WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Clustering with Local Density Peaks Based Spanning Tree

V.Sneha[#1], R.Angelin Preethi[#2]

[#1] M.Sc, Department of Computer Science, Kamban College of Arts and Science for Women, Tiruvannamalai-606603.

[#2] Assistant professor, Department of Computer Science, Kamban College of Arts and Science for Women, Tiruvannamalai-606603.

ABSTRACT:

Clustering analysis has been widely used in statistics, machine learning, pattern recognition, image processing, and so on. It is a great challenge for most existing clustering algorithms to discover clusters with arbitrary shapes. Clustering algorithms based on Minimum spanning tree (MST) are able to discover clusters with arbitrary shapes, but they are time consuming and susceptible to noise points. In this paper, we employ local density peaks (LDP) to represent the whole data set and define a shared neighbors-based distance between local density peaks to better measure the dissimilarity between objects on manifold data. On the basis of local density peaks and the new distance, we propose a novel MST-based clustering algorithm called LDP-MST. It first uses local density peaks to construct MST and then repeatedly cuts the longest edge until a given number of clusters are found. The experimental results on synthetic data sets and real data sets show that our algorithm is competent with state-of-the-art methods when discovering clusters with complex structures

1.INTRODUCTION:

Clustering, as an important unsupervised learning method, has been widely studied and applied in statistics, machine learning, pattern recognition and image processing. It aims to classify objects into several groups, so that objects in the same group are as similar as possible, while objects in different groups are as distinct as possible. Many clustering algorithms have been proposed. The MST-based clustering algorithm proposed in defines inconsistent edges as those whose weights are significantly larger than the average weight of nearby edges in the tree.However, only making use of the edges contained in an MST to partition a data set, it may not detect clusters with complex structures and is easily affected by noise points. The algorithm in uses a graph composed of two rounds of minimum spanning trees to cluster. There are other MST-based clustering algorithms that maximize or minimize the degrees of link of the vertices In order to avoid the influence of noise points and reduce the running time of MST-based clustering methods, a potential way is to select some points as representatives so that the representatives roughly retain the shape of clusters and exclude the interference of noise points. Moreover, constructing MST on representatives greatly reduces the running time compared with using all points. Inspired by this idea, we propose a minimum spanning tree-based clustering with local density peaks, called LDP-MST, which is both computationally efficient and competent with other state-of-the-art clustering approaches when discovering complex clusters. In LDP-MST, we first find local density peaks and the remaining points are assigned to the corresponding local density peaks. Then, we define a new distance between local density peaks based on shared neighbors and use the new distance to construct minimum spanning tree on the local density peaks. We obtain the final clusters by continually removing the longest edge. In order to demonstrate the effectiveness of our method, we do experiments by comparing LDP-MST with partitioning method Kmeans

2. LITERATURE SURVEY

[1].**Delbert Dueck**Clustering data by identifying a subset of representative examples is important for processing sensory signals and detecting patterns in data. Such "exemplars" can be found by randomly choosing an initial subset of data points and then iteratively refining it, but this works well only if that initial choice is close to a good solution. We devised a method called "affinity propagation," which takes as input measures of similarity between pairs of data points. Real-valued messages are exchanged between data points until a high-quality set of exemplars and corresponding clusters gradually emerges. We used affinity propagation to cluster images of faces, detect genes in microarray data, identify representative sentences in this manuscript, and identify cities that are efficiently accessed by airline travel. Affinity propagation found clusters with much lower error than other methods, and it did so in less than one-hundredth the amount of time.

[2].**ALEXRODRIGUEZ AND ALESSANDRO LAIO**Cluster analysis is aimed at classifying elements into categories on the basis of their similarity. Its applications range from astronomy to bioinformatics, bibliometrics, and pattern recognition. We propose an approach based on the idea that cluster centers are characterized by a higher density than their neighbors and by a relatively large distance from points with higher densities. This idea forms the basis of a clustering procedure in which the number of clusters arises intuitively, outliers

are automatically spotted and excluded from the analysis, and clusters are recognized regardless of their shape and of the dimensionality of the space in which they are embedded. We demonstrate the power of the algorithm on several test cases.

