Available online at www.sciencedirect.com

SciVerse ScienceDirect

Fuzzy Sets and Systems ••• (••••) •••–•••

FUZZY sets and systems

www.elsevier.com/locate/fss

Kernel fuzzy *c*-means with automatic variable weighting

Marcelo R.P. Ferreira [a,b], 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 50.740-560, Recife, PE, Brazil*

Received 5 October 2011; received in revised form 3 May 2013; accepted 4 May 2013

Abstract

This paper presents variable-wise kernel fuzzy *c*-means clustering methods in which dissimilarity measures are obtained as sums of Euclidean distances between patterns and centroids computed individually for each variable by means of kernel functions. The advantage of the proposed approach over the conventional kernel clustering methods is that it allows us to use adaptive distances which change at each algorithm iteration and can either be the same for all clusters or 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. Another advantage of this approach is that it allows the introduction of various fuzzy partition and cluster interpretation tools. Experiments with synthetic and benchmark datasets show the usefulness of the proposed algorithms and the merit of the fuzzy partition and cluster interpretation tools.

Keywords: Kernel fuzzy *c*-means; Variable-wise algorithms; Adaptive distances; Interpretation indexes

1. Introduction

Clustering methods are useful tools to explore data structures and have emerged as popular techniques for unsupervised pattern recognition. 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 [1–3]. These methods have been widely applied in various areas such as taxonomy, image processing, data mining, information retrieval, etc.

The most popular clustering techniques may be divided into hierarchical and partitioning methods. Hierarchical methods furnish an output represented by a complete structure of hierarchy, i.e., a nested sequence of partitions of the input data; their output is a hierarchical structure of groups known as *dendrogram*. On the other hand, 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 hyper-surfaces among clusters. Partitioning clustering methods were performed in two different ways: hard and fuzzy. In hard clustering, the clusters are disjoint

* 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.A.T. de Carvalho).

0165-0114/$ – see front matter
http://dx.doi.org/10.1016/j.fss.2013.05.004

Please cite this article in press as: M.R.P. Ferreira, F.A.T. de Carvalho, Kernel fuzzy *c*-means with automatic variable weighting, Fuzzy Sets and Systems (2013), http://dx.doi.org/10.1016/j.fss.2013.05.004

and non-overlapping in nature. In this case, any pattern may belong to one and only one cluster. In the case of 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 Ref. [4]. Moreover, a survey of the various clustering methods can be found, for example, in Ref. [1].

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 the 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-hyper-spherical 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, kernel-based clustering methods.

Since Girolami's first development of kernel k-means algorithm [5], several clustering methods such as fuzzy c-means [6], self-organizing maps (SOM) [7–9], the mountain method [10] and neural gas [11] have been modified to incorporate kernels and a variety of kernel methods to clustering have been proposed [12]. Such modifications have been developed under two main approaches: 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. [13–16]. Kernel-based fuzzy clustering methods have been proposed in Refs. [17–20]. The authors of Refs. [21,22] developed a kernelized version of SOM. In [23] a kernel mountain method was presented and in [24] a kernel version of neural gas algorithm was proposed. 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 non-linear separating hyper-surfaces among clusters [12,13,25–29].

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, SOM, etc., 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 cluster may have a different set of important variables.

In Ref. [30] the authors developed a kernel-based fuzzy clustering algorithm able to learn the weights of the variables during the clustering process dynamically. This algorithm belongs to the class of methods based on the kernelization of the metric and can be viewed as a clustering scheme based on a local adaptive distance that changes at each algorithm iteration and is different from one cluster to another. The changes are defined by the weights of the variables within each cluster. They also proved convergence of their algorithm and proposed a slightly modified version for clustering incomplete datasets.

In this paper we propose variable-wise kernel fuzzy c-means clustering methods where dissimilarity measures are obtained as sums of Euclidean distances between patterns and centroids computed individually for each variable by means of kernel functions. The advantage of the proposed approach over the conventional kernel clustering methods is that it allows us to use adaptive distances which change at each algorithm iteration and can either be the same for all clusters (global adaptive distances) or different from one cluster to another (local adaptive distances). 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 method presented in [30] was developed based only in local adaptive distances with the constraint that the weights must sum one and considering only the approach of kernelization of the metric. In some situations local adaptive distances may not be appropriate because they can lead the algorithm to fall into local minima, providing suboptimal solutions. For this reason, we developed adaptive methods based on both types of adaptive distances, local and global, and we also took into account the approach of clustering in feature space. Moreover, the derivation of the expressions of the relevance weights of the variables was done considering two cases: one assumes that the weights must sum one, whereas the other assumes that the product of the weights must be one [31]. Another advantage of this approach is that it allows the introduction of various fuzzy partition and cluster interpretation tools.

The remainder of the paper is organized as follows. In Section 2 a brief review about kernels and kernel functions is presented and the conventional kernel fuzzy clustering algorithms are described. Section 3 introduces the proposed

Please cite this article in press as: M.R.P. Ferreira, F.A.T. de Carvalho, Kernel fuzzy c-means with automatic variable weighting, Fuzzy Sets and Systems (2013), http://dx.doi.org/10.1016/j.fss.2013.05.004

variable-wise kernel fuzzy c-means clustering methods based on both non-adaptive and adaptive distances and considering both approaches: kernelization of the metric and clustering in feature space. In Section 3.1.1 we present the non-adaptive and adaptive methods based on kernelization of the metric, including the method proposed in [30], and in Section 3.1.3 we present the non-adaptive and adaptive methods in feature space. Convergence of the variable-wise kernel fuzzy c-means clustering algorithms is discussed in Section 4. In Section 5 we propose additional tools based on suitable dispersion measures for the interpretation of the fuzzy partition and the fuzzy clusters given by the proposed methods: indexes for evaluating the overall quality of a fuzzy partition, the homogeneity of the individual fuzzy clusters, as well as the role of the different variables in the cluster formation process. In Section 6 we demonstrate the effectiveness of the proposed methods through experiments with synthetic datasets as well as benchmark datasets. Finally, a summary is given to conclude the paper in Section 7.

2. Conventional kernel fuzzy clustering methods

Recently, a number of researchers have shown interest in kernel clustering methods [12,25]. The main idea behind these methods is the use of a non-linear mapping Φ from the input space to a higher-dimensional (possibly infinite) space, called feature space.

In this section we briefly recall the basic theory about kernel functions and the conventional kernel fuzzy clustering methods. Let $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ be a non-empty set where $\mathbf{x}_k \in \mathbb{R}^p$. A function $K : X \times X \to \mathbb{R}$ is called a *positive definite kernel* (or *Mercer kernel*) if and only if K is symmetric (i.e. $K(\mathbf{x}_k, \mathbf{x}_j) = K(\mathbf{x}_j, \mathbf{x}_k)$) and the following equation holds [32]:

$$\sum_{i=1}^{n}\sum_{j=1}^{n} c_i c_j K(\mathbf{x}_k, \mathbf{x}_j) \geqslant 0 \quad \forall n \geqslant 2, \tag{1}$$

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

By non-linearly mapping a set of non-linearly separable patterns into a higher-dimensional feature space, it is possible to separate these patterns linearly [33]. 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 $\mathbf{x}_k^\top \mathbf{x}_j$ in the input space is mapped to $\Phi(\mathbf{x}_k)^\top \Phi(\mathbf{x}_j)$ 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:

$$K(\mathbf{x}_k, \mathbf{x}_j) = \Phi(\mathbf{x}_k)^\top \Phi(\mathbf{x}_j), \tag{2}$$

that is usually referred to as kernel trick [34,35].

Because of Eq. (2), it is possible to compute Euclidean distances in $\mathcal{F}$ as follows [34,35]:

$$\begin{aligned} \left\|\Phi(\mathbf{x}_k) - \Phi(\mathbf{x}_j)\right\|^2 &= \left(\Phi(\mathbf{x}_k) - \Phi(\mathbf{x}_j)\right)^\top \left(\Phi(\mathbf{x}_k) - \Phi(\mathbf{x}_j)\right) \\ &= \Phi(\mathbf{x}_k)^\top \Phi(\mathbf{x}_k) - 2\Phi(\mathbf{x}_k)^\top \Phi(\mathbf{x}_j) + \Phi(\mathbf{x}_j)^\top \Phi(\mathbf{x}_j) \\ &= K(\mathbf{x}_k, \mathbf{x}_k) - 2K(\mathbf{x}_k, \mathbf{x}_j) + K(\mathbf{x}_j, \mathbf{x}_j). \end{aligned} \tag{3}$$

Examples of commonly used kernel functions are as follows:

- Linear: $K(\mathbf{x}_i, \mathbf{x}_k) = \mathbf{x}_i^\top \mathbf{x}_k$,
- Polynomial of degree d: $K(\mathbf{x}_i, \mathbf{x}_k) = (\gamma \mathbf{x}_i^\top \mathbf{x}_k + \theta)^d$, $\gamma > 0$, $\theta \geqslant 0$, $d \in \mathbb{N}$,
- Gaussian: $K(\mathbf{x}_i, \mathbf{x}_k) = e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_k\|^2}{2\sigma^2}}$, $\sigma > 0$,
- Laplacian: $K(\mathbf{x}_i, \mathbf{x}_k) = e^{-\gamma\|\mathbf{x}_i - \mathbf{x}_k\|}$, $\gamma > 0$,
- Sigmoid: $K(\mathbf{x}_i, \mathbf{x}_k) = \tanh(\gamma \mathbf{x}_i^\top \mathbf{x}_k + \theta)$, $\gamma > 0$, $\theta \geqslant 0$,

where γ, σ, θ and d are kernel parameters.

There are two major variations of kernel clustering methods which are based, respectively, on kernelization of the metric, and clustering in feature space. Clustering algorithms based on kernelization of the metric seeks for centroids in the input space and the distances between patterns and centroids are obtained by means of kernels:

$$\left\|\Phi(\mathbf{x}_k) - \Phi(\mathbf{v}_i)\right\|^2 = K(\mathbf{x}_k, \mathbf{x}_k) - 2K(\mathbf{x}_k, \mathbf{v}_i) + K(\mathbf{v}_i, \mathbf{v}_i).$$

Please cite this article in press as: M.R.P. Ferreira, F.A.T. de Carvalho, Kernel fuzzy c-means with automatic variable weighting, Fuzzy Sets and Systems (2013), http://dx.doi.org/10.1016/j.fss.2013.05.004

On the other hand, clustering algorithms in feature space proceed by mapping each pattern by means of a non-linear function Φ and then obtaining the centroids in the feature space. Let $\mathbf{v}_i^{\Phi}$ be the i-th centroid in the feature space. We will see that it is possible to obtain $\|\Phi(\mathbf{x}_k) - \mathbf{v}_i^{\Phi}\|^2$, without the need of calculating $\mathbf{v}_i^{\Phi}$, by means of the kernel trick (Eq. (2)).

2.1. Kernel fuzzy c-means with kernelization of the metric

The basic idea in kernel fuzzy c-means with kernelization of the metric (here labeled KFCM-K) is to minimize the following objective function [12,20,36]:

$$\begin{aligned} J &= \sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ik})^m \left\|\Phi(\mathbf{x}_k) - \Phi(\mathbf{v}_i)\right\|^2 \\ &= \sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ik})^m \left\{K(\mathbf{x}_k,\mathbf{x}_k) - 2K(\mathbf{x}_k,\mathbf{v}_i) + K(\mathbf{v}_i,\mathbf{v}_i)\right\}, \end{aligned} \tag{4}$$

subject to

$$\begin{cases} u_{ik} \in [0,1] & \forall i,k, \\ \displaystyle\sum_{i=1}^{c} u_{ik} = 1 & \forall k, \end{cases} \tag{5}$$

where $\mathbf{v}_i \in \mathbb{R}^p$ is the centroid of the i-th cluster ($i = 1,\ldots,c$), u_{ik} is the fuzzy membership degree of the pattern k to the i-th cluster ($i = 1,\ldots,c$, $k = 1,\ldots,n$) and $m \in (1,\infty)$ is a parameter that controls the fuzziness of membership for each pattern k.

The derivation of the centroids depends on the specific selection of the kernel function. If we consider the Gaussian kernel, then $K(\mathbf{x}_k,\mathbf{x}_k) = 1$ ($k = 1,\ldots,n$) and the functional (4) can be expressed as [29]:

$$J = 2\sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ik})^m\left(1 - K(\mathbf{x}_k,\mathbf{v}_i)\right). \tag{6}$$

Then, the cluster centroids are obtained as:

$$\mathbf{v}_i = \frac{\sum_{k=1}^{n}(u_{ik})^m K(\mathbf{x}_k,\mathbf{v}_i)\mathbf{x}_k}{\sum_{k=1}^{n}(u_{ik})^m K(\mathbf{x}_k,\mathbf{v}_i)}, \quad i = 1,\ldots,c. \tag{7}$$

In the updating of the partition matrix $\mathbf{U}$, the centroids $\mathbf{v}_i$ ($i = 1,\ldots,c$) are fixed and we need to find the fuzzy membership degrees u_{ik}, $i = 1,\ldots,c$, $k = 1,\ldots,n$, that minimize the clustering criterion J under the constraints given in (5). Using the technique of Lagrange multipliers we have the following solution [12,29]:

$$u_{ik} = \left[\sum_{h=1}^{c}\left(\frac{1 - K(\mathbf{x}_k,\mathbf{v}_i)}{1 - K(\mathbf{x}_k,\mathbf{v}_h)}\right)^{\frac{1}{m-1}}\right]^{-1}. \tag{8}$$

2.1.1. Algorithm

The KFCM-K clustering algorithm is executed in the following steps:

(1) Fix c, $2 \leqslant c < n$; fix m, $1 < m < \infty$; fix T (an iteration limit); and fix $0 < \varepsilon \ll 1$; initialize the fuzzy membership degrees u_{ik} ($i = 1,\ldots,c$, $k = 1,\ldots,n$) such that $u_{ik} \geqslant 0$ and $\sum_{i=1}^{c} u_{ik} = 1$.
(2) $t = 1$.
(3) Update the cluster centroids $\mathbf{v}_i$ ($i = 1,\ldots,c$) according to Eq. (7).
(4) Update the fuzzy membership degrees u_{ik} ($i = 1,\ldots,c$, $k = 1,\ldots,n$) according to Eq. (8).
(5) If $|J^{t+1} - J^t| \leqslant \varepsilon$ or $t > T$ stop, else $t = t + 1$ and go to step (3).

Please cite this article in press as: M.R.P. Ferreira, F.A.T. de Carvalho, Kernel fuzzy c-means with automatic variable weighting, Fuzzy Sets and Systems (2013), http://dx.doi.org/10.1016/j.fss.2013.05.004

2.2. Kernel fuzzy c-means in feature space

The kernel fuzzy c-means algorithm in feature space (here labeled KFCM-F) iteratively searches for c clusters by minimizing the following objective function [12,17,27,37]:

$$J = \sum_{h=1}^{k}\sum_{k=1}^{n}(u_{ik})^m \left\| \Phi(\mathbf{x}_k) - \mathbf{v}_i^{\Phi} \right\|^2, \tag{9}$$

subject to the constraints given in Eq. (5), where u_{ik} and m are defined as before and $\mathbf{v}_i^{\Phi}$ is the i-th cluster centroid in feature space.

Optimization of the criterion given in (9) with respect to $\mathbf{v}_i^{\Phi}$ furnishes the following expression for the cluster centroids in feature space [12,17,27,37]:

$$\mathbf{v}_i^{\Phi} = \frac{\sum_{k=1}^{n}(u_{ik})^m \Phi(\mathbf{x}_k)}{\sum_{k=1}^{n}(u_{ik})^m}, \quad i = 1, \ldots, c. \tag{10}$$

The next step is to minimize (9) with respect to u_{ik}, $i = 1, \ldots, c$, $k = 1, \ldots, n$, under the constraints given in (5). To do so, we apply the method of Lagrange multipliers, which leads to the following solution [12,17,27,37]:

$$u_{ik} = \left[\sum_{h=1}^{c} \left(\frac{\| \Phi(\mathbf{x}_k) - \mathbf{v}_i^{\Phi} \|^2}{\| \Phi(\mathbf{x}_k) - \mathbf{v}_h^{\Phi} \|^2} \right)^{\frac{1}{m-1}} \right]^{-1}. \tag{11}$$

The distance between $\Phi(\mathbf{x}_k)$ and $\mathbf{v}_i^{\Phi}$ in the feature space is calculated through the kernel in the original space:

$$\begin{aligned} \left\| \Phi(\mathbf{x}_k) - \mathbf{v}_i^{\Phi} \right\|^2 &= \Phi(\mathbf{x}_k)^{\top}\Phi(\mathbf{x}_k) - 2\Phi(\mathbf{x}_k)^{\top}\left(\mathbf{v}_i^{\Phi}\right) + \left(\mathbf{v}_i^{\Phi}\right)^{\top}\left(\mathbf{v}_i^{\Phi}\right) \\ &= \Phi(\mathbf{x}_k)^{\top}\Phi(\mathbf{x}_k) - \frac{2\sum_{l=1}^{n}(u_{il})^m \Phi(\mathbf{x}_l)^{\top}\Phi(\mathbf{x}_k)}{\sum_{l=1}^{n}(u_{il})^m} + \frac{\sum_{r=1}^{n}\sum_{s=1}^{n}(u_{ir})^m (u_{is})^m \Phi(\mathbf{x}_r)^{\top}\Phi(\mathbf{x}_s)}{(\sum_{r=1}^{n}(u_{ir})^m)^2} \\ &= K(\mathbf{x}_k, \mathbf{x}_k) - \frac{2\sum_{l=1}^{n}(u_{il})^m K(\mathbf{x}_l, \mathbf{x}_k)}{\sum_{l=1}^{n}(u_{il})^m} + \frac{\sum_{r=1}^{n}\sum_{s=1}^{n}(u_{ir})^m (u_{is})^m K(\mathbf{x}_r, \mathbf{x}_s)}{(\sum_{r=1}^{n}(u_{ir})^m)^2}. \end{aligned} \tag{12}$$

Additionally, the criterion J given in Eq. (9) can be rewritten as

$$\begin{aligned} J &= \sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ik})^m \left\| \Phi(\mathbf{x}_k) - \mathbf{v}_i^{\Phi} \right\|^2 \\ &= \sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ik})^m \left\{ K(\mathbf{x}_k, \mathbf{x}_k) - \frac{2\sum_{l=1}^{n}(u_{il})^m K(\mathbf{x}_l, \mathbf{x}_k)}{\sum_{l=1}^{n}(u_{il})^m} + \frac{\sum_{r=1}^{n}\sum_{s=1}^{n}(u_{ir})^m (u_{is})^m K(\mathbf{x}_r, \mathbf{x}_s)}{(\sum_{r=1}^{n}(u_{ir})^m)^2} \right\}. \end{aligned} \tag{13}$$

The kernel fuzzy c-means in feature space lacks the step in which cluster centroids are updated. The updating of the fuzzy partition matrix can be done without calculating the centroids due to the implicit mapping via the kernel function in Eq. (12).

2.2.1. Algorithm

The KFCM-F clustering algorithm is executed in the following steps:

(1) Fix c, $2 \leqslant c < n$; fix m, $1 < m < \infty$; fix T (an iteration limit); and fix $0 < \varepsilon \ll 1$; initialize the fuzzy membership degrees u_{ik} ($i = 1, \ldots, c$, $k = 1, \ldots, n$) such that $u_{ik} \geqslant 0$ and $\sum_{i=1}^{c} u_{ik} = 1$.
(2) $t = 1$.
(3) Update the fuzzy membership degrees u_{ik} ($i = 1, \ldots, c$, $k = 1, \ldots, n$) according to Eq. (11).
(4) If $|J^{t+1} - J^t| \leqslant \varepsilon$ or $t > T$ stop, else $t = t + 1$ and go to step (3).

Please cite this article in press as: M.R.P. Ferreira, F.A.T. de Carvalho, Kernel fuzzy c-means with automatic variable weighting, Fuzzy Sets and Systems (2013), http://dx.doi.org/10.1016/j.fss.2013.05.004

3. Variable-wise kernel fuzzy *c*-means clustering methods

Conventional kernel clustering methods (kernel k-means, kernel fuzzy c-means, etc.) do not take into account the relevance weights of the variables, i.e., these methods consider that all variables are equally important to the clustering process in the sense that all have the same relevance weight. However, in most areas we typically have to deal with high-dimensional datasets. Therefore, some variables may be irrelevant and, among the relevant ones, some may be more or less relevant than others. Furthermore, the relevance weight of each variable to each cluster may be different, i.e., each cluster may have a different set of relevant variables. If we consider that there may exist differences in the relevance weights among variables and if we can compute these weights, then the performance of the kernel clustering methods can be improved.

This section introduces variable-wise kernel fuzzy c-means clustering methods, which are kernel-based fuzzy clustering algorithms where dissimilarity measures are obtained as sums of euclidean distances between patterns and centroids computed at the level of each variable. The main idea of these methods is that we can write a kernel function as a sum of kernel functions applied at each variable.

Proposition 3.1. *If* $K_1 : X_1 \times X_1 \to \mathbb{R}$ *and* $K_2 : X_2 \times X_2 \to \mathbb{R}$ *are kernels, then the sum* $K_1(\mathbf{x}_1, \mathbf{x}'_1) + K_2(\mathbf{x}_2, \mathbf{x}'_2)$ *is a kernel on* $(X_1 \times X_1) \times (X_2 \times X_2)$*, where* $\mathbf{x}_1, \mathbf{x}'_1 \in X_1$ *and* $\mathbf{x}_2, \mathbf{x}'_2 \in X_2$*,* $X_1, X_2 \subset \mathbb{R}^p$.

Proof. The proof can be obtained as presented in [38]. □

Proposition 3.1 can be particularly useful if the patterns are represented by sets of variables with different meanings, and should be dealt differently. More specifically, if a pattern is represented by a p-dimensional vector (p variables), we can split it in p parts and use p different kernel functions for these parts.

Because of Proposition 3.1, a dissimilarity function between an object $\mathbf{x}_k$ and a cluster centroid $\mathbf{v}_i$ defined as

$$\begin{aligned}\varphi^2(\mathbf{x}_k, \mathbf{v}_i) &= \sum_{j=1}^{p} \left\| \Phi(x_{kj}) - \Phi(v_{ij}) \right\|^2 \\ &= \sum_{j=1}^{p} \left\{ K_j(x_{kj}, x_{kj}) - 2K_j(x_{kj}, v_{ij}) + K_j(v_{ij}, v_{ij}) \right\} \end{aligned} \tag{14}$$

is also a kernel on $(X_1 \times X_1) \times \cdots \times (X_p \times X_p)$, where $K_j : X_j \times X_j \to \mathbb{R}$ are kernel functions and X_j is the space of the j-th variable.

3.1. Variable-wise kernel fuzzy c-means clustering algorithms

In this section, we present variable-wise kernel fuzzy c-means clustering algorithms considering the approaches of kernelization of the metric and clustering in feature space. For both approaches, kernelization of the metric and clustering in feature space, we considered non-adaptive and adaptive distances. Adaptive distances are considered depending on whether they are parameterized by a unique and same vector of weights or by a different weight vector for each cluster. Moreover, the derivation of the expressions of the weights was done according to two kinds of constraints: first, assuming that the sum of the weights of the variables must be equal to one; and, the second, assuming that the product of the weights of the variables must be equal to one. This last kind of constraint was motivated by the Gustafson and Kessel [39] algorithm which is based on a quadratic distance defined by a positive definite symmetric matrix M_i associated with the cluster i ($i = 1, \ldots, c$) under $\det(M_i) = 1$. If M_i ($i = 1, \ldots, c$) is diagonal, then we have that the fuzzy clustering algorithm is based on a local adaptive distance with the constraint that the product of the weights of the variables on each cluster must be equal to one. Moreover, if M_i ($i = 1, \ldots, c$) is diagonal and M_i is also the same to each cluster ($M_i = M$, $i = 1, \ldots, c$), then we have that the fuzzy clustering algorithm is based on a global adaptive distance with the constraint that the product of the weights of the variables must be equal to one.

3.1.1. Variable-wise kernel fuzzy c-means clustering algorithms with kernelization of the metric

In this section we present variable-wise kernel fuzzy c-means clustering algorithms with kernelization of the metric.

Please cite this article in press as: M.R.P. Ferreira, F.A.T. de Carvalho, Kernel fuzzy *c*-means with automatic variable weighting, Fuzzy Sets and Systems (2013), http://dx.doi.org/10.1016/j.fss.2013.05.004

The algorithms presented hereafter optimize an adequacy criterion J measuring the fit between the clusters and their centroids, which can be generally defined as

$$J = \sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ik})^m \varphi^2(\mathbf{x}_k, \mathbf{v}_i), \tag{15}$$

subject to the constraints given in Eq. (5), where $\varphi^2(\mathbf{x}_k, \mathbf{v}_i)$ is a suitable squared distance between the pattern $\mathbf{x}_k$ and the cluster centroid $\mathbf{v}_i$ computed by means of kernels, u_{ik} $(i = 1, \ldots, c, k = 1, \ldots, n)$ and m are defined as before.

According to the distance φ^2, there are different variable-wise kernel fuzzy c-means clustering algorithms. As we have seen, the term $\|\Phi(x_{kj}) - \Phi(v_{ij})\|^2$ can be computed as

$$\left\|\Phi(x_{kj}) - \Phi(v_{ij})\right\|^2 = K(x_{kj}, x_{kj}) - 2K(x_{kj}, v_{ij}) + K(v_{ij}, v_{ij}). \tag{16}$$

The distances considered in this case are:

(a) Non-adaptive distance:

$$\varphi^2(\mathbf{x}_k, \mathbf{v}_i) = \sum_{j=1}^{p}\left\|\Phi(x_{kj}) - \Phi(v_{ij})\right\|^2. \tag{17}$$

In this paper, the variable-wise kernel clustering algorithm considering the distance given in Eq. (17) was labeled VKFCM-K.

(b) Local adaptive distance with the constraint that the sum of the weights of the variables for each cluster must be equal to one:

$$\varphi^2(\mathbf{x}_k, \mathbf{v}_i) = \varphi^2_{\boldsymbol{\lambda}_i}(\mathbf{x}_k, \mathbf{v}_i) = \sum_{j=1}^{p}(\lambda_{ij})^\beta \left\|\Phi(x_{kj}) - \Phi(v_{ij})\right\|^2, \tag{18}$$

in which $\boldsymbol{\lambda}_i = (\lambda_{i1}, \ldots, \lambda_{ip})$, subject to

$$\begin{cases} \lambda_{ij} \in [0, 1] \quad \forall i, j, \\ \displaystyle\sum_{j=1}^{p}\lambda_{ij} = 1 \quad \forall i, \end{cases} \tag{19}$$

is the vector of weights concerning to cluster i, and $\beta \in (1, \infty)$ is a parameter that controls the degree of influence of the weight of each variable to each cluster so that if β is large enough, then all variables have the same importance to all clusters; on the other hand, if $\beta \to 1$, then the influence of the weights of the variables will be the highest. It is important to note that in this case the set of variables that are important/unimportant may not be the same for all clusters, i.e., each cluster may have a different set of important variables.

