Volume 1, Issue 8, November 2016 ISSN: 2456-0006

International Journal of Science Technology Management and Research

Available online at: www.ijstmr.com

An Efficient Way to Search Multiple Objects in Multidimensional Dataset

Namrata S. Shinde[1]
Department of Computer Engg.
Sir Vishweswaraya Institute of Technology
Nashik, India

Manisha M. Naoghare[2]
Department of Computer *Engg.*
Sir Vishweswaraya Institute of Technology
Nashik, India

Abstract: ***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.***

Keywords: Projection, Hashing, Multiscale-index, clustering, NKS, ProMiSH etc.

I. INTRODUCTION

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.

In today's digital world the amount of data which is developed is increasing day by day. There are 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 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. Moreover, these techniques do not provide concrete guidelines on how to enable efficient processing for the type of queries where query coordinates are missing .Second, 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.

II. METHODOLOGY OF PROPOSED SYSTEM

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. Moreover, these techniques do not provide concrete guidelines on how to enable efficient processing for the type of queries where query coordinates are missing .Second, 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.

A. Datasets

1. Data collection: The dataset used in this work was collected from Twitter by a snowball sampling based crawler. We first manually selected a set of highly followed Twitter users in Singapore. They include the accounts of local sport and entertainment celebrities, political parties, politicians, mass media and bloggers, etc.. We expanded this set of users by adding more Singapore based users1 that are at most two hops away from some user in the original set. Using Twitter Stream APIs2, we then obtained all tweets and retweets by the users in the set. In this work, we use all tweets in October 2014 to simulate a live tweet stream. This set includes 35,491,260 tweets and retweets posted by 525,632 users.

2. Item adoption and propagation: We again use as an item. We consider a user u adopts a hash tag when u posts an original tweet containing the hash tag. Also, if user v retweets original tweets from u that contains a hash tag h, u is said to propagate h to v. We filtered away hash tags shorter than 2 characters excluding the # symbol. These short hash tags do not have clear semantics and are often the prefix of other truncated hash tags due to 140 characters length constraint. We also excluded hash tags longer than 20 characters as such hash tags are unpopular.

