2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

A Kernel K-means Clustering Algorithm Based on an Adaptive Mahalanobis Kernel

Marcelo R.P. Ferreira and Francisco de A.T. de Carvalho

***Abstract*— In this paper, a kernel K-means algorithm based on an adaptive Mahalanobis kernel is proposed. This kernel is built based on an adaptive quadratic distance defined by a symmetric positive definite matrix that changes at each algorithm iteration and takes into account the correlations between variables, allowing the discovery of clusters with non-hyperspherical shapes. The effectiveness of the proposed algorithm is demonstrated through experiments with synthetic and benchmark datasets.**

I. Introduction

CLUSTERING is an excellent state-of-the-art tool to knowledge discovering, and it is applied in a wide variety of fields including taxonomy, data mining, pattern recognition, computer vision, information retrieval, etc. 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. The most popular clustering algorithms are hierarchical and partitioning methods [1], [2], [3], [4]. Hierarchical methods delivers an output represented by a hierarchical structure of groups known as *dendrogram*, i.e., a nested sequence of partitions of the input data, whereas partitioning methods aims to obtain a single partition of the input data in a fixed number of clusters, typically by optimizing (usually locally) an objective function; the result is a creation of separation hypersurfaces among clusters. Partitioning clustering methods were performed in two different ways: hard and fuzzy. In hard clustering, the clusters are disjoint and non-overlapping in nature. In this case, any pattern may belong to one and only one cluster. On the other hand, in fuzzy clustering a pattern may belong to all clusters with a certain fuzzy membership degree. A good review of the main fuzzy clustering algorithms can be found in [5]. Moreover, a survey of the various clustering methods can be found, for example, in [2], [4].

Several clustering methods have been modified to incorporate kernels and a variety of kernel methods to clustering have been proposed [6]. The core of kernel-based algorithms is the kernel function. It measures the similarity between two patterns in a p-dimensional space. Over the different kernels used in clustering algorithms, the Gaussian kernel is the most commonly used [6], [7]. The Gaussian kernel usually gives good results and has only one parameter to be tuned. Despite the good characteristics of this kernel, it is based on the Euclidean distance, that is, algorithms based on the Gaussian kernel assume that patterns are more likely distributed within an hyperspherical region (in other words, each variable has the same variance and there is no covariance between variables). However, patterns in two different groups are more likely distributed within two different hyperellipsoidal regions, respectively. The Mahalanobis distance, which takes into account the correlations between variables and is scale-invariant, is a better choice to deal with hyperellipsoidal regions. Support vector machines (SVMs) have been modified through the use of the Mahalanobis kernel [8], [9], [10], [11].

In this paper, we propose, under the kernelization of the metric approach, a kernel K-means algorithm based on an adaptive Mahalanobis kernel. This kernel is built based on an adaptive quadratic distance defined by a symmetric positive definite matrix that changes at each iteration of the algorithm. The adaptive Mahalanobis kernel takes into account the correlations between variables, allowing the discovery of clusters with non-hyperspherical shapes. The effectiveness of the proposed clustering algorithm is demonstrated through experiments with synthetic and benchmark datasets.

The rest of the paper is organized as follows: Section II briefly reviews the theory about kernels and also presents the standard kernel K-means with kernelization of the metric considering the well known Gaussian kernel. Section III introduces the kernel K-means with kernelization of the metric based on an adaptive Mahalanobis kernel. Section IV deals with the experimental results and Section V concludes this paper.

II. Related work

With the introduction of the kernel K-means algorithm [12], several clustering methods such as fuzzy c-means [13], self organizing maps (SOM) [14], [15], the mountain method [16] and neural gas [17] have been modified to incorporate kernels and a variety of kernel methods to clustering have been proposed [6]. 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 feature space, in which centroids are obtained in the feature space. Important hard clustering algorithms based on kernels were developed in Refs. [18], [19], [20]. Kernel-based fuzzy clustering methods

Marcelo R.P. Ferreira is with the Department of Statistics, Federal University of Paraíba, João Pessoa, Paraíba, Brazil (email: marcelo@de.ufpb.br).
Francisco de A.T. de Carvalho is with the Center of Informatics, Federal University of Pernambuco, Recife, Pernambuco, Brazil (email: fatc@cin.ufpe.br).

The authors would like to thank CAPES (Brazilian Agency) for its financial support and the anonymous referees for their helpful comments and suggestions to improve the paper.

978-1-4799-1484-5/14/$31.00 ©2014 IEEE

have been proposed in Refs. [21], [22], [23]. The authors of Refs. [24], [25] developed a kernelized version of SOM. In [26] a kernel mountain method was presented and in [27] a kernel version of neural gas algorithm was proposed. A semi-supervised kernel-based clustering method with metric learning was proposed in Ref. [28]. Moreover, various studies have demonstrated that the kernel clustering methods outperforms the conventional clustering approaches when the data have a complex structure, because these algorithms may produce non-linear separating hypersurfaces among clusters [18], [6], [7], [29].

Since the beginning of the last decade a number of researchers have shown interest in kernel-based clustering methods [6]. The main idea supporting these methods is the use of a non-linear mapping Φ from the input space to a high dimensional (possibly infinite) space, called feature space.