In this paper, the variable-wise kernel clustering algorithm considering the distance given in Eq. (18) was labeled VKFCM-K-LS. In this case, the clustering algorithm is the same that has been presented in [30].

(c) Global adaptive distance with the constraint that the sum of the weights of the variables must be equal to one:

$$\varphi^2(\mathbf{x}_k, \mathbf{v}_i) = \varphi^2_{\boldsymbol{\lambda}}(\mathbf{x}_k, \mathbf{v}_i) = \sum_{j=1}^{p}(\lambda_j)^\beta \left\|\Phi(x_{kj}) - \Phi(v_{ij})\right\|^2, \tag{20}$$

in which $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_p)$, subject to

$$\begin{cases} \lambda_j \in [0, 1] \quad \forall j, \\ \displaystyle\sum_{j=1}^{p}\lambda_j = 1, \end{cases} \tag{21}$$

is the vector of weights and β is defined as before. It is important to note that in this case we are assuming that the set of variables that are important/unimportant is the same for all clusters.

Please cite this article in press as: M.R.P. Ferreira, F.A.T. de Carvalho, Kernel fuzzy c-means with automatic variable weighting, Fuzzy Sets and Systems (2013), http://dx.doi.org/10.1016/j.fss.2013.05.004

In this paper, the variable-wise kernel clustering algorithm considering the distance given in Eq. (20) was labeled VKFCM-K-GS.

(d) Local adaptive distance with the constraint that the product of the weights of the variables for each cluster must be equal to one:

$$\varphi^2(\mathbf{x}_k, \mathbf{v}_i) = \varphi^2_{\boldsymbol{\lambda}_i}(\mathbf{x}_k, \mathbf{v}_i) = \sum_{j=1}^{p} \lambda_{ij} \left\| \Phi(x_{kj}) - \Phi(v_{ij}) \right\|^2, \tag{22}$$

in which $\boldsymbol{\lambda}_i = (\lambda_{i1}, \ldots, \lambda_{ip})$, subject to

$$\begin{cases} \lambda_{ij} > 0 & \forall i, j, \\ \prod_{j=1}^{p} \lambda_{ij} = 1 & \forall i, \end{cases} \tag{23}$$

is the vector of weights concerning to cluster i. Also in this case it is important to note that the set of variables that are important/unimportant may not be the same for all clusters.

In this paper, the variable-wise kernel clustering algorithm considering the distance given in Eq. (22) was labeled VKFCM-K-LP.

(e) Global adaptive distance with the constraint that the product of the weights of the variables must be equal to one:

$$\varphi^2(\mathbf{x}_k, \mathbf{v}_i) = \varphi^2_{\boldsymbol{\lambda}}(\mathbf{x}_k, \mathbf{v}_i) = \sum_{j=1}^{p} \lambda_j \left\| \Phi(x_{kj}) - \Phi(v_{ij}) \right\|^2, \tag{24}$$

in which $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_p)$, subject to

$$\begin{cases} \lambda_j > 0 & \forall j, \\ \prod_{j=1}^{p} \lambda_j = 1, \end{cases} \tag{25}$$

is the vector of weights. Also in this case we are assuming that the set of variables that are important/unimportant is the same for all clusters.

The derivation of the centroids depends on the specific selection of the kernel function. In this paper we consider the Gaussian kernel, that is the most commonly used in the literature. In this case, $K(x_{kj}, x_{kj}) = 1$ $(k = 1, \ldots, n;$ $j = 1, \ldots, p)$ and the term given in Eq. (16) can be expressed as

$$\left\| \Phi(x_{kj}) - \Phi(v_{ij}) \right\|^2 = 2\big(1 - K(x_{kj}, v_{ij})\big). \tag{26}$$

At this step, the fuzzy membership degrees u_{ik} $(i = 1, \ldots, c, k = 1, \ldots, n)$ and, in the case of adaptive distances, the weights of the variables are fixed.

Proposition 3.2. *Whichever the distance function (Eqs. (17), (18), (20), (22) and (24)), and if $K(\cdot,\cdot)$ is the Gaussian kernel, then the cluster centroid $\mathbf{v}_i = (v_{i1}, \ldots, v_{ip})$ $(i = 1, \ldots, c)$, which minimizes the criterion J given in Eq. (15), has its components v_{ij} $(j = 1, \ldots, p)$ updated according to the following expression*:

$$v_{ij} = \frac{\sum_{k=1}^{n} (u_{ik})^m K(x_{kj}, v_{ij}) x_{kj}}{\sum_{k=1}^{n} (u_{ik})^m K(x_{kj}, v_{ij})}. \tag{27}$$

Proof. The proof is given in Appendix A. □

If we are considering the algorithms based on adaptive distances the next step is to determine the weights of the variables. Now, the fuzzy partition matrix $\mathbf{U}$ and the centroids $\mathbf{v}_i$, $i = 1, \ldots, c$, are fixed and the problem is to find the weights of the variables which minimizes the criterion J under the suitable constraints.

Please cite this article in press as: M.R.P. Ferreira, F.A.T. de Carvalho, Kernel fuzzy c-means with automatic variable weighting, Fuzzy Sets and Systems (2013), http://dx.doi.org/10.1016/j.fss.2013.05.004

Proposition 3.3. *The weights of the variables, which minimizes the criterion J given in Eq. (15), are calculated according to the adaptive distance function used*:

(a) *If the adaptive distance function is given by Eq. (18), the vector of weights $\boldsymbol{\lambda}_i = (\lambda_{i1}, \ldots, \lambda_{ip})$ which minimizes the criterion J given in Eq. (15) under $\lambda_{ij} \in [0, 1]$ $\forall i$, j and $\sum_{j=1}^{p} \lambda_{ij} = 1$ $\forall i$, have their components λ_{ij} $(i = 1, \ldots, c,\ j = 1, \ldots, p)$ updated according to the following expression*:

$$\lambda_{ij} = \left[\sum_{l=1}^{p} \left(\frac{\sum_{k=1}^{n} (u_{ik})^m \|\Phi(x_{kj}) - \Phi(v_{ij})\|^2}{\sum_{k=1}^{n} (u_{ik})^m \|\Phi(x_{kl}) - \Phi(v_{il})\|^2} \right)^{\frac{1}{\beta - 1}} \right]^{-1}. \tag{28}$$

(b) *If the adaptive distance function is given by Eq. (20), the vector of weights $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_p)$ which minimizes the criterion J given in Eq. (15) under $\lambda_j \in [0, 1]$ $\forall j$ and $\sum_{j=1}^{p} \lambda_j = 1$, have their components λ_j $(j = 1, \ldots, p)$ updated according to the following expression*:

$$\lambda_j = \left[\sum_{l=1}^{p} \left(\frac{\sum_{i=1}^{c} \sum_{k=1}^{n} (u_{ik})^m \|\Phi(x_{kj}) - \Phi(v_{ij})\|^2}{\sum_{i=1}^{c} \sum_{k=1}^{n} (u_{ik})^m \|\Phi(x_{kl}) - \Phi(v_{il})\|^2} \right)^{\frac{1}{\beta - 1}} \right]^{-1}. \tag{29}$$

(c) *If the adaptive distance function is given by Eq. (22), the vector of weights $\boldsymbol{\lambda}_i = (\lambda_{i1}, \ldots, \lambda_{ip})$ which minimizes the criterion J given in Eq. (15) under $\lambda_{ij} > 0$ $\forall i$, j and $\prod_{j=1}^{p} \lambda_{ij} = 1$ $\forall i$, have their components λ_{ij} $(i = 1, \ldots, c,\ j = 1, \ldots, p)$ updated according to the following expression*:

$$\lambda_{ij} = \frac{\{\prod_{l=1}^{p} (\sum_{k=1}^{n} (u_{ik})^m \|\Phi(x_{kl}) - \Phi(v_{il})\|^2)\}^{\frac{1}{p}}}{\sum_{k=1}^{n} (u_{ik})^m \|\Phi(x_{kj}) - \Phi(v_{ij})\|^2}. \tag{30}$$

(d) *If the adaptive distance function is given by Eq. (24), the vector of weights $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_p)$ which minimizes the criterion J given in Eq. (15) under $\lambda_j > 0$ $\forall j$ and $\prod_{j=1}^{p} \lambda_j = 1$, have their components λ_j $(j = 1, \ldots, p)$ updated according to the following expression*:

$$\lambda_j = \frac{\{\prod_{l=1}^{p} (\sum_{i=1}^{c} \sum_{k=1}^{n} (u_{ik})^m \|\Phi(x_{kl}) - \Phi(v_{il})\|^2)\}^{\frac{1}{p}}}{\sum_{i=1}^{c} \sum_{k=1}^{n} (u_{ik})^m \|\Phi(x_{kj}) - \Phi(v_{ij})\|^2}. \tag{31}$$

Proof. The proof is given in Appendix B. □

Remark. Note that for the local adaptive distances, the closer the objects are to the prototype of a given fuzzy cluster concerning a given variable, the higher is the relevance weight of this variable on this fuzzy cluster. Moreover, for the global adaptive distances, the closer the objects are to the set of cluster prototypes, the higher is the relevance weight of this variable.

Finally, the centroids and, in the case of adaptive distances, the weights of the variables are fixed and the criterion J can be viewed as a function of the fuzzy partition matrix $\mathbf{U}$. Then, the problem is to find the best fuzzy partition matrix. To do so, we need to find the membership degrees u_{ik}, $k = 1, \ldots, n$, $i = 1, \ldots, c$, under $u_{ik} \in [0, 1]$ and $\sum_{i=1}^{c} u_{ik} = 1$ $\forall k$, which minimizes the criterion J.

Proposition 3.4. *Whichever the distance function (Eqs. (17), (18), (20), (22) and (24)), the fuzzy membership degree u_{ik} $(i = 1, \ldots, c,\ k = 1, \ldots, n)$, which minimizes the clustering criterion J given in Eq. (15), under $u_{ik} \in [0, 1]$ $\forall i, k$ and $\sum_{i=1}^{c} u_{ik} = 1$ $\forall k$, is updated according to the following expression*:

$$u_{ik} = \left[\sum_{h=1}^{c} \left(\frac{\varphi^2(\mathbf{x}_k, \mathbf{v}_i)}{\varphi^2(\mathbf{x}_k, \mathbf{v}_h)} \right)^{\frac{1}{m-1}} \right]^{-1}. \tag{32}$$

Proof. The proof follows the same scheme of that developed in the classical fuzzy c-means algorithm [6]. □

Please cite this article in press as: M.R.P. Ferreira, F.A.T. de Carvalho, Kernel fuzzy c-means with automatic variable weighting, Fuzzy Sets and Systems (2013), http://dx.doi.org/10.1016/j.fss.2013.05.004

3.1.2. Algorithm

The variable-wise kernel fuzzy clustering algorithms with kernelization of the metric are executed in the following steps:

(1) Fix c (the number of clusters), $2 \leqslant c < n$; fix m, $1 < m < \infty$; fix β, $1 < \beta < \infty$ (if we are considering adaptive distances with the restriction that the sum of the weights of the variables must be equal to one); fix T (an iteration limit); and fix $0 < \varepsilon \ll 1$; randomly initialize the fuzzy membership degrees u_{ik} $(i = 1, \ldots, c,\ k = 1, \ldots, n)$ such that $u_{ik} \geqslant 0$ and $\sum_{i=1}^{c} u_{ik} = 1$; set the weights of the variables all equal to $1/p$, if we are considering the restriction that the sum of the weights of the variables must be equal to one, or all equal to one, if we are considering the restriction that the product of the weights of the variables must be equal to one.
(2) $t = 1$.
(3) Update the cluster centroids $\mathbf{v}_i$ $(i = 1, \ldots, c)$ according to Eq. (27).
(4) If the distance considered is non-adaptive, go to step (5). Else, update the weights of the variables, depending on the adaptive distance considered (Eqs. (17), (18), (20) and (22)), according to Eqs. (28), (29), (30) or (31).
(5) Update the fuzzy membership degrees u_{ik} $(i = 1, \ldots, c,\ k = 1, \ldots, n)$ according to Eq. (32).
(6) If $|J^{t+1} - J^t| \leqslant \varepsilon$ or $t > T$ stop, else $t = t + 1$ and go to step (3).

3.1.3. Variable-wise kernel fuzzy c-means clustering algorithms in feature space

In this section we present variable-wise kernel fuzzy c-means clustering algorithms in feature space.

The algorithms presented hereafter optimize an adequacy criterion J measuring the fit between the clusters and their centroids, which can be generally defined as

$$J = \sum_{i=1}^{c} \sum_{k=1}^{n} (u_{ik})^m \varphi^2\left(\mathbf{x}_k, \mathbf{v}_i^{\Phi}\right), \tag{33}$$

subject to the constraints given in Eq. (5), where $\varphi^2(\mathbf{x}_k, \mathbf{v}_i^{\Phi})$ is a suitable squared distance between the pattern $\mathbf{x}_k$ and the cluster centroid in feature space $\mathbf{v}_i^{\Phi}$ computed by means of kernels, u_{ik} $(i = 1, \ldots, c,\ k = 1, \ldots, n)$ and m are defined as before.

The distances considered in this case are:

(a) Non-adaptive distance:

$$\varphi^2\left(\mathbf{x}_k, \mathbf{v}_i^{\Phi}\right) = \sum_{j=1}^{p} \left\| \Phi(x_{kj}) - v_{ij}^{\Phi} \right\|^2. \tag{34}$$

In this paper, the variable-wise kernel clustering algorithm considering the distance given in Eq. (34) was labeled VKFCM-F.

(b) Local adaptive distance with the constraint that the sum of the weights of the variables for each cluster must be equal to one:

$$\varphi^2\left(\mathbf{x}_k, \mathbf{v}_i^{\Phi}\right) = \varphi_{\boldsymbol{\lambda}_i}^2\left(\mathbf{x}_k, \mathbf{v}_i^{\Phi}\right) = \sum_{j=1}^{p} (\lambda_{ij})^{\beta} \left\| \Phi(x_{kj}) - v_{ij}^{\Phi} \right\|^2, \tag{35}$$

in which $\boldsymbol{\lambda}_i = (\lambda_{i1}, \ldots, \lambda_{ip})$, subject to the constraints given in Eq. (19), is the vector of weights concerning to cluster i, and β is defined as before.

In this paper, the variable-wise kernel clustering algorithm considering the distance given in Eq. (35) was labeled VKFCM-F-LS.

(c) Global adaptive distance with the constraint that the sum of the weights of the variables must be equal to one:

$$\varphi^2\left(\mathbf{x}_k, \mathbf{v}_i^{\Phi}\right) = \varphi_{\boldsymbol{\lambda}}^2\left(\mathbf{x}_k, \mathbf{v}_i^{\Phi}\right) = \sum_{j=1}^{p} (\lambda_j)^{\beta} \left\| \Phi(x_{kj}) - v_{ij}^{\Phi} \right\|^2, \tag{36}$$

Please cite this article in press as: M.R.P. Ferreira, F.A.T. de Carvalho, Kernel fuzzy c-means with automatic variable weighting, Fuzzy Sets and Systems (2013), http://dx.doi.org/10.1016/j.fss.2013.05.004

in which $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_p)$, subject to the constraints given in Eq. (21), is the vector of weights and β is defined as before. It is important to note that in this case we are assuming that the set of variables that are important/unimportant is the same for all clusters.

In this paper, the variable-wise kernel clustering algorithm considering the distance given in Eq. (36) was labeled VKFCM-F-GS.

(d) Local adaptive distance with the constraint that the product of the weights of the variables for each cluster must be equal to one:

$$\varphi^2\left(\mathbf{x}_k, \mathbf{v}_i^{\Phi}\right) = \varphi^2_{\boldsymbol{\lambda}_i}\left(\mathbf{x}_k, \mathbf{v}_i^{\Phi}\right) = \sum_{j=1}^{p} \lambda_{ij} \left\| \Phi(x_{kj}) - v_{ij}^{\Phi} \right\|^2, \tag{37}$$

in which $\boldsymbol{\lambda}_i = (\lambda_{i1}, \ldots, \lambda_{ip})$, subject to the constraints given in Eq. (23) is the vector of weights concerning to cluster i. Also in this case it is important to note that the set of variables that are important/unimportant may not be the same for all clusters.

In this paper, the variable-wise kernel clustering algorithm considering the distance given in Eq. (37) was labeled VKFCM-F-LP.

(e) Global adaptive distance with the constraint that the product of the weights of the variables must be equal to one:

$$\varphi^2\left(\mathbf{x}_k, \mathbf{v}_i^{\Phi}\right) = \varphi^2_{\boldsymbol{\lambda}}\left(\mathbf{x}_k, \mathbf{v}_i^{\Phi}\right) = \sum_{j=1}^{p} \lambda_{j} \left\| \Phi(x_{kj}) - v_{ij}^{\Phi} \right\|^2, \tag{38}$$

in which $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_p)$, subject to the constraints given in Eq. (25), is the vector of weights. Also in this case we are assuming that the set of variables that are important/unimportant is the same for all clusters.

In this paper, the variable-wise kernel clustering algorithm considering the distance given in Eq. (38) was labeled VKFCM-F-GP.

At first, we need to obtain the cluster centroids in feature space $\mathbf{v}_i^{\Phi}$, $i = 1, \ldots, c$, which minimizes the criterion J. At this step, the fuzzy membership degrees u_{ik} ($i = 1, \ldots, c$, $k = 1, \ldots, n$) and, in the case of adaptive distances, the weights of the variables are fixed.

Proposition 3.5. *Whichever the distance function (Eqs. (34), (35), (36), (37) and (38)), the cluster centroid* $\mathbf{v}_i^{\Phi} = (v_{i1}^{\Phi}, \ldots, v_{ip}^{\Phi})$ $(i = 1, \ldots, c)$*, which minimizes the criterion J given in Eq. (15), has its components* v_{ij}^{Φ} $(j = 1, \ldots, p)$ *updated according to the following expression*:

$$v_{ij}^{\Phi} = \frac{\sum_{k=1}^{n} (u_{ik})^m \Phi(x_{kj})}{\sum_{k=1}^{n} (u_{ik})^m}. \tag{39}$$

Proof. The proof is given in Appendix C. □

If we are considering the algorithms based on adaptive distances the next step is to determine the weights of the variables. Now, the fuzzy partition matrix $\mathbf{U}$ and the cluster centroids in feature space $\mathbf{v}_i^{\Phi}$, $i = 1, \ldots, c$, are fixed and the problem is to find the weights of the variables which minimizes the criterion J under the suitable constraints.

Proposition 3.6. *The weights of the variables, which minimizes the criterion J given in Eq. (15), are calculated according to the adaptive distance function used*:

(a) *If the adaptive distance function is given by Eq. (35), the vector of weights* $\boldsymbol{\lambda}_i = (\lambda_{i1}, \ldots, \lambda_{ip})$ *which minimizes the criterion J given in Eq. (15) under* $\lambda_{ij} \in [0, 1]$ $\forall i, j$ *and* $\sum_{j=1}^{p} \lambda_{ij} = 1$ $\forall i$*, have their components* λ_{ij} $(i = 1, \ldots, c,\ j = 1, \ldots, p)$ *updated according to the following expression*:

$$\lambda_{ij} = \left[\sum_{l=1}^{p} \left(\frac{\sum_{k=1}^{n} (u_{ik})^m \| \Phi(x_{kj}) - v_{ij}^{\Phi} \|^2}{\sum_{k=1}^{n} (u_{ik})^m \| \Phi(x_{kl}) - v_{il}^{\Phi} \|^2} \right)^{\frac{1}{\beta - 1}} \right]^{-1}. \tag{40}$$

(b) *If the adaptive distance function is given by Eq.* (36)*, the vector of weights* $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_p)$ *which minimizes the criterion* J *given in Eq.* (15) *under* $\lambda_j \in [0, 1]\ \forall j$ *and* $\sum_{j=1}^{p} \lambda_j = 1$*, have their components* λ_j $(j = 1, \ldots, p)$ *updated according to the following expression*:

$$\lambda_j = \left[\sum_{l=1}^{p} \left(\frac{\sum_{i=1}^{c} \sum_{k=1}^{n} (u_{ik})^m \| \Phi(x_{kj}) - v_{ij}^{\Phi} \|^2}{\sum_{i=1}^{c} \sum_{k=1}^{n} (u_{ik})^m \| \Phi(x_{kl}) - v_{il}^{\Phi} \|^2} \right)^{\frac{1}{\beta - 1}} \right]^{-1}. \tag{41}$$

(c) *If the adaptive distance function is given by Eq.* (37)*, the vector of weights* $\boldsymbol{\lambda}_i = (\lambda_{i1}, \ldots, \lambda_{ip})$ *which minimizes the criterion* J *given in Eq.* (15) *under* $\lambda_{ij} > 0\ \forall i, j$ *and* $\prod_{j=1}^{p} \lambda_{ij} = 1\ \forall i$*, have their components* λ_{ij} $(i = 1, \ldots, c,$ $j = 1, \ldots, p)$ *updated according to the following expression*:

$$\lambda_{ij} = \frac{\{\prod_{l=1}^{p} (\sum_{k=1}^{n} (u_{ik})^m \| \Phi(x_{kl}) - v_{il}^{\Phi} \|^2)\}^{\frac{1}{p}}}{\sum_{k=1}^{n} (u_{ik})^m \| \Phi(x_{kj}) - v_{ij}^{\Phi} \|^2}. \tag{42}$$

(d) *If the adaptive distance function is given by Eq.* (38)*, the vector of weights* $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_p)$ *which minimizes the criterion* J *given in Eq.* (15) *under* $\lambda_j > 0\ \forall j$ *and* $\prod_{j=1}^{p} \lambda_j = 1$*, have their components* λ_j $(j = 1, \ldots, p)$ *updated according to the following expression*:

$$\lambda_j = \frac{\{\prod_{l=1}^{p} (\sum_{i=1}^{c} \sum_{k=1}^{n} (u_{ik})^m \| \Phi(x_{kl}) - v_{il}^{\Phi} \|^2)\}^{\frac{1}{p}}}{\sum_{i=1}^{c} \sum_{k=1}^{n} (u_{ik})^m \| \Phi(x_{kj}) - v_{ij}^{\Phi} \|^2}. \tag{43}$$

Proof. The proof can be obtained in a similar way as presented in Proposition 3.3. □

Remark. Note that for the local adaptive distances, the closer the objects are to the prototype of a given fuzzy cluster concerning a given variable, the higher is the relevance weight of this variable on this fuzzy cluster. Moreover, for the global adaptive distances, the closer the objects are to the set of cluster prototypes, the higher is the relevance weight of this variable.

It is important to note that the cluster centroids do not need to be computed because the centroid information is implicitly considered, i.e., the term $\| \Phi(x_{kj}) - v_{ij}^{\Phi} \|^2$ in the feature space is computed through the kernel in the original space:

$$\begin{aligned}
\left\| \Phi(x_{kj}) - v_{ij}^{\Phi} \right\|^2 &= \Phi(x_{kj}) \cdot \Phi(x_{kj}) - 2\Phi(x_{kj}) \cdot v_{ij}^{\Phi} + v_{ij}^{\Phi} \cdot v_{ij}^{\Phi} \\
&= \Phi(x_{kj}) \cdot \Phi(x_{kj}) - 2\Phi(x_{kj}) \cdot \frac{\sum_{l=1}^{n} (u_{il})^m \Phi(x_{lj})}{\sum_{l=1}^{n} (u_{il})^m} \\
&\quad + \frac{\sum_{r=1}^{n} (u_{ir})^m \Phi(x_{rj})}{\sum_{r=1}^{n} (u_{ir})^m} \cdot \frac{\sum_{s=1}^{n} (u_{is})^m \Phi(x_{sj})}{\sum_{s=1}^{n} (u_{is})^m} \\
&= \Phi(x_{kj}) \cdot \Phi(x_{kj}) - \frac{2 \sum_{l=1}^{n} (u_{il})^m \Phi(x_{lj}) \cdot \Phi(x_{kj})}{\sum_{l=1}^{n} (u_{il})^m} \\
&\quad + \frac{\sum_{r=1}^{n} \sum_{s=1}^{n} (u_{ir})^m (u_{is})^m \Phi(x_{rj}) \cdot \Phi(x_{sj})}{(\sum_{r=1}^{n} (u_{ir})^m)^2} \\
&= K(x_{kj}, x_{kj}) - \frac{2 \sum_{l=1}^{n} (u_{il})^m K(x_{lj}, x_{kj})}{\sum_{l=1}^{n} (u_{il})^m} \\
&\quad + \frac{\sum_{r=1}^{n} \sum_{s=1}^{n} (u_{ir})^m (u_{is})^m K(x_{rj}, x_{sj})}{(\sum_{r=1}^{n} (u_{ir})^m)^2}.
\end{aligned} \tag{44}$$

Finally, the centroids and, in the case of adaptive distances, the weights of the variables are fixed and the criterion J can be viewed as a function of the fuzzy partition matrix $\mathbf{U}$. Then, the problem is to find the best fuzzy partition matrix. To do so, we need to find the membership degrees $u_{ik}, k = 1, \ldots, n, i = 1, \ldots, c$, under $u_{ik} \in [0, 1]$ and $\sum_{i=1}^{c} u_{ik} = 1$ $\forall k$, which minimizes the criterion J.

Please cite this article in press as: M.R.P. Ferreira, F.A.T. de Carvalho, Kernel fuzzy c-means with automatic variable weighting, Fuzzy Sets and Systems (2013), http://dx.doi.org/10.1016/j.fss.2013.05.004

Proposition 3.7. *Whichever the distance function (Eqs. (34), (35), (36), (37) and (38)), the fuzzy membership degree* u_{ik} $(i = 1, \ldots, c,\ k = 1, \ldots, n)$*, which minimizes the clustering criterion* J *given in Eq. (15), under* $u_{ik} \in [0, 1]\ \forall i, k$ *and* $\sum_{i=1}^{c} u_{ik} = 1\ \forall k$*, is updated according to the following expression*:

$$u_{ik} = \left[\sum_{h=1}^{c} \left(\frac{\varphi^2(\mathbf{x}_k, \mathbf{v}_i)}{\varphi^2(\mathbf{x}_k, \mathbf{v}_h)} \right)^{\frac{1}{m-1}} \right]^{-1}. \tag{45}$$

Proof. The proof follows the same scheme of that developed in the classical fuzzy c-means algorithm [6]. □

3.1.4. Algorithm

The variable-wise kernel fuzzy clustering algorithms in feature space are executed in the following steps:

(1) Fix c (the number of clusters), $2 \leqslant c < n$; fix m, $1 < m < \infty$; fix β, $1 < \beta < \infty$ (if we are considering adaptive distances with the restriction that the sum of the weights of the variables must be equal to one); fix T (an iteration limit); and fix $0 < \varepsilon \ll 1$; randomly initialize the fuzzy membership degrees u_{ik} $(i = 1, \ldots, c,\ k = 1, \ldots, n)$ such that $u_{ik} \geqslant 0$ and $\sum_{i=1}^{c} u_{ik} = 1$; set the weights of the variables all equal to $1/p$, if we are considering the restriction that the sum of the weights of the variables must be equal to one, or all equal to one, if we are considering the restriction that the product of the weights of the variables must be equal to one.
(2) $t = 1$.
(3) If the distance considered is non-adaptive, go to step (4). Else, update the weights of the variables, depending on the adaptive distance considered (Eqs. (35), (36), (37) and (38)), according to Eqs. (40), (41), (42) or (43).
(4) Update the fuzzy membership degrees u_{ik} $(i = 1, \ldots, c,\ k = 1, \ldots, n)$ according to Eq. (11).
(5) If $|J^{t+1} - J^t| \leqslant \varepsilon$ or $t > T$ stop, else $t = t + 1$ and go to step (3).

