ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 7, July 2017

Improving the Keyword Searching Performance for Multi-Dimensional Dataset through an Efficient Random Projection and Hashing Method

N.Naveen Kumar[1], Yasmeen Anjum[2]

Assistant Professor, Department of CSE, School of Information Technology, JNTUH, Kukatpally, District Ranga Reddy, Telangana, India[1]

M.Tech Student, Department of CSE, School of Information Technology, JNTUH, Kukatpally, District Ranga Reddy, Telangana, India[2]

ABSTRACT: In this paper we take a look at nearest keyword set Queries on textual content high multidimensional dataset. Although existing techniques using tree-based indexes advise viable solutions to NKS queries on multidimensional datasets, the overall performance of those algorithms deteriorates sharply with the growth of length or dimensionality in datasets. In this paper, we suggest ProMiSH to permit fast processing for NKS queries. In particular, we expand an authentic ProMiSH (known as ProMiSH-E) that usually retrieves the best top-k consequences, and an approximate ProMiSH (called ProMiSH-A) that is more efficient in terms of time and area, and is able to acquire close to-ideal outcomes in exercise.

KEYWORDS: Querying, Multi-dimensional Data, Indexing, Hashing.

I. INTRODUCTION

Data mining is that the process of determine patterns in massive information sets involving strategies at the intersection of artificial intelligence, machine learning, statistics, and information systems. It is a knowledge base subfield of technology. Today, the ever-developing volume and value of digital information have raised a crucial and mounting call for lengthy-time period information protection through huge-scale and high-performance backup and archiving systems. The overall goal of data mining method is to extract information from an information set and transform it into an evident structure for additional use. Other than the raw analysis step, it involves information and knowledge management aspects, knowledge pre-processing, model and logical thinking concerns, interestingness metrics, quality concerns, post-processing of discovered structures, visualization, and on-line change. Data processing is that the analysis step of the "knowledge discovery in databases" process, or KDD. In today's digital world the number of knowledge that is developed is increasing day by day. There is different transmission within which information is saved. It's terribly difficult to search the massive dataset for a given query similarly to archive additional accuracy on user question. Within the same time query can search on dataset for actual keyword match and it will not realize the closest keyword for accuracy For instance: Flickr. The number of knowledge that is developed is increasing day by day; therefore it is terribly difficult to search massive dataset for a given query similarly to realize additional accuracy on user query. Thus we have implemented a way of efficient search in multidimensional dataset. This can be related to pictures as an input. Pictures are usually characterized by a set of relevant features, and are commonly delineate as points during a multi-dimensional feature house. For example, pictures are represented using color feature vectors, and

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 7, July 2017

frequently have descriptive text data for instance tags or keywords related to them. We tend to contemplate multi-dimensional datasets wherever every data point features a set of keywords. The presence of keywords in feature space permits for the event of latest tools to question and explore these multidimensional datasets. Our main contributions are summarized as follows. We tend to propose a unique multi-scale index for actual and approximate NKS query process. We tend to develop efficient search algorithms that employment with the multi-scale indexes for quick query process. We tend to conduct intensive experimental studies to demonstrate the performance of the projected techniques. Professional poses Pro-MISH (short for Projection and Multi-scale Hashing) to alter quick process for NKS Queries. Specially, developed a definite Pro-MiSH (refer to as Pro-MiSH-E) that continually retrieves the best top-k results and an approximate Pro-MiSH (referred to as Pro-MiSH-A) that is additional efficient in term of time and space, and is in a position to obtained near-optimal leads to follow. Pro-MiSH-E uses a collection of hash tables and inverted indexes to perform a localized search. The hashing technique is impressed by Sensitive Hashing (LSH), which is progressive technique for nearest neighbor search in high-dimensional spaces. Unlike LHS-based technique that enable only approximate search with probabilistic guarantees, the index structure in ProMiSH-E supports correct search. ProMiSH-E creates hash tables at multiple bin-widths, known as index levels. one spherical of search in a very Hash table yield set of points that contain query results, and Pro-MiSH-E explores every set using a quick pruning based algorithm Pro-MiSH-A is an approximate variation of Pro-MiSH-E for better time and space efficiency. Using this algorithm evaluates the performance of Pro-MiSH on each real and synthetic datasets and use state-of-art Vb-R*-Tree and Co-SKQ as baselines. The empirical results revel that Pro-MiSH systematically outperforms the baseline algorithms and Pro-MiSH-A in contrast to LSH-based strategies that enable only approximate search with probabilistic guarantees, the index structure in Pro-MiSH-E supports correct search. Pro-MiSH-E creates hash tables at multiple bin- widths, referred to as index levels. A single round of search in a very hash table yields subsets of points that contain question results, and Pro-MiSH-E explores every set using a fast pruning-based algorithm.

