International Journal of Research
Available at https://edupediapublications.org/journals

e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 12
April 2018

# Projection and Multi-Scale Hashing approach for Engineered Datasets

T. Swathi
Assistant professor Department of Computer Science and Engineering
**Email:tswathi2020@gmail.com**
Jagruti Institute of Engineering and Technology,

***Abstract***:

*Catchphrase based hunt in content rich multi-dimensional datasets encourages numerous novel applications and devices. In this paper, we consider objects that are labeled with watchwords and are inserted in a vector space. For these datasets, we examine inquiries that request the most secure gatherings of focuses fulfilling a given arrangement of watchwords. We propose a novel strategy called ProMiSH (Projection and Multi Scale Hashing) that utilizations irregular projection and hash-based file structures, and accomplishes high adaptability and speedup. We show a correct and an inexact variant of the calculation. Our test comes about on genuine and engineered datasets demonstrate that ProMiSH has up to 60 times of speedup over cutting edge tree-based strategies.*

***Keywords***

*ProMiSH, Vector Space, Keyword, Multi-Dimension, Datasets...*

## 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 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 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 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.

We propose a novel method called ProMiSH (Projection and Multi Scale Hashing) that uses random projection and hash-based index structures, and achieves high scalability and speedup.

We present an exact and an approximate version of the algorithm.

Our experimental 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 (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.

The proposed techniques use location information as an integral part to perform a best-first search on the IR-Tree, and query coordinates play a fundamental role in almost every step of the algorithms to prune the search space. 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 random 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.

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.

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.

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.

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.

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.

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 IR2-Tree (Information Retrieval R-Tree) which combines an R-Tree with superimposed text signatures.

We present algorithms that construct and maintain an IR2-Tree and use it to answer top-k spatial keyword queries.

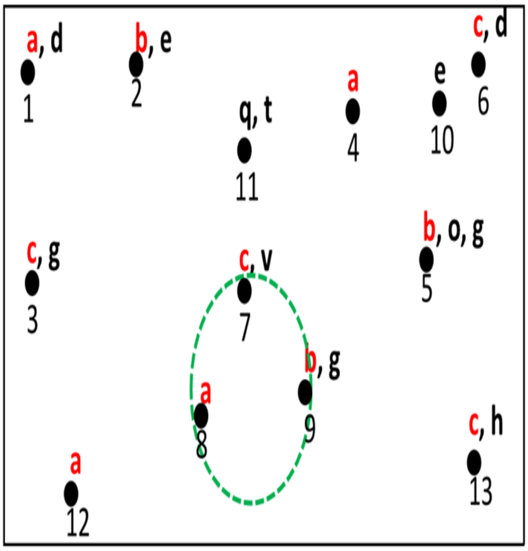
Our algorithms are experimentally compared to current methods and are shown to have superior performance and excellent scalability.

For example, consider a real estate agency office that holds a database with available flats for lease.

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.

## II. Design and Implementation.

### *System Architecture*



***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.

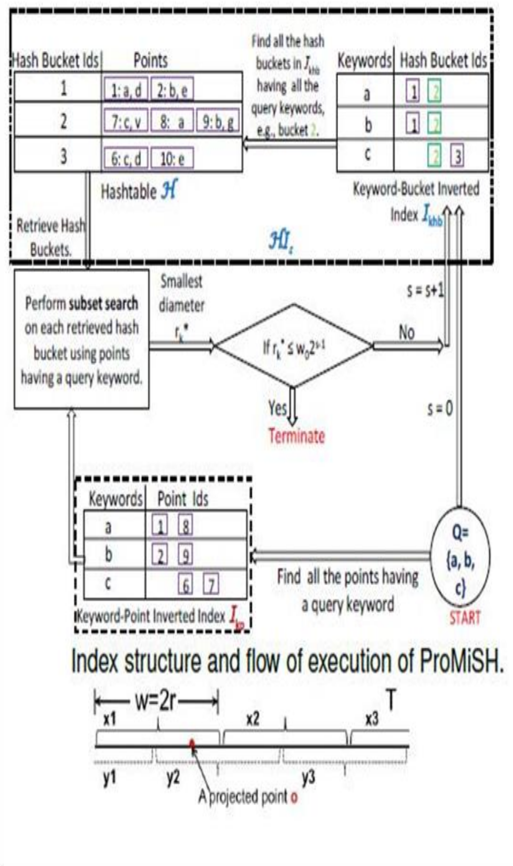
Cong et al. integrated R-tree and inverted file to answer a query similar to Felipe et al. using a different ranking function.