In this section we briefly recall the basic theory about kernel functions and the conventional kernel clustering methods. Let $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ be a set of n patterns indexed by i and described by p real-valued variables, i.e., $\mathbf{x}_i \in \mathbb{R}^p$. 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 following inequality holds [30]:

$$\sum_{i=1}^{n}\sum_{k=1}^{n} c_i c_k \mathcal{K}(\mathbf{x}_i, \mathbf{x}_k) \geq 0 \quad \forall n \geq 2, \tag{1}$$

where $c_r \in \mathbb{R}\ \forall r = 1, \ldots, n$.

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 inner 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 doesn't need to be explicitly specified because each Mercer kernel can be expressed as $\mathcal{K}(\mathbf{x}_i, \mathbf{x}_k) = \Phi(\mathbf{x}_i)^\top \Phi(\mathbf{x}_k)$, that is usually referred to as kernel trick [31], [32].

Because of the kernel trick, it is possible to compute Euclidean distances in $\mathcal{F}$ as follows [31], [32]:

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

The most commonly used kernel in the literature is the Gaussian kernel, which is given by

$$\mathcal{K}^{(g)}(\mathbf{x}_i, \mathbf{x}_k) = \exp\left\{-\frac{(\mathbf{x}_i - \mathbf{x}_k)^\top (\mathbf{x}_i - \mathbf{x}_k)}{2\sigma^2}\right\}, \tag{3}$$

where $\sigma > 0$ is a kernel hyperparameter that, roughly speaking, controls how two patterns are considered as close or similar in $\mathbb{R}^p$.

There are two major variations of kernel clustering methods which are based, respectively, on: kernelization of the metric, and clustering in the feature space. Clustering algorithms based on kernelization of the metric seeks for cluster centroids in the input space and the distance between a pattern $\mathbf{x}_i$ and a cluster centroid $\mathbf{y}_k$ is obtained by means of kernels: $||\Phi(\mathbf{x}_i) - \Phi(\mathbf{y}_k)||^2 = \mathcal{K}(\mathbf{x}_i, \mathbf{x}_i) - 2\mathcal{K}(\mathbf{x}_i, \mathbf{y}_k) + \mathcal{K}(\mathbf{y}_k, \mathbf{y}_k)$. On the other hand, clustering algorithms in the feature space proceed by mapping each pattern by means of a non-linear function Φ and then obtain the centroids in the feature space. In this paper, only the kernelization of the metric approach will be considered.

A. The kernel K-means algorithm

The kernel K-means under the kernelization of the metric approach (here labeled KKMeans) is an iterative two-steps algorithm (representation and allocation steps) that gives a partition $P = \{P_1, \ldots, P_K\}$ of X into K clusters and their corresponding cluster centroids $\mathbf{y}_k \in \mathbb{R}^p$ $(k = 1, \ldots, K)$ such that is minimized the following objective function [33], [34], [6]:

$$\begin{aligned} W &= \sum_{k=1}^{K}\sum_{\mathbf{x}_i \in P_k} ||\Phi(\mathbf{x}_i) - \Phi(\mathbf{y}_k)||^2 \\ &= \sum_{k=1}^{K}\sum_{\mathbf{x}_i \in P_k} \left\{\mathcal{K}(\mathbf{x}_i, \mathbf{x}_i) - 2\mathcal{K}(\mathbf{x}_i, \mathbf{y}_k) + \mathcal{K}(\mathbf{y}_k, \mathbf{y}_k)\right\}. \end{aligned} \tag{4}$$

During the representation step, the partition if kept fixed. The derivation of the cluster centroids depends on the specific selection of the kernel function. If we consider the Gaussian kernel, then $\mathcal{K}^{(g)}(\mathbf{x}_i, \mathbf{x}_i) = 1\ \forall i$, and the functional (4) can be expressed as [7]:

$$\begin{aligned} W &= 2\sum_{k=1}^{K}\sum_{\mathbf{x}_i \in P_k} (1 - \mathcal{K}^{(g)}(\mathbf{x}_i, \mathbf{y}_k)) \\ &= 2\sum_{k=1}^{K}\sum_{\mathbf{x}_i \in P_k} \left(1 - \exp\left\{-\frac{(\mathbf{x}_i - \mathbf{x}_k)^\top (\mathbf{x}_i - \mathbf{x}_k)}{2\sigma^2}\right\}\right). \end{aligned} \tag{5}$$

It can be shown that the cluster centroids are obtained as:

$$\mathbf{y}_k = \frac{\sum_{\mathbf{x}_i \in P_k} \mathcal{K}^{(g)}(\mathbf{x}_i, \mathbf{y}_k)\mathbf{x}_i}{\sum_{\mathbf{x}_i \in P_k} \mathcal{K}^{(g)}(\mathbf{x}_i, \mathbf{y}_k)}, \quad k = 1, \ldots, K. \tag{6}$$

In the allocation step, the cluster centroids $\mathbf{y}_k$ $(k = 1, \ldots, K)$ are kept fixed. The clusters P_k $(k = 1, \ldots, K)$, which minimizes the clustering criterion W are updated according to the following allocation rule:

$$\begin{aligned} P_k = \big\{&\mathbf{x}_i \in X : ||\Phi(\mathbf{x}_i) - \Phi(\mathbf{y}_k)||^2 \leq ||\Phi(\mathbf{x}_i) - \Phi(\mathbf{y}_h)||^2, \\ &\forall h \neq k, h = 1, \ldots, K\big\}. \end{aligned} \tag{7}$$

The KKMeans algorithm is executed in the following steps:

(1) **Initialization**
Fix K (the number of clusters), $2 \leq K < n$; randomly choose a initial partition P of X into K clusters $P_1, \ldots, P_K$ or choose K distinct patterns $\mathbf{y}_1, \ldots, \mathbf{y}_K$ belonging to X as the initial centroids

and assign each pattern i to the closest centroid $\mathbf{y}_h$ ($h = \arg\min_{1\leq k\leq K} ||\Phi(\mathbf{x}_i) - \Phi(\mathbf{y}_k)||^2$) to construct the initial partition P of X into K clusters $P_1, \ldots, P_K$.