As in the conventional approach, the variable-wise kernel fuzzy clustering algorithms in feature space lack the step in which cluster centroids are updated, due to the implicit mapping via the kernel function in Eq. (44).

4. Convergence of the variable-wise kernel fuzzy clustering algorithms

In Ref. [30] was proved the convergence of the VKFCM-K-LS algorithm, which can be directly extended to the algorithms described in this paper that are based on adaptive distances under the restriction that the sum of the weights of the variables must be equal to one.

To prove the convergence of the algorithms that are based on adaptive distances under the restriction that the product of the weights of the variables must be equal to one, let us consider the VKFCM-K-LP algorithm (Section 3.1.1) whose adequacy criterion is given by

$$\begin{aligned} J(\mathbf{V}, \boldsymbol{\Lambda}, \mathbf{U}) &= \sum_{i=1}^{c} \sum_{k=1}^{n} (u_{ik})^m \sum_{j=1}^{p} \lambda_{ij} \left\| \Phi(x_{kj}) - \Phi(v_{ij}) \right\|^2 \\ &= \sum_{i=1}^{c} \sum_{k=1}^{n} (u_{ik})^m \varphi^2_{\boldsymbol{\lambda}_i}(\mathbf{x}_k, \mathbf{v}_i), \end{aligned}$$

where

$$\varphi^2_{\boldsymbol{\lambda}_i}(\mathbf{x}_k, \mathbf{v}_i) = \sum_{j=1}^{p} \lambda_{ij} \left\| \Phi(x_{kj}) - \Phi(v_{ij}) \right\|^2 = 2 \sum_{j=1}^{p} \lambda_{ij} \big(1 - K(x_{kj}, v_{ij})\big),$$

if we restrict ourselves to the Gaussian kernel.

Let $\mathbf{U}^* = (\mathbf{u}_1^*, \ldots, \mathbf{u}_n^*)$, a corresponding c-tuple of cluster centroids $\mathbf{V}^* = (\mathbf{v}_1^*, \ldots, \mathbf{v}_c^*)$, and c adaptive distances parametrized by a c-tuple of vectors of weights $\Lambda^* = (\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_c)$ such that

$$J\big(\mathbf{V}^*, \boldsymbol{\Lambda}^*, \mathbf{U}^*\big) = \min\big\{ J(\mathbf{V}, \boldsymbol{\Lambda}, \mathbf{U}):\ \mathbf{V} \in \mathbb{V}^c,\ \boldsymbol{\Lambda} \in \mathbb{L}^c,\ \mathbf{U} \in \mathbb{U}^n \big\}, \tag{46}$$

where

- $\mathbb{U}$ is the space of fuzzy partition membership degrees such that $\mathbf{u}_i \in \mathbb{U}$ $(i = 1, \ldots, c)$. In this paper $\mathbb{U} = \{\mathbf{u} = (u_1, \ldots, u_c) \in [0, 1] \times \cdots \times [0, 1] = [0, 1]^c \colon \sum_{i=1}^{c} u_i = 1\}$ and $\mathbf{U} \in \mathbb{U}^n = \mathbb{U} \times \cdots \times \mathbb{U}$;
- $\mathbb{V}$ is the representation space of centroids such that $\mathbf{v}_i \in \mathbb{V}$ $(i = 1, \ldots, c)$. In this paper, $\mathbb{V} = \mathbb{R} \times \cdots \times \mathbb{R} = \mathbb{R}^p$ and $\mathbf{V} \in \mathbb{V}^c = \mathbb{V} \times \cdots \times \mathbb{V} = \mathbb{R}^{c \times p}$;
- $\mathbb{L}$ is the space of vectors of weights that parameterize the adaptive distances such that $\boldsymbol{\lambda_i} \in \mathbb{L}$ $(i = 1, \ldots, c)$. In this paper, $\mathbb{L} = \{\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_p) \in \mathbb{R}_+^* \times \cdots \times \mathbb{R}_+^* = (\mathbb{R}_+^*)^p \colon \prod_{j=1}^{p} \lambda_j = 1\}$ and $\boldsymbol{\Lambda} \in \mathbb{L}^c = \mathbb{L} \times \cdots \times \mathbb{L} = (\mathbb{R}_+^*)^{c \times p}$.

According to [40], the properties of convergence of this kind of algorithm can be studied from two series: $y_t = (\mathbf{V}^t, \boldsymbol{\Lambda}^t, \mathbf{U}^t)$ and $z_t = J(y_t) = J(\mathbf{V}^t, \boldsymbol{\Lambda}^t, \mathbf{U}^t)$ $(t = 0, 1, \ldots)$. From an initial term $y_0 = (\mathbf{V}^0, \boldsymbol{\Lambda}^0, \mathbf{U}^0)$, the algorithm computes the different terms of the series y_t until the convergence when the criterion J achieves a stationary value.

Proposition 4.1. *The series $z_t = J(y_t)$ decreases at each iteration and converges.*

Proof. First, we will show that the inequalities (I), (II) and (III)

$$\underbrace{J\left(\mathbf{V}^t, \boldsymbol{\Lambda}^t, \mathbf{U}^t\right)}_{z_t} \overbrace{\geqslant}^{\text{(I)}} J\left(\mathbf{V}^{t+1}, \boldsymbol{\Lambda}^t, \mathbf{U}^t\right)$$

$$\overbrace{\geqslant}^{\text{(II)}} J\left(\mathbf{V}^{t+1}, \boldsymbol{\Lambda}^{t+1}, \mathbf{U}^t\right) \overbrace{\geqslant}^{\text{(III)}} \underbrace{J\left(\mathbf{V}^{t+1}, \boldsymbol{\Lambda}^{t+1}, \mathbf{U}^{t+1}\right)}_{z_{t+1}}$$

hold (i.e., the series decreases at each iteration).

The inequality (I) holds because

$$J\left(\mathbf{V}^t, \boldsymbol{\Lambda}^t, \mathbf{U}^t\right) = \sum_{i=1}^{c} \sum_{k=1}^{n} \left(u_{ik}^{(t)}\right)^m \varphi^2_{\boldsymbol{\lambda}_i^{(t)}}\left(\mathbf{x}_k, \mathbf{v}_i^{(t)}\right),$$

and

$$J\left(\mathbf{V}^{t+1}, \boldsymbol{\Lambda}^t, \mathbf{U}^t\right) = \sum_{i=1}^{c} \sum_{k=1}^{n} \left(u_{ik}^{(t)}\right)^m \varphi^2_{\boldsymbol{\lambda}_i^{(t)}}\left(\mathbf{x}_k, \mathbf{v}_i^{(t+1)}\right),$$

and according to Proposition 3.2,

$$\mathbf{V}^{t+1} = \left(\mathbf{v}_1^{t+1}, \ldots, \mathbf{v}_c^{t+1}\right) = \underbrace{\operatorname{argmin}}_{\mathbf{V}=(\mathbf{v}_1,\ldots,\mathbf{v}_c)\in\mathbb{V}^c} \sum_{i=1}^{c} \sum_{k=1}^{n} \left(u_{ik}^{(t)}\right)^m \varphi^2_{\boldsymbol{\lambda}_i^{(t)}}(\mathbf{x}_k, \mathbf{v}_i).$$

Moreover, inequality (II) holds because

$$J\left(\mathbf{V}^{t+1}, \boldsymbol{\Lambda}^{t+1}, \mathbf{U}^t\right) = \sum_{i=1}^{c} \sum_{k=1}^{n} \left(u_{ik}^{(t)}\right)^m \varphi^2_{\boldsymbol{\lambda}_i^{(t+1)}}\left(\mathbf{x}_k, \mathbf{v}_i^{(t+1)}\right),$$

and according to Proposition 3.3,

$$\boldsymbol{\Lambda}^{t+1} = \left(\boldsymbol{\lambda}_1^{t+1}, \ldots, \boldsymbol{\lambda}_c^{t+1}\right) = \underbrace{\operatorname{argmin}}_{\Lambda=(\boldsymbol{\lambda}_1,\ldots,\boldsymbol{\lambda}_c)\in\mathbb{L}^c} \sum_{i=1}^{c} \sum_{k=1}^{n} \left(u_{ik}^{(t)}\right)^m \varphi^2_{\boldsymbol{\lambda}_i}\left(\mathbf{x}_k, \mathbf{v}_i^{(t+1)}\right).$$

The inequality (III) also holds because

$$J\left(\mathbf{V}^{t+1}, \boldsymbol{\Lambda}^{t+1}, \mathbf{U}^{t+1}\right) = \sum_{i=1}^{c} \sum_{k=1}^{n} \left(u_{ik}^{(t+1)}\right)^m \varphi^2_{\boldsymbol{\lambda}_i^{(t+1)}}\left(\mathbf{x}_k, \mathbf{v}_i^{(t+1)}\right),$$

Please cite this article in press as: M.R.P. Ferreira, F.A.T. de Carvalho, Kernel fuzzy c-means with automatic variable weighting, Fuzzy Sets and Systems (2013), http://dx.doi.org/10.1016/j.fss.2013.05.004

and according to Proposition 3.4,

$$\mathbf{U}^{t+1} = \left(\mathbf{u}_1^{t+1}, \ldots, \mathbf{u}_c^{t+1}\right) = \underbrace{\operatorname{argmin}}_{\mathbf{U}=(\mathbf{u}_1,\ldots,\mathbf{u}_c)\in\mathbb{U}^n} \sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ik})^m \varphi^2_{\boldsymbol{\lambda}_i^{(t+1)}}\left(\mathbf{x}_k, \mathbf{v}_i^{(t+1)}\right).$$

Finally, because the series z_t decreases and it is bounded ($J(y_t) \geqslant 0$), it converges.

Proposition 4.2. *The series $y_t = (\mathbf{V}^t, \boldsymbol{\Lambda}^t, \mathbf{U}^t)$ converges.*

Proof. Assume that the stationarity of the series z_t is achieved in the iteration $t = T$. Then, we have that $z_T = z_{T+1}$.

From $z_T = z_{T+1}$, we have that $J(y_T) = J(y_{T+1})$, i.e., $J(\mathbf{V}^T, \boldsymbol{\Lambda}^T, \mathbf{U}^T) = J(\mathbf{V}^{T+1}, \Lambda^{T+1}, \mathbf{U}^{T+1})$ and this equality, according to Proposition 4.1, can be rewritten as the equalities (I), (II) and (III):

$$\underbrace{J\left(\mathbf{V}^T, \boldsymbol{\Lambda}^T, \mathbf{U}^T\right)}_{z_T} \overbrace{=}^{\text{(I)}} J\left(\mathbf{V}^{T+1}, \boldsymbol{\Lambda}^T, \mathbf{U}^T\right)$$

$$\overbrace{=}^{\text{(II)}} J\left(\mathbf{V}^{T+1}, \boldsymbol{\Lambda}^{T+1}, \mathbf{U}^T\right) \overbrace{=}^{\text{(III)}} \underbrace{J\left(\mathbf{V}^{T+1}, \boldsymbol{\Lambda}^{T+1}, \mathbf{U}^{T+1}\right)}_{z_{T+1}}.$$

From (I), we have that $\mathbf{V}^T = \mathbf{V}^{T+1}$ because $\mathbf{V}$ is unique minimizing J when $\mathbf{U}^T$ and $\boldsymbol{\Lambda}^T$ are fixed. From (II), we have that $\boldsymbol{\Lambda}^T = \boldsymbol{\Lambda}^{T+1}$ because $\boldsymbol{\Lambda}$ is unique minimizing J when $\mathbf{U}^T$ and $\mathbf{V}^{T+1}$ are fixed. Furthermore, from (III), we have that $\mathbf{U}^T = \mathbf{U}^{T+1}$ because $\mathbf{U}$ is unique minimizing J when $\mathbf{V}^{T+1}$ and $\boldsymbol{\Lambda}^{T+1}$ are fixed.

Finally, we conclude that $y_T = y_{T+1}$. This conclusion holds for all $t \geqslant T$ and $y_t = y_T\ \forall t \geqslant T$ and it follows that the series y_t converges. □

5. Fuzzy partition and fuzzy cluster interpretation

Fuzzy partition and fuzzy cluster interpretation tools allow the user to evaluate the overall heterogeneity of the data, the intra-cluster and inter-cluster data heterogeneity, the contribution of each variable to the cluster formation, etc.

For quantitative data partitioned by the classical k-means algorithm, Celeux et al. [41] have introduced a family of indexes for cluster and partition interpretation that are based on the sum of squares (SSQ). For the family of indexes presented in [41], the overall dispersion decomposes into the overall dispersion within clusters plus the overall dispersion between clusters. Unfortunately, however, this decomposition is not always valid; it depends on the distance that defines the dispersion measures.

Ref. [42] introduced an approach to compute cluster and partition interpretation indexes even when the overall dispersion does not decompose into the overall dispersion within clusters plus the overall dispersion between clusters. In this section we adapt these indexes for the variable-wise kernel fuzzy c-means clustering methods introduced in this paper by considering corresponding suitable definitions of overall and within clusters dispersion measures, as well as their corresponding decompositions according to clusters, variables and both clusters and variables.

Let $P = (P_1, \ldots, P_c)$ be a fuzzy partition of $\Omega = \{1, \ldots, n\}$ into c clusters obtained from one of the variable-wise kernel fuzzy clustering algorithms presented in Section 3.

5.1. Dispersion measures defined for the algorithms with kernelization of the metric

In the following, two dispersion measures defined for the variable-wise kernel fuzzy c-means clustering algorithms with kernelization of the metric are introduced: overall dispersion and within-clusters dispersion.

Please cite this article in press as: M.R.P. Ferreira, F.A.T. de Carvalho, Kernel fuzzy c-means with automatic variable weighting, Fuzzy Sets and Systems (2013), http://dx.doi.org/10.1016/j.fss.2013.05.004

The overall heterogeneity of all n patterns is measured by the overall dispersion, which is computed according to the distance function used, Eqs. (16), (17), (18), (20) and (22) replacing the cluster centroids $\mathbf{v}_i$ by the overall centroid $\mathbf{v}$, and given by

$$T = \sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ik})^m \varphi^2(\mathbf{x}_i, \mathbf{v}). \tag{47}$$

It is important to observe that Eq. (47) represents the dispersion when the cluster centroids are replaced by the overall centroid, i.e., T measures the overall heterogeneity of all patterns when we don't consider any information about clusters. In other words, we are measuring how dispersed the patterns are with respect to the overall centroid.

Proposition 5.1. *If K is the Gaussian kernel, then the overall cluster centroid $\mathbf{v} = (v_1, \ldots, v_p)$, which minimizes the overall dispersion T given in Eq. (47), has its components v_j $(j = 1, \ldots, p)$ updated according to the distance function used*:

(a) *If the adaptive distance function is given by Eqs. (16), (18) or (22), the overall centroid $\mathbf{v}$ has its components v_j $(j = 1, \ldots, p)$ updated according to the following expression*:

$$v_j = \frac{\sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ik})^m K(x_{kj}, v_j)x_{kj}}{\sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ik})^m K(x_{kj}, v_j)}. \tag{48}$$

(b) *If the adaptive distance function is given by Eq. (17), the overall centroid $\mathbf{v}$ has its components v_j $(j = 1, \ldots, p)$ updated according to the following expression*:

$$v_j = \frac{\sum_{i=1}^{c}(\lambda_{ij})^\beta \sum_{k=1}^{n}(u_{ik})^m K(x_{kj}, v_j)x_{kj}}{\sum_{i=1}^{c}(\lambda_{ij})^\beta \sum_{k=1}^{n}(u_{ik})^m K(x_{kj}, v_j)}. \tag{49}$$

(c) *If the adaptive distance function is given by Eq. (20), the overall centroid $\mathbf{v}$ has its components v_j $(j = 1, \ldots, p)$ updated according to the following expression*:

$$v_j = \frac{\sum_{i=1}^{c}\lambda_{ij} \sum_{k=1}^{n}(u_{ik})^m K(x_{kj}, v_j)x_{kj}}{\sum_{i=1}^{c}\lambda_{ij} \sum_{k=1}^{n}(u_{ik})^m K(x_{kj}, v_j)}. \tag{50}$$

Proof. The proof can be obtained in a similar way as presented in Proposition 3.2. □

We can easily see that the overall dispersion T, given in Eq. (47), decomposes according to variables ($T = \sum_{j=1}^{p} T_j$), and according to clusters ($T = \sum_{i=1}^{c} T_i$), as well as according to clusters and variables ($T = \sum_{i=1}^{c}\sum_{j=1}^{p} T_{ij}$).

Similarly, the overall heterogeneity within-clusters is measured by the within-cluster dispersion, which is given in Eq. (15).

This measure also decomposes according to variables ($J = \sum_{j=1}^{p} J_j$), and according to clusters ($J = \sum_{i=1}^{c} J_i$), as well as according to clusters and variables ($J = \sum_{i=1}^{c}\sum_{j=1}^{p} J_{ij}$).

5.2. Dispersion measures defined for the algorithms in feature space

In the following, two dispersion measures defined for the variable-wise kernel fuzzy c-means clustering algorithms in feature space are introduced: overall dispersion and within-clusters dispersion.

The overall heterogeneity of all n patterns is measured by the overall dispersion, which is computed according to the distance function used, Eqs. (34), (35), (36), (37) and (38) replacing the cluster centroids $\mathbf{v}_i^{\Phi}$ by the overall centroid $\mathbf{v}^{\Phi}$, and given by

$$T = \sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ik})^m \varphi^2\left(\mathbf{x}_i, \mathbf{v}^{\Phi}\right). \tag{51}$$

Please cite this article in press as: M.R.P. Ferreira, F.A.T. de Carvalho, Kernel fuzzy c-means with automatic variable weighting, Fuzzy Sets and Systems (2013), http://dx.doi.org/10.1016/j.fss.2013.05.004

Proposition 5.2. *The overall cluster centroid* $\mathbf{v}^{\Phi} = (v_1^{\Phi}, \ldots, v_p^{\Phi})$*, which minimizes the overall dispersion* T *given in Eq. (51), has its components* v_j^{Φ} $(j = 1, \ldots, p)$ *updated according to the distance function used*:

(a) *If the adaptive distance function is given by Eqs. (34), (36) or (38), the overall centroid* $\mathbf{v}^{\Phi}$ *has its components* v_j^{Φ} $(j = 1, \ldots, p)$ *updated according to the following expression*:

$$v_j^{\Phi} = \frac{\sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ik})^m \Phi(x_{kj})}{\sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ik})^m}. \tag{52}$$

(b) *If the adaptive distance function is given by Eq. (35), the overall centroid* $\mathbf{v}^{\Phi}$ *has its components* v_j^{Φ} $(j = 1, \ldots, p)$ *updated according to the following expression*:

$$v_j^{\Phi} = \frac{\sum_{i=1}^{c}(\lambda_{ij})^{\beta}\sum_{k=1}^{n}(u_{ik})^m \Phi(x_{kj})}{\sum_{i=1}^{c}(\lambda_{ij})^{\beta}\sum_{k=1}^{n}(u_{ik})^m}. \tag{53}$$

(c) *If the adaptive distance function is given by Eq. (37), the overall centroid* $\mathbf{v}^{\Phi}$ *has its components* v_j^{Φ} $(j = 1, \ldots, p)$ *updated according to the following expression*:

$$v_j^{\Phi} = \frac{\sum_{i=1}^{c}\lambda_{ij}\sum_{k=1}^{n}(u_{ik})^m \Phi(x_{kj})}{\sum_{i=1}^{c}\lambda_{ij}\sum_{k=1}^{n}(u_{ik})^m}. \tag{54}$$

Proof. The proof can be obtained in a similar way as presented in Proposition 3.5. □

It is important to note that the overall cluster centroid does not need to be computed because the centroid information is implicitly considered, i.e., the term $\|\Phi(x_{kj}) - v_j^{\Phi}\|^2$ in the feature space is computed through the kernel in the original space in a similar way to that presented in Eq. (44). Thus, adequately replacing v_j^{Φ} in the distance functions presented in Section 3.1.3 (Eqs. (34), (35), (36), (37) or (38)), we can rewrite T only through kernel functions.

If the adaptive distance function is given in the form of Eqs. (34), (36) or (38), then the overall centroid is given by Eq. (52) and the distance $\|\Phi(x_{kj}) - v_j^{\Phi}\|^2$ is computed in the following way:

$$\begin{aligned}
\left\|\Phi(x_{kj}) - v_j^{\Phi}\right\|^2 &= \Phi(x_{kj})\cdot\Phi(x_{kj}) - 2\Phi(x_{kj})\cdot v_j^{\Phi} + v_j^{\Phi}\cdot v_j^{\Phi} \\
&= \Phi(x_{kj})\cdot\Phi(x_{kj}) - 2\Phi(x_{kj})\cdot\frac{\sum_{h=1}^{c}\sum_{l=1}^{n}(u_{hl})^m\Phi(x_{lj})}{\sum_{h=1}^{c}\sum_{l=1}^{n}(u_{hl})^m} \\
&\quad + \frac{\sum_{h=1}^{c}\sum_{r=1}^{n}(u_{hr})^m\Phi(x_{rj})}{\sum_{h=1}^{c}\sum_{r=1}^{n}(u_{hr})^m}\cdot\frac{\sum_{w=1}^{c}\sum_{s=1}^{n}(u_{ws})^m\Phi(x_{sj})}{\sum_{w=1}^{c}\sum_{s=1}^{n}(u_{ws})^m} \\
&= \Phi(x_{kj})\cdot\Phi(x_{kj}) - \frac{2\sum_{h=1}^{c}\sum_{l=1}^{n}(u_{hl})^m\Phi(x_{lj})\cdot\Phi(x_{kj})}{\sum_{h=1}^{c}\sum_{l=1}^{n}(u_{hl})^m} \\
&\quad + \frac{\sum_{h=1}^{c}\sum_{w=1}^{c}\sum_{r=1}^{n}\sum_{s=1}^{n}(u_{hr})^m(u_{ws})^m\Phi(x_{rj})\cdot\Phi(x_{sj})}{(\sum_{h=1}^{c}\sum_{r=1}^{n}(u_{hr})^m)^2} \\
&= K(x_{kj}, x_{kj}) - \frac{2\sum_{h=1}^{c}\sum_{l=1}^{n}(u_{hl})^m K(x_{lj}, x_{kj})}{\sum_{h=1}^{c}\sum_{l=1}^{n}(u_{hl})^m} \\
&\quad + \frac{\sum_{h=1}^{c}\sum_{w=1}^{c}\sum_{r=1}^{n}\sum_{s=1}^{n}(u_{hr})^m(u_{ws})^m K(x_{rj}, x_{sj})}{(\sum_{h=1}^{c}\sum_{r=1}^{n}(u_{hr})^m)^2}.
\end{aligned} \tag{55}$$

Similarly, if the adaptive distance function is given in the form of Eq. (35), then the overall centroid is given by Eq. (53) and the distance $\|\Phi(x_{kj}) - v_j^{\Phi}\|^2$ is computed as:

$$\begin{aligned}
\left\|\Phi(x_{kj}) - v_j^{\Phi}\right\|^2 &= K(x_{kj}, x_{kj}) - \frac{2\sum_{h=1}^{c}(\lambda_{hj})^{\beta}\sum_{l=1}^{n}(u_{hl})^m K(x_{lj}, x_{kj})}{\sum_{h=1}^{c}(\lambda_{hj})^{\beta}\sum_{l=1}^{n}(u_{hl})^m} \\
&\quad + \frac{\sum_{h=1}^{c}(\lambda_{hj})^{\beta}\sum_{w=1}^{c}(\lambda_{wj})^{\beta}\sum_{r=1}^{n}\sum_{s=1}^{n}(u_{hr})^m(u_{ws})^m K(x_{rj}, x_{sj})}{(\sum_{h=1}^{c}(\lambda_{hj})^{\beta}\sum_{r=1}^{n}(u_{hr})^m)^2}.
\end{aligned} \tag{56}$$

And finally, if the adaptive distance function is given in the form of Eq. (37), then the overall centroid is given by Eq. (54) and the distance $\|\Phi(x_{kj}) - v_j^{\Phi}\|^2$ is computed in the following way:

$$\left\|\Phi(x_{kj}) - v_j^{\Phi}\right\|^2 = K(x_{kj}, x_{kj}) - \frac{2\sum_{h=1}^{c}\lambda_{hj}\sum_{l=1}^{n}(u_{hl})^m K(x_{lj}, x_{kj})}{\sum_{h=1}^{c}\lambda_{hj}\sum_{l=1}^{n}(u_{hl})^m} + \frac{\sum_{h=1}^{c}\lambda_{hj}\sum_{w=1}^{c}\lambda_{wj}\sum_{r=1}^{n}\sum_{s=1}^{n}(u_{hr})^m(u_{ws})^m K(x_{rj}, x_{sj})}{(\sum_{h=1}^{c}\lambda_{hj}\sum_{r=1}^{n}(u_{hr})^m)^2}. \tag{57}$$

We can easily see that the overall dispersion T, given in Eq. (51), also decomposes according to variables ($T = \sum_{j=1}^{p} T_j$), and according to clusters ($T = \sum_{i=1}^{c} T_i$), as well as according to clusters and variables ($T = \sum_{i=1}^{c}\sum_{j=1}^{p} T_{ij}$).

Similarly, the overall heterogeneity within-clusters is measured by the within-cluster dispersion, which is given in Eq. (33).

This measure also decomposes according to variables ($J = \sum_{j=1}^{p} J_j$), and according to clusters ($J = \sum_{i=1}^{c} J_i$), as well as according to clusters and variables ($J = \sum_{i=1}^{c}\sum_{j=1}^{p} J_{ij}$).

5.3. Interpretation indexes

In this section we present suitable adaptations to the indexes proposed in [42] to the variable-wise kernel fuzzy clustering algorithms presented in Section 3.

We can easily see that: $T \geqslant J$, $T_i \geqslant J_i$ $(i = 1, \ldots, c)$, $T_j \geqslant J_j$ $(j = 1, \ldots, p)$, and $T_{ij} \geqslant J_{ij}$ $(i = 1, \ldots, c,\ j = 1, \ldots, p)$.

5.3.1. Fuzzy partition interpretation indexes

Interpreting the overall quality of a fuzzy partition after having applied a clustering algorithm to the data is an important problem in clustering analysis.

