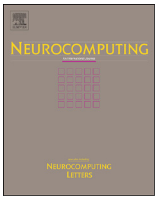
Neurocomputing 175 (2016) 473–491

Contents lists available at ScienceDirect

Neurocomputing

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



A density-based noisy graph partitioning algorithm

Jaehong Yu, Seoung Bum Kim*

Department of Industrial Management Engineering, Korea University, Anam-dong, Seoungbuk-Gu, Seoul 136-713, Republic of Korea

ARTICLE INFO

Article history:
Received 8 March 2015
Received in revised form
13 October 2015
Accepted 24 October 2015
Communicated by: Hung-Yuan Chung
Available online 9 November 2015

Keywords:
Clustering algorithm
Nonlinearity
Density coefficient
Maximizing connectivity

ABSTRACT

Clustering analysis can facilitate the extraction of implicit patterns in a dataset and elicit its natural groupings without requiring prior classification information. Numerous researchers have focused recently on graph-based clustering algorithms because their graph structure is useful in modeling the local relationships among observations. These algorithms perform reasonably well in their intended applications. However, no consensus exists about which of them best satisfies all the conditions encountered in a variety of real situations. In this study, we propose a graph-based clustering algorithm based on a novel density-of-graph structure. In the proposed algorithm, a density coefficient defined for each node is used to classify dense and sparse nodes. The main structures of clusters are identified through dense nodes and sparse nodes that are assigned to specific clusters. Experiments on various simulation datasets and benchmark datasets were conducted to examine the properties of the proposed algorithm and to compare its performance with that of existing spectral clustering and modularity-based algorithms. The experimental results demonstrated that the proposed clustering algorithm performed better than its competitors; this was especially true when the cluster structures in the data were inherently noisy and nonlinearly distributed.

1. Introduction

Modern industrial processes generate an unprecedented wealth of data that overwhelms traditional analytical approaches. Clustering analysis can facilitate the extraction of implicit patterns from these huge datasets and thus elicit their natural groupings. Clustering algorithms systematically partition the dataset by minimizing within-group variation and maximizing between-group variation [1]. Clustering analysis has been applied in various fields, such as text mining [2], image segmentation [3], bioinformatics [4], Web mining [5], and manufacturing [6].

Numerous clustering algorithms have been developed [7]. The most prominent of these are *k*-means [8], density-based spatial clustering of applications with noise (DBSCAN; [9]), and modularity-based clustering [10,11].

Although most of the existing algorithms perform reasonably well within the situations for which they were designed, no consensus exists about which is the best all-around performer in real-life situations. Most existing clustering algorithms perform poorly when the cluster structures inherent in the dataset have nonlinear patterns and different densities [12].

To address these limitations, the technique of transforming from a feature space to a graph space has been adapted to the design of clustering algorithms. By expressing the data as a graph structure, the local relationships between observations can be effectively modeled [13]. In a graph, nodes and edges express the observations and their relationships. In other words, graphs, by their topological nature, are more naturally suited to expressing certain dataset relationships and structures [13,14]. Because of these advantages, graphing techniques have been widely applied in various machine learning areas, such as manifold learning [15], semi-supervised learning [16], and clustering [17–19].

A graph-based clustering algorithm discovers the intrinsic groupings of a dataset by extracting topological information of relative adjacency among observations [20]. A number of graph-based clustering algorithms have been proposed to capitalize on the benefits described above [21]. In graph-based clustering, a subgraph can be considered as a cluster to maximize the intra-connectivity within subgraphs [22,23]. Various objective functions have been proposed to properly discover the clusters in a graph. These include cut [24], ratio cut [25], normalized cut [18,26], and conductance [27]. However, the optimization issues raised by these objective functions are hard to solve because they are non-deterministic polynomial time-hard (NP-Hard) problems [28].

To deal with this computational issue and solve the problem more efficiently, a proposed spectral clustering method eases the

* Corresponding author. Tel.: +82 2 3290 3397; fax: +82 2 929 5888.
E-mail address: sbkim1@korea.ac.kr (S.B. Kim).

http://dx.doi.org/10.1016/j.neucom.2015.10.085
0925-2312/© 2015 Elsevier B.V. All rights reserved.

optimization difficulties by adopting a spectral decomposition technique [29]. Moreover, several variants of these spectral clustering methods have been developed [30]. Among these, an algorithm proposed by Ng et al. [26] is notable. In Ng's algorithm, the normalized adjacency matrix is computed from the graph structure, and the resulting matrix is partitioned by using spectral decomposition and k-means clustering methods. This algorithm has been widely used because of its simplicity in implementation and outstanding performance in many situations [31]. However, despite its success, this algorithm has several limitations. First, the number of clusters must be determined in advance. This requirement may cause problems, especially when explicit knowledge of the data is not readily available [32]. Furthermore, spectral clustering algorithms do not work well on datasets that contain the noisy clusters common to many real situations [30,33].

In their analyses of graph clustering, many researchers in recent years have focused on modularity-based algorithms [10,34]. Modularity measures the significance of the connection between the nodes within a cluster. High modularity implies that clusters are properly constructed. Modularity is known as an effective measure for examining the adequacy of intrinsic clusters in a graph [35]. However, maximizing modularity often produces unsatisfactory results because it may lead to inadequate partitioning of nonlinear patterns, such as S-curves and Swiss roll shapes [36].

