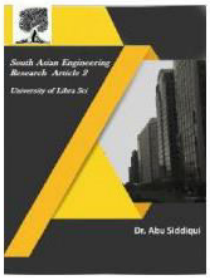
International Journal For Recent Developments in Science & Technology
2581-4575
Crossref
A Peer Reviewed Research Journal



NKS IN MULTI-DIMENSIONAL DATASETS

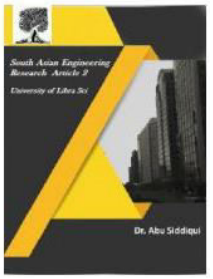
[1]MR. G. VENENDRA, [2]R. SAI RAMYA, [3]S. VENKATA SIVA SAI TEJA, [4]M. MANOJ
[1]Assistant Professor, CSE Department NRI INSTITUTE OF TECHNOLOGY, POTHAVARAPPADU
[2,3,4]UG Student, CSE Department, NRI INSTITUTE OF TECHNOLOGY, POTHAVARAPPADU

ABSTRACT
In computer Data set analysis, many files are usually examined. Much of the info in those files consists of unstructured text, whose analysis by computer examiners is difficult to be performed. During this context, automated methods of study are of great interest. Especially, algorithms for clustering documents can facilitate the invention of latest 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 work. We illustrate the proposed approach and obtain the lines and clustering word matching lines. We also present and discuss several practical results which will be useful for researchers and practitioners of knowledge set.
Keyword - Filtering, Multi-dimensional data, Indexing, Hashing.

I.INTRODUCTION

Objects (e.g., images, chemical compounds, documents, or experts in collaborative networks) are often characterized by a collection of relevant features, and are commonly represented as points during a multi- dimensional feature space. For instance, images are represented using color feature vectors, and typically have descriptive text information (e.g., tags or keywords) related to them. During this paper, we consider multi- dimensional datasets where each datum features a set of keywords. The presence of keywords in feature space allows for the event of latest tools to question 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 may be a set of user-provided keywords, and therefore the results of the query may include k sets of points each of which contains all the query keywords and forms one among the top-k tightest cluster within the multi-dimensional space. Fig. 1 illustrates an NKS query over a group of 2- dimensional data points. Each point is tagged with a group of keywords. For a question 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 the other set of points covering all the query keywords. Therefore, the set f7;8; 9g is that the top- 1 result for the query Q. NKS queries are useful for several applications, such as photo- sharing in social networks, graph pattern search, geo-location search in GIS systems1[1], [2], and so on. The subsequent are a couple of examples. Consider a photo-sharing social network (e.g., Facebook), where photos are

International Journal For Recent Developments in Science & Technology
2581-4575
Crossref
A Peer Reviewed Research Journal



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 that the set of points f7; 8; 9g. locations. These photos are often embedded during a high dimensional feature space of texture, color, or shape [3], [4]. Here an NKS query can find a gaggle of comparable photos which contains a group of individuals. NKS queries are useful for graph pattern search, where labeled graphs are embedded during a high dimensional space (e.g., through Lipschitz embedding [5]) for scalability. During this cases, an enquiry for a sub graph with a group of specified labels are often answered by an NKS query within the embedded space [6].

II.LITERATURE SURVEY

Vishwakarma Singh, Bo Zong, and Ambuj K. Singh In this paper, nearest keyword set referred to as NKS queries on text-rich multi-dimensional datasets. NKS query is a set of user-provided keywords, and the result of the query 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 NKS queries are useful for many applications, such as photo-sharing in social networks, graph pattern search, geolocation search in GIS systems and so on [1].

D. Zhang, B. C. Ooi, and A. K. H. Tung it explain about Web 2.0, it focus on the fundamental application of locating geographical resources. In Web 2.0, tagging is a popular means to annotate various resources, including news, blogs, speeches, photos and videos. Users are encouraged to add extra textual terms as semantic description or summarization for the objects. With human intelligence involved, the tags are well phrased so that much cost can be saved from handling term ambiguities. An inverted index is built along with the R-tree. It maintains inverted lists for all the tags in the database. Each element in the list consists of the node label derived from the construction of the R-tree and the actual location. Note that the list of locations are ordered by the label so that the data points close to each other in geographical space are probably still close in the inverted lists. Such an index is scalable in terms of both the number of locations and tags. Advantage of this system is an efficient tag centric query processing strategy but drawback is 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 [2].

