Nearest Keyword Set Search In Multi-Dimensional Datasets

[1]R. ANITHA, [2]R. JAYA SUNDARI, [3]V. KANIMOZHI, [4]K. MUMTAJ BEGAM
[5]Mr. D.SATHYAMURTHY ME
[1,2,3,4]Students, [5]Assistant Professor
U.G Scholar MRK Institute of Technology Dept. of Computer Science

Abstract— *In computer Data set analysis, hundreds of files are usually examined. Much of the data in those files consists of unstructured text, whose analysis by computer examiners is difficult to be performed. In this context, automated methods of analysis are of great interest. In particular, algorithms for clustering documents can facilitate the discovery of new and useful knowledge from the documents under analysis we present an approach that applies document clustering algorithms to forensic analysis of computers seized in police investigation. We illustrate the proposed approach and get the lines and clustering word matching lines. We also present and discuss several practical results that can be useful for researchers and practitioners of Data set.*

Keywords—*Clustring,Filtering,Multi-dimensional data, Indexing, Hashing***.**

I. INTRODUCTION

Objects (e.g., images, chemical compounds, documents, orexperts in collaborative networks) are often characterized by acollection of relevant features, and are commonly repre-sentedas points in a multi- dimensional feature space. For example, images are represented using color feature vectors, and usually have descriptive text information (e.g., tags or keywords) associated with them. 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 thesemulti- dimensional datasets.

we study nearest keyword set (referred to as) 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. Fig. 1 illustrates an NKS query over a set of 2-dimensional data points. Each point is tagged with

a set of keywords. For a query Q = fa; b; cg, the set of points f7; 8; 9g contains all the query keywords fa; b; cg and forms the tightest cluster compared with any other set of points covering all the query keywords. Therefore, the set f7;8; 9g is the top-1 result for the query Q.

NKS queries are useful for many applications, such asphoto- sharing in social networks, graph pattern search,geo-location search in GIS systems1[1], [2], and so on. The following are a few examples. Consider a photo-sharing social network (e.g., Facebook),where photos are tagged with people names and Fig. 1. An example of an NKS query on a keyword tagged multi-dimensional dataset. The top-1 result for query fa; b; cg is the set of points f7; 8; 9g. locations. These photos can be embedded in a high dimensional feature space of texture, color, or shape [3], [4]. Here an NKS query can find a group of similar photos which contains a set of people.

NKS queries are useful for graph pattern search, where labeled graphs are embedded in a high dimensional space (e.g., through Lipschitz embedding [5]) for scalability. In this case, a search for a subgraph with a set of specified labels can be answered by an NKS query in the embedded space [6].

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 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 ProMiSHA) 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. The hashing technique is inspired by Locality Sensitive Hashing (LSH) [10], which is a state-of-the-art method for nearest neighbor search in high-dimensional spaces.

Unlike LSH-based methods that allow only approximate search with probabilistic guarantees, the index structure in ProMiSH-E supports accurate search. ProMiSH-E creates hashtables at multiple bin- widths, called index levels. A single round of search in a hashtable yields subsets of points that contain query results, and ProMiSH-E explores each subset using a fast pruning-based algorithm.

ProMiSH-A is an approximate variation of ProMiSH-E for better time and space efficiency. We evaluate the performance of ProMiSH on both real and synthetic datasets and employ state-of-the-art VbR - Tree [2] and CoSKQ [8] as baselines. The empirical results reveal that ProMiSH consistently outperforms the baseline algorithms with up to 60 times of speedup, and ProMiSH-A is up to 16 times faster than ProMiSH-E obtaining near-optimal results.

II. LITERATURE SURVEY

Z. Li, H. Xu, Y. Lu, and A. Qian, "Aggregate nearest keyword search in spatial databases," in Asia-Pacific Web Conference, 2010. Keyword search on relational databases is useful and popular for many users without technical background. Recently, aggregate keyword search on relational databases was proposed and has attracted interest. However, two important problems still remain. First, aggregate keyword search can be very costly on large relational databases, partly due to the lack of effcient in-dexes. Second, the top-k answers to an aggregate keyword query has not been addressed systematically, including both the ranking model and the effcient evaluation methods. We also report a systematic performance evaluation using real data sets.

