International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 16
June 2018

An Effective and Secure Keyword Set Search in Large Amount of Datasets

P.Sowjanya & K.Srujana
1PG Scholar, Dept of CSE, Prakasam Engineering College, Prakasam(Dt), AP, India.
2Associative Professor, Dept of CSE, Prakasam Engineering College, Prakasam(Dt), AP, India.

Abstract— *In computer Data set analysis, hundreds of files are usually examined. Much of the data in those files consists of unstructured text, whose analysis by computer examiners is difficult to be performed. In this context, automated methods of analysis are of great interest. In particular, algorithms for clustering documents can facilitate the discovery of new and useful knowledge from the documents under analysis we present an approach that applies document clustering algorithms to forensic analysis of computers seized in police investigation. We illustrate the proposed approach and get the lines and clustering word matching lines. We also present and discuss several practical results that can be useful for researchers and practitioners of Data set.*

Keywords—Clustring,Filtering,Multi-dimensional data, Indexing, Hashing**.**

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 repre-sentedas 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) 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. Fig. 1 illustrates an NKS query over a set of 2-dimensional data points. Each point is tagged with a set of keywords. For a query Q = fa; b; cg, the set of points f7; 8; 9g contains all the query keywords fa; b; cg and forms the tightest cluster compared with any other set of points covering all the query keywords. Therefore, the set f7;8; 9g is the top-1 result for the query Q.

NKS queries are useful for many applications, such as photo- sharing in social networks, graph pattern search, geo-location search in GIS systems1[1], [2], and so on. The following are a few examples. Consider a photo-sharing social network (e.g., Facebook),where photos are tagged with people names and Fig. 1. An example of an NKS query on a keyword tagged multi-dimensional dataset. The top-1 result for query fa; b; cg is the set of points f7; 8; 9g. locations. These photos can be embedded in a high dimensional feature space of texture, color, or shape [3], [4]. Here an NKS query can find a group of similar photos which contains a set of people.

NKS queries are useful for graph pattern search, where labeled graphs are

embedded in a high dimensional space (e.g., through Lip schitz embedding [5]) for scalability. In this case, a search for a sub graph with a set of specified labels can be answered by an NKS query in the embedded space [6].

NKS queries can also reveal geographic patterns. GIS can characterize a region by a high-dimensional set of attributes, such as pressure, humidity, and soil types. Meanwhile, these regions can also be tagged with information such as diseases. An epidemiologist can formulate NKS queries to discover patterns by finding a set of similar regions with all the diseases of her interest. 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 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) [10], which is a state-of-the-art method for nearest neighbor search in high- imensional spaces. Unlike LSH-based methods that allow only approximate search with probabilistic guarantees, the index structure in ProMiSH-E supports accurate search. ProMiSH-E creates hash tables at multiple bin- widths, called index levels. A single round of search in a hash table yields subsets of points that contain query results, and ProMiSH-E explores each subset using a fast pruning-based algorithm. ProMiSH-A is an approximate variation of ProMiSH-E for better time and space efficiency. We evaluate the performance of ProMiSH on both real and synthetic datasets and employ state-of-the-art VbR - Tree [2] and CoSKQ [8] as baselines. The empirical results reveal that ProMiSH consistently outperforms the baseline algorithms with up to 60 times of speedup, and ProMiSH-A is up to 16 times faster than ProMiSH-E obtaining near-optimal results.

II.LITERATURE SURVEY

Z. Li, H. Xu, Y. Lu, and A. Qian, ―Aggregate nearest keyword search in spatial databases,‖ in Asia-Pacific Web Conference, 2010. Keyword search on relational databases is useful and popular for many users without technical background. Recently, aggregate keyword search on relational databases was proposed and has attracted interest. However, two important problems still remain. First, aggregate keyword search can be very costly on large relational databases, partly due to the lack of effcient in-dexes. Second, the top-k answers to an aggregate keyword query has not been addressed systematically, including both the ranking model and the effcient evaluation methods. We also report a systematic performance evaluation using real data sets.

De Felipe, V. Hristidis, and N. Rishe, "Keyword search on spatial databases," in ICDE, 2008, pp. 656–665.

Many applications require finding objects closest to aspecified 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.

M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, ―Locality-sensitivehashing scheme based on p-stable distributions,‖ in SCG, 2004.