II. RELATED WORK

Felipe et al. present an economical technique to answer top-k special keyword queries. To do so, we have a tendency to introduce a classification structure known as IR^2-Tree (Information Retrieval R-Tree) which combines an R-Tree with superimposed text signatures. Author presents algorithms that construct and maintain an IR^2-Tree and use it to answer top-k spatial keyword queries. Its show superior performance and excellent scalability. Top-k spatial keyword queries which is based on tight integration of data structure and algorithm used in special database search and information retrieval R-Tree (IR^2-Tree) which is structure based on the R-Tree at query time and incremental algorithm is employed that uses IR^2-Tree which is structure based on the R-Tree at query time and incremental algorithms. Is employed that uses IR^2-Tree which is structure based on the R-Tree at query time and incremental algorithm; other related queries include aggregate nearest keyword search in spatial database, in this query retrieves k objects from Q with minimum sum of distances to its nearest point in D such that each nearest point matches at least one query keyword for processing this query several algorithm proposed using IR2-Tree as index structure. Another song of associated works cope with m-closest key-word queries; bR*-Tree is evolved based totally on R*-tree that stores bitmaps and minimum bounding rectangles (MBRs) of key phrases in each node at the side of factors MBRs. BR-Tree additionally suffers from a high storage cost; consequently Zang et al Modified bR*-Tree to create virtual bR*-tree in reminiscence at run time. Virtual bR*-tree is constituted of a pre-saved r*- Tree, which indexes all the factors, and an inverted index which saved key-word facts and direction from the root node in R*-Tree for each factor. Both bR*-Tree and digital bR*-Tree stocks similar overall performance weaknesses as bR*- Tree. Tree-based indexes, such as M-tree, are proposed to organize and search large dataset from generic. M-Tree always balanced several heuristic split alternatives are considered and experimentally evaluated. This M-Tree has been extensively investigated for nearest neighbor search in high dimensional spaces. This index fails to scale to dimensions greater than 10 because of the curse of dimensionality.
Jon M. Kleinberg Develop new approach to the nearest-neighbor problem, based on a technique for combining randomly chosen one dimensional projections of the underlying purpose set. Two algorithms are introduce during this 1st for locating epsilon approximate nearest neighbors and second epsilon approximate nearest-neighbor algorithmic program with near-linear storage and question time improves asymptotically linear search in all dimensions. Aristides

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 7, July 2017

Gionis examine a completely unique scheme for approximate similarity search supported hashing. The basic plan is to hash the points from the info therefore on ensure that the likelihood of collision is far higher for objects that are near one another than for people who are way apart. The method provides important improvement in running time over different strategies for searching in high dimensional spaces supported hierarchical tree de-composition. This scheme scales well even for comparatively sizable amount of dimensions (more than 50). previous technique solve this problem with efficiency just for the approximate case correct and economical close to neighbor Search in High Dimensional Spaces during this are style to resolve r-near neighbor queries for a hard and fast query vary or for set of query ranges with probabilistic guarantees .and then extend for nearest neighbor queries.

III. FRAME WORK

A. Overview of Proposed System

In our projected system the important data set is collected from exposure sharing websites. During which we tend to collect images from descriptive tags from Flickr and therefore the images are transformed into grayscale and associate every datum, with a collection of keyword that are derived from tags. we are able to collect variety of datasets, suppose we tend to collect five datasets (R1, R2, R3, R4, R5) with up to million data points, we are able to produce multiple dataset to analyze performance. The query co-ordinates play a basic role in each step of formula to prune search house. Our work deals with providing keyword as an input. . We tend to propose a unique multi scale index for precise and approximate NKS query processing. We tend to develop economical search algorithms that job with the multi-scale indexes for quick query processing. Distance browsing is simple with R-trees. In fact, the best-first algorithm is precisely designed to output information points in ascending order of their distances.

USER Module:

User provides the input keyword as a picture.