(2) **Representation step**
Compute the cluster centroids $\mathbf{y}_k$ ($k = 1, \ldots, K$) according to Eq. (6).

(3) **Alocation step**
Compute the best partition
$test \leftarrow 0$
for $i = 1$ to n do
  compute the winning cluster P_h such that
    $h = \arg\min_{1\leq k\leq K} ||\Phi(\mathbf{x}_i) - \Phi(\mathbf{y}_k)||^2$
  if $i \in P_k$ and $h \neq k$
    $test \leftarrow 1$
    $P_h \leftarrow P_h \cup \{\mathbf{x}_i\}$
    $P_k \leftarrow P_k \setminus \{\mathbf{x}_i\}$

(4) **Stopping criterion**
If $test = 0$ then STOP, otherwise go to (2).

The computational complexity of the KKMeans algorithm for a single iteration is $O(nKp)$ [7], where n is number of patterns, K is the number of clusters and p is the number of variables.

III. Kernel K-means based on an adaptive Mahalanobis kernel

The most commonly used kernel function in the literature is the Gaussian kernel. In general, this kernel delivers good results and requires tuning only for one parameter. The Gaussian kernel is based on the Euclidean distance between two patterns $\mathbf{x}_i$ and $\mathbf{x}_k$ in the input space $\mathbb{R}^p$ and it is known that the Euclidean distance performs well with data sets in which natural clusters are nearly hyperspherical, which means that each variable has the same variance and there is no covariance between variables. However, patterns in two different groups are more likely distributed within two different hyper-ellipsoidal regions, respectively. The Mahalanobis distance, which takes into account the correlations between variables and is scale-invariant, is a better choice to deal with clusters with hyper-ellipsoidal shapes, which arise in a number of practical and experimental situations.

The Mahalanobis distance between a pattern $\mathbf{x}_i$ and the overall centroid $\mathbf{y}$ in the input space $\mathbb{R}^p$ is given by

$$d^2_{\mathbf{\Sigma}^{-1}}(\mathbf{x}_i, \mathbf{y}) = (\mathbf{x}_i - \mathbf{y})^\top \mathbf{\Sigma}^{-1} (\mathbf{x}_i - \mathbf{y}), \tag{8}$$

in which

$$y = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i,$$

and

$$\mathbf{\Sigma} = \frac{1}{n}\sum_{i=1}^{n} (\mathbf{x}_i - \mathbf{y})(\mathbf{x}_i - \mathbf{y})^\top.$$

Inspired by the definition of the Mahalanobis distance, we can define an adaptive Mahalanobis kernel by substituting the Euclidean distance with a Mahalanobis-type distance on the Gaussian kernel [8] :

$$\mathcal{K}^{(m)}(\mathbf{x}_i, \mathbf{x}_k) = \exp\left\{-\frac{(\mathbf{x}_i - \mathbf{x}_k)^\top \mathbf{M} (\mathbf{x}_i - \mathbf{x}_k)}{2\sigma^2}\right\}, \tag{9}$$

where $d^2_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_k) = (\mathbf{x}_i - \mathbf{x}_k)^\top \mathbf{M} (\mathbf{x}_i - \mathbf{x}_k)$ is a quadratic distance defined by a symmetric positive defined matrix $\mathbf{M}$. Here, the Mahalanobis distance is computed between two patterns $\mathbf{x}_i$ and $\mathbf{x}_k$ in the input space $\mathbb{R}^p$ instead of between a pattern $\mathbf{x}_i$ and the overall centroid $\mathbf{y}$. The Mahalanobis kernel is an extension of the Gaussian kernel. Namely, by setting $\mathbf{M} = \mathbf{I}$, where $\mathbf{I}$ is the $p \times p$ identity matrix, we obtain the Gaussian kernel.

The kernel K-means based on an adaptive Mahalanobis kernel (here labeled AMKKMeans) is an iterative two-steps algorithm (representation and allocation steps) that gives a partition $P = \{P_1, \ldots, P_K\}$ of X into K clusters and their corresponding cluster centroids $\mathbf{y}_k \in \mathbb{R}^p$ ($k = 1, \ldots, K$) such that is minimized the criterion J measuring the fitting between the clusters and their centroids, which is defined as

$$\begin{aligned} J &= 2\sum_{k=1}^{K}\sum_{\mathbf{x}_i \in P_k} (1 - \mathcal{K}^{(m)}(\mathbf{x}_i, \mathbf{y}_k)) \\ &= 2\sum_{k=1}^{K}\sum_{\mathbf{x}_i \in P_k} \left(1 - \exp\left\{-\frac{(\mathbf{x}_i - \mathbf{y}_k)^\top \mathbf{M} (\mathbf{x}_i - \mathbf{y}_k)}{2\sigma^2}\right\}\right). \end{aligned} \tag{10}$$