Overall heterogeneity index The quality of a fuzzy partition P is measured by the difference between the overall dispersion without clustering (T) and the overall dispersion after clustering (within-cluster dispersion J) normalized by the overall dispersion without clustering:

$$Q(P) = \frac{T - J}{T} = 1 - \frac{J}{T}. \tag{58}$$

This index takes its values between 0 and 1. It is equal to one when all the fuzzy clusters have just a single pattern and is equal to 0 for a fuzzy partition of a single cluster or when the fuzzy clusters centroids ($\mathbf{v}_i$ in the approach of kernelization of the metric, or $\mathbf{v}_i^{\Phi}$ in the approach of clustering in feature space, $i = 1, \ldots, c$) are equal to the overall centroid ($\mathbf{v}$, in the approach of kernelization of the metric, or $\mathbf{v}^{\Phi}$, in the approach of clustering in feature space). A value of Q closer to 1 means more homogeneous clusters and a better representation of the elements of a fuzzy cluster P_i by its centroid. This means that the set of variables have a high discriminant power, they are able to well separate the data set into homogeneous clusters. A value of Q closer to 0 means fuzzy clusters centroids very similar to the overall centroid and a poor representation of the elements of a fuzzy cluster P_i by its centroid. This means that the set of variables have a low discriminant power (the fuzzy clusters centroids are very similar to the overall centroid), they are not able to well separate the data set into homogeneous clusters. Moreover, for a given clustering method, because this index decreases with the number of clusters, it can only be used to compare partitions having the same number of clusters: a partition P in c clusters is better than a partition P' also in c clusters if $Q(P) > Q(P')$.

Overall heterogeneity index regarding single variables The quality of a fuzzy partition P concerning the j-th variable is measured by the difference between the variable-specific overall dispersion without clustering concerning the j-th variable (T_j) and the variable-specific overall dispersion after clustering concerning the j-th variable (variable-specific within-cluster dispersion J_j) normalized by the variable-specific overall dispersion without clustering concerning the j-th variable:

$$Q_j(P) = \frac{T_j - J_j}{T_j} = 1 - \frac{J_j}{T_j}, \quad j = 1, \ldots, p. \tag{59}$$

Please cite this article in press as: M.R.P. Ferreira, F.A.T. de Carvalho, Kernel fuzzy c-means with automatic variable weighting, Fuzzy Sets and Systems (2013), http://dx.doi.org/10.1016/j.fss.2013.05.004

This index takes also its values between 0 and 1. A value of Q_j closer to 1 denotes better quality of a fuzzy partition P concerning the j-th variable. This means that the j-th variable has a high discriminant power, the data set is well separated into homogeneous clusters for that variable. A value of Q_j closer to 0 denotes poor quality of a fuzzy partition P concerning the j-th variable. This means that the j-th variable has a low discriminant power (the fuzzy clusters centroids are very similar to the overall centroid concerning the j-th variable), the data set is not well separated into homogeneous clusters for that variable.

By comparing the value of Q_j with the value of the general index Q one may determine whether the discriminant power of the j-th variable is above or below the discriminant power of the set of variables.

5.3.2. Fuzzy cluster interpretation indexes

Another important problem in clustering analysis is evaluating the homogeneity of the individual clusters of a partition after having applied a clustering method to the data.

Fuzzy cluster heterogeneity indexes The relative contribution of the fuzzy cluster P_i to the overall within-cluster dispersion is given by

$$J(i) = \frac{J_i}{J}, \quad i = 1, \ldots, c. \tag{60}$$

Note that $0 \leqslant J(i) \leqslant 1$ and $\sum_{i=1}^{c} J(i) = 1$. A value of $J(i)$ closer to 1 indicates that there is a greater contribution from the fuzzy cluster P_i to the overall within-cluster dispersion, i.e., a relatively large value of $J(i)$ indicates that the fuzzy cluster P_i is relatively heterogeneous in comparison with the other clusters.

The quality of a fuzzy cluster P_i $(i = 1, \ldots, c)$ is measured by the difference between the cluster-specific overall dispersion without clustering (T_i) and the cluster-specific overall dispersion after clustering (cluster-specific within-cluster dispersion J_i) normalized by the cluster-specific overall dispersion without clustering:

$$Q(P_i) = \frac{T_i - J_i}{T_i} = 1 - \frac{J_i}{T_i}, \quad i = 1, \ldots, c. \tag{61}$$

This index takes also its values between 0 and 1. This index measures the gain in homogeneity of the fuzzy cluster P_i obtained when replacing the overall centroid ($\mathbf{v}$, in the approach of kernelization of the metric, or $\mathbf{v}^{\Phi}$, in the approach of clustering in feature space) by the cluster-specific centroid ($\mathbf{v}_i$ in the approach of kernelization of the metric, or $\mathbf{v}_i^{\Phi}$ in the approach of clustering in feature space). A value of $Q(P_i)$ closer to 0 means that the fuzzy cluster centroid is very similar to the overall centroid and denotes a fuzzy cluster of poor quality. A value of $Q(P_i)$ closer to 1 denotes a fuzzy cluster of better quality.

Fuzzy cluster heterogeneity index regarding single variables The quality of the fuzzy cluster P_i $(i = 1, \ldots, c)$ concerning the j-th variable is measured by the difference between the cluster-variable-specific overall dispersion without clustering concerning the j-th variable (T_{ij}) and the cluster-variable-specific overall dispersion after clustering concerning the j-th variable (cluster-variable-specific within-cluster dispersion J_{ij}) normalized by the cluster-variable-specific overall dispersion without clustering concerning the j-th variable:

$$Q_j(P_i) = \frac{T_{ij} - J_{ij}}{T_{ij}} = 1 - \frac{J_{ij}}{T_{ij}}, \quad i = 1, \ldots, c, \ j = 1, \ldots, p. \tag{62}$$

This index takes also its values between 0 and 1. This index measures the gain in homogeneity of the fuzzy cluster P_i for the j-th variable obtained when replacing the j-th component of the overall centroid (v_j, in the approach of kernelization of the metric, or v_j^{Φ}, in the approach of clustering in feature space) by the j-th component of the cluster-specific centroid (v_{ij} in the approach of kernelization of the metric, or v_{ij}^{Φ} in the approach of clustering in feature space). A value of $Q_j(P_i)$ closer to 0 means that the j-th component of the fuzzy cluster centroid is very similar to the j-th component of the overall centroid and denotes a fuzzy cluster of poor quality concerning the j-th variable. A value of $Q_j(P_i)$ closer to 1 denotes better quality of the cluster P_i concerning the j-th variable.

Moreover, this index helps the user find the variables that characterize the cluster P_i. The values of the index $Q_j(P_i)$ have to be interpreted by a comparison with the value $Q(P_i)$: the j-th variable characterizes the cluster P_i if $Q_j(P_i) > Q(P_i)$.

Please cite this article in press as: M.R.P. Ferreira, F.A.T. de Carvalho, Kernel fuzzy c-means with automatic variable weighting, Fuzzy Sets and Systems (2013), http://dx.doi.org/10.1016/j.fss.2013.05.004

6. Experimental evaluation

To evaluate the performance of the proposed algorithms in comparison with its conventional counterparts, applications with synthetic datasets as well as benchmark datasets selected from the UCI Machine Learning Repository were considered.

To compare the clustering results furnished by the clustering algorithms considered in this paper, an external index, the Corrected Rand (CR) index [43], as well as the F-measure and the Overall Error Rate of Classification (OERC) [44] were computed for the best results selected according to the clustering adequacy criterion.

The CR index assesses the degree of agreement (similarity) between an a priori partition and a partition furnished by a clustering algorithm. Furthermore, the CR index is not sensitive to the number of classes in the partitions or the distribution of the items in the clusters. Finally, 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 [45].

The F-measure index takes its values from the $[0, 1]$ interval, in which the value 1 indicates perfect agreement between partitions.

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.

6.1. Synthetic datasets

To illustrate the ability of the variable-wise kernel fuzzy c-means clustering methods based on adaptive distances to dynamically learn appropriate relevance weights of the variables, we considered two configurations of three-dimensional real-valued data with four and three classes, respectively, each class of size 100. For each synthetic dataset, the CR index, F-measure and OERC were estimated in the framework of a Monte Carlo simulation with 100 replications. The average and the standard deviation of these measures based on 100 Monte Carlo replications were computed. 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 method was selected according to the adequacy criterion. The fuzzification parameter m was set equal to 2.0. Several values were tested for the parameter β and the value which furnished the best results was $\beta = 3.0$. The term $2\sigma^2$ in the Gaussian kernel used in the conventional kernel fuzzy clustering methods was estimated as the mean of the 0.1 and 0.9 quantiles of $\|\mathbf{x}_i - \mathbf{x}_k\|^2$, $i \neq k$ [46]. We set $\varepsilon = 10^{-10}$ as the tolerance for the convergence of the adequacy criterion. In the variable-wise kernel fuzzy c-means clustering methods, the terms $2\sigma_j^2$ $(j = 1, \ldots, p)$ in the Gaussian kernels were estimated for each variable as the mean of the 0.1 and 0.9 quantiles of $\|x_{ij} - x_{kj}\|^2$, $i \neq k$. For each dataset, the number of clusters is set equal the number of classes. From the fuzzy partition given by these clustering algorithms it is obtained a hard partition by assigning each object to a hard cluster as follows: object $\mathbf{x}_k$ is assigned to cluster P_i if $i = \arg\max_{1 \leqslant h \leqslant K} u_{hk}$.

6.1.1. Synthetic dataset 1

The synthetic dataset 1 (Fig. 1a) has 400 instances in three dimensions and is divided in four classes of 100 instances drawn from 3-dimensional Gaussian distributions with a specific mean vector and a specific covariance matrix for each class. Each class exists in only two of the three dimensions. The mean vector and the variance structure of each class are shown in Table 1. The correlation between the variables was set equal to zero in all classes.

It can be noted from Fig. 1 that the variables x_1 and x_2 are relevant to define the classes 1 and 2 (Fig. 1b), whereas classes 3 and 4 are clearly defined by the variables x_2 and x_3 (Fig. 1c). This means that, for classes 1 and 2, variable x_3 represents noise, whereas for classes 3 and 4, variable x_1 represents noise, as we can observe in Fig. 1d.

Table 2 shows the performance of the FCM, KFCM-K and KFCM-F clustering algorithms, as well as the performance of the variable-wise kernel fuzzy c-means clustering algorithms proposed in this paper on the synthetic dataset 1 according to the CR index, F-measure and OERC. The performance of the clustering algorithms based on local adaptive distances (VKFCM-K-LS, VKFCM-K-LP, VKFCM-F-LS and VKFCM-F-LP) was clearly superior when there is a specific set of relevant variables to each cluster, in comparison with all the other algorithms.

Please cite this article in press as: M.R.P. Ferreira, F.A.T. de Carvalho, Kernel fuzzy c-means with automatic variable weighting, Fuzzy Sets and Systems (2013), http://dx.doi.org/10.1016/j.fss.2013.05.004

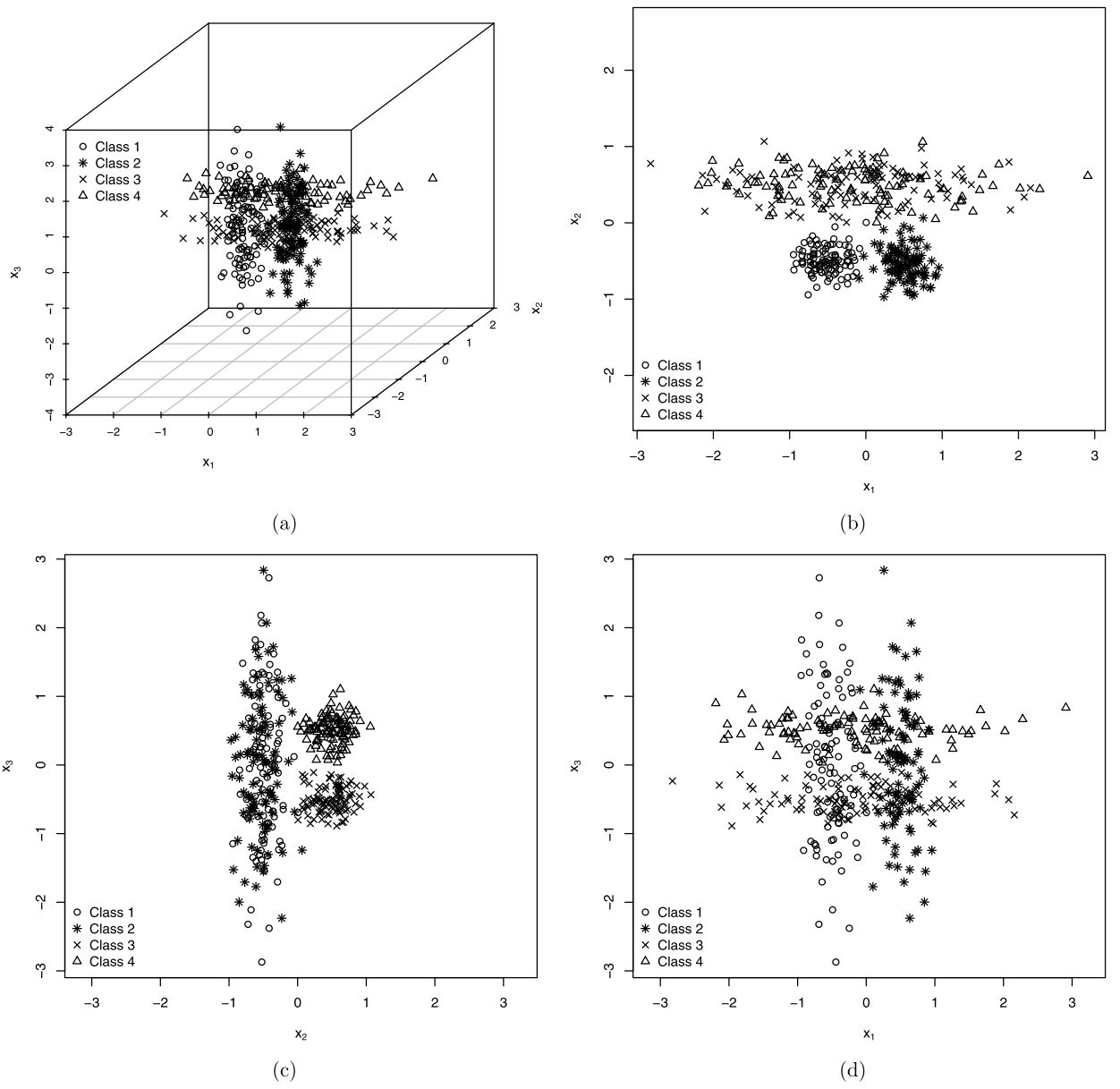

Fig. 1. Synthetic dataset 1.

Table 1
Configuration of the synthetic dataset 1.

$\boldsymbol{\mu}$	Class 1	Class 2	Class 3	Class 4
μ_1	−0.5	0.5	0.0	0.0
μ_2	−0.5	−0.5	0.5	0.5
μ_3	0.0	0.0	−0.5	0.5
$\boldsymbol{\Sigma}$	Class 1	Class 2	Class 3	Class 4
σ_1^2	0.04	0.04	1.00	1.00
σ_2^2	0.04	0.04	0.04	0.04
σ_3^2	1.00	1.00	0.04	0.04

Please cite this article in press as: M.R.P. Ferreira, F.A.T. de Carvalho, Kernel fuzzy c-means with automatic variable weighting, Fuzzy Sets and Systems (2013), http://dx.doi.org/10.1016/j.fss.2013.05.004

Table 2
Performance of the algorithms on the synthetic dataset 1: average and standard deviation (in parenthesis) of the CR index, F-measure and OERC.

	CR index	F-measure	OERC
FCM	0.2338 (0.0273)	0.5441 (0.0352)	0.4558 (0.0351)
KFCM-K	0.3368 (0.0651)	0.6549 (0.0483)	0.3455 (0.0476)
VKFCM-K	0.4553 (0.0887)	0.7052 (0.0596)	0.2946 (0.0595)
VKFCM-K-LS	**0.9452 (0.0444)**	**0.9779 (0.0266)**	**0.0220 (0.0263)**
VKFCM-K-LP	**0.9729 (0.0149)**	**0.9898 (0.0057)**	**0.0102 (0.0057)**
VKFCM-K-GS	0.3260 (0.0070)	0.5332 (0.0162)	0.4698 (0.0181)
VKFCM-K-GP	0.5252 (0.1106)	0.7132 (0.0620)	0.2871 (0.0618)
KFCM-F	0.1661 (0.0667)	0.4810 (0.0456)	0.5403 (0.0671)
VKFCM-F	0.5632 (0.1464)	0.7549 (0.0973)	0.2450 (0.0973)
VKFCM-F-LS	**0.9345 (0.0637)**	**0.9719 (0.0401)**	**0.0283 (0.0413)**
VKFCM-F-LP	**0.9736 (0.0149)**	**0.9900 (0.0057)**	**0.0010 (0.0057)**
VKFCM-F-GS	0.3260 (0.0069)	0.5334 (0.0169)	0.4690 (0.0184)
VKFCM-F-GP	0.5813 (0.1089)	0.7476 (0.0707)	0.2523 (0.0707)

6.1.2. Synthetic dataset 2

The synthetic dataset 2 (Fig. 2a) has 300 instances in three dimensions and is divided in three classes of 100 instances. Variables x_1 and x_2 were drawn from bivariate Gaussian distributions with a specific mean vector and a specific covariance matrix for each class. Variable x_3 was obtained as a linear combination of the variables x_1 and x_2 plus a noise drawn from a standard Gaussian distribution, i.e., $x_{3i} = 2x_{1i} - 1.5x_{2i} + u_i$, where $u_i \sim N(0, 1)$, $i = 1, \ldots, n$. The mean vector and the variance structure of each class concerning the variables x_1 and x_2 are shown in Table 3. The correlation between the variables x_1 and x_2 was set equal to zero in all classes. Obviously, x_1 and x_3 are positively correlated on each class, whereas x_2 and x_3 are negatively correlated on each class.

It can be noted from Fig. 2 that the variables x_1 and x_2 are relevant to define the classes 1, 2 and 3 (Fig. 2b), i.e., the set of relevant variables is the same to all clusters. It can be also noted that, for all classes, variable x_3 represents noise, as we can observe in Figs. 2c and 2d.

Table 4 shows the performance of the FCM, KFCM-K and KFCM-F clustering algorithms, as well as the performance of the variable-wise kernel fuzzy c-means clustering algorithms proposed in this paper on the synthetic dataset 2 according to the CR index, F-measure and OERC. It can be seen that when there are noisy or irrelevant variables and the set of relevant variables is the same to all clusters, the variable-wise kernel fuzzy clustering algorithms performs better than the traditional clustering methods considered in this paper. The performance of the clustering algorithms based on global adaptive distances (VKFCM-K-GS, VKFCM-K-GP, VKFCM-F-GS and VKFCM-F-GP) was superior in comparison with all the other algorithms.

The significance of the differences between the average of the CR index, F-measure and OERC in the framework of the Monte Carlo experiment was tested using a suitable one-sided Student's t-test for paired samples and a 5% level of significance was adopted. The results of these comparisons between the clustering algorithms on the synthetic datasets 1 and 2 are shown in Table 5. Suitable tests were performed with the clustering algorithms listed in the first column against all other clustering algorithms listed in the second column. In this table, the symbol "=" means that the difference between the average of the indexes concerning a pair of clustering algorithms is not statistically significant. The symbol "−" means that the performance of the clustering algorithm in the first column is inferior to the performance of the clustering algorithm in the second column. Finally, the symbol "+" means that the performance of the clustering algorithm in the first column is superior to the performance of the clustering algorithm in the second column.

As was pointed out, the variable-wise kernel fuzzy c-means algorithms based on local adaptive distances (VKFCM-K-LS, VKFCM-K-LP, VKFCM-F-LS and VKFCM-F-LP) achieved a better performance than the other methods on the synthetic dataset 1. On this dataset, according to the paired t-tests, the VKFCM-K-LP and VKFCM-F-LP algorithms presented the best performances, whereas the FCM, KFCM-K and KFCM-F algorithms presented the worst performances. Moreover, the variable-wise kernel fuzzy c-means algorithms based on global adaptive distances (VKFCM-K-GS, VKFCM-K-GP, VKFCM-F-GS and VKFCM-F-GP) achieved a better performance than the other methods on the synthetic dataset 2. On this dataset, according to the paired t-tests, the VKFCM-K-GP and

Please cite this article in press as: M.R.P. Ferreira, F.A.T. de Carvalho, Kernel fuzzy c-means with automatic variable weighting, Fuzzy Sets and Systems (2013), http://dx.doi.org/10.1016/j.fss.2013.05.004

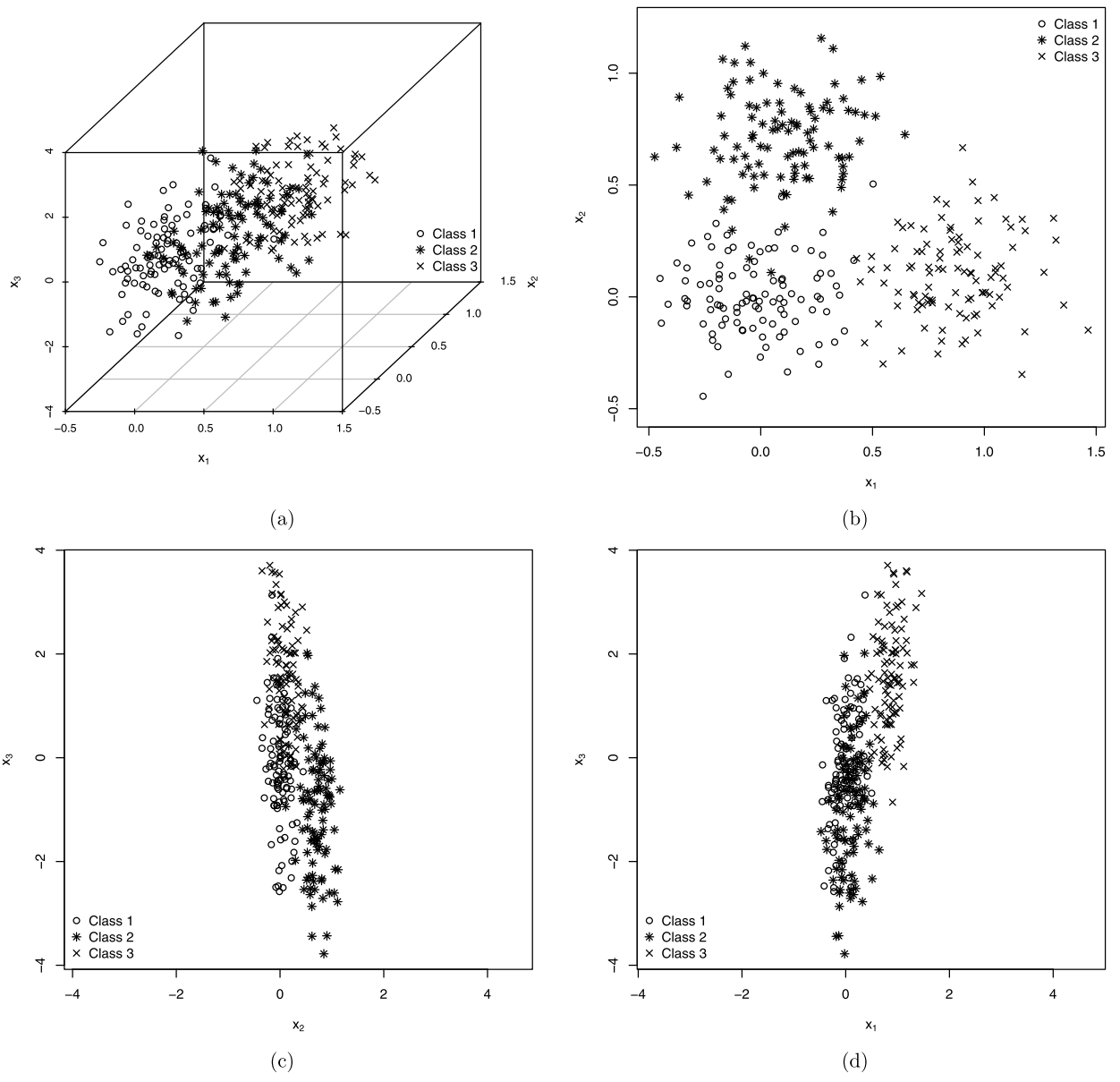

Fig. 2. Synthetic dataset 2.

Table 3
Configuration of the synthetic dataset 2.

$\boldsymbol{\mu}$	Class 1	Class 2	Class 3
μ_1	0.0	0.1	0.9
μ_2	0.0	0.7	0.1
$\boldsymbol{\Sigma}$	Class 1	Class 2	Class 3
σ_1^2	0.04	0.04	0.04
σ_2^2	0.04	0.04	0.04

Please cite this article in press as: M.R.P. Ferreira, F.A.T. de Carvalho, Kernel fuzzy c-means with automatic variable weighting, Fuzzy Sets and Systems (2013), http://dx.doi.org/10.1016/j.fss.2013.05.004

Table 4
Performance of the algorithms on the synthetic dataset 2: average and standard deviation (in parenthesis) of the CR index, F-measure and OERC.

	CR index	F-measure	OERC
FCM	0.2396 (0.0374)	0.3682 (0.0325)	0.6393 (0.0312)
KFCM-K	0.2566 (0.0377)	0.3593 (0.0325)	0.6464 (0.0312)
VKFCM-K	0.8172 (0.0489)	0.0644 (0.0183)	0.9357 (0.0183)
VKFCM-K-LS	0.7297 (0.1721)	0.1222 (0.1003)	0.8782 (0.0991)
VKFCM-K-LP	0.8490 (0.0437)	0.0530 (0.0162)	0.9470 (0.0162)
VKFCM-K-GS	**0.8731 (0.0392)**	**0.0441 (0.0142)**	**0.9559 (0.0141)**
VKFCM-K-GP	**0.8594 (0.0407)**	**0.0490 (0.0148)**	**0.9510 (0.0148)**
KFCM-F	0.2579 (0.0380)	0.3587 (0.0324)	0.6470 (0.0311)
VKFCM-F	0.8221 (0.0475)	0.0626 (0.0177)	0.9375 (0.0177)
VKFCM-F-LS	0.6269 (0.1738)	0.1807 (0.1072)	0.8213 (0.1040)
VKFCM-F-LP	0.8410 (0.0515)	0.0562 (0.0196)	0.9438 (0.0195)
VKFCM-F-GS	**0.8633 (0.0626)**	**0.0489 (0.0315)**	**0.9512 (0.0310)**
VKFCM-F-GP	**0.8581 (0.0423)**	**0.0495 (0.0154)**	**0.9505 (0.0154)**

Table 5
Comparison between the clustering algorithms on the synthetic datasets 1 and 2 according to a Student's t-test for paired samples at a 5% significance level.