***Disadvantages 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.

***Proposed System:***

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.


Hash Bucket Ids
Points
1
1:a, d
2:b, e
2
7:c, v
8: a
9:b, g
3
6:c, d
10: e
Hashtable $\mathcal{H}$
Find all the hash buckets in $I_{khb}$ having all the query keywords, e.g., bucket 2.
Keywords
Hash Bucket Ids
a
1 2
b
1 2
c
2 3
Keyword-Bucket Inverted Index $I_{khb}$
Retrieve Hash Buckets.
$\mathcal{HI}_s$
Perform subset search on each retrieved hash bucket using points having a query keyword.
Smallest diameter $r_k^*$
If $r_k^* \leq w_0 2^{s+1}$
No
s = s+1
Yes
Terminate
s = 0
Keywords
Point Ids
a
1 8
b
2 9
c
6 7
Keyword-Point Inverted Index $I_{kp}$
Find all the points having a query keyword
Q= {a, b, c}
START
Index structure and flow of execution of ProMiSH.
w=2r
x1
x2
x3
T
y1
y2
y3
A projected point o

Index Structure and Flow of Execution of ProMiSH

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.

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.

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

## III. Results

a) Home Page

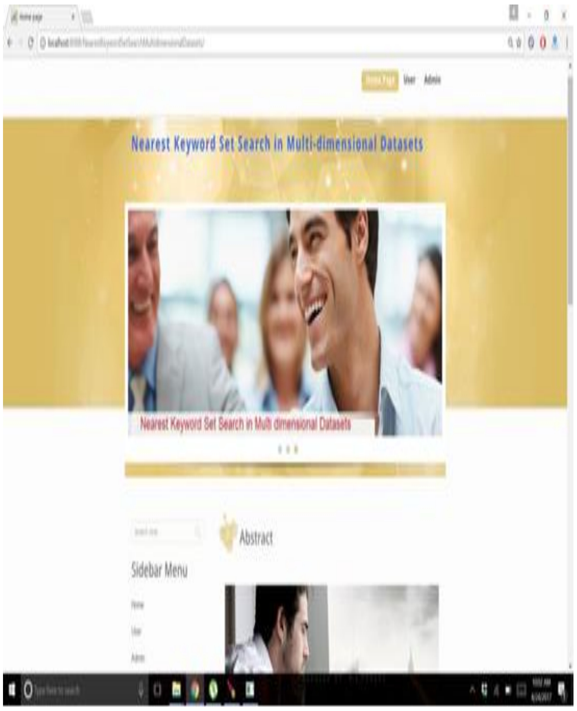

Home page
localhost:8080/NearestKeywordSetSearchMultidimensionalDatasets/
Home Page
User
Admin
Nearest Keyword Set Search in Multi-dimensional Datasets
Nearest Keyword Set Search in Multi dimensional Datasets
Abstract
Sidebar Menu
Home
User
Admin

