Expert Systems with Applications 42 (2015) 4965–4981

Contents lists available at ScienceDirect

Expert Systems with Applications

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

Clustering by growing incremental self-organizing neural network

Hao Liu [a,b], Xiao-juan Ban [a,*]

[a] Department of Computer Science and Technology, University of Science and Technology Beijing, No. 30 Xueyuan Road, Haidian District, Beijing 100083, China
[b] Graduate School of Information Science and Technology, Hokkaido University, Kita 14, Nishi 9, Kita-ku, Sapporo, Hokkaido 060-0814, Japan

ARTICLE INFO

Article history:
Available online 16 February 2015

Keywords:
Clustering
Unsupervised learning
Self-organizing neural networks
Incremental learning
Data visualization

ABSTRACT

This paper presents a new clustering algorithm that detects clusters by learning data distribution of each cluster. Different from most existing clustering techniques, the proposed method is able to generate a dynamic two-dimensional topological graph which is used to explore both partitional information and detailed data relationship in each cluster. In addition, the proposed method is also able to work incrementally and detect arbitrary-shaped clusters without requiring the number of clusters as a prerequisite. The experimental data sets including five artificial data sets with various data distributions and an original hand-gesture data set are used to evaluate the proposed method. The comparable experimental results demonstrate the superior performance of the proposed algorithm in learning robustness, efficiency, working with outliers, and visualizing data relationships.

1. Introduction

Clustering is a fundamental and important technique for data analysis, which has been applied in variety of circumstances, such as data mining, pattern recognition and image segmentation. During the past decades, many clustering algorithms have been proposed, which have been successfully applied as solutions for different kinds of clustering or classification problems. However, sometimes, given a completely unknown data set, we have to face the following problems or challenges (Du, 2010; Jain, Murty, & Flynn, 1999; Rui & Donald, 2005).

1. How many clusters (Jain & Dubes, 1988)? The biggest problem with many classical clustering algorithms is that they require the number of clusters, *k*, as an input parameter (Jain, 2010). In this case, users have to own expertise and select an accurate number for the value of *k*. Although there are some estimation methods for determining *k*. it is very common that the value of *k* cannot be exactly predicted, which often directly causes a low quality of data analysis.
2. How to discover clusters with arbitrary shapes (Rui & Donald, 2005)? With most clustering algorithms, the shape of the detected clusters are limited to convex shapes, e.g. spheres, polygons, etc. But a cluster, especially in a spatial data set, may not have a convex shape, but may be represented in an arbitrary shape (Su & Liu, 2005). Moreover, the shape of a cluster might be more complicated in higher-dimensional space.
3. How to reduce negative effects caused by outliers (Rui & Donald, 2005)? It is very common that a real world data set contains outliers (Ester, Kriegel, Sander, & Xu, 1996). In general, it is preferable that a clustering algorithm is less sensitive to the outliers and has a capability of removing them.
4. How to analyze data relationship in detail (Jain et al., 1999; Rui & Donald, 2005)? Most clustering algorithms show data relationship only by categories, which means they do not provide any detailed information about clusters, i.e. the data relationship inside a cluster. The techniques of data visualization or dimensionality reduction can help us understand data relationship in a lower-dimensional space, including the detailed information about the data relationship in each cluster, but they usually do not provide the information about cluster categories.
5. How can a clustering algorithm classify data incrementally (Rui & Donald, 2005)? Most clustering algorithms works with a "batch mode", which means that the entire data set is required to provide before clustering. But, sometimes, it is very difficult to meet this requirement, e.g., a the real time systems collects data sequentially. In this case, working with an "incremental" (non-batch) mode has much benefits.

For each of the problems or challenges mentioned above, there are already many specific solutions. Unfortunately, to the authors' knowledge, there is no solution to meet the combination of all

* Corresponding author. Tel.: +86 10 62334980.
E-mail addresses: liuhao@complex.ist.hokudai.ac.jp (H. Liu), banxj@ustb.edu.cn (X.-j. Ban).

http://dx.doi.org/10.1016/j.eswa.2015.02.006
0957-4174/

these requirements, which means that it will be a difficult task to choose a suitable clustering algorithm, if an unknown data set is given. The purpose of this paper is to propose a novel technique called growing incremental self-organizing neural network (GISONN) to offer such a solution. GISONN considers clustering as a incremental learning task of the data distribution of each cluster. Meanwhile, GISONN automatically maintains a two-dimensional topological graph which is used to represent the information about both cluster categories and the detailed relationship inside a cluster. The main contributions of this method are summarized as follows.

1. The algorithm does not require the number of clusters as an input parameter.
2. The algorithm can discover clusters with arbitrary shapes.
3. The algorithm has a capability to remove outliers.
4. The algorithm can visualize data relationship in a two-dimensional topological graph where data categories and the detailed data relationship in each cluster can be clearly reported.
5. The algorithm can work incrementally.

The remaining parts of this paper are organized as follows. In Section 2, we review the state of the art of clustering techniques. In Section 3, the GISONN algorithm is proposed. In Section 4, the experimental results are presented. Finally, we summarize the features of GISONN and give conclusions in Section 5.

2. Related work

A rough but widely agreed frame is to classify existing clustering techniques into partitional clustering, hierarchical clustering or density-based clustering (Rui & Donald, 2005).