We present a novel Locality-Sensitive Hashing scheme for the Approximate Nearest Neighbor Problem under lp norm, based on pstable distributions. Our scheme improves the running time of the earlier algorithm for the case of the l2 norm. It also yields the first known provably efficient approximate NN algorithm for the case p<1. We also show that the algorithm finds the *exact* near neigbhor in O(log n) time for data satisfying certain ―bounded growth‖ condition. Unlike earlier schemes, our LSH scheme works directly on points in the Euclidean space without embeddings. Consequently, the resulting query time bound is free of large factors and is simple and easy to implement. Our experiments (on synthetic data sets) show that the our data structure is up to 40 times faster than kd-tree. Our algorithm also inherits two very convenient properties of LSH schemes. The first one is that it works well on data that isextremely high-dimensional but sparse. Specifically, the running time bound remains unchanged if d denotes the maximum numberof non-zero elements in vectors. To our knowledge, this propertyis not shared by other known spatial data structures.

III. PROPOSED METHODOLOGY

Spatial queries with keywords have not been extensively explored. In the past years, the community has sparked enthusiasm in studying keyword search in relational databases. It is until recently that attention was diverted to multidimensional data. The best method to date for nearest neighbor search with keywords is due to Felipe et al. They nicely integrate two well-known concepts: R-tree, a popular spatial index, and signature file, an effective method for keyword-based document retrieval. By doing so they develop a structure called the IR2 -tree, which has the strengths of both R-trees and signature files. Like R-trees, the IR2 - tree preserves objects‟ spatial proximity, which is the key to solving spatial queries efficiently. On the other hand, like signature files, the IR2 -tree is able to filter a considerable portion of the objects that do not contain all the query keywords, thus significantly reducing the number of objects to be examined.

In this work, here design a variant of inverted index that is optimized for multidimensional points, and is thus named the spatial inverted index (SI-index). This access method successfully incorporates point coordinates into a conventional inverted index with small extra space, owing to a delicate compact storage scheme. Meanwhile, an SI-index preserves the spatial locality of data points, and comes with an R-tree built on every inverted list at little space overhead. As a result, it offers two competing ways for query processing. We can (sequentially) merge multiple lists very much like merging traditional inverted lists by ids. Alternatively, we can also leverage the R-trees to browse the points of all relevant lists in ascending order of their distances to the query point. As demonstrated by experiments, the SI-

index significantly outperforms the IR2 -tree in query efficiency, often by a factor of orders of magnitude.

This work addresses a novel clustering algorithm to discover the latent semantics in a text corpus from a fuzzy linguistic perspective. Besides the applicability in text domains, it can be extended to the applications, such as data mining, bioinformatics, content-based or collaborative information filtering, and so forth.Web documents can constitute several latent semantic topics equipped with numerical coefficients (fuzzy linguistic coefficients) that indicate the significance levels of these inherent situations. A collection of documents and its corresponding fuzzy linguistic topological space L are two finite and discrete topological spaces, where L = {C1, C2, · · · ,Cn} and Ci denotes a semantic topological category. A discrete topological category is composed of all discrete features, that is, attribute-value pairs. The features in a document are extracted by using semi-supervised learning schemes called named entities. Named entity recognition (NER) can identify one item from a set of features that have similar attributes, i.e., named categories.Examples of named categories are person, affiliations, location, and so on. Consider the polyse my like the term “jaguar” can be classified as “animal,” “vehicle,” and so forth. If the term “jaguar” is associated with the items, such as “cat,”“tiger,” and “feline,” the term “jaguar” is more possible to be classified into the named category “animal.” Fuzzy linguistic coefficient is given to measure the possibilities of a term belonging to every category where the term is associated with other co-occurring terms.

The general framework of our clustering method consists of two phases. The first phase, feature extraction, is to extract key named entities from a collection of “indexed” documents; the second phrase, fuzzy clustering, is to determine relations between features and identify their linguistic categories.The kernel of the first phrase is to identify the key features and their named categories. In order to identify features in documents, we deployed the named entity recognition method.From a given sentence, NER method first finds out the segmented entities composed of a sequence of words, and then classifies the entities by a type or named category, such as person, organization, location, and so on. This work considers only noun entities, especially some representative entities. Therefore, discriminative linear chain conditional random field (CRF) was used to choose the particular features in the corpus. A CRF is a simple framework for labeling and segmenting data that models a conditional distribution P(z|x) by selecting the label sequences to label a novel observation sequence x with an associated undirected graph structure that obeys the Markov property. When conditioned on the observations that are given in a particular observation sequence, the CRF defines a single log-linear distribution over the labeled sequence. CRF model does not need to explicitly present the dependencies of input variables x affording the use of rich and global features of the input, thus allows relaxation of the strong independent assumptions made by HMMs.

