregawa

This work addresses a novel spatial keyword query called the m-closest keywords (mCK) query. Given a database of spatial objects, each tuple is associated with some descriptive information represented in the form of keywords. The mCK query aims to find the spatially closest tuples which match m user-specified keywords. Given a set of keywords from a document, mCK query can be very useful in geotagging the document by comparing the keywords to other geotagged documents in a database. To answer mCK queries efficiently, we introduce a new index called the bR*-tree, which is an extension of the R*-tree. Based on bR*-tree, we exploit a priori-based search strategies to effectively reduce the search space. We also propose two monotone constraints, namely the distance mutex and keyword mutex, as our a priori properties to facilitate effective pruning. Our performance study demonstrates that our search strategy is indeed efficient in reducing query response time and demonstrates remarkable scalability in terms of the number of query keywords which is essential for our main application of searching by document.

4) Keyword Search on Spatial Databases Sign In or Purchase

AUTHORS: I. De Felipe, V. Hristidis, and N. Rishe

Many applications require finding objects closest to a specified location that contains a set of keywords. For example online yellow pages allow users to specify an address and a set of keywords. In return the user obtains a list of businesses whose description contains these keywords ordered by their distance from the specified address. The problems of nearest neighbor search on spatial data and keyword search on text data have been extensively studied separately. However to the best of our knowledge there is no efficient method to answer spatial keyword queries that is queries that specify both a location and a set of keywords. In this work we present an efficient method to answer top-k spatial keyword queries. To do so we introduce an indexing structure called IR^2-Tree (Information Retrieval R-Tree) which combines an R-Tree with superimposed text signatures. We present algorithms that construct and maintain an IR^2-Tree and use it to answer top-k spatial keyword queries. Our algorithms are experimentally compared to current methods and are shown to have superior performance and excellent scalability.

III. EXISTING SYSTEM

Location-specific keyword queries on the web and in the GIS systems were earlier answered using a combination of R-Tree and inverted index.
Felipe et al. developed IR2-Tree to rank objects from spatial datasets based on a combination of their distances to the query locations and the relevance of their text descriptions to the query keywords.
Cong et al. integrated R-tree and inverted file to answer a query similar to Felipe et al. using a different ranking function.

DISADVANTAGES OF EXISTING SYSTEM:

These techniques do not provide concrete guidelines on how to enable efficient processing for the type of queries where query coordinates are missing.
In multi-dimensional spaces, it is difficult for users to provide meaningful coordinates, and our work deals with another type of queries where users can only provide keywords as input.
Without query coordinates, it is difficult to adapt existing techniques to our problem.
Note that a simple reduction that treats the coordinates of each data point as possible query coordinates suffers poor scalability

IV.PROPOSED WORK

In our proposed system the real data set is collected from photo sharing websites. In which we collect images from descriptive tags from Flickr and the images are transformed into grayscale and associate each data point, with a set of keyword that are derived from tags. We can collect number of datasets, suppose we collect five datasets (R1, R2, R3, R4, R5) with up to million data points, we can create multiple dataset to investigate performance. The query co-ordinates play a fundamental role in every step of algorithm to prune search space. Our work deals with providing keyword as an input. . We propose a novel multi-scale index for exact and approximate NKS query processing. We develop efficient search algorithms that work with the multi-scale indexes for fast query processing. Distance browsing is easy with R-trees. In fact, the best-first algorithm is exactly designed to output data points in ascending order of their distances.

- In order to run the application efficiently the user must have following characteristics.
- USER Module: User provides the input keyword as an image. SYSTEM Module:
- The system module retrieves all images from the database, and then it analyzes keywords.
- The positive point relation is undertaken by the system.
- It analyzes image keyword relation between points.
- It filters the image based on the relations.
- Applying nearest neighbor method retrieved images.
- Displays nearest image as an output.

ARCHITECTURE

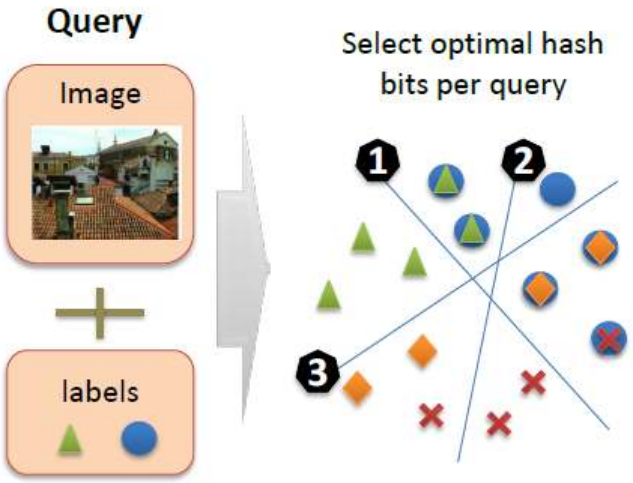

Fig 1: system Architecture

IMPLEMENTATION

The Index Structure for Exact Search (ProMiSH-E):

- In This Project we start with the index for exact ProMiSH (ProMiSH-E). This index consists of two main components.
- **Inverted Index Ikp**: The first component is an inverted index referred to as Ikp. In Ikp, we treat keywords as keys, and each keyword points to a set of data points that are associated with the keyword. Let D be a set of data points and V be a dictionary that contains all the keywords appearing in D. We build Ikp for D as follows. (1) For each ,we create a key entry in I kp, and this key entry points to a set of data points (i.e., a set includes all data points in D that contain keyword v). (2) We repeat (1) until all the keywords in V are processed.
- **Hashtable-Inverted Index Pairs HI**: The second component consists of multiple hashtables and inverted indexes referred to as HI. HI is controlled by three parameters: (1) (Index level) L, (2) (Number of random unit vectors) m, and (3) (hashtable size) B. All the three parameters are non-negative integers. These three parameters control the construction of HI.