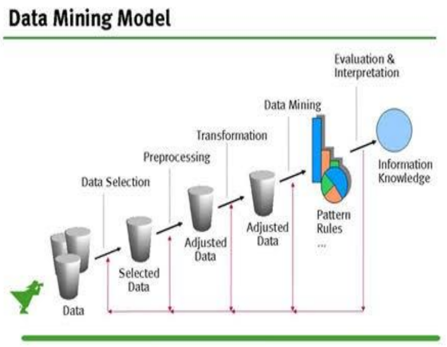
De Felipe, V. Hristidis, and N. Rishe, "Keyword search on spatial databases," in ICDE, 2008, pp. 656–665.

Many applications require finding objects closest to aspecified 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.

M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, —Locality-sensitivehashing scheme based on p-stable distributions,‖ in SCG, 2004.

We present a novel Locality-Sensitive Hashing scheme for the Approximate Nearest Neighbor Problem under lp norm, based on pstable distributions. Our scheme improves the running time of the earlier algorithm for the case of the l2 norm. It also yields the first known provably efficient approximate NN algorithm for the case p < 1. We also show that the algorithm finds the *exact* near neigbhor in O(log n) time for data satisfying certain "bounded growth" condition. Unlike earlier schemes, our LSH scheme works directly on points in the Euclidean space without embeddings. Consequently, the resulting query time bound is free of large factors and is simple and easy to implement. Our experiments (on synthetic data sets) show that the our data structure is up to 40 times faster than kd-tree. Our algorithm also inherits two very convenient properties of LSH schemes. The first one is that it works well on data that isextremely high-dimensional but sparse. Specifically, the running time bound remains unchanged if d denotes the maximum numberof non-zero elements in vectors. To our knowledge, this propertyis not shared by other known spatial data structures.

III. GENERAL DIAGRAM FOR DATA MINING



IV. EXISTING SYTSEM

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.

V. DISADVANTAGES OF EXISTING SYSTEM

These techniques do not provide concrete guidelines on how to enable efficient processing for the typ e 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.

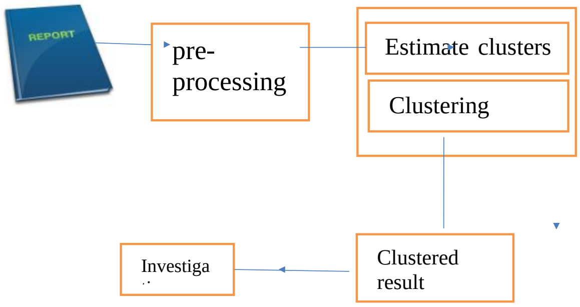
VI. PROPOSED SYSTEM

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. 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. we propose ProMiSH (short for Projection and Multi-Scale Hashing) to enable fast processing for NKS queries ProMiSH- E uses a set of hash tables and inverted indexes to perform a localized search.

VII. 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. We conduct extensive experimental studies to demonstrate the performance of the proposed techniques.

VIII. SYSTEM ARCHITECTURE AND MODULES



Estimate the Number of Clusters From Data

In order to estimate the number of clusters, a widely used approach consists of getting a set of data partitions with different numbers of clusters and then selecting that particular partition that provides the best result according to a specific quality criterion. Such a set of partitions may result directly from a hierarchical clustering dendrogram or, alternatively, from multiple runs of a partitional algorithm (e.g., K-means) starting from different numbers and initial positions of the cluster prototypes.

Applying Clustering Algorithms.

The clustering algorithms adopted in our study—the partitional K-means and K-medoids , the hierarchical Single/Complete/Average Link and the cluster ensemble based algorithm known as Distance are popular in the machine learning and data mining fields, and therefore they have been used in our study.