		Synthetic dataset 1			Synthetic dataset 2		
		CR	F-measure	OERC	CR	F-measure	OERC
FCM	KFCM-K	−	−	−	−	−	−
	VKFCM-K	−	−	−	−	−	−
	VKFCM-K-LS	−	−	−	−	−	−
	VKFCM-K-LP	−	−	−	−	−	−
	VKFCM-K-GS	−	+	+	−	−	−
	VKFCM-K-GP	−	−	−	−	−	−
	KFCM-F	+	+	+	−	−	−
	VKFCM-F	−	−	−	−	−	−
	VKFCM-F-LS	−	−	−	−	−	−
	VKFCM-F-LP	−	−	−	−	−	−
	VKFCM-F-GS	−	+	+	−	−	−
	VKFCM-F-GP	−	−	−	−	−	−
KFCM-K	VKFCM-K	−	−	−	−	−	−
	VKFCM-K-LS	−	−	−	−	−	−
	VKFCM-K-LP	−	−	−	−	−	−
	VKFCM-K-GS	=	+	+	−	−	−
	VKFCM-K-GP	−	−	−	−	−	−
	KFCM-F	+	+	+	−	−	−
	VKFCM-F	−	−	−	−	−	−
	VKFCM-F-LS	−	−	−	−	−	−
	VKFCM-F-LP	−	−	−	−	−	−
	VKFCM-F-GS	=	+	+	−	−	−
	VKFCM-F-GP	−	−	−	−	−	−
VKFCM-K	VKFCM-K-LS	−	−	−	+	+	+
	VKFCM-K-LP	−	−	−	−	−	−
	VKFCM-K-GS	+	+	+	−	−	−
	VKFCM-K-GP	−	=	=	−	−	−
	KFCM-F	+	+	+	+	+	+
	VKFCM-F	−	−	−	−	−	−
	VKFCM-F-LS	−	−	−	+	+	+
	VKFCM-F-LP	−	−	−	−	−	−
	VKFCM-F-GS	+	+	+	−	−	−
	VKFCM-F-GP	−	−	−	−	−	−

(*continued on next page*)

Please cite this article in press as: M.R.P. Ferreira, F.A.T. de Carvalho, Kernel fuzzy c-means with automatic variable weighting, Fuzzy Sets and Systems (2013), http://dx.doi.org/10.1016/j.fss.2013.05.004

Table 5 (*Continued.*)

		Synthetic dataset 1			Synthetic dataset 2		
		CR	F-measure	OERC	CR	F-measure	OERC
VKFCM-K-LS	VKFCM-K-LP	−	−	−	−	−	−
	VKFCM-K-GS	+	+	+	−	−	−
	VKFCM-K-GP	+	+	+	−	−	−
	KFCM-F	+	+	+	+	+	+
	VKFCM-F	+	+	+	−	−	−
	VKFCM-F-LS	+	=	=	+	+	+
	VKFCM-F-LP	−	−	−	−	−	−
	VKFCM-F-GS	+	+	+	−	−	−
	VKFCM-F-GP	+	+	+	−	−	−
VKFCM-K-LP	VKFCM-K-GS	+	+	+	−	−	−
	VKFCM-K-GP	+	+	+	−	−	−
	KFCM-F	+	+	+	+	+	+
	VKFCM-F	+	+	+	+	+	+
	VKFCM-F-LS	+	+	+	+	+	+
	VKFCM-F-LP	=	=	=	+	+	+
	VKFCM-F-GS	+	+	+	−	=	=
	VKFCM-F-GP	+	+	+	−	−	−
VKFCM-K-GS	VKFCM-K-GP	−	−	−	+	+	+
	KFCM-F	+	+	+	+	+	+
	VKFCM-F	−	−	−	+	+	+
	VKFCM-F-LS	−	−	−	+	+	+
	VKFCM-F-LP	−	−	−	+	+	+
	VKFCM-F-GS	=	=	=	+	=	=
	VKFCM-F-GP	−	−	−	+	+	+
VKFCM-K-GP	KFCM-F	+	+	+	+	+	+
	VKFCM-F	−	−	−	+	+	+
	VKFCM-F-LS	−	−	−	+	+	+
	VKFCM-F-LP	−	−	−	+	+	+
	VKFCM-F-GS	+	+	+	=	=	=
	VKFCM-F-GP	−	−	−	=	=	=
KFCM-F	VKFCM-F	−	−	−	−	−	−
	VKFCM-F-LS	−	−	−	−	−	−
	VKFCM-F-LP	−	−	−	−	−	−
	VKFCM-F-GS	−	−	−	−	−	−
	VKFCM-F-GP	−	−	−	−	−	−
VKFCM-F	VKFCM-F-LS	−	−	−	+	+	+
	VKFCM-F-LP	−	−	−	−	−	−
	VKFCM-F-GS	+	+	+	−	−	−
	VKFCM-F-GP	=	=	=	−	−	−
VKFCM-F-LS	VKFCM-F-LP	−	−	−	−	−	−
	VKFCM-F-GS	+	+	+	−	−	−
	VKFCM-F-GP	+	+	+	−	−	−
VKFCM-F-LP	VKFCM-F-GS	+	+	+	−	−	−
	VKFCM-F-GP	+	+	+	−	−	−
VKFCM-F-GS	VKFCM-F-GP	−	−	−	=	=	=

VKFCM-F-GP algorithms presented the best performances, whereas the FCM, KFCM-K and KFCM-F algorithms presented the worst performances.

Please cite this article in press as: M.R.P. Ferreira, F.A.T. de Carvalho, Kernel fuzzy c-means with automatic variable weighting, Fuzzy Sets and Systems (2013), http://dx.doi.org/10.1016/j.fss.2013.05.004

Table 6
Performance of the algorithms on the synthetic dataset 1 with 10 additional irrelevant variables: average and standard deviation (in parenthesis) of the CR index, F-measure and OERC.

	CR index	F-measure	OERC
FCM	0.0036 (0.0071)	0.3301 (0.0190)	0.6983 (0.0203)
KFCM-K	0.0027 (0.0061)	0.3220 (0.0175)	0.6983 (0.0176)
VKFCM-K	**0.4864** (**0.0062**)	**0.6627** (**0.0021**)	**0.5005** (**0.0010**)
VKFCM-K-LS	**0.4843** (**0.0064**)	**0.6620** (**0.0021**)	**0.5017** (**0.0015**)
VKFCM-K-LP	**0.4861** (**0.0069**)	**0.6626** (**0.0023**)	**0.5007** (**0.0014**)
VKFCM-K-GS	**0.4824** (**0.0087**)	**0.6613** (**0.0030**)	**0.5013** (**0.0022**)
VKFCM-K-GP	**0.4844** (**0.0059**)	**0.6620** (**0.0020**)	**0.5010** (**0.0016**)
KFCM-F	0.0027 (0.0049)	0.3213 (0.0147)	0.7000 (0.0131)
VKFCM-F	**0.4240** (**0.0488**)	**0.6308** (**0.0266**)	**0.4753** (**0.0323**)
VKFCM-F-LS	**0.2335** (**0.1057**)	**0.5293** (**0.0691**)	**0.5225** (**0.0455**)
VKFCM-F-LP	**0.4093** (**0.0406**)	**0.6260** (**0.0255**)	**0.4855** (**0.0280**)
VKFCM-F-GS	**0.3006** (**0.1093**)	**0.5655** (**0.0711**)	**0.4831** (**0.0650**)
VKFCM-F-GP	**0.3505** (**0.1090**)	**0.5930** (**0.0679**)	**0.4806** (**0.0560**)

Table 7
Performance of the algorithms on the synthetic dataset 2 with 10 additional irrelevant variables: average and standard deviation (in parenthesis) of the CR index, F-measure and OERC.

	CR index	F-measure	OERC
FCM	0.2550 (0.0260)	0.6220 (0.0132)	0.4176 (0.0126)
KFCM-K	0.2619 (0.0344)	0.6256 (0.0165)	0.4155 (0.0164)
VKFCM-K	0.2951 (0.0743)	0.6261 (0.0432)	0.3814 (0.0339)
VKFCM-K-LS	**0.4013** (**0.0314**)	**0.6750** (**0.0195**)	**0.3514** (**0.0123**)
VKFCM-K-LP	**0.3850** (**0.0574**)	**0.6669** (**0.0331**)	**0.3568** (**0.0266**)
VKFCM-K-GS	**0.3192** (**0.0723**)	**0.6335** (**0.0367**)	**0.3710** (**0.0339**)
VKFCM-K-GP	**0.3292** (**0.0860**)	**0.6386** (**0.0417**)	**0.3734** (**0.0387**)
KFCM-F	0.2567 (0.0327)	0.6220 (0.0178)	0.4151 (0.0135)
VKFCM-F	**0.4148** (**0.0245**)	**0.6861** (**0.0145**)	**0.3457** (**0.0087**)
VKFCM-F-LS	**0.4067** (**0.0921**)	**0.6831** (**0.0538**)	**0.3472** (**0.0480**)
VKFCM-F-LP	**0.4324** (**0.0278**)	**0.7006** (**0.0172**)	**0.3469** (**0.0096**)
VKFCM-F-GS	**0.3891** (**0.0700**)	**0.6685** (**0.0391**)	**0.3464** (**0.0415**)
VKFCM-F-GP	**0.4158** (**0.0276**)	**0.6884** (**0.0149**)	**0.3433** (**0.0190**)

6.1.3. Further evaluation

To evaluate the performance of the clustering algorithms in the case where the number of irrelevant variables is much greater than the number of the relevant variables, we carried out another Monte Carlo experiment considering again the synthetic datasets 1 and 2, but including 10 additional irrelevant variables. These additional irrelevant variables were drawn from standard Gaussian distributions and doesn't have cluster information.

Table 6 shows the performance of the clustering algorithms on the synthetic dataset 1 with 10 additional irrelevant variables according to the CR index, F-measure and OERC. The performance of the variable-wise kernel fuzzy c-means algorithms on this dataset was superior in comparison with all the other algorithms, even if the number of irrelevant variables is much greater than the number of relevant variables. However, there was a considerable decrease in the performance of the variable-wise algorithms based on local adaptive distances, specially in the performance of the VKFCM-F-LS algorithm.

Table 7 shows the performance of the clustering algorithms on the synthetic dataset 2 with 10 additional irrelevant variables according to the CR index, F-measure and OERC. Once again, it can be seen that there was a remarkable decrease in the performance of the variable-wise kernel fuzzy c-means algorithms.

The comparisons between the clustering algorithms on the synthetic datasets 1 and 2 with 10 additional irrelevant variables performed via suitable one-sided Student's t-tests for paired samples at a 5% significance level are shown in Table 8. The variable-wise kernel fuzzy c-means algorithms under the approach of kernelization of the metric (VKFCM-K, VKFCM-K-LS, VKFCM-K-LP, VKFCM-K-GS and VKFCM-K-GP) presented the best performances on the synthetic dataset 1, according to the paired t-tests, whereas the FCM, KFCM-K and KFCM-F algorithms pre-

Please cite this article in press as: M.R.P. Ferreira, F.A.T. de Carvalho, Kernel fuzzy c-means with automatic variable weighting, Fuzzy Sets and Systems (2013), http://dx.doi.org/10.1016/j.fss.2013.05.004

Table 8
Comparison between the clustering algorithms on the synthetic datasets 1 and 2 with 10 additional irrelevant variables according to a Student's t-test for paired samples at a 5% significance level.

		Synthetic dataset 1			Synthetic dataset 2		
		CR	F-measure	OERC	CR	F-measure	OERC
FCM	KFCM-K	=	=	=	=	=	=
	VKFCM-K	−	−	−	−	=	−
	VKFCM-K-LS	−	−	−	−	−	−
	VKFCM-K-LP	−	−	−	−	−	−
	VKFCM-K-GS	−	−	−	−	=	−
	VKFCM-K-GP	−	−	−	−	=	−
	KFCM-F	=	=	=	=	=	=
	VKFCM-F	−	−	−	−	−	−
	VKFCM-F-LS	−	−	−	−	−	−
	VKFCM-F-LP	−	−	−	−	−	−
	VKFCM-F-GS	−	−	−	−	−	−
	VKFCM-F-GP	−	−	−	−	−	−
KFCM-K	VKFCM-K	−	−	−	−	=	−
	VKFCM-K-LS	−	−	−	−	−	−
	VKFCM-K-LP	−	−	−	−	−	−
	VKFCM-K-GS	−	−	−	−	=	−
	VKFCM-K-GP	−	−	−	−	=	−
	KFCM-F	=	=	=	=	=	=
	VKFCM-F	−	−	−	−	−	−
	VKFCM-F-LS	−	−	−	−	−	−
	VKFCM-F-LP	−	−	−	−	−	−
	VKFCM-F-GS	−	−	−	−	−	−
	VKFCM-F-GP	−	−	−	−	−	−
VKFCM-K	VKFCM-K-LS	=	=	=	−	−	−
	VKFCM-K-LP	=	=	=	−	−	−
	VKFCM-K-GS	=	=	=	=	=	=
	VKFCM-K-GP	=	=	=	=	=	=
	KFCM-F	+	+	+	+	=	+
	VKFCM-F	+	+	=	−	−	−
	VKFCM-F-LS	+	+	+	−	−	−
	VKFCM-F-LP	+	+	=	−	−	−
	VKFCM-F-GS	+	+	=	−	−	−
	VKFCM-F-GP	+	+	=	−	−	−
VKFCM-K-LS	VKFCM-K-LP	=	=	=	+	=	=
	VKFCM-K-GS	=	=	=	+	+	+
	VKFCM-K-GP	=	=	=	+	+	+
	KFCM-F	+	+	+	+	+	+
	VKFCM-F	+	+	=	−	−	−
	VKFCM-F-LS	+	+	+	=	=	=
	VKFCM-F-LP	+	+	=	−	−	−
	VKFCM-F-GS	+	+	=	=	=	=
	VKFCM-F-GP	+	+	=	−	−	−
VKFCM-K-LP	VKFCM-K-GS	=	=	=	+	+	=
	VKFCM-K-GP	=	=	=	+	+	+
	KFCM-F	+	+	+	+	+	+
	VKFCM-F	+	+	=	−	−	−
	VKFCM-F-LS	+	+	+	=	=	=
	VKFCM-F-LP	+	+	=	−	−	−
	VKFCM-F-GS	+	+	=	=	=	=
	VKFCM-F-GP	+	+	=	−	−	−

(*continued on next page*)

Please cite this article in press as: M.R.P. Ferreira, F.A.T. de Carvalho, Kernel fuzzy c-means with automatic variable weighting, Fuzzy Sets and Systems (2013), http://dx.doi.org/10.1016/j.fss.2013.05.004

Table 8 (*Continued.*)

		Synthetic dataset 1			Synthetic dataset 2		
		CR	F-measure	OERC	CR	F-measure	OERC
VKFCM-K-GS	VKFCM-K-GP	=	=	=	=	=	=
	KFCM-F	+	+	+	+	=	+
	VKFCM-F	+	+	=	−	−	−
	VKFCM-F-LS	+	+	+	−	−	−
	VKFCM-F-LP	+	+	=	−	−	−
	VKFCM-F-GS	+	+	=	−	−	−
	VKFCM-F-GP	+	+	=	−	−	−
VKFCM-K-GP	KFCM-F	+	+	+	+	=	+
	VKFCM-F	+	+	=	−	−	−
	VKFCM-F-LS	+	+	+	−	−	−
	VKFCM-F-LP	+	+	=	−	−	−
	VKFCM-F-GS	+	+	=	−	−	−
	VKFCM-F-GP	+	+	=	−	−	−
KFCM-F	VKFCM-F	−	−	−	−	−	−
	VKFCM-F-LS	−	−	−	−	−	−
	VKFCM-F-LP	−	−	−	−	−	−
	VKFCM-F-GS	−	−	−	−	−	−
	VKFCM-F-GP	−	−	−	−	−	−
VKFCM-F	VKFCM-F-LS	+	+	+	=	=	=
	VKFCM-F-LP	=	=	=	−	−	=
	VKFCM-F-GS	+	+	=	=	+	=
	VKFCM-F-GP	+	+	=	=	=	=
VKFCM-F-LS	VKFCM-F-LP	−	−	−	=	=	=
	VKFCM-F-GS	−	−	−	=	=	=
	VKFCM-F-GP	−	−	−	=	=	=
VKFCM-F-LP	VKFCM-F-GS	+	+	=	+	+	=
	VKFCM-F-GP	+	+	=	+	+	=
VKFCM-F-GS	VKFCM-F-GP	−	=	=	=	−	=

sented the worst performances. Moreover, the variable-wise kernel fuzzy c-means algorithms in the feature space based on both local and global adaptive distances with the constraint that the product of the weights of the variables must be equal to one (VKFCM-F-LP and VKFCM-F-GP) presented the best performances, according to the paired t-tests, on the synthetic dataset 2, whereas the FCM, KFCM-K and KFCM-F algorithms presented the worst performances over again.

The authors of Refs. [47] and [48] reported some problems with the fuzzy c-means and similar algorithms when we are dealing with high-dimensional datasets and/or a large number of clusters. The main problem reported by the authors was that, in such situations, the cluster centers tend to run into the center of gravity of the entire dataset. They showed that the fuzzy c-means algorithm can be successfully applied to high-dimensional datasets if the centroids are initialized very close to the actual cluster centers. Another way to reduce the effects of high dimensions on the fuzzy c-means algorithm is to appropriately adjust the fuzzification parameter, depending on the number of variables as $m = (2 + p)/p$, where p is the number of variables.

Aiming to obtain better results than that presented when we have a majority of irrelevant variables, we considered $m = (2 + p)/p$ and carried out a Monte Carlo experiment again. In this case, $m \approx 1.15$ and the fuzzy membership matrix became almost hard.

Tables 9 and 10, respectively, show the performance of the FCM, KFCM-K and KFCM-F clustering algorithms, as well as the performance of the variable-wise kernel fuzzy c-means clustering algorithms proposed in this paper setting $m = (2 + p)/p$ on the synthetic datasets 1 and 2 with 10 additional irrelevant variables according to the CR index, F-measure and OERC. Note that, setting the fuzzification parameter as proposed by the authors of Refs. [47] and [48], we have a similar behavior to that presented by the algorithms on the synthetic datasets 1 and 2 without the

Please cite this article in press as: M.R.P. Ferreira, F.A.T. de Carvalho, Kernel fuzzy c-means with automatic variable weighting, Fuzzy Sets and Systems (2013), http://dx.doi.org/10.1016/j.fss.2013.05.004

Table 9
Performance of the algorithms on the synthetic dataset 1 with 10 additional irrelevant variables and considering $m = (2 + p)/p$: average and standard deviation (in parenthesis) of the CR index, F-measure and OERC.

	CR index	F-measure	OERC
FCM	0.0039 (0.0048)	0.3075 (0.0145)	0.6942 (0.0141)
KFCM-K	0.0057 (0.0058)	0.3112 (0.0165)	0.6894 (0.0165)
VKFCM-K	0.2612 (0.0284)	0.5437 (0.0338)	0.4556 (0.0340)
VKFCM-K-LS	**0.9120** (**0.1052**)	**0.9548** (**0.0718**)	**0.0446** (**0.0705**)
VKFCM-K-LP	**0.9704** (**0.0122**)	**0.9888** (**0.0046**)	**0.0111** (**0.0046**)
VKFCM-K-GS	0.3260 (0.0057)	0.5349 (0.0130)	0.4710 (0.0147)
VKFCM-K-GP	0.4940 (0.1584)	0.6562 (0.1080)	0.3476 (0.1096)
KFCM-F	0.0055 (0.0057)	0.3117 (0.0174)	0.6884 (0.0177)
VKFCM-F	0.4198 (0.1281)	0.6308 (0.0914)	0.3710 (0.0910)
VKFCM-F-LS	**0.6960** (**0.2653**)	**0.8220** (**0.1657**)	**0.1905** (**0.1792**)
VKFCM-F-LP	**0.9595** (**0.0577**)	**0.9814** (**0.0395**)	**0.0193** (**0.0438**)
VKFCM-F-GS	0.3262 (0.0063)	0.5355 (0.0136)	0.4710 (0.0154)
VKFCM-F-GP	0.3591 (0.0951)	0.5592 (0.0676)	0.4460 (0.0675)

Table 10
Performance of the algorithms on the synthetic dataset 2 with 10 additional irrelevant variables and considering $m = (2 + p)/p$: average and standard deviation (in parenthesis) of the CR index, F-measure and OERC.

	CR index	F-measure	OERC
FCM	0.2192 (0.0381)	0.5837 (0.0263)	0.4122 (0.0278)
KFCM-K	0.2178 (0.0358)	0.5820 (0.0270)	0.4113 (0.0251)
VKFCM-K	**0.6251** (**0.1276**)	**0.8510** (**0.0594**)	**0.1486** (**0.0591**)
VKFCM-K-LS	**0.4775** (**0.0333**)	**0.7195** (**0.0318**)	**0.2783** (**0.0341**)
VKFCM-K-LP	**0.8684** (**0.0386**)	**0.9542** (**0.0139**)	**0.0457** (**0.0139**)
VKFCM-K-GS	**0.7753** (**0.1827**)	**0.8987** (**0.1042**)	**0.1016** (**0.1049**)
VKFCM-K-GP	**0.8718** (**0.0358**)	**0.9556** (**0.0128**)	**0.0444** (**0.0129**)
KFCM-F	0.2221 (0.0355)	0.5846 (0.0280)	0.4080 (0.0245)
VKFCM-F	**0.8330** (**0.0468**)	**0.9413** (**0.0174**)	**0.0586** (**0.0174**)
VKFCM-F-LS	**0.4719** (**0.0799**)	**0.7142** (**0.0536**)	**0.2918** (**0.0596**)
VKFCM-F-LP	**0.8554** (**0.0762**)	**0.9465** (**0.0431**)	**0.0535** (**0.0432**)
VKFCM-F-GS	**0.6584** (**0.2082**)	**0.8286** (**0.1216**)	**0.1722** (**0.1226**)
VKFCM-F-GP	**0.8793** (**0.0346**)	**0.9582** (**0.0123**)	**0.0417** (**0.0123**)

additional irrelevant variables. Nevertheless, there was a remarkable decrease in the performance of the FCM, KFCM-K, VKFCM-K, KFCM-F, VKFCM-F and VKFCM-F-LS algorithms on the synthetic dataset 1, whereas there was a notable decrease in the performance of the variable-wise fuzzy c-means algorithms based on adaptive distances (local and global) with the constraint that the sum of the weights must be equal to one (VKFCM-K-LS, VKFCM-K-GS, VKFCM-F-LS, VKFCM-F-GS) on the synthetic dataset 2.

The comparisons between the clustering algorithms on the synthetic datasets 1 and 2 with 10 additional irrelevant variables and considering $m = (2 + p)/p$ performed via suitable one-sided Student's t-tests for paired samples at a 5% significance level are shown in Table 11. The kernel fuzzy c-means algorithms based on local adaptive distances (VKFCM-K-LS, VKFCM-K-LP, VKFCM-F-LS and VKFCM-F-LP) presented the best performances on the synthetic dataset 1. In this case, according to the paired t-tests, the VKFCM-K-LP and VKFCM-F-LP algorithms presented the best performances, whereas the FCM, KFCM-K and KFCM-F algorithms presented the worst performances. Moreover, the variable-wise kernel fuzzy c-means algorithms based on both local and global adaptive distances with the constraint that the product of the weights of the variables must be equal to one (VKFCM-K-LP, VKFCM-K-GP, VKFCM-F-LP and VKFCM-F-GP) achieved a better performance than the other methods on the synthetic dataset 2, according to the paired t-tests.

In conclusion, the performance of the variable-wise kernel fuzzy c-means algorithms based on local adaptive distances was superior when there were a specific set of relevant variables to each cluster, in comparison with all the others algorithms. Moreover, the performance of the variable-wise kernel fuzzy c-means clustering algorithms based on global adaptive distances was superior when the set of variables is almost the same to all clusters, in comparison

Please cite this article in press as: M.R.P. Ferreira, F.A.T. de Carvalho, Kernel fuzzy c-means with automatic variable weighting, Fuzzy Sets and Systems (2013), http://dx.doi.org/10.1016/j.fss.2013.05.004

Table 11
Comparison between the clustering algorithms on the synthetic datasets 1 and 2 with 10 additional irrelevant variables and considering $m = (2 + p)/p$ according to a Student's t-test for paired samples at a 5% significance level.

