ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

An Efficient Nearest Keyword Set Query Search in Multi-Dimensional Datasets

Adina Karunasri
M.Tech Student,
Department of CSE,
BVC Engineering College, Odalarevu.

B.S.N.Murthy, M.Tech, (Ph.D)
Associate Professor,
Department of CSE,
BVC Engineering College, Odalarevu.

Abstract:
We focused on multi-dimensional dataset where each data point has set of keywords in feature space allows for the development of new tools to query and explore these multidimensional dataset. Here we study nearest keyword set Queries on text rich multidimensional dataset. We propose a new method called ProMiSH (Projection and Multi scale Hashing) that uses random projection and hash-based index structure. Our experimental result shows that ProMiSH has Speedup over state-of-art-tree-based techniques. Keyword-based search in text-rich multi-dimensional datasets facilitates many novel applications and tools. In this work, we consider objects that are tagged with keywords and are embedded in a vector space. For these datasets, we study queries that ask for the tightest groups of points satisfying a given set of keywords.

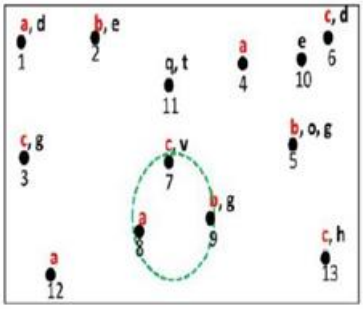
Keywords:
NKS Querying, multi-dimensional data, indexing, ProMiSH.

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. so we have implemented a method of nearest keyword set search in multi-dimensional datasets. 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. In this paper We study nearest keyword set queries on text-rich multi-dimensional datasets. we consider multi-dimensional datasets where each data point has a set of keywords.

Multi-Dimensional Data Sets:

The multi-dimensional points in the dataset are represented by dots. Each point has a unique identifier and is tagged with a set of keywords. For a query Q={a; b; c}, the set of points {7, 8,9} contains all the query keywords {a; b; c} and are nearest to each other compared to any other set of points containing these query keywords. Therefore, the set {7, 8, 9} is the top-1 Result for the query Q.



LITERATURE SURVEY:

[1]Mapping mash ups are emergent Web 2.0 applications in which data objects like blogs, images and videos from dissimilar sources are added together and marked in a map using APIs that are released by

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

online mapping solutions like Google and Yahoo Maps. These objects are mainly connected with a set of tags capturing the embedded semantic and a set of coordinates showing 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. In place of, 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 report, we concentrate on the basic 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 show its scalability while the experiments using the real life data set confirm its utility.

[2] 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. They 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. Author's 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] This work covers 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 proposes 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, they bring in a new index called the bR*-tree, which is an extension of the R*-tree. Based on bR*-tree, they exploit a priori-based search strategies to effectively reduce the search space. They 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] Many applications need finding objects closest to a specified location that have a set of keywords. For example online yellow pages allow users to specify an address and a set of keywords. In return the user gets 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 author's knowledge there are no efficient methods to answer spatial keyword queries that are queries that specify both a location and a set of keywords. In this work the author present an efficient method to answer top-k spatial keyword queries. To do so they introduced an indexing structure called IR2-Tree (Information Retrieval R-Tree) which combines an R-Tree with superimposed text signatures. they present

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

algorithms that construct and maintain an IR2-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] 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 rightness of their location, defined after combining the qualities of other features (e.g., restaurants, cafes, hospital, market, etc.) within a distance range from them. In this paper, the authors defined 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..

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:

- 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.

PROPOSED SYSTEM:

- In this report, we take 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.
- In this report, we study nearest keyword set (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 has all the query keywords and forms 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. Particularly , 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 hash tables and inverted indexes to execute a localized search.

Advantages:

- 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.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

SYSTEM ARCHITECTURE:

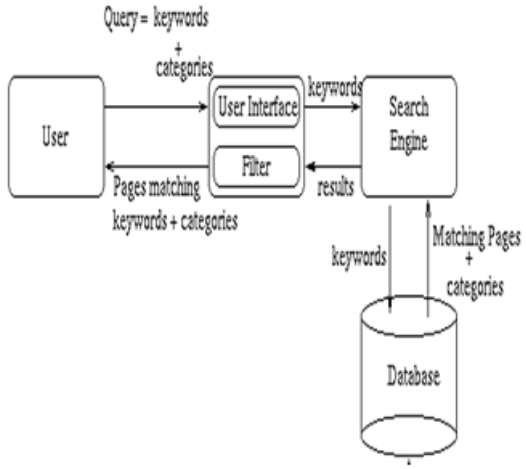


Fig system architecture

MODULES DESCRIPTION:

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

Algorithm:

In: Q: query keywords; k: number of top results
In: w0: initial bin-width
1: PQ ←[e([],+∞)]: priority queue of top-k results
2: HC: hashtable to check duplicate candidates
3: BS: bitset to track points having a query keyword
4: for all o ϵ U ʊ$_{vQ}$ϵQI$_{kp}$[v$_Q$] do
5: BS[o] ←true /* Find points having query keyword*/
6: end for
7: for all s ϵ{0,…, L-1}do
8: Get HI at s
9: E[]←0/* List of hash buckets*/
10: for all v$_Q$ ϵ Q do
11: for all bId ϵ I$_{khb}$[v$_Q$]do
12: E[bId] ←E[bid]+1
13: end for
14: end for
15: for all i ϵ(0,…, Size Of (E)) do
16: if E(i)= SizeOf(Q) then
17: F' ←Ø /* Obtain a subset of points*/
18: for all o ϵ H[i] do
19: if BS[o]= true then
20: F'← F' U o
21: end if
22: end for
23: if checkDuplicateCand(F', HC)=false then
24: searchInSubset(F', PQ)
25: end if
26: end if
27: end if
28: /* check termination condition */
29: if PQ[k].r <= $w_0 2^{s-1}$ then
30: Return PQ
31: end if
32: end for
33: /* Perform search on D If algorithm has not terminated */
34: for all o ϵ D do
35: if BS[o]=true then
36: F' ←F' U o
37: end if
38: end for
39: searchInSubset(F',PQ)
40: Return PQ