Partitional clustering, in general, is a statistical way of clustering (Chiang, Tsai, & Yang, 2011; Moody & Darken, 1989; Ng & Han, 2002). A classical technique in this category is the *k*-means clustering algorithm (MacQueen, 1967). Given a data set and an integer *k* as the number of expected clusters, the standard *k*-means algorithm first initializes *k* centers. Next, it calculates *k* corresponding clusters, then updates the center of each cluster and reassign data to the cluster with the closest center. This process is repeated until the Mean Squared Error (MSE) converges. Since *k*-means is simple and efficient, it is still widely used even though over 50 years have passed since it was first proposed (Jain, 2010). Motivated by *k*-means, many improved algorithms have been developed (Ghosh & Dubey, 2013; Celebi, Kingravi, & Vela, 2013; Lin & Chen, 2005).

Hierarchical clustering algorithms, in general, organize data with some specific hierarchical structures which are usually realized by a binary clustering tree or a dendrogram (Murtagh & Contreras, 2012). Hierarchical clustering can be implemented by an agglomerative method or a divisive method (Du, 2010). A classical agglomerative clustering algorithm is initialized with *N* clusters, where *N* usually equals the number of data in the data set, *n*. Based on a specific measure of similarity, such as calculating the Euclidean distance or Manhattan Distance between two data, and nested merge operations, new clusters will be formed. Similarly, a classical divisive clustering algorithm goes in an opposite way. The most well known hierarchical algorithms are single-linkage and mean-linkage (Song, Jin, & Shen, 2011; Theodoridis & Koutroumbas, 2009). CURE is an improved single-linkage algorithm, which can form clusters with arbitrary shapes (Guha, Rastogi, & Shim, 1998). However, most typical hierarchical algorithm requires an $O(n^2 \log n)$ complexity, which is a significant cost for a large data set (Rui & Donald, 2005). The approach (Bouguettaya, Yu, Liu, Zhou, & Song, 2015) first finds a group of "centroids" which represent similar data points in the dataset, then it builds a hierarchy based on "centroids" instead of the original data, so that the actual computational cost can be reduced.

Some hybrid algorithms which combine the advantages of partitional and hierarchical clustering have been developed. CHAMELEON (Karypis, Han, & Kumar, 1999), clusters a data set by considering both interconnectivity (the number of links between tow clusters) and closeness (the length of those links) in identifying the most similar pair of clusters, which makes CHAMELEON powerful in discovering clusters with arbitrary shapes and different sizes. Another hybrid approach, CSM (Lin & Chen, 2005), is a two-phases clustering algorithm which partitions a data set into several small sub-clusters in the first phase, and then merges these sub-clusters in the second phase based on a similarity measure of the inter-cluster distances in a hierarchical manner. Recently, P.Y. Mok et al. introduced a method which first obtains many clustering results from a partitional clustering algorithm, and integrates these different results as a judgement matrix. Then the algorithm find the final result with an iterative graph-partitioning process (Mok, Huang, Kwok, & Au, 2012).

Density-based clustering algorithm typically work with several concepts, e.g., *ε-neighborhood*, *pcore-point*, *directly density-reachable*, *density-reachable* and *density-connected* (Kriegel, Kröger, Sander, & Zimek, 2011). DBSCAN (Ester et al., 1996) is the first implementation of such concepts. It is insensitive to noisy data, and able to discover arbitrary-shaped clusters. However, the main difficulties of using DBSCAN are the detection of clusters with different densities and parameter determination (Cassisi, Ferro, Giugno, Pigola, & Pulvirenti, 2013). The OPTICS (Ankerst, Breunig, Kriegel, & Sander, 1999) builds an augmented ordering of data so that it is able to deal with clusters of different densities. However, the performance of OPTICS is generally 1.6 time slower than DBSCAN (Berkhin, 2006). Based on different concepts and techniques on density, many density-based clustering algorithms have been developed in the literature, as found in Hinneburg and Gabriel (2007), Ren, Liu, and Liu (2012) and Huang, Sun, Song, Deng, and Han (2013).

Out of this frame, another important technique for clustering is the self-organizing neural networks. One of the most well-known techniques is the self-organizing map (SOM) (Kohonen, 1982), also known as the Kohonen network. The standard SOM has two layers: input layer and competitive layer. Competitive layer consists of a set of units (also called neurons or nodes) with lateral connections, which is usually constructed as a two-dimensional topological structure. A weight vector is assigned to each unit and is updated during training procedures according to the simple competitive learning (SCL) (Kohonen, 2001) strategy. When the training procedure is finished, SOM divides the input space into several regions, i.e., Voronoi regions, and each best matching unit (BMU) is the site of the corresponding Voronoi region. This indicates that SOM is suitable for clustering tasks. Another powerful feature of SOM is that high-dimensional data can be projected to a low-dimensional topological structure, which means that SOM can be applied to data visualization. Unfortunately, SOM has some drawbacks, one of which is its fixed structure in its competitive layer. The size of the network must be predefined and is unchangeable during training procedures, which is similar to providing an integer number *k* as a parameter in the *k*-means clustering algorithm. In general, it is difficult to determine *k* without any prior knowledge of the given data set (Mangiameli, Chen, & West, 1996). In addition, the fixed structure limits SOM to the detection of clusters which are presented as complex shapes.

With respect to this problem, a series of "growing-type" self-organizing neural networks have been developed. These techniques usually have dynamic network structures. Growing neural