III. SYSTEM ARCHITECTURE

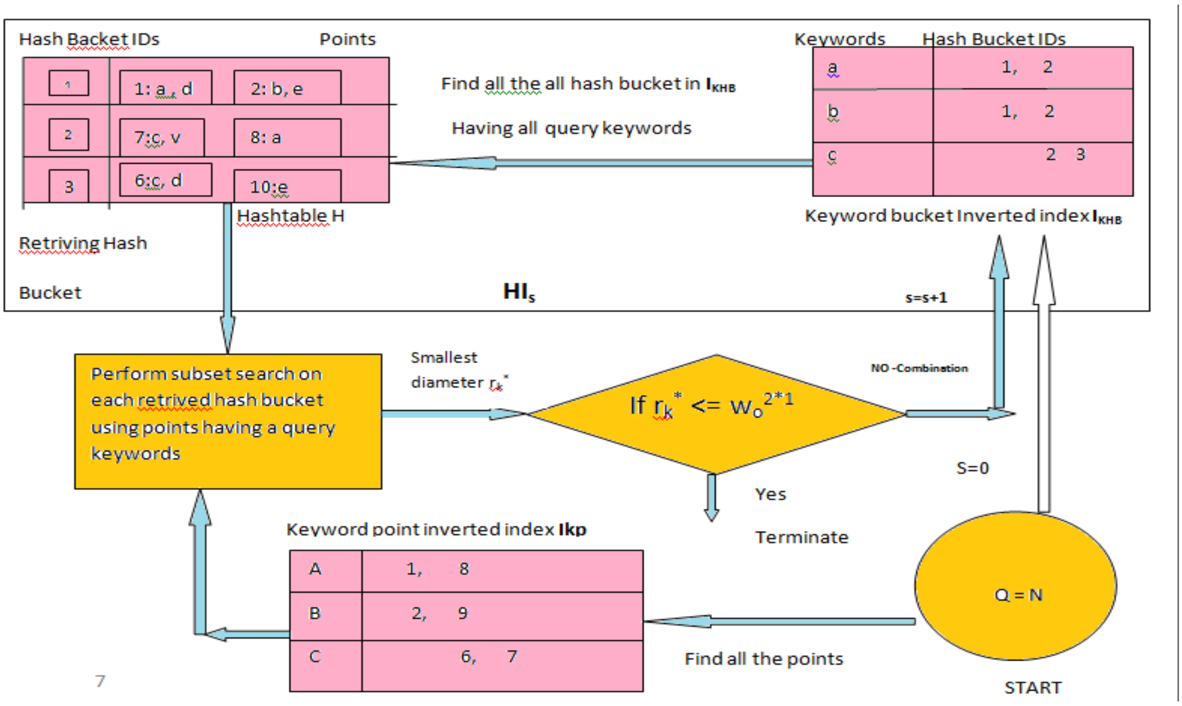

Fig. 1 System Architecture

In today's digital world the amount of data which is developed is increasing day by day. There are 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**. It** uses keyword density and exact matching for keyword parsing technique on given user query.

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.

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.

IV. RELATEDWORK

A variety of related queries have been studied in literature on text-rich spatial datasets. Location-specific keyword queries on the web and in the GIS systems [11], [12], [13], [14] were earlier answered using a combination of R-Tree and inverted index. Felipe et al. developed IR 2-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. Martins et al. computed text relevancy and location proximity independently, and then combined the two ranking scores. Cao et al. [7] and Long et al. [8] proposed algorithms to retrieve a group of spatial web objects such that the group's keywords cover the query's keywords and the objects in the group are nearest to the query location and have the lowest inter-object distances. Other related queries include aggregate nearest keyword search in spatial databases, top-k preferential query, top-ksites in a spatial data based on their influence on feature points, and optimal location queries.

CONCLUSION

In this paper, we proposed solutions to the problem of top-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.

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.

ACKNOWLEDGMENT

I take this opportunity to express my sincere thanks and deep sense of gratitude to Prof. M.M. Naoghare for valuable guidance. I am also thankful to all faculties of computer department, my colleagues and family who encouraged me to do such a research**.**

REFERENCE

[1] Vishwakarma Singh, Bo Zong, and Ambuj K. Singh "Nearest Keyword Set Search inMulti-Dimensional Datasets", in IEEE transactions on knowledge and data engineering, vol. 28, no. 3, march 2016.

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

[3] 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.

[4] V. Singh, A. Bhattacharya, and A. K. Singh, "Querying spatial patterns," inProc. 13th Int. Conf. Extending Database Technol.: Adv. Database Technol., 2010, pp. 418–429

[5] X. Cao, G. Cong, C. S. Jensen, and B. C. Ooi, "Collective spatial keyword querying," inProc. ACM SIGMOD Int. Conf. Manage. Data, 2011, pp. 373–384.

[6] T. Ibaraki and T. Kameda, “On the optimal nesting order for computing N-relational joins,”ACM Trans. Database Syst., vol. 9, pp. 482–502, 1984

[7] D. Comer, “The ubiquitous b-tree,”ACM Comput. Surveys, vol. 11, no. 2, pp. 121–137, 1979.

[8] H.-H. Park, G.-H. Cha, and C.-W. Chung, “Multi-way spatial joinsusing r-trees: Methodology and performance evaluation,” inProc. 6th Int. Symp. Adv. Spatial Databases, 1999, pp. 229–250.

[9] V. Singh and A. K. Singh, “SIMP: Accurate and efficient near neighbor search in high dimensional spaces,” inProc. 15th Int. Conf. Extending Database Technol., 2012, pp. 492–503.

[10] A. Gionis, P. Indyk, and R. Motwani, “Similarity search in high dimensions via hashing,” inProc. 25th Int. Conf. Very Large Databases, 1999, pp. 518–529.

[11] W. Johnson and J. Lindenstrauss, “Extensions of Lipschitz mappings into a Hilbert Space,”Contemporary Math., vol. 26, pp. 189–206, 1984.

[12] P. Ciaccia, M. Patella, and P. Zezula, “M-tree: An efficient access method for similarity search in metric spaces,” inProc. 23rd Int.Conf. Very Large Databases, 1997, pp. 426–435.

[13] R. Agrawal and R. Srikant, “Fast algorithms for mining association rules in large databases,” in Proc. 20th Int. Conf. Very Large Databases, 1994, pp. 487–499.

[14] N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger, “The R*-tree: An efficient and robust access method for points and rectangles,” in Proc. ACM SIGMOD Int. Conf. Manage. Data, 1990,pp. 322–331.