Pattern Recognition 47 (2014) 3082–3095

Contents lists available at ScienceDirect

Pattern Recognition

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

Kernel-based hard clustering methods in the feature space with automatic variable weighting

Marcelo R.P. Ferreira [a], Francisco de A.T. de Carvalho [b,*]

[a] Departamento de Estatística, Centro de Ciências Exatas e da Natureza, Universidade Federal da Paraíba, CEP 58051-900 João Pessoa (PB), Brazil
[b] Centro de Informática, Universidade Federal de Pernambuco, Av. Jornalista Anibal Fernandes, s/n - Cidade Universitária, CEP 50740-560 Recife (PE), Brazil

ARTICLE INFO

Article history:
Received 25 March 2013
Received in revised form
31 January 2014
Accepted 26 March 2014
Available online 4 April 2014

Keywords:
Kernel clustering
Feature space
Adaptive distances
Clustering analysis

ABSTRACT

This paper presents variable-wise kernel hard clustering algorithms in the feature space in which dissimilarity measures are obtained as sums of squared distances between patterns and centroids computed individually for each variable by means of kernels. The methods proposed in this paper are supported by the fact that a kernel function can be written as a sum of kernel functions evaluated on each variable separately. The main advantage of this approach is that it allows the use of adaptive distances, which are suitable to learn the weights of the variables on each cluster, providing a better performance. Moreover, various partition and cluster interpretation tools are introduced. Experiments with synthetic and benchmark datasets show the usefulness of the proposed algorithms and the merit of the partition and cluster interpretation tools.

1. Introduction

Clustering is one of the most useful tools to explore data structures and has been widely applied in various areas, including taxonomy, image processing, data mining, and information retrieval. Clustering means the task of organizing a set of patterns into clusters such that patterns within a given cluster have a high degree of similarity, whereas patterns belonging to different clusters have a high degree of dissimilarity [17,25].

An important component of a clustering algorithm is the dissimilarity (or similarity) measure. Distance measures are important examples of dissimilarity measures and the Euclidean distance is most commonly used in conventional partitioning (hard and fuzzy) clustering algorithms, which perform well with datasets in which natural clusters are nearly hyper-spherical and linearly separable. However, when the data structure is complex (i.e., clusters with non-hyperspherical shapes and/or linearly non-separable patterns), these algorithms may have poor performance. Because of this limitation, several methods that are able to handle complex data have been proposed, among them, the kernel-based clustering methods.

With the development of the kernel *K*-means algorithm [16], several clustering methods such as fuzzy *c*-means [3], self-organizing maps (SOM) [29,30], the mountain method [46] and neural gas [33] have been modified to incorporate kernels and a variety of kernel methods for clustering have been proposed [14]. Two main approaches have guided such modifications: kernelization of the metric, where the centroids are obtained in the original space and the distances between patterns and centroids are computed by means of kernels, and clustering in the feature space, in which centroids are obtained in the feature space. Important hard clustering algorithms based on kernels were developed in Refs. [5,13,20]. Kernel-based fuzzy clustering methods have been proposed in Refs. [10,44,49]. The authors of Refs. [23,32] developed a kernelized version of SOM. In [28] a kernel mountain method was presented and in [39] a kernel version of neural gas algorithm was proposed. A semi-supervised kernel-based clustering method with metric learning was proposed in Ref. [47]. Moreover, various studies have demonstrated that the kernel clustering methods outperform the conventional clustering approaches when the data have a complex structure, because these algorithms may produce nonlinear separating hypersurfaces among clusters [5,14,19,27].

In clustering analysis the patterns to be clustered are usually represented as vectors where each component is a measurement of a variable. Conventional clustering algorithms, such as *K*-means, fuzzy *c*-means and SOM, and their kernelized counterparts consider that all variables are equally important in the sense that all have the same weight in the construction of the clusters. Nevertheless, in most areas, especially if we are dealing with high-dimensional datasets, some variables may be irrelevant and, among the relevant ones, some may be more or less important than others to the clustering procedure. Moreover, the contribution of each variable to each cluster may be different, i.e., each

* Corresponding author. Tel.: +55 81 21268430; fax: +55 81 21268438.
E-mail addresses: marcelo@de.ufpb.br (M.R.P. Ferreira), fatc@cin.ufpe.br (F.d.A.T. de Carvalho).

http://dx.doi.org/10.1016/j.patcog.2014.03.026
0031-3203/

cluster may have a different set of important variables. A number of modifications of the *K*-means algorithm have been proposed in the literature to automatically learn the weights of the variables and improve the performance of the *K*-means algorithm [1,21,26,31,43].

