Pattern Recognition 48 (2015) 3703–3713

Contents lists available at ScienceDirect

Pattern Recognition

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

Subspace clustering with automatic feature grouping

Guojun Gan [a,*], Michael Kwok-Po Ng [b]

[a] Department of Mathematics, The Institute for Systems Genomics, and The Center for Health, Intervention, and Prevention (CHIP), University of Connecticut, 196 Auditorium Rd U-3009, Storrs, CT 06269, USA
[b] Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Kowloon, Hong Kong

ARTICLE INFO

Article history:
Received 30 December 2014
Received in revised form
12 April 2015
Accepted 16 May 2015
Available online 29 May 2015

Keywords:
Data clustering
Subspace clustering
k-means
Feature group

ABSTRACT

This paper proposes a subspace clustering algorithm with automatic feature grouping for clustering high-dimensional data. In this algorithm, a new component is introduced into the objective function to capture the feature groups and a new iterative process is defined to optimize the objective function so that the features of high-dimensional data are grouped automatically. Experiments on both synthetic data and real data show that the new algorithm outperforms the FG-*k*-means algorithm in terms of accuracy and choice of parameters.

1. Introduction

As one of the major tasks of data mining, data clustering is a process that aims to identify homogeneous groups or clusters of objects from a set of objects. Given a set of multi-dimensional data points, clustering algorithms can be used to find a partition of the points into clusters such that the points within a cluster are similar to each other and the points from different clusters are quite distinct [1,2]. Data clustering can be applied to a wide range of areas such as bioinformatics [3], pattern recognition [4], health care [5], insurance [6], to just name a few.

In the past six decades, many clustering algorithms have been developed. The *k*-means algorithm is one of the oldest and most widely used clustering algorithm [7]. In the *k*-means algorithm, the number of clusters is a required input. Given a dataset and a number *k* of clusters, the *k*-means algorithm starts from *k* initial cluster centers and then repeats updating the cluster memberships and the cluster centers until some stop criterion is met [8]. A key problem of the *k*-means algorithm and other conventional clustering algorithms is that they suffer from the curse of dimensionality. In high-dimensional data, clusters are usually embedded in subspaces of the original data space and different clusters might be embedded in different subspaces. As a result, these conventional clustering algorithms are not efficient to deal with high-dimensional data.

To address this problem, subspace clustering algorithms have been developed to identify clusters embedded in subspaces of the original data space. Agrawal et al. proposed a clustering algorithm called CLIQUE to find dense subspace clusters [9]. Parsons et al. presented a review of subspace clustering algorithms developed up to that time [10]. In [11], Huang et al. proposed a subspace clustering algorithm called W-*k*-means by introducing feature weighting to the *k*-means algorithm. Gan and Wu proposed the FSC algorithm and proved its convergence [12]. In [13], Jing et al. proposed a subspace clustering algorithm named EWKM by extending the *k*-means algorithm to include weight entropy in the objective function. In [14], Domeniconi et al. proposed the LAC algorithm, which is similar to EWKM. Kriegel et al. presented a comprehensive survey of high-dimensional data clustering, including subspace clustering [15]. In [16], Deng et al. extended the EWKM algorithm to a new subspace clustering algorithm named EEW-SC by considering between-cluster separation. In [17], Favaro et al. treated the subspace clustering problem as a rank minimization problem and proposed a closed-form solution. Müller et al. studied the scalability issue of clustering high-dimensional data and proposed a density-based subspace clustering algorithm [18]. Elhamifar and Vidal presented a sparse subspace clustering (SSC) algorithm using the idea of sparse representation [19]. The correctness of the SSC algorithm was proved by Soltanolkotabi et al. [20]. Timmerman et al. proposed a subspace *k*-means algorithm by modeling the centers and cluster residuals in reduced spaces [21]. In [22], Mcwilliams and Montana proposed a predictive subspace clustering (PSC) algorithm by assuming that each cluster can be approximated well by a linear subspace estimated by a principal component analysis. Zhu et al.