SYSTEM Module:

the system module retrieves all pictures from the database, so it analyzes keywords the positive purpose relation is undertaken by the system.

It analyzes image keyword relation between points. It filters the image supported the relations, Applying nearest neighbor technique retrieved pictures, Displays nearest image as an output. We tend to start with the index for precise search. There are a pair of main element enclosed i.e. Inverted index ikp and Hash-table inverted index pairs (HI). We tend to treat keyword as keys and provide it as an input to our system. There are hash bucket IDs and several points related to the keywords, it will realize all the hash buckets in Ikhb (keyword bucket inverted index), having all query keywords. In our system we tend to are playing set search on every retrieved hash bucket mistreatment points having query keywords. These indexes fail to scale dimension larger than 10 as a result of its dimensionality therefore random projection with hashing and categorization has come back up within the technique of nearest keyword search in multidimensional datasets. for instance consider there are three keywords a, b, c. we will be looking out the points related to the hash bucket IDS i.e. there'll be hunt for all the keywords, if there is no precise match for the keyword, then it will explore for two keywords i.e. the multiple combination of the keywords, so for the one keyword. Therefore all the keywords are searched with efficiency with less time and additional accuracy in multidimensional datasets, and that we projected resolution re-implementing multiple rounds within the top k nearest set in multidimensional datasets. By exploitation this approach the subsequent benefits are Distance browsing is straightforward with R-trees.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 7, July 2017

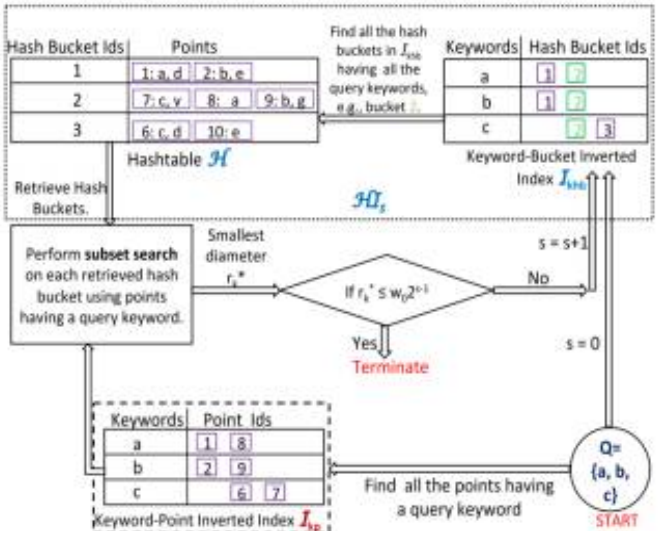


Fig1: Architecture of Proposed System

In fact, the best-first algorithm is precisely designed to output information points in ascending order of their distances. It is easy to increase our compression scheme- to any dimensional space. By exploitation this approach the subsequent drawbacks are failing to produce real time answers on tough inputs. The important nearest neighbor lies quite distant from the query point, whereas all the closer neighbors are missing a minimum of one amongst the question keywords.

B. Implementation of Proposed System

Search Algorithm

ProMiSH referred to as ProMiSH-A. We start with the algorithm description of ProMiSH-A, after which examine its approximation satisfactory. ProMiSH-E fairly relies upon on a immature search set of rules that unearths pinnacle-okay results from a subset of data points.

HI Construction

It includes multiple hash tables and inverted indexes referred to as HI. HI is managed by way of 3 parameters:

- **Index degree (L)**

HI at all the index stage then it performs a seek inside the entire dataset D.

- **Number of random unit vectors (m)**

We don't forget its projection area as a section [0, pMax] and partition the segment into 2(L-s+1) +1 overlapping boxes, where each bin has width and is equally overlapped with two other boxes.