Removingthe Outliers

We assess a simple approach to remove outliers. This approach makes recursive use of the silhouette. Fundamentally, if the best partition chosen by the silhouette has singletons (i.e., clusters formed by a single object only), these are removed. Then, the clustering process is repeated over and over again until a partition without singletons is found. At the end of the process, all singletons are incorporated into the resulting data partition (for evaluation purposes) as single clusters.

IX. Experimental evaluation

We Evaluate Datasetsand provide Evaluative measure about the result by analysing the data.

SCREENSHOTS

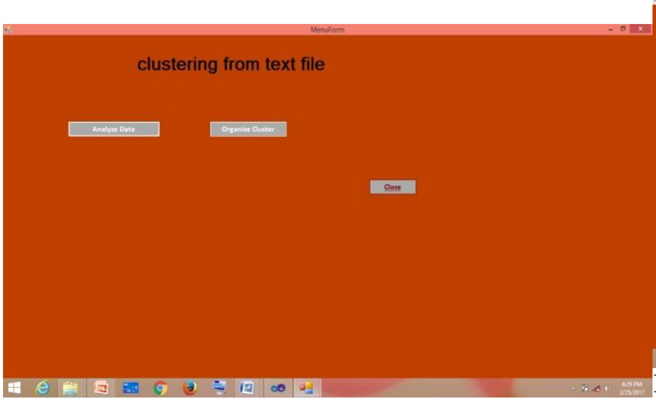
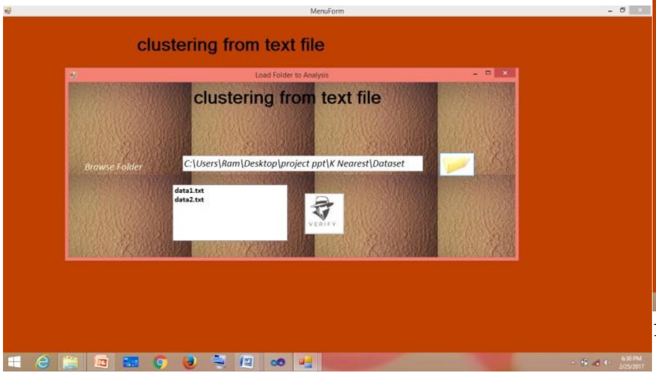


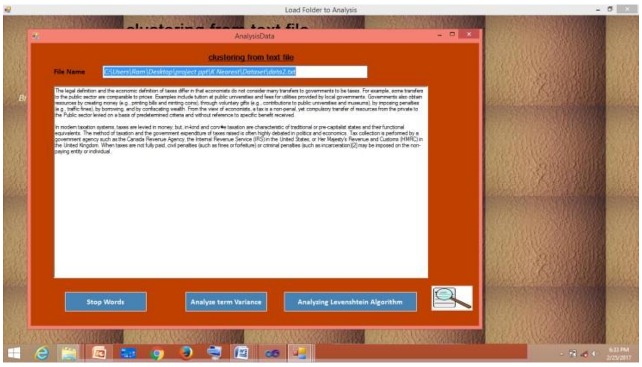
Fig.9.1: clustering from text file



C:\Users\Ram\Desktop\project ppt\K Nearest\Dataset

data1.txt
data2.txt

Fig.9.2: analyze the text file



File Name C:\Users\Ram\Desktop\project ppt\K Nearest\Dataset\data2.txt

The legal definition and the economic definition of taxes differ in that economists do not consider many transfers to governments to be taxes. For example, some transfers to the public sector are comparable to prices. Examples include tuition at public universities and fees for utilities provided by local governments. Governments also obtain resources by creating money (e.g., printing bills and minting coins), through voluntary gifts (e.g., contributions to public universities and museums), by imposing penalties (e.g., traffic fines), by borrowing, and by confiscating wealth. From the view of economists, a tax is a non-penal, yet compulsory transfer of resources from the private to the Public sector levied on a basis of predetermined criteria and without reference to specific benefit received.