The representation step has now two stages. In the first stage, the partition and the matrix $\mathbf{M}$ are kept fixed. As in the case of the Gaussian kernel, it can be shown that, also considering the adaptive Mahalanobis kernel, the cluster centroids are obtained as:

$$\mathbf{y}_k = \frac{\sum_{\mathbf{x}_i \in P_k} \mathcal{K}^{(m)}(\mathbf{x}_i, \mathbf{y}_k)\mathbf{x}_i}{\sum_{\mathbf{x}_i \in P_k} \mathcal{K}^{(m)}(\mathbf{x}_i, \mathbf{y}_k)}, \quad k = 1, \ldots, K. \tag{11}$$

In the second stage, the partition and the prototypes $\mathbf{y}_k$ of the corresponding clusters P_k ($k = 1, \ldots, K$) are kept fixed. It can be shown that the symmetric positive definite matrix $\mathbf{M}$, which minimizes the clustering criterion J under $\det(\mathbf{M}) = 1$, is updated according to the following expression:

$$\begin{aligned} &\mathbf{M} = [\det(\mathbf{Q})]^{\frac{1}{p}} \mathbf{Q}^{-1}, \text{ in which } \mathbf{Q} = \sum_{k=1}^{K} \mathbf{Q}_k \text{ and} \\ &\mathbf{Q}_k = \sum_{\mathbf{x}_i \in P_k} \mathcal{K}^{(m)}(\mathbf{x}_i, \mathbf{y}_k)(\mathbf{x}_i - \mathbf{y}_k)(\mathbf{x}_i - \mathbf{y}_k)^\top. \end{aligned} \tag{12}$$

The proof of Eq. (12) can be achieved through the Lagrange multiplyers method considering the constraint that $\det(\mathbf{M}) = 1$.

In the allocation step, the cluster centroids $\mathbf{y}_k$ ($k = 1, \ldots, K$) and the matrix $\mathbf{M}$ are kept fixed. The clusters P_k ($k = 1, \ldots, K$), which minimizes the clustering criterion J

are updated according to the following allocation rule:

$$P_k = \left\{ \mathbf{x}_i \in X : ||\Phi(\mathbf{x}_i) - \Phi(\mathbf{y}_k)||^2 \leq ||\Phi(\mathbf{x}_i) - \Phi(\mathbf{y}_h)||^2, \right. \\ \left. \forall h \neq k, h = 1, \ldots, K \right\}. \tag{13}$$

The AMKKMeans algorithm is executed in the following steps:

(1) **Initialization**
Fix K (the number of clusters), $2 \leq K < n$; randomly choose a initial partition P of X into K clusters $P_1, \ldots, P_K$ or choose K distinct patterns $\mathbf{y}_1, \ldots, \mathbf{y}_K$ belonging to X as the initial centroids and assign each pattern i to the closest centroid $\mathbf{y}_h$ ($h = \arg\min_{1 \leq k \leq K} ||\Phi(\mathbf{x}_i) - \Phi(\mathbf{y}_k)||^2$) to construct the initial partition P of X into K clusters $P_1, \ldots, P_K$.

(2) **Representation step**
   (a) Stage 1: Compute the cluster centroids $\mathbf{y}_k$ ($k = 1, \ldots, K$) according to Eq. (11).
   (b) Stage 2: Compute the matrix $\mathbf{M}$ according to Eq. (12).

(3) **Allocation step**
Compute the best partition
$test \leftarrow 0$
for $i = 1$ to n do
  compute the winning cluster P_h such that
    $h = \arg\min_{1 \leq k \leq K} ||\Phi(\mathbf{x}_i) - \Phi(\mathbf{y}_k)||^2$
  if $i \in P_k$ and $h \neq k$
    $test \leftarrow 1$
    $P_h \leftarrow P_h \cup \{\mathbf{x}_i\}$
    $P_k \leftarrow P_k \setminus \{\mathbf{x}_i\}$

(4) **Stopping criterion**
If $test = 0$ then STOP, otherwise go to (2).

For a single iteration the complexity of the AMKKMeans algorithm for computing the cluster centroids is $O(nKp)$ and the complexity for computing the partition is $O(nKp^2)$. The complexity of computing M depends on the method of matrix inversion used in the implementation of the clustering algorithm.

IV. Empirical results

To show the usefulness of the adaptive Mahalanobis kernel K-means algorithm in comparison with the kernel K-means algorithm with the usual Gaussian kernel, two configurations of synthetic quantitative datasets in $\mathbb{R}^2$, and four benchmark datasets selected from the UCI Machine Learning Repository[1] [35] are considered. The standard K-means algorithm (here labeled KMeans) is also considered.

To assess the performance of the different clustering algorithms and compare them, we assume that the ground truth (the *a priori* partition) is known and we use two external evaluation measures: the corrected Rand index (CR) [36], as well as the overall error rate of classification (OERC) [37].

The CR index takes its values from the interval $[1, 1]$, in which the value 1 indicates perfect agreement between partitions, whereas values near 0 (or negatives) correspond to cluster agreement found by chance [38]. The OERC index aims to measure the ability of a clustering algorithm to find out a priori classes present in a dataset and takes its values from the interval $[0, 1]$ in which lower OERC values indicate better clustering results.

[1] `http://archive.ics.uci.edu/ml/`

A. Synthetic datasets

Each synthetic dataset was simulated considering classes with different sizes and shapes. The synthetic datasets have 500 points each, divided into four classes with sizes 200, 150, 50 and 100, respectively. Each class in these data were drawn according to a bivariate normal distribution with mean vector and covariance matrix represented by

$$\boldsymbol{\mu} = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} \text{ and } \boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix}.$$

Two configurations of synthetic datasets are considered: (1) the class covariance matrices are diagonal and almost the same; (2) the class covariance matrices are not diagonal but almost the same.

