Expert Systems with Applications 42 (2015) 2785–2797

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Efficient agglomerative hierarchical clustering

Athman Bouguettaya [a,*], Qi Yu [b], Xumin Liu [b], Xiangmin Zhou [c], Andy Song [a]

[a] RMIT University, Australia
[b] Rochester Institute of Technology, USA
[c] Victoria University, Australia

ARTICLE INFO

Article history:
Available online 22 October 2014

Keywords:
Clustering analysis
Hybrid clustering
Data mining
Data distribution
Coefficient of correlation

ABSTRACT

Hierarchical clustering is of great importance in data analytics especially because of the exponential growth of real-world data. Often these data are unlabelled and there is little prior domain knowledge available. One challenge in handling these huge data collections is the computational cost. In this paper, we aim to improve the efficiency by introducing a set of methods of agglomerative hierarchical clustering. Instead of building cluster hierarchies based on raw data points, our approach builds a hierarchy based on a group of centroids. These centroids represent a group of adjacent points in the data space. By this approach, feature extraction or dimensionality reduction is not required. To evaluate our approach, we have conducted a comprehensive experimental study. We tested the approach with different clustering methods (i.e., UPGMA and SLINK), data distributions, (i.e., normal and uniform), and distance measures (i.e., Euclidean and Canberra). The experimental results indicate that, using the centroid based approach, computational cost can be significantly reduced without compromising the clustering performance. The performance of this approach is relatively consistent regardless the variation of the settings, i.e., clustering methods, data distributions, and distance measures.

1. Introduction

Clustering is an important means of data analytics in real-world scenarios because manual tagging of the data is usually expensive. Furthermore prior knowledge required to facilitate manual tagging is often unavailable or insufficient. Under such circumstances clustering is a more suitable option over supervised learning approaches, such as classification and regression.

Efficient techniques for data clustering has been studied for decades due to its significant implication in real-world applications where the amount of data are often very large and the accumulation of data is often accelerating (Jain, Murty, & Flynn, 1999; Romesburg, 1990). A clustering method which requires less computational cost can be beneficial in general data mining and knowledge discovery, as well as in specific domains e.g. bio-informatics, web usage monitoring and social network analysis. Due to the widespread of web applications, mobile devices and network of sensors, the volume of data to be analyzed grows much faster than computational power, especially in recent years. This flood of data makes efficiency a high priority in developing clustering methods.

In this study we address the efficiency issue of hierarchical clustering which is one of the main stream clustering methods as it is generally applicable to most types of data. In comparison with partitional clustering algorithms such as K-means, hierarchical approaches have higher cost, with a complexity of $O(N^2 logN)$, but they do not require any predefined parameter hence are more suitable for handling real-world data where finding a suitable set of parameters can be tricky.

Hierarchical clustering can go both ways, aggregating from individual points to the most high-level cluster or dividing from a top cluster to atomic data objects. Our focus is the bottom-up approach which is known as the agglomerative approach, because computational cost can be reduced if the bottom-up process starts from somewhere in the middle of the hierarchy and the lower part of the hierarchy is built by a less expensive method such as partitional clustering. This idea would not work well on the top-down approach which is known as divisive hierarchical clustering because it is notorious for its high cost, $O(2^N)$, and verifying middle level sub-clusters by individual data points would still be expensive.

It is possible to use a hierarchical approach to generate middle-level sub-clusters then apply partitional algorithms on these sub-clusters. However predefined parameters like K still need to be determined. Another possible way to improve efficiency in hierarchical clustering is to perform feature extraction or selection,

* Corresponding author. Tel.: +61 399252169; fax: +61 396621617.

E-mail addresses: Athman.Bouguettaya@rmit.edu.au (A. Bouguettaya), qi.yu@rit.edu (Q. Yu), xumin.liu@rit.edu (X. Liu), Xiangmin.zhou@vu.edu.au (X. Zhou), andy.song@rmit.edu.au (A. Song).

http://dx.doi.org/10.1016/j.eswa.2014.09.054
0957-4174/© 2014 Elsevier Ltd. All rights reserved.

which may reduce data dimensionality. However that process often requires domain knowledge of the data. It also makes the clustering outcomes dependent on the performance of the feature extraction or selection algorithms.

In this paper we present an efficient agglomerative hierarchical method which does not require feature extraction or selection. The main goals of this study are:

1. Presenting a methodology of combining agglomerative hierarchical clustering and partitional clustering to reduce the overall computational cost. By this method the number of output clusters needs not to be determined beforehand.
2. Studying the behaviors of our methods with different distributions.
3. Evaluating the performance of our methods based on the coefficients of correlation.

The paper is organized as follows. In Section 2, we briefly discuss the related works. In Section 3, we describe the proposed methodology with associated approaches. In Section 4 the datasets used in this study are described. Section 5 shows the experimental settings and the results. The further discussion on the results is presented in Section 6. Section 7 concludes this study with a brief outlook for further studies.

