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

Implementation and Analysis of ProMiSH (Projection and Multi Scale Hashing) in Multi-Dimensional Datasets

A. Nagarjuna
M.Tech,
Academic Consultant,
S. V. U. C. E,
S. V. University.

M. Damodhar
M.Tech,
Academic Consultant,
S. V. U. C. E,
S. V. University.

ABSTRACT:
Consider objects that are tagged with keywords and are embedded in a vector space. The presence of keywords in feature space allows for the development of new tools to query and explore these multi-dimensional datasets. We propose a method called Projection and Multi Scale Hashing that uses random projection and hash-based index structures, and achieves high scalability and speedup. 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. Images are represented using color feature vectors, and usually have descriptive text information (e.g., tags or keywords) associated with them. Our system is based on real datasets shows that we can show the efficient searching of keywords in multidimensional datasets.

Key Words: ***Querying, Multi-dimensional Data, Indexing, Hashing.***

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.

The amount of data which is developed is increasing day by day, thus it is very difficult to search large dataset for a given query as well to achieve more accuracy on user query.so we have implemented a method of efficient search in multidimensional dataset. This is associated with images as an input. Images are often characterized by a collection of relevant features, and are commonly represented as points in a multi-dimensional feature space. For example, images are represented using colour feature vectors, and usually have descriptive text information (e.g., tags or keywords) associated with them. 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.

Our main contributions are summarized as follows.
(1)We propose a novel multi-scale index for exact and approximate NKS query processing.
(2)We develop efficient search algorithms that work with the multi-scale indexes for fast query processing.
(3)We conduct extensive experimental studies to demonstrate the performance of the proposed techniques.
1. Filename: It is based on image filename.
2.CBIR (Content based image search): Content-based image retrieval (CBIR), also known as query by image content (QBIC) and content-based visual information retrieval (CBVIR) is the application of computer vision techniques to the image retrieval problem, that is, the problem of searching for digital images in large databases. Content- based image retrieval is opposed to traditional concept- based approaches (see Concept-based image indexing).

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

3.TBIR (Text based image search): Concept-based image indexing, also variably named as "description-based" or "text-based" image indexing/retrieval, refers to retrieval from text-based indexing of images that may employ keywords, subject headings, captions, or natural language text. It is opposed to Content-based image retrieval. Indexing is a technique used in CBIR.

We formally define NKS queries as follows.

Nearest keyword set. Let $\mathcal{D} \subset \mathcal{R}^d$ be a d-dimensional data-set with N points. For any $o \in \mathcal{D}$, it is tagged with a set of keywords $\sigma(o)=\{v_1,..,v_t\} \subseteq \mathcal{V}$, where $\mathcal{V}$ is a dictionary of U unique keywords. For any $o_i, o_j \in \mathcal{D}$, the distance between o_i and o_j is measured by their L_2-norm (i.e., euclidean distance) as $dist(o_i, o_j) = \|o_i - o_j\|_2$. Given a set of data points $A \subset \mathcal{D}$, $r(A)$ is the diameter A and is defined by the maximum distance between any two points in A,

$$r(A) = \max_{\forall o_i, o_j \in A} \|o_i - o_j\|_2.$$

A smaller $r(A)$ implies the points in A are more similar to each other.

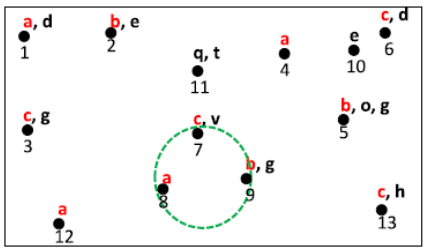


Fig. 1. An example of an NKS query on a keyword tagged multi-dimensional dataset. The top-1 result for query $\{a, b, c\}$ is the set of points $\{7, 8, 9\}$.

Given an NKS query with q keywords $Q = \{v_1, \ldots, v_q\}$, $A \subseteq \mathcal{D}$ is a candidate result of Q if it covers all the keywords in Q by $Q \subseteq \bigcup_{o \in A} \sigma(o)$. Let $\mathcal{S}$ be the set including all candidates of Q. The top-1 result A^* of Q is obtained by

$$A^* = \arg\min_{A \in \mathcal{S}} r(A).$$

Similarly, a top-k NKS query retrieves the top-k candidates with the least diameter. If two candidates have equal diameters, then they are further ranked by their cardinality.

Existing System:

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 clusters in the multi-dimensional space. NKS query over a set of two-dimensional data points. In tree-based indexes suggest possible solutions to NKS queries on multidimensional datasets, the performance of these algorithms deteriorates sharply with the increase of size or dimensionality in datasets.

Disadvantages:

1. NKS queries are useful for graph pattern search, where labeled graphs are embedded in a high dimensional space.
2. Nearest neighbor queries usually require coordinate information for queries, which makes it difficult to develop an efficient method to solve NKS queries by existing techniques for nearest neighbor search.