Patterns of each class in data configuration 1 (Figure 1(a)) were drawn from a bivariate normal distribution with, respectively, the following parameters:
Class 1: $\mu_1 = 45$, $\mu_2 = 30$, $\sigma_1^2 = 100$, $\sigma_2^2 = 9$, $\rho = 0.0$
Class 2: $\mu_1 = 70$, $\mu_2 = 38$, $\sigma_1^2 = 81$, $\sigma_2^2 = 16$, $\rho = 0.0$
Class 3: $\mu_1 = 45$, $\mu_2 = 42$, $\sigma_1^2 = 100$, $\sigma_2^2 = 16$, $\rho = 0.0$
Class 4: $\mu_1 = 42$, $\mu_2 = 20$, $\sigma_1^2 = 81$, $\sigma_2^2 = 9$, $\rho = 0.0$

Patterns of each class in data configuration 2 (Figure 2(a)) were drawn from a bivariate normal distribution with, respectively, the following parameters:
Class 1: $\mu_1 = 45$, $\mu_2 = 30$, $\sigma_1^2 = 100$, $\sigma_2^2 = 9$, $\rho = 0.7$
Class 2: $\mu_1 = 70$, $\mu_2 = 38$, $\sigma_1^2 = 81$, $\sigma_2^2 = 16$, $\rho = 0.8$
Class 3: $\mu_1 = 45$, $\mu_2 = 42$, $\sigma_1^2 = 100$, $\sigma_2^2 = 16$, $\rho = 0.7$
Class 4: $\mu_1 = 42$, $\mu_2 = 20$, $\sigma_1^2 = 81$, $\sigma_2^2 = 9$, $\rho = 0.8$

Figures 1(b), 1(c) and 1(d) show the clustering results furnished, respectively, by the KMeans, KKMeans and AMKKMeans algorithms on the synthetic dataset 1. The KMeans and KKMeans algorithms cannot categorize the synthetic dataset 1, as it can be seen in Figures 1(b) and 1(c), respectively. As it can be seen in Figure 1(d), the AMKKMeans algorithm furnishes a better categorization of the data when the class covariance matrices are diagonal and almost the same.

Figures 2(b), 2(c) and 2(d) show the clustering results furnished, respectively, by the KMeans, KKMeans and AMKKMeans algorithms on the synthetic dataset 1. Also in this case, the KMeans and KKMeans algorithms cannot categorize the synthetic dataset 2, as it can be seen in Figures 2(b) and 2(c), respectively. As it can be seen in Figure 2(d), the AMKKMeans algorithm furnishes a better categorization of the data when the class covariance matrices are not diagonal but almost the same.

For each synthetic dataset, the CR index and the OERC were estimated in the framework of a Monte Carlo simulation with 100 replications. In each replication the clustering algorithms were run (until the convergence to a stationary

value of the adequacy criterion) 100 times and the best result for each clustering algorithm was selected according to the adequacy criterion. The average and the standard deviation of these measures based on 100 Monte Carlo replications were computed. The term $2\sigma^2$ in the Gaussian and adaptive Mahalanobis kernels was estimated as the average of the 0.1 and 0.9 quantiles of $||\mathbf{x}_i - \mathbf{x}_k||^2$, $i \neq k$ [39]. The algorithms were applied to the synthetic datasets 1 and 2 to obtain 4-cluster partitions. The 4-cluster partitions furnished by the clustering algorithms were compared with the known a priori 4-class partition.

Table I presents the performance of the KMeans and KKMeans clustering algorithms, as well as the performance of the AMKKMeans algorithm on the synthetic dataset 1 according to the CR index and the OERC. As was pointed out, the performance of the AMKKMeans algorithm was clearly superior when the class covariance matrices are diagonal and almost the same, in comparison with all the other algorithms.

TABLE I

PERFORMANCE OF THE ALGORITHMS ON THE SYNTHETIC DATASET 1: AVERAGE AND STANDARD DEVIATION (IN PARENTHESIS) OF THE CR INDEX AND OERC.

	CR	OERC
KMeans	0.340 (0.055)	0.385 (0.044)
KKMeans	0.338 (0.061)	0.389 (0.048)
AMKKMeans	0.646 (0.100)	0.177 (0.068)

TABLE II

PERFORMANCE OF THE ALGORITHMS ON THE SYNTHETIC DATASET 2: AVERAGE AND STANDARD DEVIATION (IN PARENTHESIS) OF THE CR INDEX AND OERC.

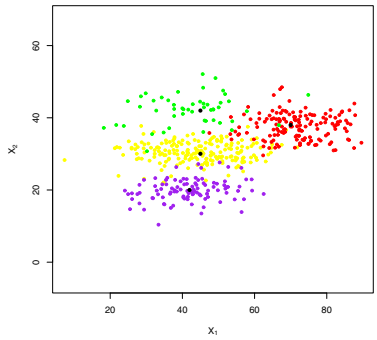
	CR	OERC
KMeans	0.254 (0.035)	0.438 (0.032)
KKMeans	0.255 (0.035)	0.438 (0.033)
AMKKMeans	0.759 (0.041)	0.083 (0.016)

Table II presents the performance of the KMeans and KKMeans clustering algorithms, as well as the performance of the AMKKMeans algorithm on the synthetic dataset 2 according to the CR index and the OERC. Once again, the performance of the AMKKMeans algorithm was clearly superior when the class covariance matrices are not diagonal but almost the same, in comparison with all the other algorithms.

