International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue 14
November 2017

Multi-Dimensional Datasets used for Keyword Search

[1]EAGA DEEPIKA [2]Dr. PULYALA MAHIPAL REDDY

[1]PG Scholar, Department of CSE, Vaagdevi College of Engineering, Bollikunta Warangal, Telangana, Mail id: eaga.deepika@gmail.com.

[2]Assistant Professor, Department of CSE, Vaagdevi College of Engineering, Bollikunta Warangal, Telangana, Mail id: mahipalreddy.pulyala@gmail.com.

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

1 INTRODUCTION

In today's digital world the amount of data which is developed is increasing day by day. There Isdifferent 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 multidimensional 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 multidimensional 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 multidimensional datasetsOur main contributionsare summarized as follows.

(1)We propose a novel multiscale index for exact andApproximate NKS query processing.

(2) We develop efficient search algorithms thatwork with

the multiscale 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): Contentbased image retrieval(CBIR), also known as query by image content(QBIC) and contentbased visual information retrieval (CBVIR) is the applicationof computer vision techniques to the image retrieval problem, that is, the problemof searching for digital images in large databases. Contentbased image retrieval isopposed to traditional conceptbased approaches (see Conceptbased image indexing).

3. TBIR (Text based image search): Conceptbased image indexing, also variably named as "descriptionbased"or"textbased"image indexing/retrieval, refers to retrieval from textbased indexing of images that may employ keywords, subject headings, captions, or natural language

text. It is opposed to Contentbased image retrieval. Indexing is a technique used in CBIR.

2. LITERATURE SURVEY

We study nearest keyword set (referred to asNKS) queries on textrich multidimensional datasets. AnNKS query is a set of userprovided 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 multidimensional space. Illustrates an NKS query over a set of two

dimensional data points. Each point is tagged with a set of keywords. For a query the set of points contains all thequery keywords and forms the tightest cluster compared with any other set of points covering all the query keywords. Therefore, the set is the International Research result for the query Q.NKS queries are useful for many applications, such as photosharing in social networks, graph pattern search, geolocation search in GISsystemsand so on.We present an exact and an approximate version of the algorithm. Our experimental results on real and synthetic datasets show that the methodhas morespeedup over stateoftheart treebased techniques.Other related queries include aggregate nearest keyword search in spatial databases, topk preferential

query, topk sites in a spatialdata based on their influence on feature points, and optimal location queries. Ourwork is different from these techniques. First, existing works mainlyfocus on thetype of queries where the coordinates of query points are known. Even though it ispossible to make their cost functions same to the cost function in NKS queries, suchtuning does not change their techniques. The proposed techniques use location information as an integral part to perform a best first search on the IRTree, and querycoordinates play a fundamental role in almost every step of the algorithms to prunethe search space. Moreover, these techniques do not provide concrete guidelinesonhow to enable efficient processing for the type of queries where query coordinates aremissing. Second, in multidimensional spaces, it is difficult for users to provide meaningful coordinates, and our work deals with another type of queries where users canonly provide keywords as input. Without query coordinates, it is difficult to adaptexisting techniques to our problem.Finding nearest neighbors in large multidimensional data has always been one of the research interests in data mining field. In this paper, we present our continuous research on similarity search problems. Previouswork on exploring the meaning of K nearest neighbors from a new perspective in Pan KNN. It redefines the distances between data points and a given query point Q, efficientlyand effectively selecting data points which are closest to Q. It can be applied in various data mining fields. A large amount of real data sets have irrelevant or obstacle information which greatly affects the effectiveness and efficiency of finding nearest neighbors for a given query data point. In this paper, we present our approach to solving the similarity search problem in the presence of obstacles. We apply the concept of obstacle points and process the similarity search problems in a different way. This approach can assist to improve the performance of existing data analysis approaches.

The similarity between two data points used to be based on a similarity function such as Euclidean distance which aggregates the difference between each dimension of thetwo data points in traditional nearest neighbor problems.In those applications, the nearest neighbor problems are solved based

on the distance between the data point and the query point over a fixed set of dimensions (features). However, such approaches only focus on full similarities, i.e.,
the similarity in full data space of the data set. Also early methods suffer from the "curse of dimensionality". In a high dimensional space the data are usually sparse, and widely used distance metric such as Euclidean distance may not work well as dimensionalitygoes higher. Recent research [8] shows that in high dimensions nearest neighbor queries become unstable: the difference of the distances of farthest and nearest points to some query point does not increase as fast as the minimum of the two, thus the distance between two data points in high dimensionality is less meaningful. Some approaches are proposed targeting partial similarities. However, they have limitations such as the requirement of the fixed subset of dimensions, or fixed number of dimensions as the input parameter(s) for the algorithms.

3 INDEX STRUCTURE FOR EXACT PROMISH

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 invertedindex referred to as Ikp. In Ikp, we treat keywords as keys,and each keyword points to a set of data points that areassociated with the keyword. Let D be a set of data pointsand V be a dictionary that contains all the keywords appearingin D. We build Ikp for D as follows. (1) For each v 2 V,we create a key entry in Ikp, and this key entry points to aset of data points Dv ¼ fo 2 Dj v 2 sðoÞg (i.e., a set includesall data points in D that contain keyword v). (2) We repeat(1) until all the keywords in V are processed. In Fig. 2, anexample for Ikp is shown in the dashed rectangle at thebottom.Hashtable-Inverted Index Pairs HI. The second componentconsists of multiple hashtables and inverted indexesreferred to as HI. HI is controlled by three parameters: (1)(Index level) L, (2) (Number of random unit vectors) m, and (3)(hashtable size) B. All the three parameters are non-negativeintegers. Next, we describe how these three parameters controlthe construction of HI.