In this paper we propose variable-wise kernel hard clustering methods in the feature space where dissimilarity measures are obtained as sums of squared distances between patterns and centroids computed individually for each variable by means of kernel functions. The main advantage of the proposed approach over the conventional kernel-based clustering methods is that it allows us to use adaptive distances which change at each algorithm iteration and can be different from one cluster to another. This kind of dissimilarity measure is suitable to learn the weights of the variables during the clustering process, improving the performance of the algorithms. The derivation of the expressions of the weights of the variables was done considering two cases: one assumes that the sum of the weights of the variables on each cluster must be equal to one, whereas the other assumes that the product of the weights of the variables on each cluster must be equal to one [12]. Another advantage of this approach is that it allows the introduction of various partition and cluster interpretation tools.

The remainder of the paper is organized as follows. In Section 2 a brief review about kernels is presented and the conventional kernel-based hard clustering algorithm in the feature space is described. Section 3 introduces variable-wise kernel hard clustering methods in the feature space based on adaptive distances. In Section 4 we introduce suitable dispersion measures in which the tools for the interpretation of the partition and the clusters are based: indexes for evaluating the overall quality of a partition, the homogeneity of the individual clusters, as well as the role of the different variables in the cluster formation process. In Section 5 we demonstrate the effectiveness of the proposed methods through experiments with synthetic and benchmark datasets. Finally, a summary is given to conclude the paper in Section 6.

2. Conventional kernel-based hard clustering method in the feature space

Recently, a number of researchers have shown interest in kernel clustering methods [14]. The main idea behind these methods is the use of a non-linear mapping Φ from the input space to a high dimensional (possibly infinite) space, called the feature space.

In this section we briefly recall the basic theory about kernel functions and the conventional kernel clustering algorithm in the feature space. Let $X=\{\mathbf{x}_1,\ldots,\mathbf{x}_n\}$ be a non-empty set where $\mathbf{x}_i\in\mathbb{R}^p$. A function $\mathcal{K}: X\times X\to\mathbb{R}$ is called a *positive definite kernel* (or *Mercer kernel*) if and only if $\mathcal{K}$ is symmetric (i.e., $\mathcal{K}(\mathbf{x}_i,\mathbf{x}_k)=\mathcal{K}(\mathbf{x}_k,\mathbf{x}_i)$) and the inequality $\sum_{i=1}^{n}\sum_{k=1}^{n}c_i c_k\mathcal{K}(\mathbf{x}_i,\mathbf{x}_k)\geq 0\ \forall n\geq 2$ holds, where $c_r\in\mathbb{R}\ \forall r=1,\ldots,n$ [34].

Let $\Phi: X\to\mathcal{F}$ be a non-linear mapping from the input space X to a high dimensional feature space $\mathcal{F}$. By applying the mapping Φ, the dot product $\mathbf{x}_i^\top\mathbf{x}_k$ in the input space is mapped to $\Phi(\mathbf{x}_i)^\top\Phi(\mathbf{x}_k)$ in the feature space. The key idea in kernel algorithms is that the non-linear mapping Φ does not need to be explicitly specified because each Mercer kernel can be expressed as $\mathcal{K}(\mathbf{x}_i,\mathbf{x}_k)=\Phi(\mathbf{x}_i)^\top\Phi(\mathbf{x}_k)$ [37,41].

One of the most relevant aspects in applications is that it is possible to compute Euclidean distances in $\mathcal{F}$ without explicitly knowing Φ. This can be done using the so-called *distance kernel trick* [14,19,37,41]:

$$\begin{aligned}\|\Phi(\mathbf{x}_i)-\Phi(\mathbf{x}_k)\|^2&=(\Phi(\mathbf{x}_i)-\Phi(\mathbf{x}_k))^\top(\Phi(\mathbf{x}_i)-\Phi(\mathbf{x}_k))\\&=\Phi(\mathbf{x}_i)^\top\Phi(\mathbf{x}_i)-2\Phi(\mathbf{x}_i)^\top\Phi(\mathbf{x}_k)+\Phi(\mathbf{x}_k)^\top\Phi(\mathbf{x}_k)\\&=\mathcal{K}(\mathbf{x}_i,\mathbf{x}_i)-2\mathcal{K}(\mathbf{x}_i,\mathbf{x}_k)+\mathcal{K}(\mathbf{x}_k,\mathbf{x}_k).\end{aligned}$$

Let $\mathbf{K}$ be an $n\times n$ matrix called kernel matrix where each element $\kappa_{il}=\mathcal{K}(\mathbf{x}_i,\mathbf{x}_l)$, $i=1,\ldots,n$, $l=1,\ldots,n$ [38]. Examples of commonly used kernel functions are the Gaussian, given by $\mathcal{K}(\mathbf{x}_i,\mathbf{x}_k)=e^{-\|\mathbf{x}_i-\mathbf{x}_k\|^2/2\sigma^2}$, $\sigma>0$ [45], and the Polynomial of degree d, given by $\mathcal{K}(\mathbf{x}_i,\mathbf{x}_k)=(\gamma\mathbf{x}_i^\top\mathbf{x}_k+\theta)^d$, $\gamma>0$, $\theta\geq 0$, $d\in\mathbb{N}$.

