ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 1, January 2019

Multi-Scale Hashing of Nearest Keyword Search in Multi-Dimensional Datasets

Chinni BalaVijaya Durga[1], Dr. G. Rama Mohan Babu.$_{\text{M.Tech.,Ph.D}}$[2]

Research Scholar, University College of Engineering & Technology, Acharya Nagarjuna University, Nagarjuna Nagar, Guntur, India[1]

Professor, Dept. of IT, RVR & JC College of Engineering, Chowdavaram, Guntur, India [2]

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 queryNKS 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.

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 MBRsProMiSH (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.

I. INTRODUCTION

Nearest keyword search is a classic problem with tremendous impacts on artificial intelligence, pattern recognition, information retrieval, and so on. Due to the popularity of keyword search, particularly on the Internet, many of these applications allow the user to provide a list of keywords thatthe spatial objects (henceforth referred to simply as objects)should contain, in their description or other attribute. Forexample, online yellow pages allow users to specify anaddress and a set of keywords, and return businesses whosedescription contains these keywords, ordered by their distanceto the specified address location. As another example, realestate web sites allow users to search for properties withspecific keywords in their description and rank themaccording to their distance from a specified location. So a method of nearest keyword set search in multi-dimensional datasets is implemented. In Existing techniques using tree based indexes suggest possible solution to NKS queries on multi-dimensional dataset, the performance of these algorithms decline sharply with the increase of size or dimensionality in dataset. Therefore there is need for an efficient algorithm that scales with dataset dimension, and yield practical query efficiency on large datasets. An NKS query is set of user-provide keywords, and result of the query may include k-sets of data points each of which contains all the query keywords and forms one of the top-k tightest cluster in the multi-dimensional space.

DOI: 10.15680/IJIRCCE.2019. 0701054

In this paper, we study 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 keywords and forms one of the top-k tightest cluster in the multi-dimensional space.

II. LITERATURE SURVEY

1) Locating mapped resources in Web 2.0, 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: 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 geotagged images taken from community contributed repositories like Flickr. Only a small subset of the images in these repositories is geotagged, 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 geotagged 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 geotags.
3) Keyword Search in Spatial Databases: Towards Searching by Document: 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: 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.

5) Top-k Spatial Preference Queries: M. L. Yiu, X. Dai, N. Mamoulis, and M. Vaitis - A spatial preference query ranks objects based on the qualities of features in their spatial neighborhood. For example, consider a real estate agency office that holds a database with available flats for lease. A customer may want to rank the flats with respect to the appropriateness of their location, defined after aggregating the qualities of other features (e.g., restaurants, cafes, hospital, market, etc.) within a distance range from them. In this paper, we formally define spatial preference queries and propose appropriate indexing techniques and search algorithms for them. Our methods are experimentally evaluated for a wide range of problem settings.

III. PROPOSED MECHANISM

- In this paper, we consider multi-dimensional datasets where each data point has a set of keywords. The presence of keywords in feature space allows for the development of new tools to query and explore these multi-dimensional datasets. we study 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 keywords and forms one of the top-k tightest cluster in the multi-dimensional space.
- In this paper, we propose 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.
- ProMiSH-E uses a set of hashtables and inverted indexes to perform a localized search.

ADVANTAGES OF PROPOSED SYSTEM:

- Better time and space efficiency.
- A novel multi-scale index for exact and approximate NKS query processing.
- It's an efficient search algorithms that work with the multi-scale indexes for fast query processing.

IV. SYSTEM ARCHITECTURE

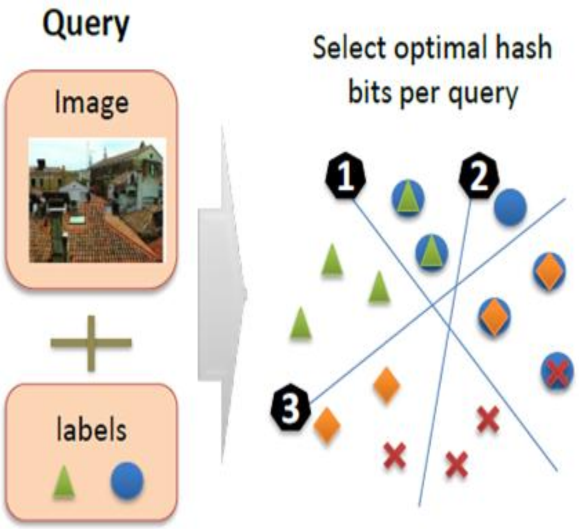

Fig 1: Architecture of the system

V. IMPLEMENTATION

- The Index Structure For Exact Search(ProMiSH-E)
- The Exact Search Algorithm
- Optimization Techniques
- The Approximate Algorithm (ProMiSH-A)

DESCRIPTION:

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) (hash table 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 pair wise 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 ofthe 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.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 1, January 2019

- 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 hash table at a given index level: It terminates if it has k entries with nonempty data point sets in its priority queue PQ.

VI. CONCLUSION

A novel index called ProMiSH based on random projections and hashing is implemented in this paper based on this index ProMiSH-E that finds an optimal subset of points and ProMiSH-A that searches near-optimal results with better efficiency. The empirical results show that ProMiSH is faster than state-of-the-art tree-based techniques, with multiple orders of magnitude performance improvement. Moreover, our techniques scale well with both real and synthetic datasets.

REFERENCES

[1] W. Li and C. X. Chen, “Efficient data modeling and querying system for multi-dimensional spatial data,” in Proc. 16th ACM SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst., 2008, pp. 58:1– 58:4.

[2] D. Zhang, B. C. Ooi, and A. K. H. Tung, “Locating mapped resources in web 2.0,” in Proc. IEEE 26th Int. Conf. Data Eng., 2010, pp. 521–532.

[3] V. Singh, S. Venkatesha, and A. K. Singh, “Geo-clustering of images with missing geotags,” in Proc. IEEE Int. Conf. Granular Comput., 2010, pp. 420–425.

[4] V. Singh, A. Bhattacharya, and A. K. Singh, “Querying spatial patterns,” in Proc. 13th Int. Conf. Extending Database Technol.: Adv.Database Technol., 2010, pp. 418–429.

[5] J. Bourgain, “On lipschitz embedding of finite metric spaces in hilbert space,” Israel J. Math., vol. 52, pp. 46–52, 1985.

[6] H. He and A. K. Singh, “GraphRank: Statistical modeling and mining of significant subgraphs in the feature space,” in Proc. 6th Int. Conf. Data Mining, 2006, pp. 885–890.

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

[8] C. Long, R. C.-W. Wong, K. Wang, and A. W.-C. Fu, “Collective spatial keyword queries: A distance owner-driven approach,” in Proc. ACM SIGMOD Int. Conf. Manage. Data, 2013, pp. 689–700.

[9] D. Zhang, Y. M. Chee, A. Mondal, A. K. H. Tung, and M. Kitsuregawa, “Keyword search in spatial databases: Towards searching by document,” in Proc. IEEE 25th Int. Conf. Data Eng., 2009, pp. 688–699.

[10] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, “Localitysensitive hashing scheme based on p-stable distributions,” in Proc. 20th Annu.Symp.Comput. Geometry, 2004, pp. 253–262.