Chameleon [37] and Markov cluster (MCL; [38]) are also well-known graph-based clustering algorithms. Chameleon algorithm starts with the k-nearest neighborhood graph and partitions the graph structure into large number of small initial clusters. The initial clusters are then merged to preserve to maximize the internal self-similarities of clusters. Chameleon algorithm works well when the clusters have nonlinear patterns and the clusters have different densities [39]. However, this algorithm suffers from the curse of dimensionality in high dimensional data and requires a number of user-defined parameters [12]. MCL partitions the dataset based on the stochastic process. This algorithm constructs the transition matrix from the adjacencies between observations and expands it until this matrix converges. The final cluster is identified from the converged transition matrix [38]. MCL is widely used for graph partitioning due to its effectiveness and robustness against the noises. In addition, this algorithm does not require the number of clusters in advance. In spite of these advantages, MCL might not be suitable for identifying the nonlinear clusters because this algorithm tends to partition the large clusters [40].

In the present study, we propose a novel graph-based clustering algorithm that is especially useful for grouping data exhibiting noisy and nonlinear patterns. To achieve robustness against background noise, the proposed algorithm differentiated between dense and sparse nodes [23]. The proposed algorithm determines the main structure of each cluster in the dense regions of a graph; then the clusters are partitioned by the sparse regions in the graph. The basic concept of the proposed algorithm derives from the density-level set approach [23,41,42]. Two types of noise treatment schemes — rough cluster and exact cluster identification — can be defined in a density-level set approach [23]. In choosing between these two noise treatment schemes, we focus on exact cluster identification because of its robustness against noisy observations.

The remainder of this paper is organized as follows. Section 2 introduces our proposed clustering algorithm. Section 3 presents a simulation study to demonstrate the advantages of the proposed algorithm over the existing algorithms. Section 4 reports the results of experiments undertaken with simulated and real data to examine the properties of the proposed algorithm and to compare it with existing graph-based clustering algorithms. Section 5 contains our concluding remarks.

2. Proposed algorithm

The proposed density-based noisy graph partition (DENGP) algorithm consists of five main steps: The first is to represent the data as a mutual k-nearest neighbor graph. In this graph, all observations are represented as nodes. Second, the density of each node (called the density coefficient) is computed to determine the dense regions in the graph structure. Having calculated the density coefficients of all nodes, they are then classified either as core nodes or surrounding nodes. Those classified as surrounding nodes are temporarily excluded from the clustering procedure. Third, the core nodes are partitioned into several initial subgroups, and these subgroups are agglomerated to maximize the intra-connectivity within the cluster. In other words, those clusters that are connected with each other are hierarchically merged until no connection between them exists. In the fourth step, the temporarily excluded surrounding nodes are assigned to one of the clusters by a weighted majority voting scheme. Finally, all nodes are examined to see whether they have been properly assigned. If a node has been assigned incorrectly, it is then reassigned to the maximally connected cluster. Fig. 1 shows a graphical illustration of the proposed algorithm.

Fig. 1 shows the overall process of the proposed DENGP algorithm with an illustrative dataset containing three clusters. As shown in Fig. 1a, the original dataset is first transformed into a mutual k-nearest neighbor graph structure. The density coefficients of all nodes are then computed, and each node is classified as either a core or a surrounding node based on a given threshold value. Section 2.2 describes the detailed process to determine the appropriate threshold value. After the classification, surrounding nodes are temporarily removed from the graph. In Fig. 1b, the core and surrounding nodes are expressed as pentagrams (five-pointed stars) and diamonds, respectively. As shown in this figure, the clusters are more clearly delineated after the surrounding nodes have been eliminated. The third step groups the core nodes into several cluster structures, as shown in Fig. 1c. This figure illustrates construction from the core of three clusters with nodes that are represented as circles, triangles, and squares. In the next step, the surrounding nodes temporarily removed earlier are assigned to the appropriate cluster, as shown in Fig. 1d. Finally, several incorrectly assigned nodes are reassigned to the appropriate cluster labels. In Fig. 1e, the nodes included in the dashed regions are incorrectly assigned nodes. A more detailed explanation of each step of the proposed algorithm is presented in the following sections.

2.1. Constructing the mutual k-nearest neighbor graph

The first step of the proposed algorithm is to represent the data as a graph structure. As mentioned in Section 1, the cluster analysis of nonlinear patterns makes frequent use of representing a dataset as a neighborhood graph structure [43,44]. Several types of neighborhood graph structures exist. These include the ε-nearest neighbor graph, the symmetric k-nearest neighbor graph, and the mutual k-nearest neighbor graph [45]. Of these, the mutual k-nearest neighbor graph is sparser than other graph schemes, a feature that leads to the minimization of noise effects [16]. This makes the boundaries between clusters clearer [12]. Hence, in this study, we use the mutual k-nearest neighbor graph to group the data. The definition of the mutual k-nearest neighborhood graph is as follows:

Definition 1. : Mutual k-Nearest Neighbor Graph. A mutual k-nearest neighbor-based graph with n nodes is constructed as follows. An edge, e_{ij}, between node i and j is defined as:

$$e_{ij} = \begin{cases} 1 & \text{if } x_i \in K(j) \text{ and } x_j \in K(i) \\ 0 & \text{otherwise} \end{cases}. \tag{1}$$