Fig 3: Home page

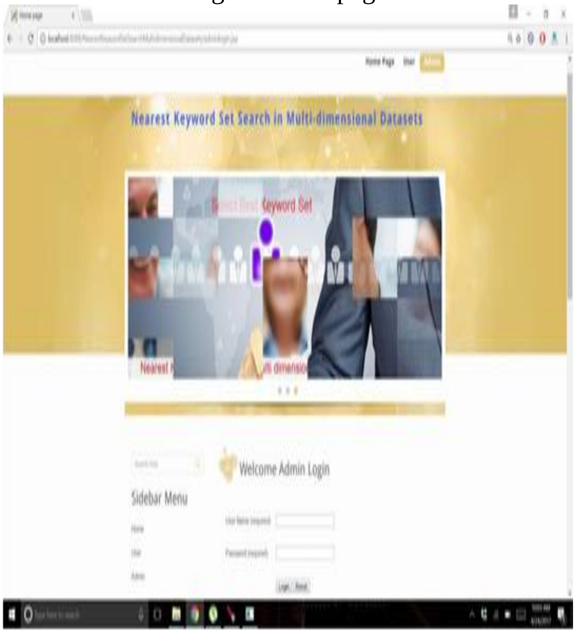
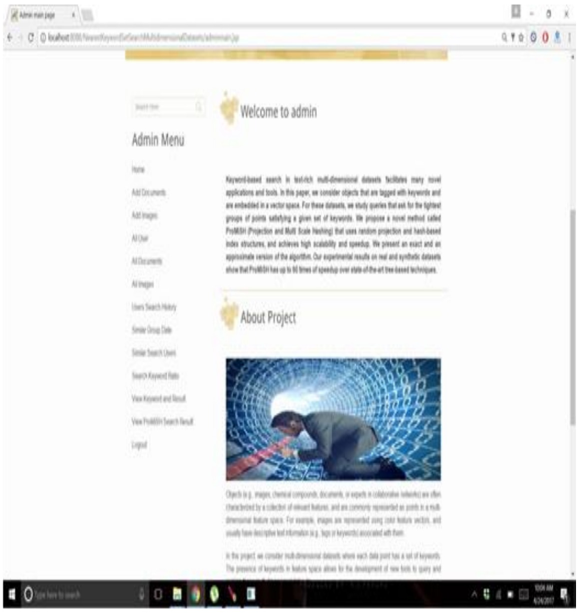


Fig: 4 Admin login

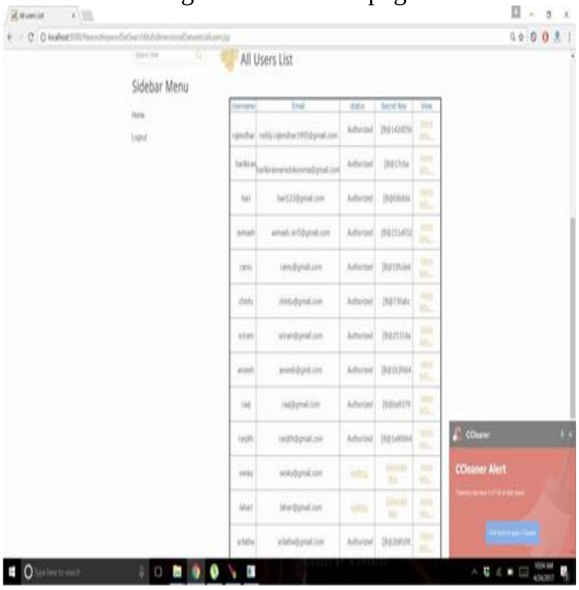


Keyword-based search in text-rich multi-dimensional datasets facilitates many novel applications and tools. In this paper, we consider objects that are tagged with keywords and are embedded in a vector space. For these datasets, we study queries that ask for the tightest groups of points satisfying a given set of keywords. We propose a novel method called ProMiSH (Projection and Multi Scale Hashing) that uses random projection and hash-based index structures, and achieves high scalability and speedup. We present an exact and an approximate version of the algorithm. Our experimental results on real and synthetic datasets show that ProMiSH has up to 60 times of speedup over state-of-the-art tree-based techniques.

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 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.

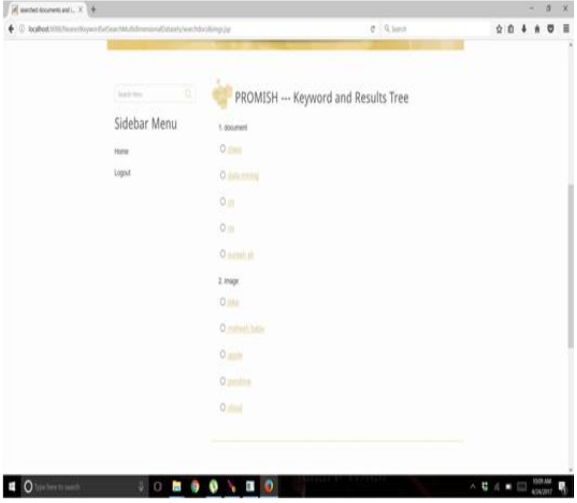
In this project we consider multi-dimensional datasets where each data point has a set of keywords. The presence of keywords in feature space allows for the development of new tools to query and