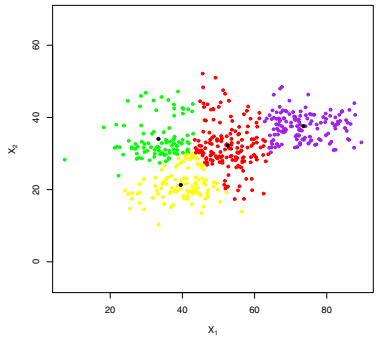
B. Benchmark datasets

The clustering algorithms were applied to four benchmark datasets obtained from the UCI Machine Learning Repository [35], namely Breast Tissue, Iris plants, Wisconsin Diagnostic Breast Cancer (WDBC) and Wine. Table III (in which n represents the number of patterns, p represents the number of variables and K represents the number of classes) describes shortly the datasets considered.

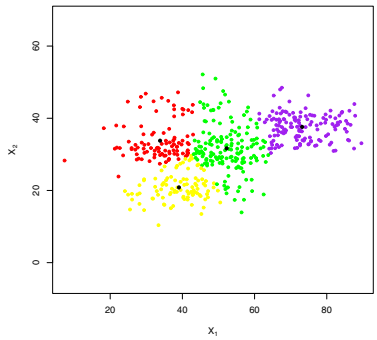
For each dataset, the number of clusters is set equal to the number of classes and the algorithms are run 100



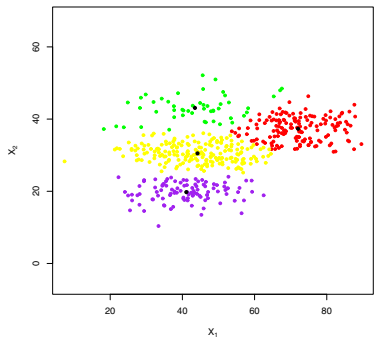
(a) Configuration 1



(b) KMeans

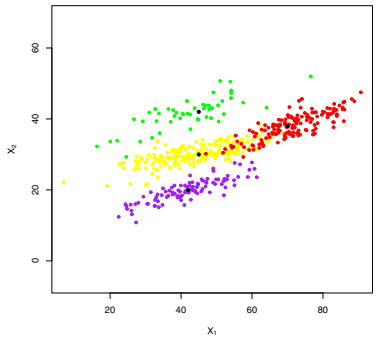


(c) KKMeans

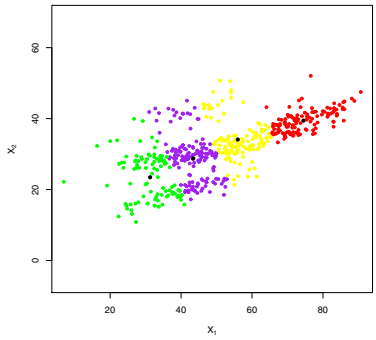


(d) AMKKMeans

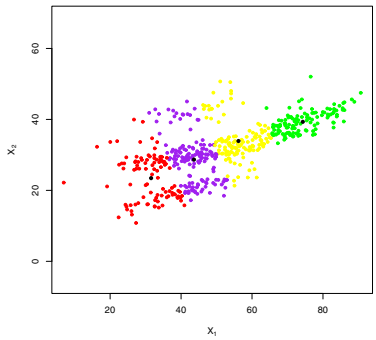
Fig. 1. Synthetic data 1.



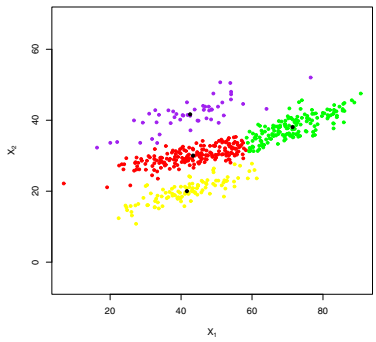
(a) Configuration 2



(b) KMeans



(c) KKMeans



(d) AMKKMeans

Fig. 2. Synthetic data 2.

TABLE III

SUMMARY OF THE DATASETS.

Dataset	n	p	K
Breast Tissue	106	9	6
Iris Plant	150	4	3
WDBC	569	30	2
Wine	178	13	3

times and the best results were selected according to the clustering adequacy criterion. The algorithms were applied to the datasets to obtain a K-cluster partition. For each dataset the known *a priori* K-class partition was assumed as being the true partition. Then, the K-cluster partitions furnished by the clustering algorithms were compared with the known *a priori* K-class partition. Once again, the term $2\sigma^2$ in the Gaussian and adaptive Mahalanobis kernels was estimated as the average of the 0.1 and 0.9 quantiles of $||\mathbf{x}_i - \mathbf{x}_k||^2$, $i \neq k$ [39].

The results presented by the clustering algorithms are summarized in Table IV.

The KMeans, KKMeans and AMKKMeans clustering algorithms were applied to the Breast tissue dataset. The 6-cluster partition obtained with these clustering algorithms was compared with the known a priori 6-cluster partition. The CR index were 0.271, 0.271 and 0.289, respectively, whereas the overall error rates of classification were 43.4%, 43.4% and 41.5% for these clustering methods, respectively. As it can be seen, the AMKKMeans algorithm furnished results slightly better than the results furnished by the other algorithms.