[*] Corresponding author. Tel.: +1 860 486 4238; fax: +1 860 486 3919.
E-mail addresses: Guojun.Gan@uconn.edu (G. Gan), mng@math.hkbu.edu.hk (M.-P. Ng).

http://dx.doi.org/10.1016/j.patcog.2015.05.016
0031-3203/© 2015 Elsevier Ltd. All rights reserved.

proposed online subspace clustering algorithm to clustering data streams [23]. In [24], the authors proposed a subspace clustering algorithm based on affinity propagation.

The aforementioned subspace clustering algorithms can be divided into two categories: hard subspace clustering and soft subspace clustering. A hard subspace clustering algorithm determines the exact subspaces in which clusters are embedded. A soft subspace clustering algorithm assigns weights to features and identify subspaces with large weights. One major challenge of the soft subspace clustering algorithms mentioned above is that the individual feature weights are sensitive to noise and missing values. To address this problem, Chen et al. introduced the idea of assigning weights to feature groups and proposed a new subspace clustering algorithm called FG-*k*-mean [25]. The FG-*k*-means algorithm is shown to outperform the *k*-means algorithm and several other subspace clustering algorithms such as W-*k*-means [11], LAC [14], and EWKM [13].

However, the FG-*k*-means algorithm requires that the feature groups are determined before the data is clusterized. In many cases, we do not know the group information of the features that describe a high-dimensional dataset. In this paper, we propose a subspace clustering algorithm, referred to as AFG-*k*-means, that is able to determine the feature groups automatically during the clustering process. The AFG-*k*-means algorithm extends the *k*-means algorithm by incorporating automatic feature group selection.

The remaining of the paper is organized as follows. In Section 2, we review the FG-*k*-means algorithm. In Section 3, we present the new subspace clustering algorithm, i.e., the AFG-*k*-means algorithm. In Section 4, we demonstrate the performance of the AFG-*k*-means algorithm using both synthetic data and real data. Section 5 concludes the paper with some remarks.

2. Related work

In this section, we give a brief introduction to the FG-*k*-means algorithm [25]. To describe these algorithms, we let $X=\{\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_n\}$ be a dataset of n points, each of which is described by a set of m features: $A=\{A_1,A_2,\ldots,A_m\}$.

In the FG-*k*-means algorithm, the features that describe the high dimensional data are divided into feature groups, each of which is associated with a group weight. Within a feature group, each feature is also associated with a feature weight. The two types of weights are updated in the clustering process to identify important feature groups and individual features in each cluster.

Suppose that the set of features is divided into T groups $G=\{G_1,G_2,\ldots,G_T\}$ such that $G_t\neq\varnothing$, $G_t\cap G_s=\varnothing$ for $1\le t,s\le T$, $t\neq s$, and $\bigcup_{t=1}^{T}G_t=A$. To cluster X into k clusters, the FG-*k*-means algorithm minimizes the following objective function:

$$P(U,Z,V,W)=\sum_{l=1}^{k}\left[\sum_{i=1}^{n}\sum_{t=1}^{T}\sum_{j\in G_t}u_{il}w_{lt}v_{lj}d(x_{ij},z_{lj})+\lambda\sum_{t=1}^{T}w_{lt}\log(w_{lt})+\eta\sum_{j=1}^{m}v_{lj}\log(v_{lj})\right] \tag{1}$$

subject to the following conditions:

$$\sum_{l=1}^{k}u_{il}=1,\quad i=1,2,\ldots,n,\ \ u_{il}\in\{0,1\} \tag{2a}$$

$$\sum_{t=1}^{T}w_{lt}=1,\quad l=1,2,\ldots,k,\ \ w_{lt}>0 \tag{2b}$$

$$\sum_{j\in G_t}v_{lj}=1,\quad l=1,2,\ldots,k,\ t=1,2,\ldots,T,\ \ v_{lj}>0, \tag{2c}$$

where $U=(u_{il})_{n\times k}$ is a hard partition matrix, $Z=\{\mathbf{z}_1,\mathbf{z}_2,\ldots,\mathbf{z}_k\}$ is a set of k cluster centers, $V=(v_{lj})_{k\times m}$ and $W=(w_{lt})_{k\times T}$ are the two weight matrices mentioned before, λ and η are two positive parameters, and $d(x_{ij},z_{lj})$ is a distance measure between the i-th object and the center of the l-th cluster in the j-th feature. If the j-th feature is numeric, the distance measure is the square Euclidean distance. If the j-th feature is categorical, the distance measure is just the simple matching distance.