Proposed System:

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 hash tables and inverted indexes to perform a localized search. 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. ProMiSH is faster than state-of-the-art tree-based techniques, with multiple orders of magnitude performance improvement.

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

Advantages:

1. The performance of ProMiSH on both real and synthetic datasets.
2. We develop efficient search algorithms that work with the multi-scale indexes for fast query processing.

System Architecture

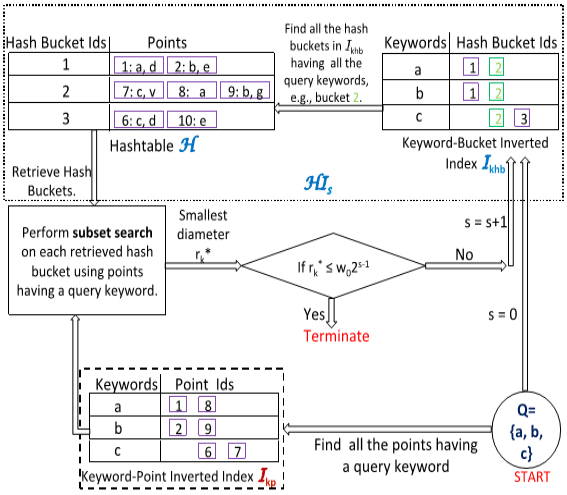


Fig. Index structure and flow of execution of ProMiSH

IMPLEMENTATION

Modules Description:

- Multi-dimensional data
- Nearest Keyword
- Indexing
- Hashing

Multi-dimensional Data:

Keyword-based search in text-rich multi-dimensional datasets facilitates many novel applications and tools. 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. these algorithms may take hours to terminate for a multi-dimensional dataset of millions of points. Therefore, there is a need for an efficient algorithm that scales with dataset dimension, and yields practical query efficiency on large datasets. 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.

Nearest Keyword:

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. Location-specific keyword queries on the web and in the GIS systems were earlier answered using a combination of R-Tree and inverted index. 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.

Indexing:

Indexing time as the metrics to evaluate the index size for ProMiSH-E and ProMiSH-A. Indexing time indicates the amount of time used to build ProMiSH variants. the memory usage and indexing time of ProMiSH-E and ProMiSH-A under different input real data. Memory usage grows slowly in both ProMiSH-E and ProMiSH-A when the number of dimensions in data points increases. ProMiSH-A is more efficient than ProMiSH-E in terms of memory usage and indexing time: it takes 80% less memory and 90% less time, and is able to obtain near-optimal results.

Hashing:

The hashing technique is inspired by Locality Sensitive Hashing (LSH), 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. Random projection with hashing has come to be the state-of-the-art method for nearest neighbor search in high-dimensional datasets.

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

ALGORITHMS:

ProMiSH:

results on real and synthetic datasets show that ProMiSH has up to 60 times of speedup over state-of-the-art tree-based techniques. ProMiSH(Projection and Multi Scale Hashing) that uses random projection and hash-based index structures, and achieves high scalability and speedup. 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).

Algorithm 1. ProMiSH-E

In: Q: query keywords; k: number of top results
In: w_0: initial bin-width
$PQ \leftarrow [e([\,], +\infty)]$: priority queue of top-$k$ results
HC: hashtable to check duplicate candidates
BS : bitset to track points having a query keyword
for all $o \in \cup_{\forall v_Q \in Q} \mathcal{I}_{kp}[v_Q]$ **do**
  $BS[o] \leftarrow$ true /* Find points having query keywords*/
end for
for all $s \in \{0, \ldots, L-1\}$ **do**
  Get $\mathcal{HI}$ at s
  $E[\,] \leftarrow 0$ /* List of hash buckets */
  for all $v_Q \in Q$ **do**
    for all $bId \in \mathcal{I}_{khb}[v_Q]$ **do**
      $E[bId] \leftarrow E[bId] + 1$
    end for
  end for
  for all $i \in (0, \ldots, SizeOf(E))$ **do**
    if $E[i] = SizeOf(Q)$ **then**
      $F' \leftarrow \emptyset$ /* Obtain a subset of points */
      for all $o \in \mathcal{H}[i]$ **do**
        if $BS[o] =$ true **then**
          $F' \leftarrow F' \cup o$
        end if
      end for
      if checkDuplicateCand(F', HC) = false **then**
        searchInSubset(F', PQ)
      end if
    end if
  end for
  /* Check termination condition */
  if $PQ[k].r \leq w_0 2^{s-1}$ **then**
    Return PQ
  end if
end for
/* Perform search on $\mathcal{D}$ if algorithm has not terminated */
for all $o \in \mathcal{D}$ **do**
  if $BS[o] =$ true **then**
    $F' \leftarrow F' \cup o$
  end if
end for
searchInSubset(F', PQ)
Return PQ

Euclidean Distance:

The Euclidean distance or Euclidean metric is the "ordinary" (i.e. straight-line) distance between two points in Euclidean space. With this distance, Euclidean space becomes a metric space. The associated norm is called the Euclidean norm. Older literature refers to the metric as Pythagorean metric.