There are two major variations of kernel clustering methods which are based, respectively, on kernelization of the metric, in which the clustering algorithms seek for centroids in the input space and the distances between patterns and centroids are obtained by means of kernels; and clustering in the feature space that proceeds by mapping each pattern by means of a non-linear function Φ and then obtaining the centroids in the feature space. Let $\mathbf{v}_k^\Phi$ be the kth cluster centroid in the feature space. It is possible to obtain $\|\Phi(\mathbf{x}_i)-\mathbf{v}_k^\Phi\|^2$ without the need for calculating $\mathbf{v}_k^\Phi$ by means of the kernel trick.

2.1. Kernel K-means in the feature space

The kernel *K*-means algorithm in the feature space (here labeled KCM-F) iteratively searches for K cluster centroids by minimizing the following objective function [11,14,18,49]:

$$J=\sum_{k=1}^{K}\sum_{i\in P_k}\|\Phi(\mathbf{x}_i)-\mathbf{v}_k^\Phi\|^2, \tag{1}$$

where $\mathbf{v}_k^\Phi$ is the kth cluster centroid in the feature space.

Optimization of the criterion given in (1) with respect to $\mathbf{v}_k^\Phi$ provides the following expression for the cluster centroids in the feature space [11,14,18,49]:

$$\mathbf{v}_k^\Phi=\frac{1}{|P_k|}\sum_{i\in P_k}\Phi(\mathbf{x}_i),\quad k=1,\ldots,K. \tag{2}$$

The non-linear mapping Φ is not known explicitly, so the cluster centroid in feature space $\mathbf{v}_k^\Phi$ ($k=1,\ldots,K$) cannot be obtained directly. The distance between $\Phi(\mathbf{x}_i)$ and $\mathbf{v}_k^\Phi$ in the feature space is calculated through the kernel in the original space:

$$\begin{aligned}&\|\Phi(\mathbf{x}_i)-\mathbf{v}_k^\Phi\|^2\\&\quad=\Phi(\mathbf{x}_i)^\top\Phi(\mathbf{x}_i)-2\Phi(\mathbf{x}_i)^\top(\mathbf{v}_k^\Phi)+(\mathbf{v}_k^\Phi)^\top(\mathbf{v}_k^\Phi)\\&\quad=\Phi(\mathbf{x}_i)^\top\Phi(\mathbf{x}_i)-\frac{2\sum_{l\in P_k}\Phi(\mathbf{x}_l)^\top\Phi(\mathbf{x}_i)}{|P_k|}\\&\qquad+\frac{\sum_{r\in P_k}\sum_{s\in P_k}\Phi(\mathbf{x}_r)^\top\Phi(\mathbf{x}_s)}{|P_k|^2}\\&\quad=\mathcal{K}(\mathbf{x}_i,\mathbf{x}_i)-\frac{2\sum_{l\in P_k}\mathcal{K}(\mathbf{x}_l,\mathbf{x}_i)}{|P_k|}\\&\qquad+\frac{\sum_{r\in P_k}\sum_{s\in P_k}\mathcal{K}(\mathbf{x}_r,\mathbf{x}_s)}{|P_k|^2}.\end{aligned} \tag{3}$$

Additionally, the criterion J given in Eq. (1) can be rewritten as

$$J=\sum_{k=1}^{K}\sum_{i\in P_k}\left\{\mathcal{K}(\mathbf{x}_i,\mathbf{x}_i)-\frac{2\sum_{l\in P_k}\mathcal{K}(\mathbf{x}_l,\mathbf{x}_i)}{|P_k|}+\frac{\sum_{r\in P_k}\sum_{s\in P_k}\mathcal{K}(\mathbf{x}_r,\mathbf{x}_s)}{|P_k|^2}\right\}. \tag{4}$$

The KCM-F algorithm lacks the step in which cluster centroids are updated. The updating of the partition can be done without calculating the centroids due to the implicit mapping via the kernel function in Eq. (3).

The clusters P_k ($k=1,\ldots,K$), which minimize the clustering criterion J given in Eq. (1), are updated according to the following