In general, HI contains L hashtable-inverted index pairs,characterized by fðHðsÞ; IðsÞ khbÞ j s 2 f0; 1; 2; . . . ; L _ 1gg,where HðsÞ and IðsÞ khb are the s-th hashtable and invertedindex, respectively.

Algorithm . SearchInSubset

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue 14
November 2017

In: F0: subset of points; Q: query keywords; q: query size
In: PQ: priority queue of top-k results
1: rk PQ½k_:r /* kth smallest diameter */
2: SL ½ðv; ½ _Þ_: list of lists to store groups per querykeyword
3: for all v 2 Q do
4: SL½v_ f8o 2 F0 : o is tagged with vg /* form groups */
5: end for
6: /* Pairwise inner joins of the groups*/
7: AL: adjacency list to store distances between points
8: M 0: adjacency list to store count of pairs betweengroups
9: for all ðvi; vjÞ 2 Q such that i q; j q; i < j do
10: for all o 2 SL½vi_ do
11: for all o0 2 SL½vj_ do
12: if jjo _ o0jj2 rk then
13: AL½o; o0_ jjo _ o0jj2
14: M½vi; vj_ M½vi; vj_ þ 1
15: end if
16: end for
17: end for
18: end for
19: /* Order groups by a greedy approach */
20: curOrder ½ _
21: while Q 6¼ ; do
22: ðvi; vjÞ removeSmallestEdge(M)
23: if vi 62 curOrder then
24: curOrder.append(vi); Q Q n vi
25: end if
26: if vj 62 curOrder then
27: curOrder.append(vj); Q Q n vj
28: end if
29: end while
30: sort(SL, curOrder) /* order groups */
31: findCandidates(q, AL, PQ, Idx, SL, curSet, curSetr, rk)

4.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.In this paper, we study nearest keyword set (referred to asNKS) queries on text-rich multi-dimensional datasets. AnNKS query is a set of user-provided keywords, and theresult of the query may include k sets of data points each ofwhich contains all the query keywords and forms one of thetop-k tightest cluster in the multi-dimensional space.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.

5 CONCLUSIONS

In this paper, we proposed solutions to the problem of top-knearest keyword set search in multi-dimensional datasets.We proposed a novel index called ProMiSH based on randomprojections and hashing. Based on this index, wedeveloped ProMiSH-E that finds an optimal subset of pointsand ProMiSH-A that searches near-optimal results with betterefficiency. Our empirical results show that ProMiSH isfaster than state-of-the-art tree-based techniques, with multipleorders of magnitude performance improvement. Moreover,our techniques scale well with both real and syntheticdatasets.Ranking functions. In the future, we plan to explore otherscoring schemes for ranking the result sets. In one scheme,we may assign weights to the keywords of a point by usingtechniques like tf-idf. Then, each group of points can bescored based on distance between points and weights ofkeywords. Furthermore, the criteria of a result containingall the keywords can be relaxed to generate results havingonly a subset of the query keywords

REFERENCES

[1] W. Li and C. X. Chen, “Efficient data modeling and queryingsystem for multi-dimensional spatial data,” in Proc. 16th ACMSIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst., 2008, pp. 58:1–58:4.

[2] D. Zhang, B. C. Ooi, and A. K. H. Tung, “Locating mapped resourcesin 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 ofimages with missing geotags,” in Proc. IEEE Int. Conf. GranularComput., 2010, pp. 420–425.

[4] V. Singh, A. Bhattacharya, and A. K. Singh, “Querying spatialpatterns,” in Proc. 13th Int. Conf. Extending Database Technol.: Adv.Database Technol., 2010, pp. 418–429.

[5] J. Bourgain, “On lipschitz embedding of finite metric spaces in Hilbert space,” Israel J. Math., vol. 52, pp. 46–52, 1985.

[6] H. He and A. K. Singh, “GraphRank: Statistical modeling andmining of significant subgraphs in the feature space,” in Proc. 6thInt. Conf. Data Mining, 2006, pp. 885–890.

[7] X. Cao, G. Cong, C. S. Jensen, and B. C. Ooi, “Collective spatialkeyword querying,”

in Proc. ACM SIGMOD Int. Conf. Manage.Data, 2011, pp. 373–384.

[8] C. Long, R. C.-W. Wong, K. Wang, and A. W.-C. Fu, "Collectivespatial keyword queries: A distance owner-driven approach," inProc. ACM SIGMOD Int. Conf. Manage. Data, 2013, pp. 689–700.

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

[10] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, "Localitysensitivehashing scheme based on p-stable distributions," inProc. 20th Annu. Symp. Comput. Geometry, 2004, pp. 253–262.

[11] Y. Zhou, X. Xie, C. Wang, Y. Gong, and W.-Y. Ma, "Hybrid indexstructures for location-based web search," in Proc. 14th ACM Int.Conf. Inf. Knowl. Manage., 2005, pp. 155–162.

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

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

[14] A. Khodaei, C. Shahabi, and C. Li, "Hybrid indexing and seamlessranking of spatial and textual features of web documents," in Proc.21st Int. Conf. Database Expert Syst. Appl., 2010, pp. 450–466.

[15] A. Guttman, "R-trees: A dynamic index structure for spatialsearching," in Proc. ACM SIGMOD Int. Conf. Manage. Data, 1984,pp. 47–57.

[16] I. De Felipe, V. Hristidis, and N. Rishe, "Keyword search on spatialdatabases," in Proc. IEEE 24th Int. Conf. Data Eng., 2008,pp. 656–665.

[17] G. Cong, C. S. Jensen, and D. Wu, "Efficient retrieval of the top-kmost relevant spatial web objects," Proc. VLDB Endowment, vol. 2,pp. 337–348, 2009.