Fig: 5 Admin homepage


All users List
localhost:8080/NearestKeywordSetSearchMultidimensionalDatasets/allusers.jsp
All Users List
Sidebar Menu
Home
Logout
CCleaner
CCleaner Alert

| Username | Email | status | Secret Key | View |
|---|---|---|---|---|
| rajendhar | reddy.rajendhar1995@gmail.com | Authorized | [B@143d056 | Click Info... |
| harikiran | harikiranreddykommu@gmail.com | Authorized | [B@17b5a | Click Info... |
| hari | hari123@gmail.com | Authorized | [B@6b6dda | Click Info... |
| avinash | avinash.sri5@gmail.com | Authorized | [B@151a932 | Click Info... |
| ramu | ramu@gmail.com | Authorized | [B@10fcde6 | Click Info... |
| chintu | chintu@gmail.com | Authorized | [B@730abc | Click Info... |
| sriram | sriram@gmail.com | Authorized | [B@25314a | Click Info... |
| anvesh | anvesh@gmail.com | Authorized | [B@1539564 | Click Info... |
| raaj | raaj@gmail.com | Authorized | [B@6ba6379 | Click Info... |
| ranjith | ranjith@gmail.com | Authorized | [B@1e90064 | Click Info... |
| venky | venky@gmail.com | waiting | Generate Key | Click Info... |
| bhari | bhari@gmail.com | waiting | Generate Key | Click Info... |
| srilatha | srilatha@gmail.com | Authorized | [B@26f659 | Click Info... |

Fig: 6 All users List


searched documents and i...
localhost:8080/NearestKeywordSetSearchMultidimensionalDatasets/searchdocsImgs.jsp
Search

PROMISH --- Keyword and Results Tree

Sidebar Menu

Home

Logout

1. document

○ java

○ data mining

○ ai

○ ai

○ suresh ai

2. image

○ bike

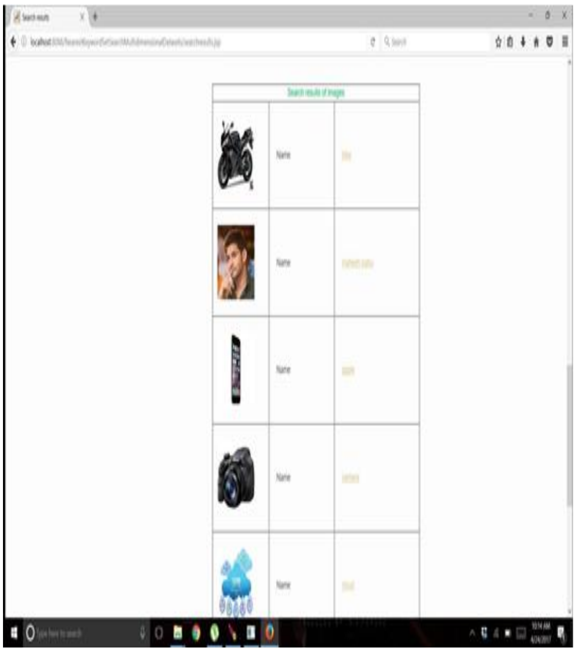
○ mahesh babu

○ apple

○ prabhas

○ cloud

Fig: 7 Promish Result



Search result of images

| | | |
|---|---|---|
| | Name | bike |
| | Name | mahesh babu |
| | Name | apple |
| | Name | camera |
| | Name | cloud |

Fig: 8 Search result of images

## IV. 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 random 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. Moreover, our techniques scale well with both real and synthetic datasets.

## V. Future Enhancement

- 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 ProMiSH to disk. ProMiSH-E sequentially reads only required buckets from Ikp to find points containing at least one query keyword.

- Therefore, Ikp can be stored on disk using a directory-file structure. We can create a directory for Ikp.

- Each bucket of Ikp 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.

## VI. References

1. The unified modelling language user guide SECOND EDITION,grady Booch,James Rumbuagh,Ivor Jacobson

2. The complete reference java seventh edition ,Herbert Schildt

3. R. Agrawal and R. Srikant, “Fast algorithms for mining association rules in large databases,” in VLDB, 1994, pp. 487–499.

4. 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.