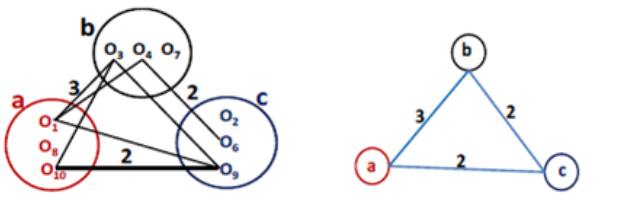
Since Euclidean space with dot product is an inner product space, we have

$$||O1z-O2z||2=|(O1-O2)z|$$
$$< ||z||2.\ ||O1-O2||2$$
$$=||O1-O2||2, \text{ since } ||z||2=1$$

$$E(x,y) = \sqrt{\sum_{i=0}^{n}(x_i - y_i)^2}$$

Algorithm 2. CheckDuplicateCand

In: F': a subset; HC: hashtable of subsets
$F' \leftarrow sort(F')$
$pr1$: list of prime numbers; $pr2$: list of prime numbers;
for all $o \in F'$ **do**
  $pr_1 \leftarrow$ randomSelect($pr1$); $pr_2 \leftarrow$ randomSelect($pr2$)
  $h_1 \leftarrow h_1 + (o \times pr_1)$; $h_2 \leftarrow h_2 + (o \times pr_2)$
end for
$h \leftarrow h_1 h_2$;
if isEmpty($HC[h]$)=false **then**
  if elementWiseMatch(F', $HC[h]$) = true **then**
    Return true;
  end if
end if
$HC[h]$.add(F');
Return false;



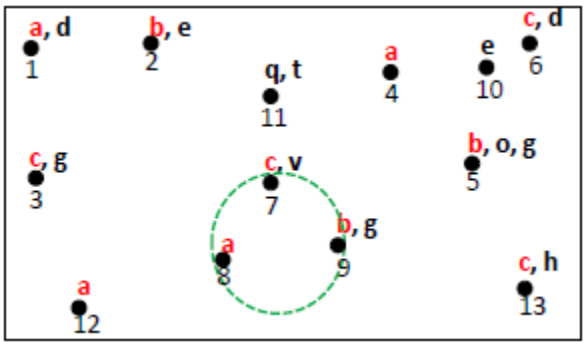
(a) Pairwise inner joins (b) A graph representation

PRUNING ALGORITHM:

Pruning is a technique in machine learning that reduces the size of decision trees by removing sections of the tree that provide little power to classify

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

instances. Pruning reduces the complexity of the final classifier, and hence improves predictive accuracy by the reduction of overfitting.



CONCLUSIONS AND FUTURE WORK

In this paper, 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 developed ProMiSH-E that finds an optimal subset of points and ProMiSH-A that searches near-optimal results with better efficiency. Our empirical results show that ProMiSH is faster than state-of-the-art tree-based techniques, with multiple orders of magnitude performance improvement. More- over, 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. 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 keywords.

Disk extension: We plan to explore the extension of ProM- iSH to disk. ProMiSH-E sequentially reads only required buckets from I_{kp} to find points containing at least one query keyword. Therefore, I_{kp} can be stored on disk using a directory file structure. We can create a directory for I_{kp}. Each bucket of I_{kp} will be stored in a separate file named after its key in the directory. Moreover, ProMiSH-E sequentially probes HI 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 hash table and the inverted index of a HI structure. Therefore, all the hash tables and the inverted indexes of HI can again be stored using a similar directory file structure as Ikp, and all the points in the dataset can be indexed into a B+ Tree using their ids and stored on the disk. In this way, subset search can retrieve the points from the disk using B+ Tree for exploring the final set of results.

References

[1] Vishwakarma Singh, Bo Zong, and Ambuj K. Singh, "Nearest Keyword Set Search in Multi-Dimensional Datasets", IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 28, NO. 3, MARCH 2016.

[2] 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.

[3] D. Zhang, B. C. Ooi, and A. K. H. Tung, "Locating mapped resour- ces in web 2.0," in Proc. IEEE 26th Int. Conf. Data Eng., 2010, pp. 521–532.

[4] 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.

[5] 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.

[6] J. Bourgain, "On lipschitz embedding of finite metric spaces in hil- bert space," Israel J. Math., vol. 52, pp. 46–52, 1985.

[7] 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.

[8] 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.

[9] 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.

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

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

[12] 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.

[13] R. Hariharan, B. Hore, C. Li, and S. Mehrotra, "Processing spatial- keyword (SK) queries in geographic information retrieval (GIR) systems," in Proc. 19th Int. Conf. Sci. Statistical Database Manage., 2007, p. 16.

[14] S. Vaid, C. B. Jones, H. Joho, and M. Sanderson, "Spatio-textual indexing for geographical search on the web," in Proc. 9th Int. Conf. Adv. Spatial Temporal Databases, 2005, pp. 218–235.

[15] 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.