- **Hash desk length (B)**

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

IV. EXPERIMENTAL RESULTS

In our experiments, any user can upload the flicker dataset into the system after the successfully uploading the data set into the system generate the inverted index and hash table of the loaded dataset after the apply the ProMiSH-E method after applying this technique enter the query based on that query result will be generate and similarity score also be generate the generating queries is known as Nearest Keyword Set (NKS) after that apply the another technique like ProMiSH-A algorithm in that we have to enter the query based on that query result will be generate after that enter the top-k value means how many records will be retrieved and after that we can find the Nearest Keyword Set based on the

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

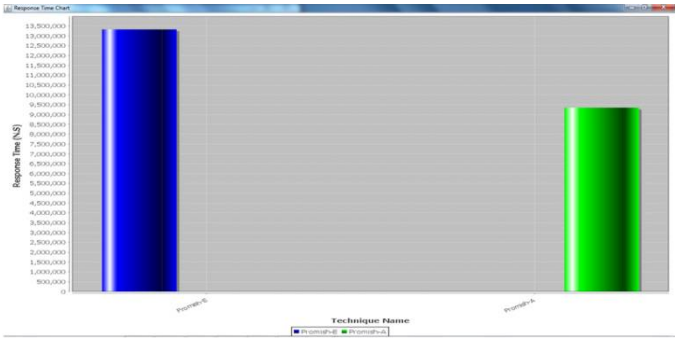
International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

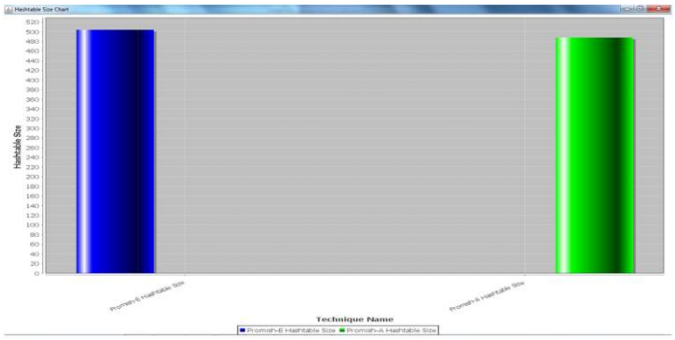
Website: www.ijirset.com

Vol. 6, Issue 7, July 2017

two schemes charts will be generate. In the below charts we can observe that first chart difference between the length of both ProMiSH-E and ProMiSH-A and second chart is difference between the length of both ProMiSH-E Hash table size and ProMiSH-A Hash table Size.



We can observe two charts in that first chart is describes that ProMiSH-E length is higher than ProMiSH-A length. The difference will be shown in the sense of Response time in milliseconds (M.S) and second chart is describes that ProMiSH-E Hash table length is higher than ProMiSH-A Hash table length. The difference will be shown in the sense of Hash table size. So we can consider that the advantage of the two schemes.



V. CONCLUSION

Finally, in this paper we proposed an efficient Random Projection and Hashing Method named as ProMiSH. From our implementation and experimental results we can say that the user is upload the data set into the system after successful uploading the data set into the system we applying the two algorithm by using this two algorithms we can the search Nearest Keyword Set with low cost and effective manner when compare to current techniques.

REFERENCES

[1] 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.
[2] A. Guttman, "R-trees: A dynamic index structure for spatial searching," inProc. ACM SIGMOD Int. Conf. Manage. Data, 1984, pp. 47–57.
[3] I. De Felipe, V. Hristidis, and N. Rishe, "Keyword search on spatial databases," in Proc. IEEE 24th Int. Conf. Data Eng., 2008, pp. 656–665.
[4] B. Martins, M. J. Silva, and L. Andrade, "Indexing and ranking in Geo-IR systems," in Proc. Workshop Geographic Inf., 2005, pp. 31–34.
[5] Z. Li, H. Xu, Y. Lu, and A. Qian, "Aggregate nearest keyword search in spatial databases," in Proc. 12th Int. Asia-Pacific Web Conf., 2010.
[6] M. L. Yiu, X. Dai, N. Mamoulis, and M. Vaitis, "Top-k spatial preference queries," in Proc. IEEE 23rd Int. Conf. Data Eng., 2007, pp. 1076–1085.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 7, July 2017

[7] T. Xia, D. Zhang, E. Kanoulas, and Y. Du, "On computing top-t most influential spatial sites," in Proc. 31st Int. Conf. Very Large Databases, 2005, pp. 946–957.

[8] Y. Du, D. Zhang, and T. Xia, "The optimal-location query," in Proc. 9th Int. Conf. Adv. Spatial Temporal Databases, 2005, pp. 163–180.

[9] D. Zhang, Y. Du, T. Xia, and Y. Tao, "Progressive computation of the min-dist optimal-location query," in Proc. 32nd Int. Conf. Very Large Databases, 2006, pp. 643–654.