[3].**YewangChena**Centroid-based clustering approaches fail to recognize extremely complex patterns that are non-isotropic. We analyze the underlying causes and find some inherent flaws in these approaches, including Shape Loss, False Distances and False Peaks, which typically cause centroid-based approaches to fail when applied to complex patterns. As an alternative to current methods, we propose a hybrid decentralized approach named DCore, which is based on finding density cores instead of centroids, to overcome these flaws. The underlying idea is that we consider each cluster to have a shrunken density core that roughly retains the shape of the cluster. Each such core consists of a set of loosely connected local density peaks of higher density than their surroundings. Borders, edges and outliers are distributed around the outsides of these cores in a hierarchical structure. Experiments demonstrate that the promise of DCore lies in its power to recognize extremely complex patterns and its high performance in real applications, for example, image segmentation and face clustering, regardless of the dimensionality of the space in which the data are embedded.

[4]. **MDaszykowskia**A density-based unsupervised clustering approach for detecting natural patterns in data (further denoted as NP) is presented, and its performance is illustrated for data sets with different types of clusters. NP works for arbitrary clusters, is a single-scan technique, requires no presumptions regarding data distribution and requires only one input parameter, which describes the minimal number of objects, considered as cluster. Moreover, a comparison of NP with partitioning approaches is demonstrated. NP can be applied not only for data clustering, but also for the identification of outliers.

[5]. Xiaochun WangDue to their ability to detect clusters with irregular boundaries, minimum spanning tree-based clustering algorithms have been widely used in practice. However, in such clustering algorithms, the search for nearest neighbor in the construction of minimum spanning trees is the main source of computation and the standard solutions take O(N^{2}) time. In this paper, we present a fast minimum spanning tree-inspired clustering algorithm, which, by using an efficient implementation of the cut and the cycle property of the minimum spanning trees, can have much better performance than O(N^{2}).

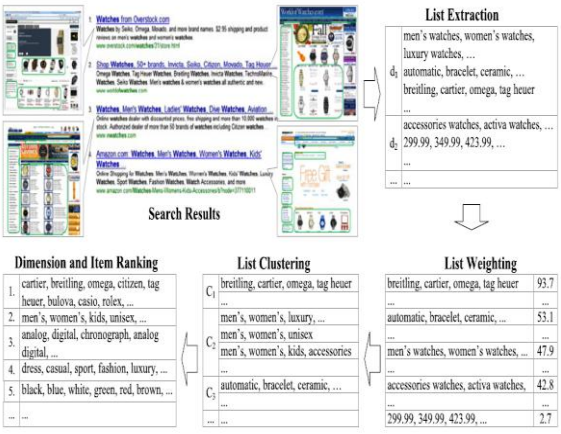
3.PROPOSED SYSTEM:

We propose aggregating frequent lists within the top search results to mine query facets and implement a system called QDMiner. More specifically, QDMiner extracts lists from free text, HTML tags, and repeat regions contained in the top search results, groups them into clusters based on the items they contain, then ranks the clusters and items based on how the lists and items appear in the top results. We propose two models, the Unique Website Model and the Context Similarity Model, to rank query facets. In the Unique Website Model, we assume that lists from the same website might contain duplicated information, whereas different websites are independent and each cancontribute a separated vote for weighting facets.We propose the Context Similarity Model, in which we model the fine-grained similarity between each pair of lists. More specifically, we estimate the degree of duplication between two lists based on their contexts and penalize facets containing lists with high duplication.

In this paper, we explore to automatically find query dependent facets for open-domain queries based on a general Web search engine. Facets of a query are automatically mined from the top web search results of the query without any additional domain knowledge required.