		Synthetic dataset 1			Synthetic dataset 2		
		CR	F-measure	OERC	CR	F-measure	OERC
FCM	KFCM-K	=	=	=	=	=	=
	VKFCM-K	−	−	−	−	−	−
	VKFCM-K-LS	−	−	−	−	−	−
	VKFCM-K-LP	−	−	−	−	−	−
	VKFCM-K-GS	−	−	−	−	−	−
	VKFCM-K-GP	−	−	−	−	−	−
	KFCM-F	=	=	=	=	=	=
	VKFCM-F	−	−	−	−	−	−
	VKFCM-F-LS	−	−	−	−	−	−
	VKFCM-F-LP	−	−	−	−	−	−
	VKFCM-F-GS	−	−	−	−	−	−
	VKFCM-F-GP	−	−	−	−	−	−
KFCM-K	VKFCM-K	−	−	−	−	−	−
	VKFCM-K-LS	−	−	−	−	−	−
	VKFCM-K-LP	−	−	−	−	−	−
	VKFCM-K-GS	−	−	−	−	−	−
	VKFCM-K-GP	−	−	−	−	−	−
	KFCM-F	=	=	=	−	=	=
	VKFCM-F	−	−	−	−	−	−
	VKFCM-F-LS	−	−	−	−	−	−
	VKFCM-F-LP	−	−	−	−	−	−
	VKFCM-F-GS	−	−	−	−	−	−
	VKFCM-F-GP	−	−	−	−	−	−
VKFCM-K	VKFCM-K-LS	−	−	−	+	+	+
	VKFCM-K-LP	−	−	−	−	−	−
	VKFCM-K-GS	−	=	+	−	−	−
	VKFCM-K-GP	−	−	−	−	−	−
	KFCM-F	+	+	+	+	+	+
	VKFCM-F	−	−	−	−	−	−
	VKFCM-F-LS	−	−	−	+	+	+
	VKFCM-F-LP	−	−	−	−	−	−
	VKFCM-F-GS	−	=	+	=	=	=
	VKFCM-F-GP	−	=	=	−	−	−
VKFCM-K-LS	VKFCM-K-LP	−	−	−	−	−	−
	VKFCM-K-GS	+	+	+	−	−	−
	VKFCM-K-GP	+	+	+	−	−	−
	KFCM-F	+	+	+	+	+	+
	VKFCM-F	+	+	+	−	−	−
	VKFCM-F-LS	+	+	+	=	=	=
	VKFCM-F-LP	−	=	=	−	−	−
	VKFCM-F-GS	+	+	+	−	−	−
	VKFCM-F-GP	+	+	+	−	−	−
VKFCM-K-LP	VKFCM-K-GS	+	+	+	+	+	+
	VKFCM-K-GP	+	+	+	=	=	=
	KFCM-F	+	+	+	+	+	+
	VKFCM-F	+	+	+	+	+	+
	VKFCM-F-LS	+	+	+	+	+	+
	VKFCM-F-LP	=	=	=	=	=	=
	VKFCM-F-GS	+	+	+	+	+	+
	VKFCM-F-GP	+	+	+	−	−	−

(*continued on next page*)

Please cite this article in press as: M.R.P. Ferreira, F.A.T. de Carvalho, Kernel fuzzy c-means with automatic variable weighting, Fuzzy Sets and Systems (2013), http://dx.doi.org/10.1016/j.fss.2013.05.004

Table 11 (*Continued.*)

		Synthetic dataset 1			Synthetic dataset 2		
		CR	F-measure	OERC	CR	F-measure	OERC
VKFCM-K-GS	VKFCM-K-GP	−	−	−	−	−	−
	KFCM-F	+	+	+	+	+	+
	VKFCM-F	−	−	−	=	−	−
	VKFCM-F-LS	−	−	−	+	+	+
	VKFCM-F-LP	−	−	−	−	−	−
	VKFCM-F-GS	=	=	=	+	+	+
	VKFCM-F-GP	=	−	−	−	−	−
VKFCM-K-GP	KFCM-F	+	+	+	+	+	+
	VKFCM-F	=	=	=	+	+	+
	VKFCM-F-LS	−	−	−	+	+	+
	VKFCM-F-LP	−	−	−	=	=	=
	VKFCM-F-GS	+	+	+	+	+	+
	VKFCM-F-GP	+	+	+	−	−	−
KFCM-F	VKFCM-F	−	−	−	−	−	−
	VKFCM-F-LS	−	−	−	−	−	−
	VKFCM-F-LP	−	−	−	−	−	−
	VKFCM-F-GS	−	−	−	−	−	−
	VKFCM-F-GP	−	−	−	−	−	−
VKFCM-F	VKFCM-F-LS	−	−	−	+	+	+
	VKFCM-F-LP	−	−	−	=	=	=
	VKFCM-F-GS	+	+	+	+	+	+
	VKFCM-F-GP	+	+	+	−	−	−
VKFCM-F-LS	VKFCM-F-LP	−	−	−	−	−	−
	VKFCM-F-GS	+	+	+	−	−	−
	VKFCM-F-GP	+	+	+	−	−	−
VKFCM-F-LP	VKFCM-F-GS	+	+	+	+	+	+
	VKFCM-F-GP	+	+	+	=	=	=
VKFCM-F-GS	VKFCM-F-GP	−	−	−	−	−	−

with all the others algorithms. It can be noted that, in the case where there are more irrelevant variables than relevant variables, there was a considerable decrease in the performance of the variable-wise kernel fuzzy c-means clustering algorithms based on local adaptive distances on the synthetic dataset 1 and a significant decrease in the performance of all variable-wise kernel fuzzy c-means algorithms on the synthetic dataset 2. Appropriate adjustment of the fuzzification parameter resulted in similar results to that obtained for the synthetic datasets 1 and 2 without additional irrelevant variables.

6.2. Benchmark datasets

The standard FCM, the conventional kernel fuzzy clustering algorithms, described in Section 2, and the variable-wise kernel fuzzy c-means clustering methods, presented in Section 3, were applied to seven UCI Machine Learning Repository datasets [49], namely E. coli, Image segmentation, Iris plant, Sonar mines versus rocks, Thyroid gland, Wisconsin diagnostic breast cancer (WDBC), and Wine, and to a dataset available in [30], the Ru-kiln glazes dataset. Table 12, in which n represents the number of instances, p represents the number of variables and c represents the number of a priori classes, describes shortly the datasets considered.

6.2.1. Performance of the fuzzy clustering algorithms

The numerical experiments were performed with the data without standardization (none), and also considering two kinds of standardization, the first based on the mean and standard deviation of each variable (a), and the second based on the minimum and maximum values of each variable (b). The fuzzification parameter m and the parameter

Please cite this article in press as: M.R.P. Ferreira, F.A.T. de Carvalho, Kernel fuzzy c-means with automatic variable weighting, Fuzzy Sets and Systems (2013), http://dx.doi.org/10.1016/j.fss.2013.05.004

β were set both equal to 2.0. The term $2\sigma^2$ in the Gaussian kernel used in the conventional kernel fuzzy clustering methods was estimated as the mean of the 0.1 and 0.9 quantiles of $\|\mathbf{x}_i - \mathbf{x}_k\|^2$, $i \neq k$ [46]. We set $\varepsilon = 10^{-10}$ as the tolerance for the convergence of the adequacy criterion. In the variable-wise kernel fuzzy c-means clustering methods, the terms $2\sigma_j^2$ $(j = 1, \ldots, p)$ in the Gaussian kernels were estimated for each variable as the mean of the 0.1 and 0.9 quantiles of $\|x_{ij} - x_{kj}\|^2$, $i \neq k$. For each dataset, the algorithms are run, until the convergence to a stationary value of the adequacy criterion, 100 times and the best results were selected according to the clustering adequacy criterion. For each dataset, the number of clusters is set equal the number of classes. From the fuzzy partition given by these clustering algorithms it is obtained a hard partition by assigning each object to a hard cluster as follows: object $\mathbf{x}_k$ is assigned to cluster P_i if $i = \arg\max_{1 \leqslant h \leqslant K} u_{hk}$.

Tables 13, 14, and 15 show, respectively, the CR index, F-measure, and the OERC computed to the hard partitions in comparison with the a priori partitions. These indexes were obtained with the standard FCM, KFCM-K, KFCM-F,

Table 12
Summary of the datasets.

Dataset	n	p	c	Dataset	n	p	c
E. coli	336	5	8	Sonar	208	60	2
Image	210	16	7	Thyroid	215	5	3
Iris	150	4	3	WDBC	569	30	2
Ru-kiln	27	10	2	Wine	178	13	3

Table 13
CR index for the datasets considered; "none" means that the clustering methods were applied to the datasets without standardization; (a) means that standardization based on the mean and standard deviation values of each variables was considered; (b) means that standardization based on the minimum and maximum values of each variable was considered.

	E. coli			Image		
	none	(a)	(b)	none	(a)	(b)
FCM	0.3619(05)	0.3640(05)	0.3589(05)	0.3832(09)	0.5264(06)	0.5114(09)
KFCM-K	0.3387(06)	0.3376(06)	0.3307(06)	0.4338(07)	**0.6105(02)**	0.5174(07)
VKFCM-K	0.3317(07)	0.3216(09)	0.3282(07)	**0.5967(01)**	**0.6403(01)**	**0.6319(01)**
VKFCM-K-LS	0.1379(12)	0.1333(13)	0.1381(13)	0.1014(13)	0.2504(13)	00991(13)
VKFCM-K-LP	0.3297(08)	0.3256(08)	0.3273(08)	0.5237(05)	0.5144(08)	0.5196(06)
VKFCM-K-GS	0.1868(11)	0.1868(11)	0.1868(11)	0.2917(11)	0.2942(11)	0.2917(11)
VKFCM-K-GP	0.3261(09)	0.3290(07)	0.3256(09)	0.5282(04)	0.5413(04)	0.5324(03)
KFCM-F	0.3947(04)	**0.4909(01)**	0.3848(04)	0.4254(08)	0.5920(03)	0.5228(05)
VKFCM-F	**0.4369(01)**	**0.4468(02)**	**0.4431(02)**	**0.5566(02)**	0.4204(09)	**0.5757(02)**
VKFCM-F-LS	0.1372(13)	0.1357(12)	0.1399(12)	0.1014(13)	0.2504(13)	0.0991(13)
VKFCM-F-LP	**0.4290(02)**	0.4242(04)	0.4110(03)	0.5384(03)	0.5312(05)	0.5258(04)
VKFCM-F-GS	0.2220(10)	0.1868(11)	0.1868(11)	0.2917(11)	0.2942(11)	0.2917(11)
VKFCM-F-GP	0.4284(03)	0.4363(03)	**0.4487(01)**	0.5161(06)	0.5195(07)	0.5161(08)

	Iris			Ru-kiln		
	none	(a)	(b)	none	(a)	(b)
FCM	0.7294(11)	0.6303(13)	0.7287(09)	0.5783(06)	0.2026(11)	0.2062(05)
KFCM-K	0.7570(10)	0.6303(13)	0.7282(11)	0.0021(13)	0.0006(12)	0.0006(10)
VKFCM-K	0.6521(13)	0.6522(10)	0.6201(13)	0.2845(12)	0.3763(09)	0.1382(07)
VKFCM-K-LS	**0.9037(02)**	0.8857(04)	**0.8857(04)**	**0.8503(02)**	**0.8503(02)**	0.4736(04)
VKFCM-K-LP	0.8341(08)	0.8508(08)	0.8508(08)	0.5855(05)	0.7139(04)	**0.7139(02)**
VKFCM-K-GS	0.8858(04)	**0.8858(03)**	**0.8857(04)**	0.3735(10)	0.4776(08)	−0.0221(13)
VKFCM-K-GP	0.8683(06)	0.8683(06)	0.8683(06)	0.2845(12)	0.4776(08)	0.0801(09)
KFCM-F	0.7570(10)	0.6303(13)	0.7282(11)	0.3763(08)	0.2845(10)	0.0801(09)
VKFCM-F	0.6521(13)	0.6522(10)	0.6412(12)	0.3735(10)	−0.0335(13)	0.1382(07)
VKFCM-F-LS	**0.9037(02)**	**0.9037(01)**	**0.8857(04)**	**0.8503(02)**	**0.8503(02)**	0.4736(04)
VKFCM-F-LP	0.8508(07)	0.8680(07)	0.8680(07)	0.5855(05)	0.7139(04)	**0.7139(02)**
VKFCM-F-GS	0.8858(04)	**0.8858(03)**	**0.8857(04)**	0.7139(03)	0.4776(08)	−0.0025(11)
VKFCM-F-GP	0.8683(06)	0.8683(06)	0.8683(06)	0.4736(07)	0.5899(05)	−0.0206(12)

(*continued on next page*)

Please cite this article in press as: M.R.P. Ferreira, F.A.T. de Carvalho, Kernel fuzzy c-means with automatic variable weighting, Fuzzy Sets and Systems (2013), http://dx.doi.org/10.1016/j.fss.2013.05.004

Table 13 (*Continued.*)

	Sonar			Thyroid		
	none	(a)	(b)	none	(a)	(b)
FCM	0.0064(08)	−0.0033(11)	0.0064(09)	0.4413(05)	**0.6592(01)**	**0.6927(02)**
KFCM-K	0.0027(11)	0.0253(03)	0.0109(08)	0.1875(11)	0.1694(10)	0.1669(11)
VKFCM-K	−0.0025(12)	**0.0401(01)**	0.0323(03)	0.0534(13)	0.0662(12)	0.0458(13)
VKFCM-K-LS	−0.0033(13)	−0.0033(11)	−0.0045(13)	0.1993(10)	0.2018(08)	0.2018(09)
VKFCM-K-LP	0.0220(04)	**0.0287(02)**	0.0253(04)	0.3780(06)	0.4339(05)	**0.6956(01)**
VKFCM-K-GS	**0.0287(03)**	0.0085(05)	0.0190(05)	0.2628(07)	0.1961(09)	0.2881(08)
VKFCM-K-GP	**0.0287(03)**	−0.0047(13)	**0.0401(02)**	0.0543(12)	0.0587(13)	0.0509(12)
KFCM-F	0.0064(08)	−0.0003(08)	0.0045(10)	0.2237(08)	0.2606(06)	0.3048(07)
VKFCM-F	0.0161(06)	0.0190(04)	0.0134(07)	**0.6172(01)**	0.5513(04)	0.3420(06)
VKFCM-F-LS	**0.0361(01)**	−0.0047(13)	**0.0401(02)**	0.2226(09)	0.2119(07)	0.2001(10)
VKFCM-F-LP	0.0027(11)	0.0027(07)	−0.0003(11)	0.5387(03)	0.5820(03)	0.6898(03)
VKFCM-F-GS	0.0027(11)	−0.0015(09)	−0.0025(12)	0.4894(04)	**0.6475(02)**	0.4118(05)
VKFCM-F-GP	0.0190(05)	0.0064(06)	0.0134(07)	**0.6109(02)**	0.1503(11)	0.5276(04)

	WDBC			Wine		
	none	(a)	(b)	none	(a)	(b)
FCM	0.4914(13)	0.6829(11)	0.7305(07)	0.3539(13)	**0.8975(03)**	**0.8498(05)**
KFCM-K	0.5286(12)	0.6896(09)	0.7493(05)	0.3749(12)	**0.8975(03)**	**0.8498(05)**
VKFCM-K	0.7074(08)	0.7073(07)	0.6898(11)	**0.8649(01)**	**0.8975(03)**	**0.8498(05)**
VKFCM-K-LS	0.7119(07)	0.6936(08)	0.6758(12)	0.3968(09)	0.3968(12)	0.3968(12)
VKFCM-K-LP	0.7736(03)	**0.7798(02)**	0.7736(03)	0.8185(03)	0.8185(05)	0.8185(06)
VKFCM-K-GS	**0.7800(01)**	**0.7861(01)**	**0.7924(01)**	0.7571(04)	0.7724(06)	**0.8498(05)**
VKFCM-K-GP	0.7554(05)	0.7554(04)	0.7615(04)	**0.8498(02)**	0.8516(04)	**0.8819(01)**
KFCM-F	0.5502(11)	0.6608(13)	0.7073(09)	0.3749(12)	0.3779(13)	0.6432(11)
VKFCM-F	0.6722(10)	0.6722(12)	0.6664(13)	0.6682(08)	0.6682(10)	0.6707(10)
VKFCM-F-LS	0.7610(04)	0.7179(06)	0.7239(08)	0.3877(10)	0.3968(12)	0.3707(13)
VKFCM-F-LP	**0.7738(02)**	0.7738(03)	**0.7738(02)**	0.7039(05)	0.6914(09)	0.7297(09)
VKFCM-F-GS	0.7190(06)	0.7371(05)	0.7492(06)	0.6921(06)	0.7191(08)	0.7899(07)
VKFCM-F-GP	0.6895(09)	0.6895(10)	0.6954(10)	0.6800(07)	0.7238(07)	0.7471(08)

Table 14
F-measure for the datasets considered; "none" means that the clustering methods were applied to the datasets without standardization; (a) means that standardization based on the mean and standard deviation values of each variables was considered; (b) means that standardization based on the minimum and maximum values of each variable was considered.

	E. coli			Image		
	none	(a)	(b)	none	(a)	(b)
FCM	0.5979(04)	0.6158(04)	0.5915(04)	0.6181(09)	0.6942(06)	0.6890(07)
KFCM-K	0.5696(05)	0.5691(05)	0.5670(05)	0.6612(07)	**0.7634(02)**	0.7051(04)
VKFCM-K	0.5571(08)	0.5543(08)	0.5545(08)	**0.7582(01)**	**0.7811(01)**	**0.7758(01)**
VKFCM-K-LS	0.4288(12)	0.4256(11)	0.4279(11)	0.3651(13)	0.4899(13)	0.3745(13)
VKFCM-K-LP	0.5403(09)	0.5398(09)	0.5408(09)	0.6865(06)	0.6755(08)	0.6876(08)
VKFCM-K-GS	0.4206(13)	0.4206(13)	0.4206(13)	0.5160(11)	0.5214(11)	0.5160(11)
VKFCM-K-GP	0.5579(07)	0.5626(07)	0.5593(07)	0.6946(04)	0.7096(05)	0.6993(06)
KFCM-F	**0.6514(02)**	**0.7114(01)**	**0.6468(02)**	0.6569(08)	0.7553(03)	0.7148(03)
VKFCM-F	**0.6571(01)**	**0.6717(02)**	**0.6679(01)**	**0.7237(02)**	0.6081(09)	**0.7305(02)**
VKFCM-F-LS	0.4288(12)	0.4277(10)	0.4302(10)	0.3651(13)	0.4899(13)	0.3745(13)
VKFCM-F-LP	0.6413(03)	0.6321(03)	0.6177(03)	0.6934(05)	0.6923(07)	0.6389(09)
VKFCM-F-GS	0.4413(10)	0.4206(13)	0.4206(13)	0.5160(11)	0.5214(11)	0.5160(11)
VKFCM-F-GP	0.5579(07)	0.5626(07)	0.5593(07)	0.6946(04)	0.7096(05)	0.6993(06)

(*continued on next page*)

Please cite this article in press as: M.R.P. Ferreira, F.A.T. de Carvalho, Kernel fuzzy c-means with automatic variable weighting, Fuzzy Sets and Systems (2013), http://dx.doi.org/10.1016/j.fss.2013.05.004

Table 14 (*Continued.*)

	Iris			Ru-kiln		
	none	(a)	(b)	none	(a)	(b)
FCM	0.8923(11)	0.8399(13)	0.8926(11)	0.8866(06)	0.7501(11)	0.7520(07)
KFCM-K	0.9061(10)	0.8399(13)	0.8929(10)	0.6102(12)	0.6086(12)	0.6086(11)
VKFCM-K	0.8533(13)	0.8533(10)	0.8332(13)	0.7869(11)	0.8228(09)	0.7163(09)
VKFCM-K-LS	**0.9667(02)**	**0.9600(02)**	**0.9600(04)**	**0.9636(02)**	**0.9636(02)**	0.8560(06)
VKFCM-K-LP	0.9399(08)	0.9466(08)	0.9466(08)	0.8907(05)	0.9280(04)	**0.9280(02)**
VKFCM-K-GS	0.9599(04)	0.9599(04)	**0.9600(04)**	0.5394(13)	0.8580(08)	0.5739(13)
VKFCM-K-GP	0.9532(06)	0.9532(07)	0.9532(07)	0.7869(11)	0.8580(08)	0.8580(04)
KFCM-F	0.9061(10)	0.8399(13)	0.8929(10)	0.8228(07)	0.7869(10)	0.6798(10)
VKFCM-F	0.8533(13)	0.8533(10)	0.8465(12)	0.8215(08)	0.5428(13)	0.7163(09)
VKFCM-F-LS	**0.9667(02)**	**0.9667(01)**	**0.9600(04)**	**0.9636(02)**	**0.9636(02)**	0.8560(06)
VKFCM-F-LP	0.9466(07)	0.9533(05)	0.9533(05)	0.8907(05)	0.9280(04)	**0.9280(02)**
VKFCM-F-GS	0.9599(04)	0.9599(04)	**0.9600(04)**	0.9280(03)	0.8580(08)	0.6073(12)
VKFCM-F-GP	0.9532(06)	0.9532(07)	0.9532(07)	0.7869(11)	0.8580(08)	0.8580(04)

	Sonar			Thyroid		
	none	(a)	(b)	none	(a)	(b)
FCM	0.5530(09)	0.5534(09)	0.5532(09)	0.7994(03)	**0.8886(01)**	0.8982(03)
KFCM-K	0.5423(12)	0.5865(07)	0.5626(08)	0.6508(09)	0.6336(08)	0.6084(10)
VKFCM-K	0.6014(04)	0.6060(03)	0.5966(04)	0.4772(11)	0.4935(11)	0.4695(13)
VKFCM-K-LS	0.5235(13)	0.5216(13)	0.5146(13)	0.7026(08)	0.6335(09)	0.6950(06)
VKFCM-K-LP	0.5822(06)	0.5918(06)	0.5869(06)	0.7777(05)	0.8059(05)	**0.9054(01)**
VKFCM-K-GS	0.6061(03)	0.6014(04)	0.5192(12)	0.7040(07)	0.6348(07)	0.7206(05)
VKFCM-K-GP	**0.6062(02)**	**0.6062(02)**	**0.6062(02)**	0.4733(13)	0.4809(13)	0.4754(12)
KFCM-F	0.5524(10)	0.5338(11)	0.5483(10)	0.6357(10)	0.6400(06)	0.6516(09)
VKFCM-F	0.5712(07)	0.5774(08)	0.5673(07)	**0.8269(02)**	0.8522(03)	0.6635(08)
VKFCM-F-LS	0.6001(05)	0.5280(12)	0.6042(03)	0.7149(06)	0.6226(10)	0.6931(07)
VKFCM-F-LP	0.5582(08)	0.5965(05)	0.5965(05)	**0.8497(01)**	**0.8657(02)**	**0.9020(02)**
VKFCM-F-GS	0.5434(11)	0.5351(10)	0.5235(11)	0.7810(04)	0.8226(04)	0.7955(04)
VKFCM-F-GP	**0.6062(02)**	**0.6062(02)**	**0.6062(02)**	0.4733(13)	0.4809(13)	0.4754(12)

	WDBC			Wine		
	none	(a)	(b)	none	(a)	(b)
FCM	0.8443(13)	0.9133(11)	0.9274(08)	0.6986(13)	**0.9660(03)**	0.9488(06)
KFCM-K	0.8589(12)	0.9161(09)	0.9333(06)	0.7204(12)	**0.9660(03)**	0.9488(06)
VKFCM-K	0.9214(09)	0.9213(08)	0.9163(11)	**0.9545(01)**	**0.9660(03)**	0.9488(06)
VKFCM-K-LS	0.9214(09)	0.9156(10)	0.9100(12)	0.7358(09)	0.7358(12)	0.7358(12)
VKFCM-K-LP	0.9398(03)	**0.9416(02)**	0.9398(03)	0.9375(04)	0.9375(06)	0.9375(07)
VKFCM-K-GS	**0.9417(01)**	**0.9434(01)**	**0.9451(01)**	0.9139(05)	0.9199(07)	0.9488(06)
VKFCM-K-GP	0.9349(06)	0.9349(05)	0.9367(05)	**0.9488(03)**	0.9489(05)	**0.9603(02)**
KFCM-F	0.8669(11)	0.9076(13)	0.9213(10)	0.7204(12)	0.6700(13)	0.8589(11)
VKFCM-F	0.9109(10)	0.9109(12)	0.9092(13)	0.8715(08)	0.8715(10)	0.8719(10)
VKFCM-F-LS	0.9361(04)	0.9232(07)	0.9250(09)	0.7290(10)	0.7358(12)	0.7154(13)
VKFCM-F-LP	**0.9401(02)**	0.9401(03)	**0.9401(02)**	0.8909(06)	0.8850(09)	0.9030(09)
VKFCM-F-GS	0.9245(07)	0.9297(06)	0.9332(07)	0.8836(07)	0.8958(08)	0.9256(08)
VKFCM-F-GP	0.9349(06)	0.9349(05)	0.9367(05)	**0.9488(03)**	0.9489(05)	**0.9603(02)**

and the variable-wise kernel fuzzy c-means clustering methods presented in Section 3, for the datasets considered and to their corresponding two standardized versions. In addition, it is shown (in parenthesis) the performance rank of each algorithm according to the datasets considered and to their corresponding two standardized versions.

Table 16 shows the average performance ranking of the fuzzy clustering algorithms, according to the indexes and datasets considered (non-standardized and standardized versions), computed from Tables 13, 14, and 15. In addition, it is shown (in parenthesis) the performance rank of each algorithm according to the average performance ranking of the fuzzy clustering algorithms showed on Table 16. Note that, as expected, the performance rank of each algorithm are very similar, respectively on the non-standardized and standardized versions of the datasets, whatever the index

Please cite this article in press as: M.R.P. Ferreira, F.A.T. de Carvalho, Kernel fuzzy c-means with automatic variable weighting, Fuzzy Sets and Systems (2013), http://dx.doi.org/10.1016/j.fss.2013.05.004

Table 15
OERC index for the datasets considered; "none" means that the clustering methods were applied to the datasets without standardization; (a) means that standardization based on the mean and standard deviation values of each variables was considered; (b) means that standardization based on the minimum and maximum values of each variable was considered.

