Journal of Classification 32:176-211 (2015)
DOI: 10.1007/s00357-015-9181-3

Kernel-Based Methods to Identify Overlapping Clusters with Linear and Nonlinear Boundaries

Chiheb-Eddine Ben N'Cir

University of Tunis, Tunisia

Nadia Essoussi

University of Carthage, Tunisia

Mohamed Limam

University of Tunis, Tunisia and Dhofar University, Oman

Abstract: Detecting overlapping structures and identifying non-linearly-separable clusters with complex shapes are two major issues in clustering. This paper presents two kernel based methods that produce overlapping clusters with both linear and nonlinear boundaries. To improve separability of input patterns, we used for both methods Mercer kernel technique. First, we propose Kernel Overlapping K-means I (KOKMI), a centroid based method, generalizing kernel K-means to produce non-disjoint clusters with nonlinear separations. Second, we propose Kernel Overlapping K-means II (KOKMII), a medoid based method improving the previous method in terms of efficiency and complexity. Experiments performed on non-linearly-separable and real multi-labeled data sets show that proposed learning methods outperform the existing ones.

Keywords: Overlapping clustering; Non-disjoint clusters; Learning multi-labels; Kernel methods; Kernel K-means; Nonlinear separations; Non-linearly-separable clusters.

Authors' Addresses: C.-E. Ben N'Cir, LARODEC, ISG, University of Tunis, Tunisia, Supérieur de Gestion de Tunis, 41 rue de la liberté, Cité Bouchoucha, 2000 Le Bardo, Tunisie, email: chiheb.benncir@isg.rnu.tn; N. Essoussi, LARODEC, FSEG Nabeul, University of Carthage, Tunisia, email: nadia.essoussi@isg.rnu.tn; Mohamed Limam, LARODEC, ISG, University of Tunis, Tunisia and Dhofar University, Oman, email: mohamed.limam@isg.rnu.tn.

Published online: 8 July 2015

1. Introduction

Clustering is an important task in data mining. It aims to divide data into groups where similar observations are assigned to the same group called cluster. It has been applied successfully in many fields such as in market segmentation (Van Hattum and Hoijtink 2009; DeSarbo and Cron 1988), in social network analysis (Wang and Fleury 2011; Pérez-Suárez, Martínez-Trinidad, Carrasco-Ochoa, and Medina-Pagola 2013) and in document classification (Chao-Liu, Wu, and Liu 2011; Aliguliyev 2009). For many clustering applications, it may be recommended to allow observations to belong to more than one cluster. This kind of applications is referred to as *overlapping clustering* (Banerjee, Krumpelman, Basu, Mooney, and Ghosh 2005; Fellows, Guo, Komusiewicz, Niedermeier, and Uhlmann 2011).

Overlapping clustering is different from crisp clustering where each observation belongs to exactly one cluster leading to K exclusive clusters representing the data. Overlapping clustering is based on the assumption that clusters are non-exclusive. In this configuration, any observation can belong to one or many clusters. Looking for non-exclusive clusters contributes to solve many real life problems that require to find overlapping clusters in order to fit the data set structure. For example, in social network analysis, community extraction algorithms need to detect overlapping clusters where an actor can belong to multiple communities (Tang and Liu 2009; Wang, Tang, Gao, and Liu 2010; Fellows, Guo, Komusiewicz, Niedermeier, and Uhlmann 2011). In video classification, overlapping clustering is a necessary requirement where videos have potentially multiple genres (Snoek, Worring, van Gemert, Geusebroek, and Smeulders 2006). In emotion detection, overlapping clustering methods need to detect different emotions for a specific piece of music (Wieczorkowska, Synak, and Ras 2006).

Several overlapping clustering methods based on different approaches are proposed in the literature. This work will focus on overlapping methods, extending or generalizing K-means algorithm, to produce non-disjoint groups. Based on the squared Euclidean distance, these methods look for linear separations between clusters and fail to produce clusters with nonlinear boundaries (Filippone, Camastra, Masulli, and Rovetta 2008; Girolami 2002). This fact makes existing methods not appropriate to detect overlapping groups in real life applications where separations are frequently nonlinear and complex. Hence, the use of nonlinear methods becomes a necessary requirement to detect relevant clusters.