- 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.
- Hash table-Inverted Index Pairs HI: The second component consists of multiple hash tables 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-

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

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 hashtable bucket Points in the subset are grouped based on the query keywords. Then, all the promising candidates are searched by a multi-way distance join of these groups. The join uses rk, the diameter of the k^{th} result obtained 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 hash table at a given index level: It terminates if it has k entries with nonempty data point sets in its priority queue PQ.

CONCLUSION AND FUTURE SCOPE:

In this report, we suggested solutions to the problem of top-k nearest keyword set search in multi-dimensional datasets. We proposed a novel index called ProMiSH based on random projections and hashing. Based on this index, we developed ProMiSH-E that finds an optimal subset of points and ProMiSH-A which searches near-optimal results with better efficiency. Our observational 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. Ranking functions. In the future, we plan to explore other scoring schemes for ranking the result sets.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

In one scheme, we may allot 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. Moreover, the criteria of a result containing all the keywords can be relaxed to generate results having only a subset of the query keywords.

REFERENCES:

[1]D. Zhang, B. C. Ooi, and A. K. H. Tung, "Finding mapped assets in web 2.0," in Proc. IEEE 26th Int. Conf. Information Eng., 2010, pp. 521–532.

[2]V. Singh, S. Venkatesha, and A. K. Singh, "Geo-bunching of pictures with missing geotags," in Proc. IEEE Int. Conf. Granular Comput., 2010, pp. 420–425.

[3]D. Zhang, Y. M. Chee, A. Mondal, A. K. H. Tung, and M. Kitsuregawa, "Watchword look in spatial databases: Towards seeking by record," in Proc. IEEE 25th Int. Conf. Information Eng., 2009, pp. 688–699.

[4]I. De Felipe, V. Hristidis, and N. Rishe, "Catchphrase seek on spatial databases," in Proc. IEEE 24th Int. Conf. Information Eng., 2008, pp. 656–665.

[5]M. L. Yiu, X. Dai, N. Mamoulis, and M. Vaitis, "Best k spatial inclination inquiries," in Proc. IEEE 23rd Int. Conf. Information Eng., 2007, pp. 1076–1085.

[6]W. Li and C. X. Chen, "Effective information displaying and questioning framework for multi-dimensional spatial information," in Proc. sixteenth ACM SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst., 2008, pp. 58:1– 58:4.

[7]V. Singh, A. Bhattacharya, and A. K. Singh, "Questioning spatial examples," in Proc. thirteenth Int. Conf. Expanding Database Technol.: Adv.Database Technol., 2010, pp. 418–429.

[8]J. Bourgain, "On lipschitz inserting of limited metric spaces in hilbert space," Israel J. Math., vol. 52, pp. 46–52, 1985.

[9]H. He and A. K. Singh, "GraphRank: Statistical demonstrating and mining of critical subgraphs in the element space," in Proc. sixth Int. Conf. Information Mining, 2006, pp. 885–890.

[10]X. Cao, G. Cong, C. S. Jensen, and B. C. Ooi, "Aggregate spatial catchphrase questioning," in Proc. ACM SIGMOD Int. Conf. Oversee. Information, 2011, pp. 373–384.

[11]C. Long, R. C.- W. Wong, K. Wang, and A. W.-C. Fu, "Aggregate spatial catchphrase inquiries: A separation proprietor driven approach," in Proc. ACM SIGMOD Int. Conf. Oversee. Information, 2013, pp. 689–700.

[12]M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, "Localitysensitive hashing plan in light of p-stable circulations," in Proc. twentieth Annu. Symp. Comput. Geometry, 2004, pp. 253–262.

[13] Y. Zhou, X. Xie, C. Wang, Y. Gong, and W.- Y. Mama, "Half and half file structures for area based web seek," in Proc. fourteenth ACM Int. Conf. Inf. Knowl. Oversee., 2005, pp. 155–162.

[14]R. Hariharan, B. Hore, C. Li, and S. Mehrotra, "Handling spatialkeyword (SK) questions in geographic data recovery (GIR) frameworks," in Proc. nineteenth Int. Conf. Sci. Measurable Database Manage., 2007, p. 16.

[15]S. Vaid, C. B. Jones, H. Joho, and M. Sanderson, "Spati o-textualindexing for topographical hunt on the eb," in Proc. ninth Int. Conf. Adv. Spatial Temporal Databases, 2005, pp. 218–235.

[16]A. Khodaei, C. Shahabi, and C. Li, "Half and half ordering and consistent positioning of spatial and

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

literary elements of web records," in Proc. 21st Int. Conf. Database Expert Syst. Appl., 2010, pp. 450–466.

[17]A. Guttman, "R-trees: A dynamic file structure for spatial looking," in Proc. ACM SIGMOD Int. Conf. Oversee. Information, 1984, pp. 47–57.

[18]G. Cong, C. S. Jensen, and D. Wu, "Proficient recovery of the top-k most pertinent spatial web objects," Proc. VLDB Endowment, vol. 2, pp. 337–348, 2009.