In the FG-*k*-means algorithm, the objective function given in Eq. (1) is optimized as follows. Given $Z=\hat{Z}$, $V=\hat{V}$, and $W=\hat{W}$, the hard partition matrix U that minimizes the objective function is given by

$$u_{il}=\begin{cases}1 & \text{if } D_{il}\le D_{is} \text{ for } 1\le s\le k;\\ 0 & \text{if otherwise,}\end{cases} \tag{3}$$

where $D_{is}=\sum_{t=1}^{T}\hat{w}_{st}\sum_{j\in G_t}\hat{v}_{sj}d(x_{ij},\hat{z}_{sj})$. Given $U=\hat{U}$, $V=\hat{V}$, and $W=\hat{W}$, the set Z of cluster centers that minimizes the objective function is given by

$$z_{lj}=\frac{\sum_{i=1}^{n}\hat{u}_{il}x_{ij}}{\sum_{i=1}^{n}\hat{u}_{il}}. \tag{4}$$

Given $U=\hat{U}$, $Z=\hat{Z}$, and $W=\hat{W}$, the weight matrix V that minimizes the objective function is given by

$$v_{lj}=\frac{\exp\left(-\frac{E_{lj}}{\eta}\right)}{\sum_{h\in G_t}\exp\left(-\frac{E_{lh}}{\eta}\right)}, \tag{5}$$

where $E_{lj}=\sum_{i=1}^{n}\hat{u}_{il}\hat{w}_{lt}d(x_{ij},\hat{z}_{lj})$ with t being the index of the feature group to which the j-th feature is assigned, i.e., $A_j\in G_t$. Given $U=\hat{U}$, $Z=\hat{Z}$, and $V=\hat{V}$, the weight matrix W that minimizes the objective function is given by

$$w_{lt}=\frac{\exp\left(-\frac{F_{lt}}{\lambda}\right)}{\sum_{s=1}^{T}\exp\left(-\frac{F_{ls}}{\lambda}\right)}, \tag{6}$$

where $F_{lt}=\sum_{i=1}^{n}\hat{u}_{il}\sum_{j\in G_t}\hat{v}_{lj}d(x_{ij},\hat{z}_{lj})$.

Note that in the FG-*k*-means algorithm, the feature group G is given as an input. The feature group weights are automatically calculated by the algorithm.

3. The AFG-*k*-means algorithm

In this section, we present the AFG-*k*-means algorithm that incorporates automatic feature grouping in the clustering process. To describe the algorithm, we let $X=\{\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_n\}$ be a dataset of n points, each of which is described by a set of m numerical features: $A=\{A_1,A_2,\ldots,A_m\}$. Let k be the desired number of clusters and let T be the desired number of feature groups.

The objective function of the AFG-*k*-means algorithm is defined as

$$Q(U,Z,W,G,V,\Gamma)=\sum_{l=1}^{k}\sum_{i=1}^{n}u_{il}\sum_{j=1}^{m}w_{lj}^2(x_{ij}-z_{lj})^2+\epsilon_1\sum_{l=1}^{k}\sum_{j=1}^{m}w_{lj}^2+\beta\left(\sum_{t=1}^{T}\sum_{j=1}^{m}g_{jt}\sum_{l=1}^{k}\gamma_{lt}^2(w_{lj}-v_{lt})^2+\epsilon_2\sum_{l=1}^{k}\sum_{t=1}^{T}\gamma_{lt}^2\right), \tag{7}$$

where $U=(u_{il})_{n\times k}$ is an $n\times k$ matrix of binary numbers. If point $\mathbf{x}_i$ belongs to the l-th cluster, then $u_{il}=1$; otherwise, $u_{il}=0$. $Z=\{\mathbf{z}_1,\mathbf{z}_2,\ldots,\mathbf{z}_k\}$ is a set of k cluster centers. $W=(w_{lj})_{k\times m}$ is a $k\times$