X. Cao, G. Cong, C. S. Jensen, and B. C. Oo it explain about the Collective spatial keyword querying considered as standard spatial keyword queries. These all involve different conditions on the spatial and textual aspects of places. In spatial databases, the arguably most fundamental queries are range queries and k nearest neighbor queries. In text retrieval, queries may be Boolean, requiring results to contain the query keywords, or ranking- based, returning the k places that rank the highest according to a text similarity function [3].

R. Hariharan, B. Hore, C. Li, and S. Mehrotra it explain the location based information is stored in GIS database.

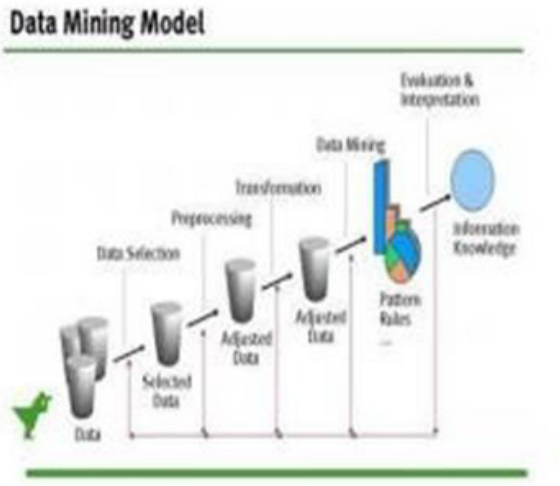
International Journal For Recent Developments in Science & Technology
2581-4575
Crossref
A Peer Reviewed Research Journal

Location-specific keyword queries on the web and in the GIS systems were earlier answered using a combination of R-Tree and inverted index. Tree-based indexes, such as R-Tree and M-Tree, have been extensively investigated for nearest neighbor search in high-dimensional spaces. These information entities of such databases have both spatial and textual descriptions. This method introduces a framework for GIR system and focus is on indexing strategies that can process spatial keyword query. It introduces two index structures to store spatial and textual information. 1) Separate index for spatial and text attributes 2) Hybrid index. But by using first structure that is separate index for spatial and text attributes, if filtering is done first, many objects may lie within a query is spatial extent, but very few of them are relevant to query keywords. This increases the disk access cost by generating a large number of candidate objects. The subsequent stage of keyword filtering becomes expensive. And by using second structure that is hybrid index there are high overhead in subsequent merging process [4].

A. Khodaei, C. Shahabi, and C. Li this paper explain the how large amount of location-based information generated and used by many applications. The Internet is the most popular source of data with location-specific information, such as documents describing schools at certain regions, Wikipedia pages containing spatial information and images with annotations and information about the places they were taken. Users of such a web-based application often need to query the system by providing requirements on a location as well as keywords in order to find relevant documents there is a significant commercial and research interest in location based web search engines. Given a number of search keywords and one or more locations that a user is interested in, a location-based web search retrieves and ranks the most textually and spatially relevant web pages. In this type of search, both the spatial and textual information should be indexed. Currently, no efficient index structure exists that can handle both the spatial and

textual aspects of data simultaneously and accurately. In this paper, it proposes a new index structure called Spatial-Keyword Inverted File to handle location-based web searches in an integrated/ efficient manner. To seamlessly find and rank relevant documents, develop a new distance measure called spatial tf-idf. Advantage is to perform top k searches but give poor performance [5].

III.GENERAL DIAGRAM FOR DATA MINING



International Journal For Recent Developments in Science & Technology
2581-4575
Crossref
A Peer Reviewed Research Journal

IV. EXISTING SYTSEM

Location-specific keyword queries on the online and within the GIS systems were earlier answered employing a combination of R-Tree and inverted index. Felipe et al. developed IR2-Tree to rank objects from spatial datasets supported a mixture of their distances to the query locations and therefore 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 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.

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. 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. 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.

VII. ADVANTAGES OF PROPOSED SYSTEM

• Better time and space efficiency.

• A novel multi-scale index for exact and approximate NKS queries processing.

• It's an efficient search algorithm that works with the multi-scale indexes for fast query processing.