	E. coli			Image		
	none	(a)	(b)	none	(a)	(b)
FCM	**0.1875(01)**	**0.1935(01)**	**0.1905(01)**	0.4000(09)	0.3286(08)	0.3095(07)
KFCM-K	**0.2113(03)**	0.2054(03)	0.2173(04)	0.3476(08)	**0.2429(02)**	0.2952(05)
VKFCM-K	0.2202(05)	0.2202(06)	0.2232(07)	**0.2524(01)**	**0.2286(01)**	**0.2333(01)**
VKFCM-K-LS	0.4226(13)	0.4226(12)	0.4226(13)	0.6381(13)	0.4952(13)	0.6333(13)
VKFCM-K-LP	**0.2113(03)**	0.2173(05)	**0.2113(02)**	0.3095(05)	0.3286(08)	0.3143(08)
VKFCM-K-GS	0.3750(11)	0.3750(11)	0.3750(11)	0.4714(11)	0.4667(11)	0.4714(11)
VKFCM-K-GP	0.2202(05)	0.2173(05)	0.2202(06)	0.3000(04)	0.2857(04)	0.2952(05)
KFCM-F	0.2232(06)	**0.1994(02)**	0.2202(06)	0.3476(08)	0.2524(03)	0.2857(03)
VKFCM-F	0.2530(09)	0.2411(07)	0.2411(09)	**0.2810(02)**	0.4143(09)	**0.2619(02)**
VKFCM-F-LS	0.4226(13)	0.4256(13)	0.4196(12)	0.6381(13)	0.4952(13)	0.6333(13)
VKFCM-F-LP	0.2411(08)	0.2440(09)	0.2411(09)	0.3000(04)	0.3238(06)	0.3048(06)
VKFCM-F-GS	0.3333(10)	0.3750(11)	0.3750(11)	0.4714(11)	0.4667(11)	0.4714(11)
VKFCM-F-GP	0.2411(08)	0.2440(09)	0.2143(03)	0.3238(06)	0.3095(05)	0.3238(09)

	Iris			Ru-kiln		
	none	(a)	(b)	none	(a)	(b)
FCM	0.1067(11)	0.1600(13)	0.1067(11)	0.1111(06)	0.2593(11)	0.2593(05)
KFCM-K	0.0933(10)	0.1600(13)	0.1067(11)	0.2963(13)	0.2963(13)	0.2963(13)
VKFCM-K	0.1467(13)	0.1467(10)	0.1667(13)	0.2222(12)	0.1852(09)	0.2963(13)
VKFCM-K-LS	**0.0333(02)**	**0.0400(04)**	**0.0400(04)**	**0.0370(02)**	**0.0370(02)**	0.1481(04)
VKFCM-K-LP	0.0600(08)	0.0533(08)	0.0533(08)	0.1111(06)	0.0741(04)	**0.0741(02)**
VKFCM-K-GS	0.0400(04)	**0.0400(04)**	**0.0400(04)**	0.1852(10)	0.1481(08)	0.2963(13)
VKFCM-K-GP	0.0467(06)	0.0467(07)	0.0467(07)	0.2222(12)	0.1481(08)	0.2963(13)
KFCM-F	0.0933(10)	0.1600(13)	0.1067(11)	0.1852(10)	0.2222(10)	0.2963(13)
VKFCM-F	0.1467(13)	0.1467(10)	0.1533(12)	0.1852(10)	0.2963(13)	0.2963(13)
VKFCM-F-LS	**0.0333(02)**	**0.0333(01)**	**0.0400(04)**	**0.0370(02)**	**0.0370(02)**	0.1481(04)
VKFCM-F-LP	0.0533(07)	0.0467(07)	0.0467(07)	0.1111(06)	0.0741(04)	**0.0741(02)**
VKFCM-F-GS	0.0400(04)	**0.0400(04)**	**0.0400(04)**	0.0741(03)	0.1481(08)	0.2963(13)
VKFCM-F-GP	0.0467(06)	0.0467(07)	0.0467(07)	0.1481(07)	0.1111(05)	0.2963(13)

	Sonar			Thyroid		
	none	(a)	(b)	none	(a)	(b)
FCM	0.4471(08)	0.4663(13)	0.4471(09)	0.2093(07)	**0.1023(01)**	**0.0930(01)**
KFCM-K	0.4567(11)	0.4135(03)	0.4375(08)	0.2465(11)	0.2140(10)	0.2093(10)
VKFCM-K	0.4663(13)	**0.3942(01)**	0.4038(03)	0.3023(13)	0.3023(13)	0.3023(13)
VKFCM-K-LS	0.4663(13)	0.4663(13)	0.4663(13)	0.2140(10)	0.1907(07)	0.2093(10)
VKFCM-K-LP	0.4183(04)	**0.4087(02)**	0.4135(04)	0.1907(04)	0.1907(07)	**0.0977(02)**
VKFCM-K-GS	**0.4087(03)**	0.4423(05)	0.4231(05)	0.2047(05)	0.2047(09)	0.2000(07)
VKFCM-K-GP	**0.4087(03)**	0.4663(13)	**0.3942(02)**	0.3023(13)	0.3023(13)	0.3023(13)
KFCM-F	0.4471(08)	0.4663(13)	0.4519(10)	0.2140(10)	0.2047(09)	0.1814(05)
VKFCM-F	0.4279(06)	0.4231(04)	0.4327(07)	**0.1721(02)**	0.1535(03)	0.1535(04)
VKFCM-F-LS	**0.3990(01)**	0.4663(13)	**0.3942(02)**	0.2140(10)	0.1907(07)	0.2047(08)
VKFCM-F-LP	0.4567(11)	0.4567(07)	0.4663(13)	**0.1628(01)**	**0.1442(02)**	0.1023(03)
VKFCM-F-GS	0.4567(11)	0.4663(13)	0.4663(13)	0.1907(04)	0.1860(04)	0.1953(06)
VKFCM-F-GP	0.4231(05)	0.4471(06)	0.4327(07)	0.2093(07)	0.3023(13)	0.2326(11)

(*continued on next page*)

Please cite this article in press as: M.R.P. Ferreira, F.A.T. de Carvalho, Kernel fuzzy c-means with automatic variable weighting, Fuzzy Sets and Systems (2013), http://dx.doi.org/10.1016/j.fss.2013.05.004

Table 15 (*Continued.*)

	WDBC			Wine		
	none	(a)	(b)	none	(a)	(b)
FCM	0.1459(13)	0.0861(11)	0.0721(07)	0.3146(13)	**0.0337(03)**	**0.0506(05)**
KFCM-K	0.1336(12)	0.0844(10)	0.0668(06)	0.2921(12)	**0.0337(03)**	**0.0506(05)**
VKFCM-K	0.0791(08)	0.0791(07)	0.0844(11)	**0.0449(01)**	**0.0337(03)**	**0.0506(05)**
VKFCM-K-LS	0.0773(07)	0.0826(08)	0.0879(12)	0.2584(09)	0.2584(12)	0.2584(12)
VKFCM-K-LP	**0.0598(03)**	**0.0580(02)**	**0.0598(03)**	0.0618(03)	0.0618(05)	0.0618(06)
VKFCM-K-GS	**0.0580(01)**	**0.0562(01)**	**0.0545(01)**	0.0843(04)	0.0787(06)	**0.0506(05)**
VKFCM-K-GP	0.0650(05)	0.0650(04)	0.0633(04)	**0.0506(02)**	0.0506(04)	**0.0393(01)**
KFCM-F	0.1265(11)	0.0931(13)	0.0791(09)	0.2921(12)	0.3989(13)	0.1348(11)
VKFCM-F	0.0896(10)	0.0896(12)	0.0914(13)	0.1236(08)	0.1236(10)	0.1236(10)
VKFCM-F-LS	0.0633(04)	0.0756(06)	0.0738(08)	0.2640(10)	0.2584(12)	0.2753(13)
VKFCM-F-LP	**0.0598(03)**	0.0598(03)	**0.0598(03)**	0.1067(05)	0.1124(09)	0.0955(09)
VKFCM-F-GS	0.0756(06)	0.0703(05)	0.0668(06)	0.1124(06)	0.1011(08)	0.0730(07)
VKFCM-F-GP	0.0844(09)	0.0844(10)	0.0826(10)	0.1180(07)	0.1011(08)	0.0899(08)

Table 16
Average performance ranking of the fuzzy clustering algorithms, according to the indexes and datasets considered (non-standardized and standardized versions); "none" means that the clustering methods were applied to the datasets without standardization; (a) means that standardization based on the mean and standard deviation values of each variables was considered; (b) means that standardization based on the minimum and maximum values of each variable was considered.

	CR			F-measure			OERC		
	none	(a)	(b)	none	(a)	(b)	none	(a)	(b)
FCM	8.75(12)	7.62(09)	6.37(04)	8.5(11)	7.25(07)	6.87(05)	8.5(10)	7.62(07)	**5.75(02)**
KFCM-K	10.25(13)	7.25(07)	7.87(09)	9.87(13)	7.37 (08)	7.5(06)	10.0(13)	7.12(05)	7.75(06)
VKFCM-K	8.37(09)	6.5(03)	7.5(08)	7.25(09)	6.62 (05)	8.12(11)	8.25(09)	6.25(03)	8.25(08)
VKFCM-K-LS	8.5(10)	8.87(13)	10.0(13)	8.5(11)	9.00 (13)	9.62(13)	8.62(11)	8.87(12)	10.12(13)
VKFCM-K-LP	**5.25(02)**	**5.25(02)**	**4.75(01)**	**5.75(02)**	**6.00(02)**	**5.5(02)**	**4.5(01)**	**5.12(01)**	**4.37(01)**
VKFCM-K-GS	6.37(05)	6.75(04)	7.25(06)	7.12(08)	6.87(06)	8.12(11)	6.12(03)	6.87(04)	7.12(05)
VKFCM-K-GP	6.62(06)	7.37(08)	5.75(03)	6.5(05)	6.5(04)	5.62(04)	6.25(04)	7.25(06)	6.37(03)
KFCM-F	8.62(11)	8.37(12)	8.25(11)	8.75(12)	8.75(12)	8.12(11)	9.37(12)	9.5(13)	8.5(10)
VKFCM-F	6.37(05)	8.00(10)	7.37(07)	6.37(03)	8.37(11)	7.75(07)	7.5(08)	8.5(11)	8.75(11)
VKFCM-F-LS	6.75(07)	8.25(11)	8.25(11)	6.75(06)	8.37(11)	8.12(11)	6.87(07)	8.37(10)	8.00(07)
VKFCM-F-LP	**4.75(01)**	**5.25(02)**	**5.12(02)**	**4.62(01)**	**4.75(01)**	**4.62(01)**	**5.62(02)**	**5.87(02)**	6.5(04)
VKFCM-F-GS	6.87(08)	7.12(06)	8.37(12)	7.12(08)	8.00(09)	8.75(12)	6.87(07)	8.00(09)	9.37(12)
VKFCM-F-GP	5.62(03)	6.87(05)	7.00(05)	6.5(05)	6.5(04)	5.62(04)	6.87(07)	7.87(08)	8.5(10)

considered. For example, the rank of the FCM on the non-standardized version of the data set is 12, 11 and 10, respectively, for CR, F-measure and OERC indexes.

Table 17 shows the average performance ranking of the fuzzy clustering algorithms, according to the datasets considered (non-standardized and standardized versions), computed from Table 16. In addition, it is shown (in parenthesis) the performance rank of each algorithm according to the average performance ranking of the fuzzy clustering algorithms showed on Table 17. This table shows the performance of the fuzzy clustering algorithms according to a synthesis of the three indexes CR, F-measure and OERC.

For the non-standardized data sets, the best performance was presented by the algorithms VKFCM-F-LP and VKFCM-K-LP (variable-wise algorithms with local adaptive distance and constraint given by the product of the weights equal to one). The worst performance was presented by the FCM and the conventional kernel fuzzy clustering algorithms (KFCM-K and KFCM-F). Moreover, the algorithms with constraint given by the product of the weights equal to one (VKFCM-F-LP, VKFCM-K-LP, VKFCM-F-GP and VKFCM-K-GP) performed better than the algorithms with constraint given by the sum of the weights equal to one (VKFCM-F-LS, VKFCM-K-LS, VKFCM-F-GS and VKFCM-K-GS).

Concerning the algorithms with kernelization of the metric, the version with non-adaptive distances (VKFCM-K) outperforms only the version with local adaptive distances and constraint given by the sum of the weights equal to

Table 17
Average performance ranking of the fuzzy clustering algorithms, according to the datasets considered (non-standardized and standardized versions); "none" means that the clustering methods were applied to the datasets without standardization; (a) means that standardization based on the mean and standard deviation values of each variable was considered; (b) means that standardization based on the minimum and maximum values of each variable was considered.

	none	(a)	(b)
FCM	11.00(11)	7.66(08)	3.66(04)
KFCM-K	13.00(13)	6.66(07)	7.00(06)
VKFCM-K	9.00(09)	3.66(03)	9.00(09)
VKFCM-K-LS	10.66(10)	12.66(13)	13.00(13)
VKFCM-K-LP	**1.66(02)**	**1.66(02)**	**1.33(01)**
VKFCM-K-GS	5.33(06)	4.66(04)	7.33(07)
VKFCM-K-GP	5.00(04)	6.00(06)	3.33(03)
KFCM-F	11.66(12)	12.33(12)	10.66(11)
VKFCM-F	5.33(06)	10.66(11)	8.33(08)
VKFCM-F-LS	6.66(07)	10.66(11)	9.66(10)
VKFCM-F-LP	**1.33(01)**	**1.66(02)**	**2.33(02)**
VKFCM-F-GS	7.66(08)	8.00(09)	12.00(12)
VKFCM-F-GP	5.00(04)	5.66(05)	6.33(05)

one (VKFCM-K-LS). Moreover, the version with global adaptive distances (VKFCM-K-GS) outperforms the version with local adaptive distances (VKFCM-K-LS) for the constraint given by the sum of the weights equal to one whereas the version with local adaptive distances (VKFCM-K-LP) outperforms the version with global adaptive distances (VKFCM-K-GP) for the constraint given by the product of the weights equal to one.

For the algorithms in feature space, the version with non-adaptive distances (VKFCM-F) outperforms the versions with constraint given by the sum of the weights equal to one (VKFCM-F-LS and VKFCM-F-GS). Moreover, the version with local adaptive distances outperforms the version with global adaptive distances for both the constraint given by the sum of the weights equal to (VKFCM-F-LS versus VKFCM-F-GS) and for the constraint given by the product of the weights equal to one (VKFCM-F-LP versus VKFCM-F-GP).

For the datasets with standardization based on the mean and standard deviation values of each variable, the algorithms VKFCM-F-LP and VKFCM-K-LP were still the best, but the worst performance now is presented by the algorithms with local adaptive distances and constraint given by the sum of the weights equal to one (VKFCM-K-LS and VKFCM-F-LS), the conventional kernel fuzzy clustering algorithm in feature space (KFCM-F) and the variable-wise algorithm in feature space with non-adaptive distances (VKFCM-F). The traditional fuzzy c-means (FCM), the conventional kernel fuzzy clustering algorithm with kernelization of the metric (KFCM-K) and the variable-wise algorithm with kernelization of the metric and non-adaptive distances (VKFCM-K) have clearly improved their performance.

Concerning the algorithms with kernelization of the metric, the version with non-adaptive distances (VKFCM-K) outperforms the versions with adaptive distances with the exception of the VKFCM-K-LP algorithm. Moreover, the version with global adaptive distances (VKFCM-K-GS) outperforms the version with local adaptive distances (VKFCM-K-LS) for the constraint given by the sum of the weights equal to one whereas the version with local adaptive distances (VKFCM-K-LP) outperforms the version with global adaptive distances (VKFCM-K-GP) for the constraint given by the product of the weights equal to one.

For the algorithms in feature space, the version with non-adaptive distances (VKFCM-F) is outperformed by the versions with adaptive distances with the exception of the VKFCM-F-LS algorithm. Moreover, the version with global adaptive distances (VKFCM-F-GS) outperforms the version with local adaptive distances (VKFCM-F-LS) for the constraint given by the sum of the weights equal to one whereas the version with local adaptive distances (VKFCM-F-LP) outperforms the version with global adaptive distances (VKFCM-F-GP) for the constraint given by the product of the weights equal to one.

For the datasets with standardization based on the minimum and maximum values of each variable, the algorithms VKFCM-F-LP and VKFCM-K-LP were still the best, but the worst performance now is presented by the algorithms with local adaptive distances and constraint given by the sum of the weights equal to one (VKFCM-K-LS and VKFCM-F-LS), by the algorithm in feature space, global adaptive distances and constraint given by the sum of the weights equal to one (VKFCM-F-GS) and the conventional kernel fuzzy clustering algorithm in feature space

Please cite this article in press as: M.R.P. Ferreira, F.A.T. de Carvalho, Kernel fuzzy c-means with automatic variable weighting, Fuzzy Sets and Systems (2013), http://dx.doi.org/10.1016/j.fss.2013.05.004

(KFCM-F). The traditional fuzzy c-means (FCM) and the conventional kernel fuzzy clustering algorithm with kernelization of the metric (KFCM-K) have clearly improved their performance. Moreover, the algorithms with constraint given by the product of the weights equal to one (VKFCM-F-LP, VKFCM-K-LP, VKFCM-F-GP and VKFCM-K-GP) performed better than the algorithms with constraint given by the sum of the weights equal to one (VKFCM-F-LS, VKFCM-K-LS, VKFCM-F-GS and VKFCM-K-GS).

Concerning the algorithms with kernelization of the metric, the version with non-adaptive distances (VKFCM-K) outperforms only the version with local adaptive distances and constraint given by the sum of the weights equal to one (VKFCM-K-LS). Moreover, the version with global adaptive distances (VKFCM-K-GS) outperforms the version with local adaptive distances (VKFCM-K-LS) for the constraint given by the sum of the weights equal to one whereas the version with local adaptive distances (VKFCM-K-LP) outperforms the version with global adaptive distances (VKFCM-K-GP) for the constraint given by the product of the weights equal to one.

For the algorithms in feature space, the version with non-adaptive distances (VKFCM-F) outperforms the versions with constraint given by the sum of the weights equal to one (VKFCM-F-LS and VKFCM-F-GS). Moreover, the version with local adaptive distances outperform the version with global adaptive distances for both the constraint given by the sum of the weights equal to one (VKFCM-F-LS versus VKFCM-F-GS) and for the constraint given by the product of the weights equal to one (VKFCM-F-LP versus VKFCM-F-GP).

Finally, whatever the datasets considered (non-standardized and standardized versions) the best performance was presented by the algorithms VKFCM-F-LP and VKFCM-K-LP (variable-wise algorithms with local adaptive distance and constraint given by the product of the weights equal to one).

6.2.2. Some remarks about the performance of the fuzzy clustering algorithms on the Sonar mines versus rocks dataset

The results presented in Tables 13, 14 and 15 show that all methods considered in this work performs poorly on the Sonar mines versus rocks dataset. This dataset consists of 208 instances distributed in two classes, with 111 and 97 instances, respectively. Each instance is described by 60 variables taking values in the $(0, 1)$ interval. As was pointed out in Refs. [47] and [48], the poor performance of the clustering algorithms on this dataset is due to the fact that the fuzzy c-means algorithm as well as the fuzzy clustering algorithms derived from it don't present good results when applied to high-dimensional datasets, because in such situation, the cluster centroids become very close to the overall centroid.

Indeed, for this dataset, the obtained cluster centroids were very similar to the overall centroid for the fuzzy c-means and for the kernel fuzzy c-means with kernelization of the metric, as well as for all variable-wise kernel fuzzy c-means algorithms under the approach of kernelization of the metric. Unfortunately, for the algorithms based on the approach of clustering in the feature space we can't compute neither the cluster centroids nor the overall centroid.

In Section 5 we presented some indexes to fuzzy partition and cluster interpretation. The overall heterogeneity index (Q), given in Eq. (58), measures the quality of a fuzzy partition P. This index can be used as a warning concerning the problem described by the authors of Refs. [47] and [48]. When the clusters centroids are very similar to the overall centroid the value of Q becomes closer to 0.

In fact, we observed that, for this dataset, the value of the index Q was almost 0 for all clustering algorithms considered in this paper, except for the standard fuzzy c-means and for the kernel fuzzy c-means with kernelization of the metric, for which the value of Q was equal to 0.0573 and 0.0799, respectively.

The authors of Refs. [47] and [48] also showed that the fuzzy c-means algorithm can be successfully applied to high-dimensional datasets if the centroids are initialized very close to the actual cluster centers or if we appropriately adjust the fuzzification parameter, depending on the number of variables. They suggest $m = (2 + p)/p$, where p is the number of variables, as a good choice for the fuzzification parameter. We followed these potential solutions, but the performance of the clustering algorithms was not significantly improved. We believe this problem needs a much more detailed study which is beyond the scope of this article and will be faced in future researches.

6.2.3. Application: the Thyroid gland dataset

This dataset consists of three classes concerning the state of the thyroid gland: normal, hyperthyroidism and hypothyroidism, with 150, 35 and 30 instances, respectively. The instances are described by five real-valued variables namely: (1) T3-resin uptake test, (2) total serum thyroxin, (3) total serum triiodothyronine, (4) basal thyroid-stimulating hormone (TSH), and (5) maximal absolute difference of TSH value.

Please cite this article in press as: M.R.P. Ferreira, F.A.T. de Carvalho, Kernel fuzzy c-means with automatic variable weighting, Fuzzy Sets and Systems (2013), http://dx.doi.org/10.1016/j.fss.2013.05.004

Table 18
The relevance weights obtained from the VKFCM-K-LP and VKFCM-F-LP algorithms for the Thyroid dataset with standardization type (b).

		x_1	x_2	x_3	x_4	x_5
VKFCM-K-LP	Cluster 1	0.25818	0.59568	**1.04938**	**4.43517**	**1.39709**
	Cluster 2	0.91276	**3.06025**	**3.65616**	0.31241	0.31343
	Cluster 3	0.15307	0.20435	0.17590	**14.92557**	**12.17677**
VKFCM-F-LP	Cluster 1	0.27170	0.58914	**1.03320**	**4.46903**	**1.35298**
	Cluster 2	0.88688	**2.59718**	**3.19632**	0.35940	0.37793
	Cluster 3	0.17679	0.22967	0.19652	**13.20950**	**9.48739**

Table 19
Cluster centroids and Overall centroid obtained from VKFCM-K-LP algorithm for the Thyroid dataset with standardization type (b).

	x_1	x_2	x_3	x_4	x_5
Cluster centroid 1	0.580642	0.338913	0.155076	0.022189	0.065346
Cluster centroid 2	0.744234	0.113013	0.079650	0.189564	0.223298
Cluster centroid 3	0.445605	0.563330	0.253028	0.017111	0.018787
Overall centroid	0.632794	0.228727	0.126105	0.019399	0.029020

Table 20
Quality of the partition concerning single variables.

	x_1	x_2	x_3	x_4	x_5
$Q_j(P)$	0.3151	**0.5653**	0.2847	0.2537	0.2747

The results presented in Tables 13, 14 and 15 show that the FCM, VKFCM-K-LP and VKFCM-F-LP algorithms applied to this dataset with standardization (b) obtained the best results. The CR index values were 0.6927, 0.6956 and 0.6898, respectively. The F-measure values were 0.8982, 0.9054 and 0.9020, respectively, and the OERC values were 0.0930, 0.0977 and 0.1023, respectively. Table 18 gives the relevance weights of the variables computed for VKFCM-K-LP and VKFCM-F-LP algorithms.

It can be viewed that the set of relevant variables is not the same to all clusters. For both methods, variables x_3, x_4 and x_5 are the most important variables in the definition of the respective clusters 1, variables x_2 and x_3 are the most important variables in the definition of the respective clusters 2, whereas variables x_4 and x_5 are the most important variables in the definition of the respective clusters 3. Moreover, for the VKFCM-K-LP, it can be observed that in cluster 1, the variable x_4 has the greatest relevance weight because for the objects belonging to this cluster, the variable x_4 assumes similar values. On contrary, variable x_1 has the smallest relevance weight in cluster 1 because for the objects belonging to this cluster, the variable x_1 assumes more differently values.

6.2.3.1. Fuzzy partition and fuzzy cluster interpretation In order to show the usefulness of the partition and cluster interpretation indexes introduced in Section 5, we consider the previous results obtained with the application of the VKFCM-K-LP on the thyroid gland dataset. Table 19 shows the clusters centroids and the overall centroid obtained from VKFCM-K-LP algorithm for the Thyroid dataset with standardization type (b).

6.2.3.2. Fuzzy partition interpretation The overall heterogeneity index $Q(P)$, which measures the quality of the partition given by the VKFCM-K-LP algorithm, was equal to 0.3652. This value indicates fuzzy clusters centroids $\mathbf{v}_i$, $i = 1, \ldots, 3$, quite similar to the overall centroid $\mathbf{v}$ and a not so good representation of the elements of a cluster P_i by its centroid. This means that the set of variables have a not so good discriminant power, they are not able to very well separate the dataset into homogeneous clusters.

Table 20 displays the overall heterogeneity indexes concerning each variable. Comparing the values of $Q_j(P)$ with the value of $Q(P)$, we may conclude that the discriminant power of variable x_2 is above the discriminant power of the set variables, whereas all the other variables have a discriminant power below the discriminant power of the set variables.

Please cite this article in press as: M.R.P. Ferreira, F.A.T. de Carvalho, Kernel fuzzy c-means with automatic variable weighting, Fuzzy Sets and Systems (2013), http://dx.doi.org/10.1016/j.fss.2013.05.004

Table 21
Cluster heterogeneity indexes.

Cluster	Cardinal	$J(i)$	$Q(P_i)$
1	108	0.3506	0.3238
2	24	**0.3284**	**0.4726**
3	83	**0.3210**	0.2608

Table 22
Cluster heterogeneity indexes concerning single variables (%).

Cluster	$Q_j(P_i)$	x_1	x_2	x_3	x_4	x_5
1	$Q_j(P_1)$	0.1168	**0.5618**	0.2463	0.0686	**0.3667**
2	$Q_j(P_2)$	0.4557	**0.6367**	0.4025	**0.4777**	0.2327
3	$Q_j(P_3)$	**0.3020**	**0.4616**	0.1625	0.0408	0.1917

6.2.3.3. Fuzzy cluster interpretation From Table 21, we can see that clusters 2 and 3 are the most homogeneous, whereas cluster 1 is slightly more heterogeneous. Moreover, cluster 2 has the best quality index, its cluster centroid is the less similar to the overall centroid, whereas cluster 3 has the worst quality index, its cluster prototype is the most similar to the overall centroid.

Table 22 displays the fuzzy cluster heterogeneity index concerning single variables. Comparing the values of $Q_j(P_i)$ with the values of $Q(P_i)$, $i = 1, \ldots, 3$, $j = 1, \ldots, 5$ (see Table 21), we can see that variables x_2 and x_5 characterize cluster 1 (its cluster centroid is less similar to the overall centroid concerning these variables), variables x_2 and x_4 characterize cluster 2 and variables x_1 and x_2 characterize cluster 3. Moreover, fuzzy clusters 1 and 3 have a poor quality concerning variable x_4, their respective clusters centroids are very similar to the overall centroid concerning this variable.

7. Concluding remarks

In this paper we proposed variable-wise kernel fuzzy clustering methods, new kernel-based fuzzy clustering algorithms where dissimilarity measures are obtained as sums of Euclidean distances between patterns and centroids computed individually for each variable by means of kernel functions. The advantage of the proposed approach over the conventional kernel clustering methods is that it allows us to use adaptive distances which changes at each algorithm iteration and can either be the same for all clusters (global adaptive distances) or different from on cluster to another (local adaptive distances). This kind of dissimilarity measure is suitable to learn the weights of the variables during the clustering process, improving the performance of the algorithms.

For each algorithm, the paper gives the solution for the best centroid of each cluster, the best relevance weight of each variable as well the best fuzzy membership matrix, according to the clustering criterion. The derivation of the expressions of the relevance weights of the variables was done considering two cases: one assuming that the weights must sum one, whereas the other assuming that the product of the weights must be one. Convergence properties of the proposed algorithms were also provided.

Moreover, another advantage of this approach is that we were able to introduce various fuzzy partition and cluster interpretations tools.

The usefulness of proposed methods was shown with synthetic datasets and with several UCI machine learning datasets and a dataset available in Ref. [30]. Concerning the synthetic datasets, we can conclude that the variable-wise kernel fuzzy clustering methods based on local adaptive distances presented better results than the traditional clustering algorithms considered in this paper when each cluster has a different set of relevant variables and the variable-wise kernel fuzzy clustering methods based on global adaptive distances presented better results in the case that the set of relevant variables is the same to all clusters but there are irrelevant or noisy variables.

Concerning the benchmark datasets, the performance of the fuzzy clustering algorithms was evaluated according to the Corrected Rand index, F-measure and OERC indexes. Whatever the datasets considered (non-standardized and standardized versions) the best performance was presented by the variable-wise fuzzy c-means clustering algorithms with local adaptive distance and constraint given by the product of the weights equal to one.

Please cite this article in press as: M.R.P. Ferreira, F.A.T. de Carvalho, Kernel fuzzy c-means with automatic variable weighting, Fuzzy Sets and Systems (2013), http://dx.doi.org/10.1016/j.fss.2013.05.004

For the non-standardized benchmark datasets, the worst performance was presented by the standard fuzzy c-means and the conventional kernel fuzzy clustering algorithms. Moreover, the algorithms with constraint given by the product of the weights equal to one performed better than the algorithms with constraint given by the sum of the weights equal to one.

For the benchmark datasets with standardization based on the mean and standard deviation values of each variable, the worst performance was presented by the algorithms with local adaptive distances and constraint given by the sum of the weights equal to one, the conventional kernel fuzzy clustering algorithm in feature space and the variable-wise algorithm in feature space with non-adaptive distances. Moreover, the traditional fuzzy c-means, the conventional kernel fuzzy clustering algorithm with kernelization of the metric and the variable-wise algorithm with kernelization of the metric and non-adaptive distances have clearly improved their performance.

Concerning the benchmark datasets with standardization based on the minimum and maximum values of each variable, the worst performance was presented by the algorithms with local adaptive distances and constraint given by the sum of the weights equal to one, by the algorithm in feature space, global adaptive distances and constraint given by the sum of the weights equal to one and the conventional kernel fuzzy clustering algorithm in feature space. The standard fuzzy c-means and the conventional kernel fuzzy clustering algorithm with kernelization of the metric have clearly improved their performance. Moreover, the algorithms with constraint given by the product of the weights equal to one performed better than the algorithms with constraint given by the sum of the weights equal to one.