2. Related work

We review some representative clustering analysis techniques in this section and highlight their difference with the proposed approach. We categorize existing approaches into three major categories: partitional, hierarchical, and hybrid clustering, to achieve a more focused discussion and comparison. We also review some important applications of clustering to demonstrate its importance in data-intensive processing environments (Altingövde, Demir, Can, & Ulusoy, 2008; Hruschka, Campello, Freitas, & de Leon F. de Carvalho, 2009; Jacinth Salome & Suresh, 2012; Jain et al., 1999; Lee, Han, & Whang, 2007; Lin, Liu, & Chen, 2005; Liu & Yu, 2005; Liu, Li, Sim, & Wong, 2007; Lee, Han, Li, & Gonzalez, 2008; Ordonez & Omiecinski, 2004; Pan, Zhang, & Wang, 2008; Rokach, 2010; Xu & Wunsch, 2010; Zhou et al., 2009).

2.1. Hierarchical clustering

A hierarchical algorithm yields a dendrogram representing the nested grouping of patterns and similarity levels at which groupings change (Jain et al., 1999). The clustering process is performed by merging the most similar patterns in the cluster set to form a bigger one. In Bouguettaya (1996) and Bouguettaya, Qi, Park, and Delis (2002), Bouguettaya et al. investigated the different hierarchical clustering algorithms, including UPGMA, WARDS, SLINK, CLINK, etc, and studied the behavior and stability of these algorithms on low-dimensional and high-dimensional data respectively. Hierarchical clustering approaches produce clusters of higher quality. However, these approaches suffer from high time cost. The efficiency of hierarchical algorithms can be improved with the support of index structures (Zhang, Ramakrishnan, & Livny, 1996). In Zhang et al. (1996), “Balancing Iterative Reducing and Clustering using Hierarchies” (BIRCH) has been proposed for minimizing the running time of clustering. BIRCH incrementally clusters very large datasets, whose sizes are much greater than the amount of available memory. Given a large dataset, the clustering process is performed by constructing a height-balanced tree, called CF tree. The algorithm continuously parses the dataset and updates the CF tree until the final result is achieved. BIRCH algorithm adopts the notion of clustering features to capture the information of a cluster. The clusters that are built so far by the algorithm are organized into the CF tree. The leaf node of the CF tree is a sub-cluster instead of a single data point. Therefore, CF tree is a concise representation of the original dataset and can fit into the memory. However, different from the proposed approach, BIRCH adopts the centroid method with fixed order of the points, which may affect the behavior of clustering results.

In recent years, evolutionary computation has been introduced into clustering. As a kind of stochastic search methods, it can often be quite effective in finding optimal solutions. However the efficient aspect is rather an issue as an evolutionary process is time-consuming (Wu, Hu, Maybank, Zhu, & Li, 2012). To speed up a hierarchical agglomerative clustering process, GPU can certainly be utilized (Shalom & Dash, 2013). This study does not involve GPU although the proposed method can include GPU to further enhance the execution time.

2.2. Partitional clustering

In contrast to the hierarchical clustering, a partitional clustering algorithm obtains a flat partition of the dataset which optimizes a predefined criterion function. The most widely used partitional clustering algorithm is K-means clustering, which repeatedly assigns each object to its closest cluster center and computes the new cluster centers accordingly until the predefined criterion is met. Based on how the distance between data points is computed, various partitional clustering algorithms have been developed and representative ones include spectral clustering (Luxburg, 2007; Ng, Jordan, & Weiss, 2001), graph-partitioning based (Dhillon, Guan, & Kulis, 2004), and non-negative matrix factorization based approaches (Li & Ding, 2006). Comparing with K-means clustering, these algorithms usually generate clusters of better quality. However, these algorithms are more computationally involved, requiring performing eigendecomposition or repetitive matrix multiplication, making them not scalable to very large datasets. Mixture model or other density based clustering algorithms output soft cluster memberships, allowing each data point to be associated with multiple clusters with different probabilities. Compared with the proposed approach and hierarchical clustering in general, partitional clustering algorithms suffer two major limitations. First, their performance heavily relies on pre-defined parameters, especially the number of clusters, so the quality of data clusters can not be guaranteed. Second, the resultant clusters have a flat structure instead of hierarchical structure that captured much richer relationship among data points. A hierarchical structure offer a more natural way to organize many real-world objects (e.g., documents and webpages) and facilitate human users to browse the data.

2.3. Hybrid clustering

Hybrid data clustering combines the hierarchical and partitional methods to obtain the good quality of the former and the efficiency of the latter. Different hybrid data clustering algorithms have been proposed (Guha, Rastogi, & Shim, 1998; Lin & Chen, 2005; Wattanachon, Suksawatchon, & Lursinsap, 2009). In Guha et al. (1998), a hybrid clustering algorithm called CURE was proposed to effectively identify the arbitrarily shaped clusters. Given a large dataset, CURE draws a set of data samples from the whole dataset by random sampling. The data samples are grouped as several partitions and those in each partition are partially clustered. The outliers are then removed from the dataset. The final clusters are obtained by further clustering over the partial clusters produced in the previous step. CURE is scalable to large datasets with a linear time complexity. However, different from the proposed approach, it