In modern taxation systems, taxes are levied in money; but, in-kind and corvée taxation are characteristic of traditional or pre-capitalist states and their functional equivalents. The method of taxation and the government expenditure of taxes raised is often highly debated in politics and economics. Tax collection is performed by a government agency such as the Canada Revenue Agency, the Internal Revenue Service (IRS) in the United States, or Her Majesty's Revenue and Customs (HMRC) in the United Kingdom. When taxes are not fully paid, civil penalties (such as fines or forfeiture) or criminal penalties (such as incarceration)[2] may be imposed on the non-paying entity or individual.

Fig. 9.3: reading in text file

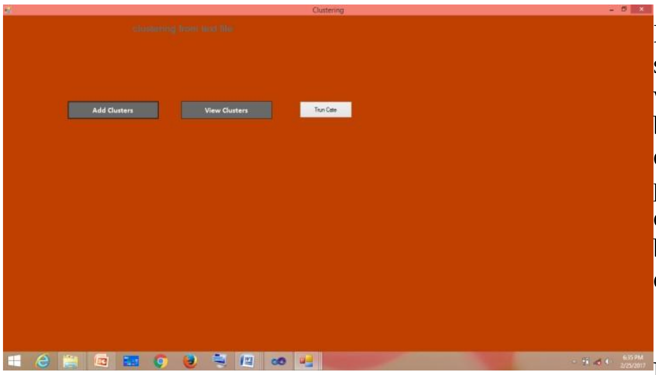
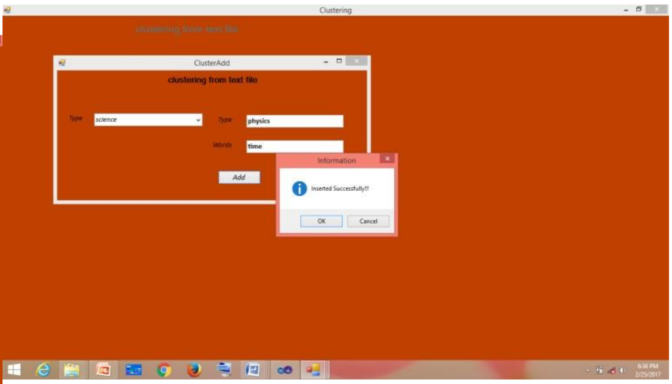


Fig.9.4: add clusters and view clusters



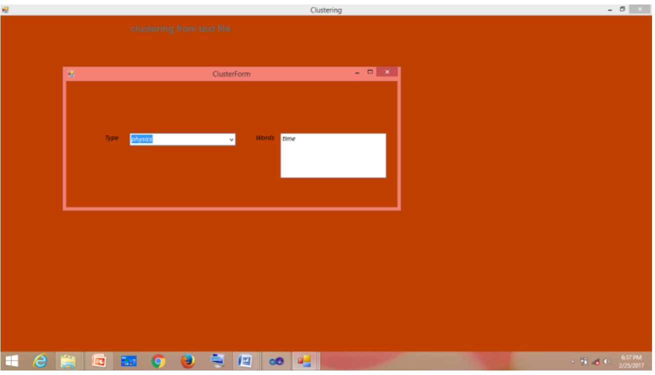
Type science

Type physics

Words time

Inserted Successfully!!

Fig. 9.5: analyze the type of cluster



Type physics

Words time

fig. 9.6: viewing from add clusters

X. CONCLUSION

we proposed solutions to the problem of top-k nearest keyword set search in multi-dimensional datasets. We proposed a novel index called ProMiSH based on ran-dom projections and hashing. Based on this index, we de-veloped ProMiSH-E that finds an optimal subset of points and ProMiSH-A that searches near-optimal results with better data structures starting at the smallest scale to generate the candidate point ids for the subset search, and it reads only required buckets from the hashtable and the inverted index of a HI structure. Therefore, all the hashtables and the inverted indexes of HI can again be stored using a similar directory-file structure

XI. FUTURE ENHANCEMENTS

In the future, we plan to explore other scoring schemes for ranking the result sets. In one scheme, we may assign weights to the keywords of a point by using techniques like tf- idf. Then, each group of points can be scored based on distance between points and weights of keywords. Furthermore, the criteria of a result containing all the keywords can be relaxed to generate results having only a subset of the query keyword.