As query facets are good summaries of a query and are potentially useful for users to understand the query and help them explore information, they are possible data sources that enable a general open-domain faceted exploratory search

ARCHITECTURE DIAGRAM



4. METHODOLOGY:

MODULES DESCRIPTION

Dataset

In the first module, We build datasets from scratch. First, we build a service for finding facets, and invite human subjects to issue queries on

topics they know well. We collect 89 queries issued by the subjects, and name them as "UserQ". As this approach might induce a bias towards topics in which lists are more useful than general web queries, we further randomly sample another set of 105 English queries from a query log of a commercial search engine, and name this set of queries as "RandQ". We first ask a subject to manually create facets and add items that are covered by the query, based on his/her knowledge after a deep survey on any related resources (such as Wikipedia, Freebase, or official web sites related to the query). We then aggregate the qualified items in the facets returned by all algorithms we want to evaluate, and ask the subject to assign unlabelled items into the created facets.

List and Context Extraction

We extract all text within document d and split it into sentences. We then employ the pattern which is similar to that in , to extract matched items from each sentence. We name this sentence based pattern as TEXTS. In Example 1, the items in italic font are extracted as a list. We further use the pattern to extract lists from some semi-structured paragraphs. It extracts lists from continuous lines that are comprised of two parts separated by a dash or a colon. The first parts of these lines are extracted as a list. We named this text-based pattern as TEXT. For a list extracted by the pattern TEXTS, its container node is the sentence containing the extracted list. similarly, for a list extracted by pattern TEXTP, its container node is the paragraph containing the items. We then add the previous and next sentence or paragraph into the context correspondingly.

List clustering Similar

An individual list may inevitably include noise. (2) An individual list usually contains a small number of items of a facet and thus it is far from complete; (3) many lists contain duplicated information. They are not exactly same, but share overlapped items. To conquer the above issues, we group similar lists together to compose facets.

The QT algorithm assumes that all data is equally important, and the cluster that has the most number of points is selected in each iteration. In our problem, lists are not equally important. Better lists should be grouped first. We modify the original QT algorithm to first group highly weighted lists.

Facet Ranking & Item Ranking

The lists in c are extracted from more unique content of search results; and the lists in c are more important, i.e., they have higher weights. Here we emphasize "unique" content, because sometimes there are duplicated content and lists among the top search results. The importance of an item depends on how many lists contain the item and its ranks in the lists. As a better item is usually ranked higher by its creator than a worse item in the original list.

Search result:

QD Miner is based on the assumption that most top results of a query are relevant. In this section, we investigate whether our facet mining algorithms are significantly affected by the quality of search results. We experiment with the following configurations: (1) Top - using the original top K results; (2) Top Shuffle - randomly shuffling the top K results; (3) Random - randomly selecting K results from the original 100 results and then shuffling them. In general, the Random method generates worse ranking than Top Shuffle, and both perform worse than Top in ranking effectiveness.

5 .CONCLUSION

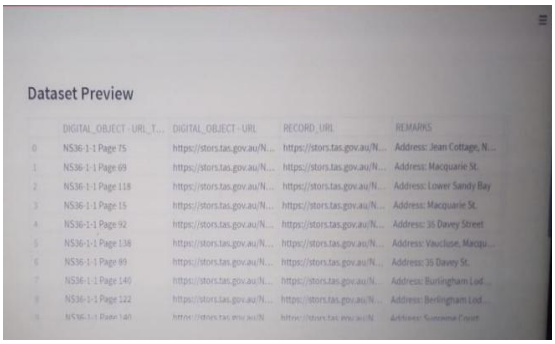
In this paper, we study the problem of finding query facets. We propose a systematic solution, which we refer to as QDMiner, to automatically mine query facets by aggregating frequent lists from free text, HTML tags, and repeat regions within top search results. We create two human annotated data sets and apply existing metrics and two new combined metrics to evaluate the quality of query facets. Experimental results show that useful query facets are mined by the approach. We further analyze the problem of duplicated lists, and find that facets can be improved by modeling fine-grained similarities between lists within a facet by comparing their similarities.