The Exact Search Algorithm:

- We present the search algorithms in ProMiSH-E that finds top-k results for NKS queries. First, we introduce two lemmas that guarantee ProMiSH-E always retrieves the optimal top-k results.
- We project all the data points in D on a unit random vector and partition the projected values into overlapping bins of bin-width. If we perform a search in each of the bins independently, that the top-1 result of query Q will be found in one of the bins. ProMiSH-E explores each selected bucket using an efficient pruning based technique to generate results. ProMiSH-E terminates after exploring HI structure at the smallest index level s such that all the top-k results have been found. The efficiency of ProMiSH-E highly depends on an efficient search algorithm that finds top-k results from a subset of data points.

Optimization Techniques

- An algorithm for finding top-k tightest clusters in a subset of points. A subset is obtained from a hash table bucket Points in the subset are grouped based on the query keywords. Then, all the promising candidates are explored by a multi-way distance join of these groups. The join uses rk, the diameter of the kth result obtained so far by ProMiSH-E, as the distance threshold.
- A suitable ordering of the groups leads to an efficient candidate exploration by a multi-way distance join. We first perform a pairwise inner joins of the groups with distance threshold rk. In inner join, a pair of points from two groups are joined only if the distance between them is at most rk.
- We propose a greedy approach to find the ordering of groups. The weight of an edge is the count of point pairs obtained by an inner join of the corresponding groups. The greedy method starts by selecting an edge having the least weight. If there are multiple edges with the same weight, then an edge is selected at random and we perform a multi-way distance join of the groups by nested loops.

The Approximate Algorithm (ProMiSH-A):

- The approximate version of ProMiSH referred to as ProMiSH-A. We start with the algorithm description of ProMiSH-A, and then analyze its approximation quality.
- ProMiSH-A is more time and space efficient than ProMiSH-E, and is able to obtain near-optimal

results in practice. The index structure and the search method of ProMiSH-A are similar to ProMiSH-E.

- The index structure of ProMiSH-A differs from ProMiSH-E in the way of partitioning projection space of random unit vectors. ProMiSH-A partitions projection space into non-overlapping bins of equal width, unlike ProMiSH-E which partitions projection space into overlapping bins. The search algorithm in ProMiSH-A differs from ProMiSH-E in the termination condition. ProMiSH-A checks for a termination condition after fully exploring a hashtable at a given index level: It terminates if it has k entries with nonempty data point sets in its priority queue PQ.

V. CONCLUSION

The proposed system provides accurate results in multiple keyword search. This is how user data can be used to enhance search list and to find interest of the user. In our project we proposed how social annotations will be useful in the field of complex word search, which gives optimization as day by day large size of data available for searching by interest will be the future for search engines. The main advantage of this system will save lacks of processor cycles used in multidimensional data sets for finding image..

REFERENCES

[1] Y. Zhou, X. Xie, C. Wang, Y. Gong, and W.-Y. Ma, "Hybrid index structures for location-based web search," in Proc. 14th ACM Int. Conf. Inf. Knowl. Manage., 2005, pp. 155–162.
[2] 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. Statistical Database Manage., 2007, p. 16.
[3] S. Vaid, C. B. Jones, H. Joho, and M. Sanderson, "Spati o-textualindexing for geographical search on the web," in Proc. 9th Int. Conf. Adv. Spatial Temporal Databases, 2005, pp. 218–235.
[4] A. Khodaei, C. Shahabi, and C. Li, "Hybrid indexing and seamless ranking of spatial and textual features of web documents," in Proc. 21st Int. Conf. Database Expert Syst. Appl., 2010, pp. 450–466.
[5] A. Guttman, "R-trees: A dynamic index structure for spatial searching," in Proc. ACM SIGMOD Int. Conf. Manage. Data, 1984, pp. 47–57.
[6] I. De Felipe, V. Hristidis, and N. Rishe, "Keyword search on spatial databases," in Proc. IEEE 24th Int. Conf. Data Eng., 2008, pp. 656–665.
[7] G. Cong, C. S. Jensen, and D. Wu, "Efficient retrieval of the top-k most relevant spatial web objects," Proc. VLDB Endowment, vol. 2, pp. 337–348, 2009.
[8] B. Martins, M. J. Silva, and L. Andrade, "Indexing and ranking in Geo-IR systems," in Proc. Workshop Geographic Inf., 2005, pp. 31–34.
[9] Z. Li, H. Xu, Y. Lu, and A. Qian, "Aggregate nearest keyword search in spatial databases," in Proc. 12th Int. Asia-Pacific Web Conf., 2010, pp. 15–21.
[10] M. L. Yiu, X. Dai, N. Mamoulis, and M. Vaitis, "Top-k spatial preference queries," in Proc. IEEE 23rd Int. Conf. Data Eng., 2007, pp. 1076–1085.