EFFICIENCY: our empirical results show that promishis faster future enhancement

REFERENCES

[1] W. Li and C. X. Chen, ―Efficient data modeling and querying system for multi-dimensional spatial data,‖ in GIS, 2008, pp. 58:1–58:4.

[2] D. Zhang, B. C. Ooi, and A. K. H. Tung, ―Locating mappedresources in web 2.0,‖ in ICDE, 2010, pp. 521–532.

[3] V. Singh, S. Venkatesha, and A. K. Singh, ―Geo-clustering ofimages with missing geotags,‖ in GRC, 2010, pp. 420–425.

[4] V. Singh, A. Bhattacharya, and A. K. Singh, ―Querying spatialpatterns,‖
in EDBT, 2010, pp. 418–429.

[5] J. Bourgain, ―On lipschitz embedding of finite metric spaces inhilbert
space,‖ Israel J. Math., vol. 52, pp. 46–52, 1985.

[6] H. He and A. K. Singh, ―Graphrank: Statistical modeling and mining ofsignificant subgraphs in the feature space,‖ in ICDM, 2006, pp. 885–890.

[7] X. Cao, G. Cong, C. S. Jensen, and B. C. Ooi, ―Collective spatialkeyword
querying,‖ in SIGMOD, 2011.

[8] C. Long, R. C.-W. Wong, K. Wang, and A. W.-C. Fu, ―Collective spatialkeyword queries: a distance owner-driven approach,‖ in SIGMOD, 2013. [9] D. Zhang, Y. M. Chee, A. Mondal, A. K. H. Tung, and M.Kitsuregawa,
―Keyword search in spatial databases: Towardssearching by document,‖ in ICDE, 2009, pp. 688–699.

[10] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, ―Locality-sensitivehashing scheme based on p-stable distributions,‖ in SCG, 2004.

[11] Y. Zhou, X. Xie, C. Wang, Y. Gong, and W.-Y. Ma, "Hybrid index structures for location-based web search," in CIKM, 2005.

[12] R. Hariharan, B. Hore, C. Li, and S. Mehrotra, "Processing spatial- keyword (SK) queries in geographic information retrieval (GIR) sys- tems," in SSDBM, 2007.

[13] S. Vaid, C. B. Jones, H. Joho, and M. Sanderson, "Spatio-textual indexing for geographical search on the web," in SSTD, 2005.

[14] A. Khodaei, C. Shahabi, and C. Li, "Hybrid indexing and seamless ranking of spatial and textual features of web documents," in DEXA,
2010, pp. 450–466.

[15] A. Guttman, "R-trees: A dynamic index structure for spatial searching,"in ACM SIGMOD, 1984, pp. 47–57.

[16] I. De Felipe, V. Hristidis, and N. Rishe, "Keyword search on spatial databases," in ICDE, 2008, pp. 656–665.

[17] G. Cong, C. S. Jensen, and D. Wu, "Efficient retrieval of the top-k most relevant spatial web objects," PVLDB, vol. 2, pp. 337–348, 2009.

[18] B. Martins, M. J. Silva, and L. Andrade, "Indexing and ranking in geo-ir systems," in workshop on GIR, 2005, pp. 31–34.

[19] Z. Li, H. Xu, Y. Lu, and A. Qian, "Aggregate nearest keyword search in spatial databases," in Asia-Pacific Web Conference, 2010.

[20] M. L. Yiu, X. Dai, N. Mamoulis, and M. Vaitis, "Top-k spatial preference queries," in ICDE, 2007, pp. 1076–1085.