As the first approach of finding query facets, QDMiner can be improved in many aspects. For example, some semisupervised bootstrapping list extraction algorithms can be used to iteratively extract more lists from the top results. Specific website wrappers can also be employed to extract high-quality lists from authoritative websites. Adding these lists may improve both accuracy and recall of query facets. Part-of-speech information can be used to further check the homogeneity of lists and improve the quality of query facets. We will explore these topics to refine facets in the future. We will also investigate some other related topics to finding query facets. Good descriptions of query facets may be helpful for users to better understandthe facets. Automatically generate meaningful descriptions is an interesting research topic.

6.RESULTS

Fig :1 Data Analysis

Fig:2 Clustering Data



REFERENCES:

[1] O. Ben-Yitzhak, N. Golbandi, N. Har'El, R. Lempel, A. Neumann, S. Ofek-Koifman, D. Sheinwald, E. Shekita, B. Sznajder, and S. Yogev, "Beyond basic faceted search," in Proc. Int. Conf. Web Search Data Mining, 2008, pp. 33–44.

[2] M. Diao, S. Mukherjea, N. Rajput, and K. Srivastava,, "Faceted search and browsing of audio content on spoken web," in Proc. 19th ACM Int. Conf. Inf. Knowl. Manage., 2010, pp. 1029–1038.

[3] D. Dash, J. Rao, N. Megiddo, A. Ailamaki, and G. Lohman, "Dynamic faceted search for discovery-driven analysis," in ACM Int. Conf. Inf. Knowl. Manage., pp. 3–12, 2008.

[4] W. Kong and J. Allan, "Extending faceted search to the general web," in Proc.ACMInt. Conf. Inf. Knowl. Manage., 2014, pp. 839–848.

[5] T. Cheng, X. Yan, and K. C.-C. Chang, "Supporting entity search: A large-scale prototype search engine," in Proc. ACM SIGMOD Int. Conf. Manage. Data, 2007, pp. 1144–1146.

[6] K. Balog, E. Meij, and M. de Rijke, "Entity search: Building bridges between two worlds," in Proc. 3rd Int. Semantic Search Workshop, 2010, pp. 9:1–9:5.

[7] M. Bron, K. Balog, and M. de Rijke, "Ranking related entities: Components and analyses," in Proc. ACM Int. Conf. Inf. Knowl. Manage., 2010, pp. 1079–1088.

[8] C. Li, N. Yan, S. B. Roy, L. Lisham, and G. Das, "Facetedpedia: Dynamic generation of query-dependent faceted interfaces for wikipedia," in Proc. 19th Int. Conf. World Wide Web, 2010, pp. 651–660.

[9] W. Dakka and P. G. Ipeirotis, "Automatic extraction of useful facet hierarchies from text databases," in Proc. IEEE 24th Int. Conf. Data Eng., 2008, pp. 466–475.

[10] A. Herdagdelen, M. Ciaramita, D. Mahler, M. Holmqvist, K. Hall, S. Riezler, and E. Alfonseca, "Generalized syntactic and semantic models of query reformulation," in Proc. 33rd Int. ACM SIGIR Conf. Res. Develop. Inf. retrieval, 2010, pp. 283–290.

[11] M. Mitra, A. Singhal, and C. Buckley, "Improving automatic query expansion," in Proc. 21st Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval, 1998, pp. 206–214.

[12] P. Anick, "Using terminological feedback for web search refinement: A log-based study," in Proc. 26th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval, 2003, pp. 88–95.

[13] S. Riezler, Y. Liu, and A. Vasserman, "Translating queries into snippets for improved query expansion," in Proc. 22nd Int. Conf. Comput. Ling., 2008, pp. 737–744.

[14] X. Xue and W. B. Croft, "Modeling reformulation using query distributions," ACM Trans. Inf. Syst., vol. 31, no. 2, pp. 6:1–6:34, May 2013.