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

VIII. PROCESS DIAGRAM AND MODULES

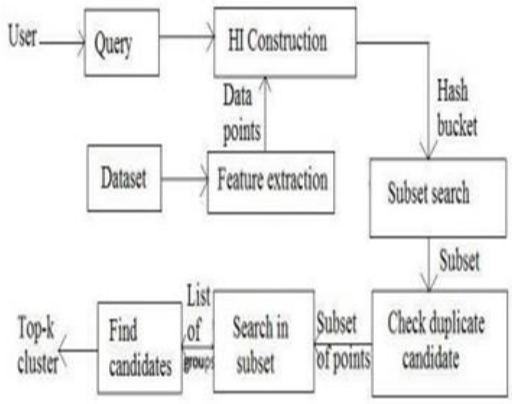


Fig: Process Diagram

International Journal For Recent Developments in Science & Technology
2581-4575
Crossref
A Peer Reviewed Research Journal

There are two modules.

• Search Algorithm Module

• HI Construction Module

1. Search Algorithm Module:

 ProMiSH referred to as ProMiSH-A. We start with the algorithm description of ProMiSH-A, and then analyse its approximation quality.

 ProMiSH-E highly depends on an efficient search algorithm that finds top- k results from a subset of data points.

2. HI Construction Module:

It consists of multiple hash tables and inverted indexes referred to as HI. HI is controlled by three parameters:

 Index level(L):

HI at all the index level then it performs a search in the complete dataset D.

 Number of random unit vectors(m)

We consider its projection space as a segment [0, pMax] and partition the segment into 2(L-s+1) +1 overlapping bins, where each bin has width and is equally overlapped with two other bins. We conduct the projection space partition on all the m random unit vectors.

 Hash table size(B)

A given a dictionary V and hash table H(s), we create the inverted index I(s)khb. In this inverted index, keys are still keywords. HI with one pair of hash table and inverted index shown in the dotted rectangle.

IX. EXPERIMENTAL EVALUATION

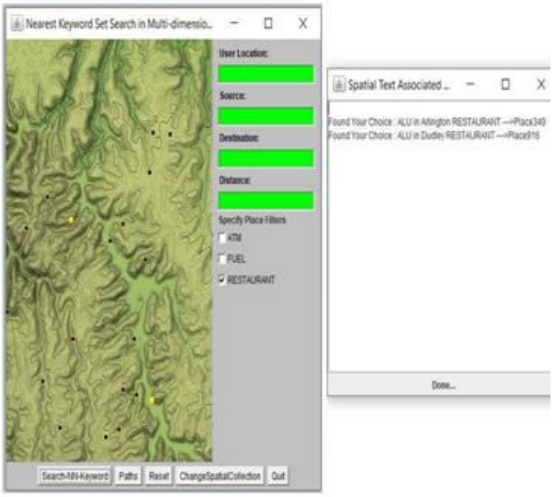


Fig: searching nearest restaruants

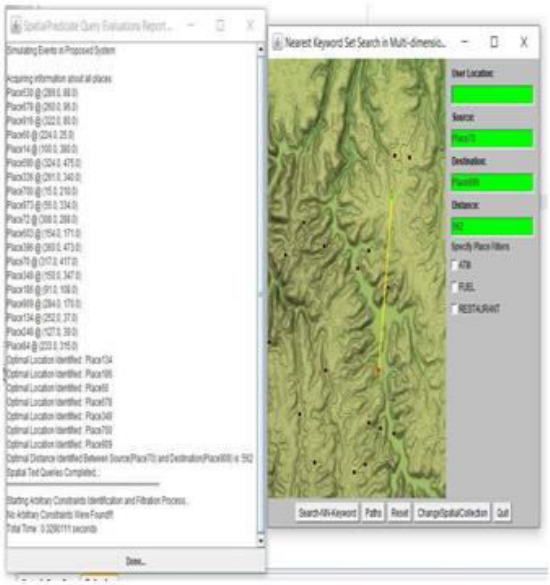
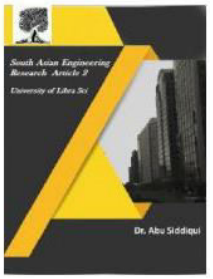


Fig: Finding nearest path

The above figure shows that the distance between the two places i.e., objects here. The window having the places visited between the two selected places and the distance between the points which is the optimal path.