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 ran-dom projections and hashing. Based on this index, we de-veloped ProMiSH-E that finds an optimal subset of points and ProMiSH-A that searches near-optimal results with better 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 hashtable and the inverted index of a HI structure. Therefore, all the hashtables and the inverted indexes of HI can again be stored using a similar directory-file structure

V.FUTURE ENHANCEMENTS

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 keyword.

REFERENCES

[1] W. Li and C. X. Chen, ―Efficient data modeling and querying system for multi-dimensional spatial data,‖ in GIS, 2008, pp. 58:1–58:4.

[2] D. Zhang, B. C. Ooi, and A. K. H. Tung, ―Locating mappedresources in web 2.0,‖ in ICDE, 2010, pp. 521–532.

[3] V. Singh, S. Venkatesha, and A. K. Singh, ―Geo-clustering ofimages with missing geotags,‖ in GRC, 2010, pp. 420–425.

[4] V. Singh, A. Bhattacharya, and A. K. Singh, ―Querying spatialpatterns,‖ in EDBT, 2010, pp. 418–429.

[5] J. Bourgain, ―On lipschitz embedding of finite metric spaces inhilbert space,‖ Israel J. Math., vol. 52, pp. 46–52, 1985.

[6] H. He and A. K. Singh, ―Graphrank: Statistical modeling and mining ofsignificant subgraphs in the feature space,‖ in ICDM, 2006, pp. 885–890.

[7] X. Cao, G. Cong, C. S. Jensen, and B. C. Ooi, ―Collective spatialkeyword querying,‖ in SIGMOD, 2011.

[8] C. Long, R. C.-W. Wong, K. Wang, and A. W.-C. Fu, ―Collective spatialkeyword queries: a distance owner-driven approach,‖ in SIGMOD, 2013.

[9] [9] D. Zhang, Y. M. Chee, A. Mondal, A. K. H. Tung, and M.Kitsuregawa,

―Keyword search in spatial databases: Towardssearching by document,‖ in ICDE, 2009, pp. 688–699.

[10] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, ―Locality-sensitivehashingscheme based on p-stable distributions,‖ in SCG, 2004.

[11] Y. Zhou, X. Xie, C. Wang, Y. Gong, and W.-Y. Ma, ―Hybrid index structures for location-based web search,‖ in CIKM, 2005.

[12] R. Hariharan, B. Hore, C. Li, and S. Mehrotra, ―Processing spatial- keyword (SK) queries in geographic information retrieval (GIR) sys- tems,‖ in SSDBM, 2007.

[13] S. Vaid, C. B. Jones, H. Joho, and M. Sanderson, ―Spatio-textual indexing for geographical search on the web,‖ in SSTD, 2005.

[14] A. Khodaei, C. Shahabi, and C. Li, ―Hybrid indexing and seamless ranking of spatial and textual features of web documents,‖ in DEXA,

2010, pp. 450–466.

[15] A. Guttman, ―R-trees: A dynamic index structure for spatial searching,‖in ACM SIGMOD, 1984, pp. 47–57.

[16] I. De Felipe, V. Hristidis, and N. Rishe, ―Keyword search on spatial databases,‖ in ICDE, 2008, pp. 656–665.

[17] 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.

[18] B. Martins, M. J. Silva, and L. Andrade, ―Indexing and ranking in geo-ir systems,‖ in workshop on GIR, 2005, pp. 31– 34.

[19] Z. Li, H. Xu, Y. Lu, and A. Qian, ―Aggregate nearest keyword search in spatial databases,‖ in Asia-Pacific Web Conference, 2010.

[20] M. L. Yiu, X. Dai, N. Mamoulis, and M. Vaitis, ―Top-k spatial preference queries,‖ in ICDE, 2007, pp. 1076–1085.