Concerning the Iris plants dataset, the 3-cluster partition obtained with the KMeans, KKMeans and AMKKMeans clustering algorithms was compared with the known a priori 3-cluster partition. The CR index were 0.716, 0.730 and 0.941, respectively, whereas the overall error rates of classification were 11.3%, 10.7% and 2% for these clustering methods, respectively. As it can be seen, the performance of the AMKKMeans algorithm was clearly superior.

The KMeans, KKMeans and AMKKMeans clustering algorithms were applied to the Wisconsin Diagnostic Breast Cancer dataset. The 2-cluster partition obtained with these clustering algorithms was compared with the known a priori 2-cluster partition. The CR index were 0.491, 0.534 and 0.613, respectively, whereas the overall error rates of classification were 14.6%, 13.2% and 10.7% for these clustering methods, respectively. As it can be seen, the performance of the AMKKMeans algorithm was superior.

Finally, concerning the Wine dataset, the 3-cluster partition obtained with the KMeans, KKMeans and AMKKMeans clustering algorithms was compared with the known a priori 3-cluster partition. The CR index were 0.371, 0.371 and 0.965, respectively, whereas the overall error rates of classification were 29.8%, 29.8% and 1.1% for these clustering algorithms, respectively. As it can be seen, the performance of the AMKKMeans algorithm was clearly superior.

In conclusion, for these real benchmark datasets, the

adaptive Mahalanobis kernel K-means clustering algorithm presented the best performance, in comparison with the standard K-means and the kernel K-means based on the well known Gaussian kernel.

TABLE IV
BENCHMARK DATASETS

	Breast Tissue		Iris plants	
	CR	OERC	CR	OERC
KMeans	0.271	0.434	0.716	0.113
KKMeans	0.271	0.434	0.730	0.107
AMKKMeans	0.289	0.415	0.941	0.020
	WDBC		Wine	
	CR	OERC	CR	OERC
KMeans	0.491	0.146	0.371	0.298
KKMeans	0.534	0.132	0.371	0.298
AMKKMeans	0.613	0.107	0.965	0.011

V. CONCLUSIONS

In this paper, we proposed a kernel K-means algorithm based on an adaptive Mahalanobis kernel. This kernel is built based on an adaptive quadratic distance defined by a symmetric positive definite matrix that changes at each algorithm iteration and takes into account the correlations between variables. The main advantage of the proposed clustering algorithm over its conventional counterpart is that the introduction of a symmetric positive definite matrix allow the discovery of clusters with non-hyperspherical shapes. The effectiveness of the proposed algorithm was demonstrated through experiments with synthetic and benchmark datasets.

Concerning the synthetic datasets considered, the AMKKMeans algorithm presented better performances when the class covariance matrices are diagonal and almost the same and also when the class covariance matrices are not diagonal but almost the same. Concerning the real benchmark datasets, the adaptive Mahalanobis kernel K-means clustering algorithm presented the best performance, in comparison with the standard K-means and the kernel K-means based on the well known Gaussian kernel.

As future works we can consider a kernel K-means algorithm based on an adaptive Mahalanobis kernel based on an adaptive quadratic distance defined by a symmetric positive definite matrix (diagonal and not diagonal) for each cluster. Moreover, a kernel K-means algorithm based on an adaptive polynomial kernel $(\mathcal{K}(\mathbf{x}_i, \mathbf{x}_k) = (\gamma \mathbf{x}_i^\top \mathbf{M} \mathbf{x}_k + \theta)^d)$ can be proposed.

REFERENCES

[1] A. D. Gordon, *Classification*, 2nd ed. Boca Raton: Chapman & Hall, 1999.

[2] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM Comput. Surveys*, vol. 31, no. 3, pp. 264–233, 1999.

[3] R. Xu and D. I. I. Wunusch, "Survey of clustering algorithms," *IEEE Trans. Neural Networks*, vol. 16, no. 3, pp. 645–678, 2005.

[4] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern Recogn. Lett.*, vol. 31, no. 8, pp. 651–666, Jun. 2010. [Online]. Available: http://dx.doi.org/10.1016/j.patrec.2009.09.011

[5] F. Höppner, F. Klawonn, R. Kruse, and T. Runkler, *Fuzzy Cluster Analysis*. John Wiley & Sons, Inc., 1999.

[6] M. Filippone, F. Camastra, F. Masulli, and S. Rovetta, "A survey of kernel and spectral methods for clustering," *Pattern Recognition*, vol. 41, pp. 176–190, 2008.

[7] D. Graves and W. Pedrycz, "Kernel-based fuzzy clustering and fuzzy clustering: A comparative experimental study," *Fuzzy Sets and Systems*, vol. 161, pp. 522–543, 2010.

[8] G. Camps-Valls, A. Rodrigo-Gonzlez, J. Muoz-Mar, L. Gmez-Chova, and J. Calpe-Maravilla, "Hyperspectral image classification with mahalanobis relevance vector machines." in *IGARSS*, 2007, pp. 3802–3805.

[9] S. Abe, "Training of support vector machines with mahalanobis kernels." in *ICANN (2)*, ser. Lecture Notes in Computer Science, W. Duch, J. Kacprzyk, E. Oja, and S. Zadrozny, Eds., vol. 3697. Springer, 2005, pp. 571–576.

[10] Y. Kamada and S. Abe, "Support vector regression using mahalanobis kernels," in *ANNPR*, 2006, pp. 144–152.