Finally, an application with the Thyroid gland dataset showed the merit of the fuzzy partition and fuzzy cluster interpretation tools.

Acknowledgements

Authors are grateful to the anonymous referees for their careful revision, valuable suggestions, and comments which improved this paper. This research was partially supported by grants from CNPq (Brazilian Agency).

Appendix A. Proof of Proposition 3.2

Whichever the distance function (Eqs. (17), (18), (20), (22) and (24)), and if $K(\cdot,\cdot)$ is the Gaussian kernel, then the cluster centroid $\mathbf{v}_i = (v_{i1}, \ldots, v_{ip})$ $(i = 1, \ldots, c)$, which minimizes the criterion J given in Eq. (15), has its components v_{ij} $(j = 1, \ldots, p)$ updated according to the following expression:

$$v_{ij} = \frac{\sum_{k=1}^{n} (u_{ik})^m K(x_{kj}, v_{ij}) x_{kj}}{\sum_{k=1}^{n} (u_{ik})^m K(x_{kj}, v_{ij})}. \tag{A.1}$$

Proof. If we restrict ourselves to the Gaussian kernel, then $K(x_{kj}, x_{kj}) = 1$ $(k = 1, \ldots, n;\ j = 1, \ldots, p)$ and $\|\Phi(x_{kj}) - \Phi(v_{ij})\|^2 = 2(1 - K(x_{kj}, v_{ij}))$. Then, for a cluster i and a variable j the problem becomes to find the centroid v_{ij} that minimizes the term

$$\sum_{k=1}^{n} (u_{ik})^m K(x_{kj}, v_{ij}). \tag{A.2}$$

The calculation of the centroids v_{ij}, $i = 1, \ldots, c$, $j = 1, \ldots, p$, proceeds as follows:

$$\frac{\partial J}{\partial v_{ij}} = \sum_{k=1}^{n} (u_{ik})^m \frac{\partial K(x_{kj}, v_{ij})}{\partial v_{ij}} = 0,$$

$$\sum_{k=1}^{n} (u_{ik})^m \frac{\partial (e^{-(x_{kj} - v_{ij})^2 / 2\sigma_j^2})}{\partial v_{ij}} = 0,$$

$$\sum_{k=1}^{n} (u_{ik})^m \frac{(x_{kj} - v_{ij})}{\sigma_j^2} e^{-(x_{kj} - v_{ij})^2 / 2\sigma_j^2} = 0,$$

$$v_{ij}\sum_{k=1}^{n}(u_{ik})^m K(x_{kj}, v_{ij}) = \sum_{k=1}^{n}(u_{ik})^m K(x_{kj}, v_{ij})x_{kj},$$

$$v_{ij} = \frac{\sum_{k=1}^{n}(u_{ik})^m K(x_{kj}, v_{ij})x_{kj}}{\sum_{k=1}^{n}(u_{ik})^m K(x_{kj}, v_{ij})}. \qquad \square$$

Appendix B. Proof of Proposition 3.3

The weights of the variables, which minimizes the criterion J given in Eq. (15), are calculated according to the adaptive distance function used:

(a) If the adaptive distance function is given by Eq. (18), the vector of weights $\boldsymbol{\lambda}_i = (\lambda_{i1}, \ldots, \lambda_{ip})$ which minimizes the criterion J given in Eq. (15) under $\lambda_{ij} \in [0, 1]\ \forall i, j$ and $\sum_{j=1}^{p}\lambda_{ij} = 1\ \forall i$, have their components λ_{ij} $(i = 1, \ldots, c,\ j = 1, \ldots, p)$ updated according to the following expression:

$$\lambda_{ij} = \left[\sum_{l=1}^{p}\left(\frac{\sum_{k=1}^{n}(u_{ik})^m \|\Phi(x_{kj}) - \Phi(v_{ij})\|^2}{\sum_{k=1}^{n}(u_{ik})^m \|\Phi(x_{kl}) - \Phi(v_{il})\|^2}\right)^{\frac{1}{\beta-1}}\right]^{-1}.$$

(b) If the adaptive distance function is given by Eq. (20), the vector of weights $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_p)$ which minimizes the criterion J given in Eq. (15) under $\lambda_j \in [0, 1]\ \forall j$ and $\sum_{j=1}^{p}\lambda_j = 1$, have their components λ_j $(j = 1, \ldots, p)$ updated according to the following expression:

$$\lambda_j = \left[\sum_{l=1}^{p}\left(\frac{\sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ik})^m \|\Phi(x_{kj}) - \Phi(v_{ij})\|^2}{\sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ik})^m \|\Phi(x_{kl}) - \Phi(v_{il})\|^2}\right)^{\frac{1}{\beta-1}}\right]^{-1}.$$

(c) If the adaptive distance function is given by Eq. (22), the vector of weights $\boldsymbol{\lambda}_i = (\lambda_{i1}, \ldots, \lambda_{ip})$ which minimizes the criterion J given in Eq. (15) under $\lambda_{ij} > 0\ \forall i, j$ and $\prod_{j=1}^{p}\lambda_{ij} = 1\ \forall i$, have their components λ_{ij} $(i = 1, \ldots, c,\ j = 1, \ldots, p)$ updated according to the following expression:

$$\lambda_{ij} = \frac{\{\prod_{l=1}^{p}(\sum_{k=1}^{n}(u_{ik})^m \|\Phi(x_{kl}) - \Phi(v_{il})\|^2)\}^{\frac{1}{p}}}{\sum_{k=1}^{n}(u_{ik})^m \|\Phi(x_{kj}) - \Phi(v_{ij})\|^2}.$$

(d) If the adaptive distance function is given by Eq. (24), the vector of weights $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_p)$ which minimizes the criterion J given in Eq. (15) under $\lambda_j > 0\ \forall j$ and $\prod_{j=1}^{p}\lambda_j = 1$, have their components λ_j $(j = 1, \ldots, p)$ updated according to the following expression:

$$\lambda_j = \frac{\{\prod_{l=1}^{p}(\sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ik})^m \|\Phi(x_{kl}) - \Phi(v_{il})\|^2)\}^{\frac{1}{p}}}{\sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ik})^m \|\Phi(x_{kj}) - \Phi(v_{ij})\|^2}.$$

Proof. (a) We want to minimize J with respect to λ_{ij} $(i = 1, \ldots, c,\ j = 1, \ldots, p)$ under $\lambda_{ij} \in [0, 1]\ \forall i, j$ and $\sum_{j=1}^{p}\lambda_{ij} = 1\ \forall i$. As the fuzzy membership degree u_{ik} $(i = 1, \ldots, c,\ k = 1, \ldots, n)$, the centroids $\mathbf{v}_i$ $(i = 1, \ldots, c)$, and the parameters m and β are fixed, we can rewrite the criterion J as

$$J(\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_c) = \sum_{i=1}^{c} J_i(\boldsymbol{\lambda}_i) = \sum_{i=1}^{c}\sum_{j=1}^{p}\lambda_{ij}^{\beta}\sum_{k=1}^{n}(u_{ik})^m \left\|\Phi(x_{kj}) - \Phi(v_{ij})\right\|^2$$

where $J_i(\boldsymbol{\lambda}_i) = J_i(\lambda_{i1}, \ldots, \lambda_{ip}) = \sum_{j=1}^{p}\lambda_{ij}^{\beta}J_{ij}$, $J_{ij} = \sum_{k=1}^{n}(u_{ik})^m \|\Phi(x_{kj}) - \Phi(v_{ij})\|^2$. The criterion J being additive, the problem becomes minimizing J_i, $i = 1, \ldots, c$. Let $g_i(\lambda_{i1}, \ldots, \lambda_{ip}) = \sum_{j=1}^{p}\lambda_{ij} - 1$. We want to determine the extremes of $J_i(\lambda_{i1}, \ldots, \lambda_{ip})$ with the restriction $g_i(\lambda_{i1}, \ldots, \lambda_{ip}) = 0$. To do so, we shall apply the Lagrange multipliers method to solve the following system

$$\nabla J_i(\lambda_{i1}, \ldots, \lambda_{ip}) = \mu\nabla g_i(\lambda_{i1}, \ldots, \lambda_{ip}).$$

Then, for $i = 1, \ldots, c$ and $j = 1, \ldots, c$, we have

Please cite this article in press as: M.R.P. Ferreira, F.A.T. de Carvalho, Kernel fuzzy c-means with automatic variable weighting, Fuzzy Sets and Systems (2013), http://dx.doi.org/10.1016/j.fss.2013.05.004

$$\frac{\partial J_i(\lambda_{i1},\ldots,\lambda_{ip})}{\partial \lambda_{ij}} = \mu \frac{\partial g_i(\lambda_{i1},\ldots,\lambda_{ip})}{\partial \lambda_{ij}},$$
$$\beta \lambda_{ij}^{\beta-1} J_{ij} = \mu,$$
$$\lambda_{ij} = \left(\frac{\mu}{\beta}\right)^{\frac{1}{\beta-1}} \cdot \frac{1}{(J_{ij})^{\frac{1}{\beta-1}}}. \tag{B.1}$$

As we know that $\sum_{l=1}^{p} \lambda_{il} = 1$, $\forall i$, we have

$$\sum_{l=1}^{p} \left(\frac{\mu}{\beta}\right)^{\frac{1}{\beta-1}} \cdot \frac{1}{(J_{il})^{\frac{1}{\beta-1}}} = 1. \tag{B.2}$$

Solving (B.2) for $(\frac{\mu}{\beta})^{\frac{1}{\beta-1}}$ and substituting in (B.1), we have that an extremum of J_i is reached when

$$\lambda_{ij} = \left[\sum_{l=1}^{p} \left(\frac{J_{ij}}{J_{il}}\right)^{\frac{1}{\beta-1}}\right]^{-1} = \left[\sum_{l=1}^{p} \left(\frac{\sum_{k=1}^{n} (u_{ik})^m \|\Phi(x_{kj}) - \Phi(v_{ij})\|^2}{\sum_{k=1}^{n} (u_{ik})^m \|\Phi(x_{kl}) - \Phi(v_{il})\|^2}\right)^{\frac{1}{\beta-1}}\right]^{-1}.$$

We have,

$$\frac{\partial J_i}{\partial \lambda_{ij}} = \beta \lambda_{ij}^{\beta-1} J_{ij}$$

then,

$$\frac{\partial^2 J_i}{\partial (\lambda_{ij})^2} = \beta(\beta-1)\lambda_{ij}^{\beta-2} J_{ij} \quad \text{and} \quad \frac{\partial^2 J_k}{\partial \lambda_{ij} \partial \lambda_{il}} = 0 \quad \forall l \neq j.$$

The Hessian matrix of J_i evaluated at $\boldsymbol{\lambda}_i = (\lambda_{i1},\ldots,\lambda_{ip})$ is

$$H(\boldsymbol{\lambda}_i) = \begin{pmatrix} \frac{\beta(\beta-1)J_{i1}}{\sum_{l=1}^{p} (\frac{J_{i1}}{J_{il}})^{\frac{\beta-2}{\beta-1}}} & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{\beta(\beta-1)J_{ip}}{\sum_{l=1}^{p} (\frac{J_{ip}}{J_{il}})^{\frac{\beta-2}{\beta-1}}} \end{pmatrix},$$

where $H(\boldsymbol{\lambda}_i)$ is positive definite so that we can conclude that this extremum is a minimum.

(b) Following a similar reasoning as in part (a) we conclude that

$$\lambda_j = \left[\sum_{l=1}^{p} \left(\frac{\sum_{i=1}^{c}\sum_{k=1}^{n} (u_{ik})^m \|\Phi(x_{kj}) - \Phi(v_{ij})\|^2}{\sum_{i=1}^{c}\sum_{k=1}^{n} (u_{ik})^m \|\Phi(x_{kl}) - \Phi(v_{il})\|^2}\right)^{\frac{1}{\beta-1}}\right]^{-1}.$$

(c) We want to minimize J with respect to λ_{ij}, $i = 1,\ldots,c$, $j = 1,\ldots,p$, under $\lambda_{ij} > 0$ $\forall i, j$ and $\prod_{j=1}^{p} \lambda_{ij} = 1$ $\forall i$. As the fuzzy membership degree u_{ik} ($i = 1,\ldots,c, k = 1,\ldots,n$), the centroids $\mathbf{v}_i$ ($i = 1,\ldots,c$), and the parameter m are fixed, we can rewrite the criterion J as

$$J(\boldsymbol{\lambda}_1,\ldots,\boldsymbol{\lambda}_c) = 2\sum_{i=1}^{c} J_i(\boldsymbol{\lambda}_i) = \sum_{i=1}^{c}\sum_{j=1}^{p} \lambda_{ij} \sum_{k=1}^{n} (u_{ik})^m \left\|\Phi(x_{kj}) - \Phi(v_{ij})\right\|^2$$

where $J_i(\boldsymbol{\lambda}_i) = J_i(\lambda_{i1},\ldots,\lambda_{ip}) = \sum_{j=1}^{p} \lambda_{ij} J_{ij}$, with $J_{ij} = \sum_{k=1}^{n} (u_{ik})^m \|\Phi(x_{kj}) - \Phi(v_{ij})\|^2$. The criterion J being additive, the problem becomes minimizing J_i, $i = 1,\ldots,c$. Let $g_i(\lambda_{i1},\ldots,\lambda_{ip}) = \prod_{j=1}^{p} \lambda_{ij} - 1 = \lambda_{i1} \times \cdots \times \lambda_{ip} - 1$. We want to determine the extremes of $J_i(\lambda_{i1},\ldots,\lambda_{ip})$ with the restriction $g_i(\lambda_{i1},\ldots,\lambda_{ip}) = 0$. To do so, we shall apply the Lagrange multipliers method to solve the following system

$$\nabla J_i(\lambda_{i1},\ldots,\lambda_{ip}) = \mu \nabla g_i(\lambda_{i1},\ldots,\lambda_{ip}).$$

Please cite this article in press as: M.R.P. Ferreira, F.A.T. de Carvalho, Kernel fuzzy c-means with automatic variable weighting, Fuzzy Sets and Systems (2013), http://dx.doi.org/10.1016/j.fss.2013.05.004

But $\nabla J_i(\lambda_{i1},\ldots,\lambda_{ip}) = (J_{i1},\ldots,J_{ip})$ and $\nabla g_i(\lambda_{i1},\ldots,\lambda_{ip}) = (\frac{1}{\lambda_{i1}},\ldots,\frac{1}{\lambda_{ip}})$, then, $(J_{i1},\ldots,J_{ip}) = \mu(\frac{1}{\lambda_{i1}},\ldots,\frac{1}{\lambda_{ip}})$. Thus, for $j = 1,\ldots,p$, $J_{ij} = \frac{\mu}{\lambda_{ij}} \Rightarrow \lambda_{ij} = \frac{\mu}{J_{ij}}$. As we know that $\prod_{l=1}^{p}\lambda_l = 1$, we have $\prod_{l=1}^{p}\frac{\mu}{J_{il}} = 1 \Rightarrow \frac{\mu^p}{\prod_{l=1}^{p}J_{il}} = 1 \Rightarrow \mu = (\prod_{l=1}^{p}J_{il})^{1/p}$ and it follows that an extremum value of J_i is reached when

$$\lambda_{ij} = \frac{\{\prod_{l=1}^{p}J_{il}\}^{1/p}}{J_{ij}} = \frac{\{\prod_{l=1}^{p}(\sum_{k=1}^{n}(u_{ik})^m\|\Phi(x_{kl}) - \Phi(v_{il})\|^2)\}^{\frac{1}{p}}}{\sum_{k=1}^{n}(u_{ik})^m\|\Phi(x_{kj}) - \Phi(v_{ij})\|^2}.$$

This extremum is $J_i(\lambda_{i1},\ldots,\lambda_{ip}) = \sum_{j=1}^{p}\lambda_{ij}J_{ij} = p\{J_{i1}\times\cdots\times J_{ip}\}^{\frac{1}{p}}$. As $J_i(1,\ldots,1) = \sum_{j=1}^{p}J_{ij} = J_{i1}+\cdots+J_{ip}$, and as it is well known that the arithmetic mean is greater than the geometric mean, i.e., $\frac{1}{p}\{J_{i1}+\cdots+J_{ip}\} > \{J_{i1}\times\cdots\times J_{ip}\}^{\frac{1}{p}}$ (the equality holds only if $J_{i1} = \cdots = J_{ip}$), we conclude that this extremum is a minimum.

(d) Following a similar reasoning as in part (c) we conclude that

$$\lambda_j = \frac{\{\prod_{l=1}^{p}(\sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ik})^m\|\Phi(x_{kl}) - \Phi(v_{il})\|^2)\}^{\frac{1}{p}}}{\sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ik})^m\|\Phi(x_{kj}) - \Phi(v_{ij})\|^2}.$$

Thus, Proposition 3.3 was proved. □

Appendix C. Proof of Proposition 3.5

Whichever the distance function (Eqs. (34), (35), (36), (37) and (38)), the cluster centroid $\mathbf{v}_i^{\Phi} = (v_{i1}^{\Phi},\ldots,v_{ip}^{\Phi})$ $(i = 1,\ldots,c)$, which minimizes the criterion J given in Eq. (15), has its components v_{ij}^{Φ} $(j = 1,\ldots,p)$ updated according to the following expression:

$$v_{ij}^{\Phi} = \frac{\sum_{k=1}^{n}(u_{ik})^m\Phi(x_{kj})}{\sum_{k=1}^{n}(u_{ik})^m}.$$

Proof. For a cluster i and a variable j the problem becomes to find the centroid v_{ij} that minimizes the term

$$\sum_{k=1}^{n}(u_{ik})^m\left\|\Phi(x_{kj}) - v_{ij}^{\Phi}\right\|^2. \tag{C.1}$$

The calculation of the centroids v_{ij}, $i = 1,\ldots,c$, $j = 1,\ldots,p$, proceeds as follows:

$$\frac{\partial J}{\partial v_{ij}^{\Phi}} = \sum_{k=1}^{n}(u_{ik})^m\frac{\partial\|\Phi(x_{kj}) - v_{ij}^{\Phi}\|^2}{\partial v_{ij}^{\Phi}} = 0,$$

$$-2\sum_{k=1}^{n}(u_{ik})^m\left(\Phi(x_{kj}) - v_{ij}^{\Phi}\right) = 0,$$

$$v_{ij}^{\Phi} = \frac{\sum_{k=1}^{n}(u_{ik})^m\Phi(x_{kj})}{\sum_{k=1}^{n}(u_{ik})^m}. \qquad \square$$

References

[1] A.K. Jain, M.N. Murty, P.J. Flynn, Data clustering: A review, ACM Comput. Surv. 31 (3) (1999) 264–323.
[2] R. Xu, D.I.I. Wunusch, Survey of clustering algorithms, IEEE Trans. Neural Netw. 16 (3) (2005) 645–678.
[3] A.D. Gordon, Classification, 2nd edition, Chapman & Hall, Boca Raton, 1999.
[4] F. Höppner, F. Klawonn, R. Kruse, T. Runkler, Fuzzy Cluster Analysis, John Wiley & Sons, Inc., 1999.
[5] M. Girolami, Mercer kernel-based clustering in feature space, IEEE Trans. Neural Netw. 13 (3) (2002) 780–784.
[6] J.C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum, New York, 1981.
[7] T. Kohonen, Self-organized formation of topologically correct feature maps, Biol. Cybern. 43 (1) (1982) 59–69.

Please cite this article in press as: M.R.P. Ferreira, F.A.T. de Carvalho, Kernel fuzzy c-means with automatic variable weighting, Fuzzy Sets and Systems (2013), http://dx.doi.org/10.1016/j.fss.2013.05.004

[8] T. Kohonen, The self-organizing map, Proc. IEEE 78 (1990) 1464–1480.
[9] T. Kohonen, Self-Organizing Maps, Springer, New York, 2001.
[10] R.R. Yager, D.P. Filev, Approximate clustering via the mountain method, IEEE Trans. Syst. Man Cybern. 24 (8) (1994) 1279–1284.
[11] T.M. Martinetz, S.G. Berkovich, K.J. Schulten, 'Neural gas' network for vector quantization and its application to time-series prediction, IEEE Trans. Neural Netw. 4 (4) (1993) 558–569.
[12] M. Filippone, F. Camastra, F. Masulli, S. Rovetta, A survey of kernel and spectral methods for clustering, Pattern Recognit. 41 (2008) 176–190.
[13] F. Camastra, A. Verri, A novel kernel method for clustering, IEEE Trans. Neural Netw. 27 (5) (2005) 801–804.
[14] D. Horn, Clustering via Hilbert space, Physica A 302 (2001) 70–79.
[15] A.S. Have, M.A. Girolami, J. Larsen, Clustering via kernel decomposition, IEEE Trans. Neural Netw. 17 (1) (2006) 256–264.
[16] I.S. Dhillon, Y. Guan, 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.
[17] D.Q. Zhang, S.C. Chen, Fuzzy clustering using kernel method, in: The 2002 International Conference on Control and Automation, 2002 ICCA, 2002, pp. 162–163.
[18] D.Q. Zhang, S.C. Chen, Z.S. Pan, K.R. Tan, Kernel-based fuzzy clustering incorporating spatial constraints for image segmentation, in: International Conference on Machine Learning and Cybernetics, vol. 4, 2003, pp. 2189–2192.
[19] S.C. Chen, D.Q. Zhang, Robust image segmentation using FCM with spatial constraints based on new kernel-induced distance measure, IEEE Trans. Syst. Man Cybern. 34 (4) (2004) 1907–1916.
[20] D.Q. Zhang, S.C. Chen, A novel kernelized fuzzy c-means algorithm with application in medical image segmentation, Artif. Intell. Med. 32 (1) (2004) 37–50.
[21] D. Macdonald, 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.
[22] R. Inokuchi, 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.
[23] D.W. Kim, K.Y. Lee, D. Lee, K.H. Lee, A kernel-based subtractive clustering method, Pattern Recognit. Lett. 26 (2005) 879–891.
[24] A.K. Qinand, 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.
[25] A. Ben-Hur, D. Horn, H.T. Siegelmann, V. Vapnik, Support vector clustering, J. Mach. Learn. Res. 2 (2001) 125–137.
[26] S. Borer, W. Gerstner, A new kernel clustering algorithm, in: Proceedings of the Ninth Internat. Conf. on Neural Information Processing, vol. 5, 2002, pp. 2527–2531.
[27] J.H. Chiang, P.Y. Hao, A new kernel-based fuzzy clustering approach: support vector clustering with cell growing, IEEE Trans. Fuzzy Syst. 11 (4) (2003) 518–527.
[28] D.W. Kim, K.Y. Lee, D. Lee, K.H. Lee, Evaluation of the performance of clustering algorithms in kernel-induced feature space, Pattern Recognit. 38 (4) (2005) 607–611.
[29] D. Graves, W. Pedrycz, Kernel-based fuzzy clustering and fuzzy clustering: A comparative experimental study, Fuzzy Sets Syst. 161 (2010) 522–543.
[30] H. Shen, J. Yang, S. Wang, X. Liu, Attribute weighted mercer kernel based fuzzy clustering algorithm for general non-spherical datasets, Soft Comput. 10 (11) (2006) 1061–1073.
[31] F.A.T. De Carvalho, C.P. Tenório, N.L. Cavalcanti Jr., Partitional fuzzy clustering methods based on adaptive quadratic distances, Fuzzy Sets Syst. 157 (2006) 2833–2857.
[32] J. Mercer, Functions of positive and negative type and their connection with the theory of integrals equations, Proc. R. Soc. Lond. 209 (1909) 415–446.
[33] S. Haykin, Neural Networks: A Comprehensive Foundation, 2nd edition, Prentice Hall PTR, Upper Saddle River, NJ, USA, 1998.
[34] B. Schölkopf, A.J. Smola, K.R. Müller, Nonlinear component analysis as a kernel eigenvalue problem, Neural Comput. 10 (5) (1998) 1299–1319.
[35] K.R. Müller, S. Mika, G. Rätsch, K. Tsuda, B. Schölkopf, An introduction to kernel-based learning algorithms, IEEE Trans. Neural Netw. 12 (2) (2001) 181–202.
[36] D.Q. Zhang, 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.
[37] T. Graepel, K. Obermayer, Fuzzy topographic kernel clustering, in: W. Brauer (Ed.), Proceedings of the Fifth GI Workshop Fuzzy Neuro Systems '98, 1998, pp. 90–97.
[38] B. Schölkopf, A.J. Smola, Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond, The MIT Press, 2002.
[39] D.E. Gustafson, W.C. Kessel, Fuzzy clustering with a fuzzy covariance matrix, in: IEEE Conference on Decision and Control Including the 17th Symposium on Adaptive Processes, vol. 17, IEEE, 1978, pp. 761–766.
[40] E. Diday, J.C. Simon, Digital Pattern Classification, Springer, Berlin, 1976, pp. 47–94 (Ch. Clustering analysis).
[41] G. Celeux, E. Diday, G. Govaert, Y. Lechevallier, H. Ralambondrainy, Classification Automatique des Données, Bordas, Paris, 1989.
[42] M. Chavent, F.D. Carvalho, Y. Lechevallier, R. Verde, New clustering methods for interval data, Comput. Stat. 21 (2) (2006) 211–229.
[43] L. Hubert, P. Arabie, Comparing partitions, J. Classif. 2 (1985) 193–218.
[44] L. Breiman, J. Friedman, C.J. Stone, R.A. Olshen, Classification and Regression Trees, Chapman & Hall/CRC, Boca Raton, 1984.
[45] G.W. Milligan, Clustering and Classification, World Scientific, Singapore, 1996, pp. 341–375 (Ch. Clustering validation: Results and implications for applied analysis).

Please cite this article in press as: M.R.P. Ferreira, F.A.T. de Carvalho, Kernel fuzzy c-means with automatic variable weighting, Fuzzy Sets and Systems (2013), http://dx.doi.org/10.1016/j.fss.2013.05.004

[46] B. Caputo, K. Sim, F. Furesjo, A. Smola, Appearance-based object recognition using SVMs: which kernel should I use? in: Proceedings of NIPS Workshop on Statistical Methods for Computational Experiments in Visual Processing and Computer Vision, 2002.
[47] R. Winkler, F. Klawonn, R. Kruse, Fuzzy c-means in high dimensional spaces, Int. J. Fuzzy Syst. Appl. 1 (1) (2011) 1–16.
[48] R. Winkler, F. Klawonn, R. Kruse, Problems of fuzzy c-means clustering and similar algorithms with high dimensional data sets, Int. J. Fuzzy Syst. Appl. 1 (1) (2011) 1–16.
[49] A. Frank, A. Asuncion, UCI machine learning repository, http://archive.ics.uci.edu/ml, 2010.

Please cite this article in press as: M.R.P. Ferreira, F.A.T. de Carvalho, Kernel fuzzy *c*-means with automatic variable weighting, Fuzzy Sets and Systems (2013), http://dx.doi.org/10.1016/j.fss.2013.05.004