International Journal For Recent Developments in Science & Technology
2581-4575
Crossref
A Peer Reviewed Research Journal



X. CONCLUSION

We proposed solutions for the matter of top-k nearest keyword set search in multi-dimensional datasets. We developed a particular (ProMiSH-E) and an approximate (ProMiSH-A) method. We designed a completely unique index supported random projections and hashing. Index is employed to seek out subset of points containing truth results. We also proposed an efficient solution to question results from a subset of knowledge points. Our empirical results show that ProMiSH is quicker than state-of-the art tree-based technique, having performance improvements of multiple orders of magnitude. These performance gains are further emphasized as dataset size and dimension increase, also as for giant query sizes. ProMiSH-A has the fastest query time. We empirically observed a linear scalability of ProMiSH with the dataset size, the dataset dimension, the query size, and therefore the result size. We also observed that ProMiSH yield practical query times on large datasets of high dimensions for queries of huge sizes.

REFERENCES

1. Vishwakarma Singh, Bo Zong, and Ambuj K. Singh, “Nearest Keyword Set Search in Multi-Dimensional Datasets”.
2. D. Zhang, B. C. Ooi, and A. K. H. Tung, “Locating mapped resources in web 2.0”, in ICDE, 2010, pp.521- 532.
3. X. Cao, G. Cong, C. S. Jensen, and B.
C. Ooi, “Collective spatial keyword querying”, in SIGMOD, 2011.pp.373-384.
4. R. Hariharan, B. Hore, C. Li, and S. Mehrotra, “Processing spatial keyword (SK) queries in geographic information retrieval (GIR) systems”, in SSDBM, 2007.
5. 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.
6. N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger, “The R*- tree: An efficient and robust access method for points and rectangles,” in SIGMOD, 1990, pp. 322–331.
7. R. Agrawal and R. Srikant, “Fast algorithms for mining association rules in large databases,” in VLDB, 1994, pp. 487–499.
8. P. Ciaccia, M. Patella, and P. Zezula, “M-tree: An efficient access method for similarity search in metric spaces,” in VLDB, 1997.
9. R. Weber, H.-J. Schek, and S. Blott, “A quantitative analysis and performance study for similarity- search methods in high-dimensional spaces,” in VLDB, 1998, pp. 194– 205.
10. H. V. Jagadish, B. C. Ooi, K.-L. Tan, C. Yu, and R. Zhang, “idistance: An adaptive B+-tree based indexing method for nearest neighbour search,” ACM TDS, vol. 30, no. 2, pp. 364–397, 2005.
11. M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, “Locality sensitive hashing scheme based on p- stable distributions,” in SCG, 2004.

12. Y. Zhou, X. Xie, C. Wang, Y. Gong, and W.-Y. Ma, "Hybrid index structures for location-based web search," in CIKM, 2005.
13. S. Vaid, C. B. Jones, H. Joho, and M. Sanderson, "Spatio-textual indexing for geographical search on the web," in SSTD, 2005.
14. Z. Li, H. Xu, Y. Lu, and A. Qian, "Aggregate nearest keyword search in spatial databases," in Asia-Pacific Web Conference, 2010.
15. Z. Li, H. Xu, Y. Lu, and A. Qian, "Aggregate nearest keyword search in spatial databases," in Asia-Pacific Web Conference, 2010.
16. M. L. Yiu, X. Dai, N. Mamoulis, and M. Vaitis, "Top-k spatial preference queries," in ICDE, 2007, pp. 1076– 1085.
17. T. Xia, D. Zhang, E. Kanoulas, and Y. Du, "On computing top-t most influential spatial sites," in VLDB, 2005, pp. 946–957.
18. Y. Du, D. Zhang, and T. Xia, "The optimal-location query," in SSTD, 2005, pp. 163–180.
19. A. Gionis, P. Indyk, and R. Motwani, "Similarity search in high dimensions via hashing," in VLDB, 1999, pp. 518–529.
20. V. Singh and A. K. Singh, "Simp: accurate and efficient near neighbour search in high dimensional spaces," in EDBT, 2012, pp. 492–503.