[11] D. Wang, D. S. Yeung, and E. C. C. Tsang, "Weighted mahalanobis distance kernels for support vector machines," *IEEE Transactions on Neural Networks*, vol. 18, no. 5, pp. 1453–1462, 2007.

[12] M. Girolami, "Mercer kernel-based clustering in feature space," *IEEE Trans. Neural Networks*, vol. 13, no. 3, pp. 780–784, 2002.

[13] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York: Plenum, 1981.

[14] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biol. Cybernet.*, vol. 43, no. 1, pp. 59–69, 1982.

[15] ——, "The self-organizing map," in *Proceedings of the IEEE*, vol. 78, 1990, pp. 1464–1480.

[16] R. R. Yager and D. P. Filev, "Approximate clustering via the mountain method," *IEEE Trans. Systems, Man, Cybernet.*, vol. 24, no. 8, pp. 1279–1284, 1994.

[17] T. M. Martinetz, S. G. Berkovich, and K. J. Schulten, "'neural gas' network for vector quantization and its application to time-series prediction," *IEEE Trans. Neural Networks*, vol. 4, no. 4, pp. 558–569, 1993.

[18] F. Camastra and A. Verri, "A novel kernel method for clustering," *IEEE Trans. Neural Networks*, vol. 27, no. 5, pp. 801–804, 2005.

[19] I. S. Dhillon, Y. Guan, and B. Kulis, "Kernel k-means spectral clustering and normalized cuts," in *Proceedings 10th ACM Internat. Conf. on Knowledge Discovery and Data Mining*, 2004, pp. 551–556.

[20] A. S. Have, M. A. Girolami, and J. Larsen, "Clustering via kernel decomposition," *IEEE Trans. Neural Networks*, vol. 17, no. 1, pp. 256–264, 2006.

[21] S. C. Chen and D. Q. Zhang, "Robust image segmentation using fcm with spatial constraints based on new kernel-induced distance measure," *IEEE Trans. Systems Man Cybernet*, vol. 34, no. 4, pp. 1907–1916, 2004.

[22] D.-M. Tsai and C.-C. Lin, "Fuzzy c-means based clustering for linearly and nonlinearly separable data," *Pattern Recogn.*, vol. 44, no. 8, pp. 1750–1760, Aug. 2011.

[23] D. Q. Zhang and S. C. Chen, "Fuzzy clustering using kernel method," in *The 2002 International Conference on Control and Automation, 2002 ICCA*, 2002, pp. 162–163.

[24] R. Inokuchi and S. Miyamoto, "Lvq clustering and som using a kernel function," in *Proceedings of IEEE International Conference on Fuzzy Systems*, vol. 3, 2004, pp. 1497–1500.

[25] D. Macdonald and C. Fyfe, "The kernel self-organizing map," in *Fourth International Conference on Knowledge-Based Intelligent Engineering Systems and Allied Technologies*, vol. 1, 2000, pp. 317–320.

[26] D. W. Kim, K. Y. Lee, D. Lee, and K. H. Lee, "A kernel-based subtractive clustering method," *Pattern Recognition Letters*, vol. 26, pp. 879–891, 2005.

[27] A. K. Qinand and P. N. Suganthan, "Kernel neural gas algorithms with application to cluster analysis," in *ICPR – 17th International Conference on Pattern Recognition (ICPR'04)*, vol. 4, 2004, pp. 617–620.

[28] X. Yin, S. Chen, E. Hu, and D. Zhang, "Semi-supervised clustering with metric learning: An adaptive kernel method," *Pattern Recogn.*, vol. 43, no. 4, pp. 1320–1333, Apr. 2010. [Online]. Available: http://dx.doi.org/10.1016/j.patcog.2009.11.005

[29] D. W. Kim, K. Y. Lee, D. Lee, and K. H. Lee, "Evaluation of the performance of clustering algorithms in kernel-induced feature space," *Pattern Recognition*, vol. 38, no. 4, pp. 607–611, 2005.

[30] J. Mercer, "Functions of positive and negative type and their connection with the theory of integrals equations," in *Proc. R. Soc. London*, vol. 209, 1909, pp. 415–446.

[31] B. Schölkopf, A. J. Smola, and K. R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Comput.*, vol. 10, no. 5, pp. 1299–1319, 1998.
[32] K. R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf, "An introduction to kernel-based learning algorithms," *IEEE Trans. Neural Networks*, vol. 12, no. 2, pp. 181–202, 2001.
[33] D. Q. Zhang and S. C. Chen, "Kernel based fuzzy and possibilistic *c*-means clustering," in *Proceedings of the International Conference in Artificial Neural Network*, 2003, pp. 122–125.
[34] ——, "A novel kernelized fuzzy *c*-means algorithm with application in medical image segmentation," *Artif. Intell. Med.*, vol. 32, no. 1, pp. 37–50, 2004.
[35] K. Bache and M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml
[36] L. Hubert and P. Arabie, "Comparing partitions," *Journal of Classification*, vol. 2, pp. 193–218, 1985.
[37] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and Regression Trees*. Boca Raton: Chapman & Hall/CRC, 1984.
[38] G. W. Milligan, *Clustering and Classification*. Singapore: World Scientific, 1996, ch. Clustering validation: results and implications for applied analysis, pp. 341–375.
[39] B. Caputo, K. Sim, F. Furesjo, and A. Smola, "Appearance-based object recognition using svms: which kernel should i use?" in *Proceedings of NIPS workshop on Statitsical methods for computational experiments in visual processing and computer vision*, 2002.