In order to identify relevant overlapping clusters with nonlinear separations, this work focuses on kernel K-means, a nonlinear variant of K-means. This method is generalized to look for overlapping clusters by modeling the possibility of overlaps in the optimized criterion and by exploring the possible space of overlapping assignments rather than that of partitions. Two variants are proposed, Kernel Overlapping K-Means I (KOKMI) and Kernel Overlapping K-Means II (KOKMII) to produce clusters in a high, possibly infinite, dimensional space based on the Mercer Kernel technique. The second method is an improvement of the first method in terms of computational complexity and efficiency.

This paper is organized as follows: Section 2 presents existing overlapping clustering methods and discusses their shortcomings in detecting nonlinear separations between clusters. Section 3 presents the Mercer Kernel technique and kernel K-means. Section 4 introduces the proposed methods KOKMI and KOKMII while Section 5 describes experiments and show results on different data sets. Finally, Section 6 presents the conclusion and future works.

2. Overlapping Clustering

2.1 Related Work

In many clustering applications, the possibility that an observation belongs to more than one cluster is usually ignored. However, few works have focused on detecting non-disjoint groups in data. First, some methods modify results of fuzzy classification to produce overlapping clusters such as the extension of clusters obtained with Fuzzy c-means by thresholding cluster memberships (Deodhar and Ghosh 2006; Lingras and West 2004; Zhang, Wang, and Zhang 2007). However, learning the prior threshold is a difficult task as the number of clusters increases. In addition, criteria to be optimized iteratively look for optimal partitions without introducing overlaps between data in the optimization step. These contributions can lead to suitable results in some applications but being not based on theoretical approaches, their extensions or improvements are limited (Banerjee, Krumpelman, Basu, Mooney, and Ghosh 2005).

Recent methods look for overlapping clusters based on theoretical approaches. The most important advantage of these methods is their ability to produce non-disjoint clusters where overlaps are introduced in their optimized criteria. There are two kinds of approaches : *SUM* and *AVERAGE*. We denote by *SUM* methods which group observations into overlapping clusters while minimizing the sum of distances between each observation and the *sum* of clusters' representatives (prototypes or centroids) to which the observation belongs to. Examples of these methods are Principal Cluster Analysis (PCL) (Mirkin 1987b) and its variants (Mirkin 1987a, 1990), the Alternating Least Square algorithms (ALS) (Depril, Van Mechelen, and Mirkin 2008; Wilderjans, Depril, and Van Mechelen 2012) and the Low-dimensional Additive Overlapping Clustering (Depril, Van Mechelen, and Wilderjans 2012).

Conversely, methods based on *AVERAGE* approach group observations into overlapping clusters while minimizing the sum of distances between each observation and the *average*, instead of the sum, of clusters' representatives to which the observation belongs to. Examples of these methods are the Overlapping K-means (OKM) (Cleuziou 2008), Overlapping K-Medoid (OKMED) (Cleuziou 2010), the Evidential C-means (ECM) (Masson and Denoeux 2008) , Overlapping Clustering with Sparseness Constraint (Lu, Hong, Street, Wang, and Tong 2012) and Overlapping Self Organizing Map (OSOM) (Cleuziou 2013).

Clustering based on *SUM* and *AVERAGE* approaches can lead to non disjoint groups. The adoption of these approaches is motivated by requirements of real life applications. Methods based on *SUM* were used in grouping patients into diseases. Each patient may suffer from more than one disease and therefore could be assigned to multiple syndrome clusters. Thus, the final symptom profile of a patient is the sum of the symptom profiles of all syndromes he is suffering from. However, to avoid false analysis, these methods need sometimes to prepare data to have zero mean. For example, if a symptom variable denotes the body temperature, then when a patient simultaneously suffers from two diseases, it does not make sense to assume that his body temperature equals to the sum of body temperatures as associated with two diseases.

Methods based on *AVERAGE* approach have been well used to group music signals into different emotions and films into several genres. These methods are based on a geometrical reasoning in the data space by considering overlapping observations resulting of intersections of boundary surfaces of overlapping clusters. For example, if a film belongs to action and horror genres, it should have some shared properties with these categories of films, however it can neither be a full action film or a full horror one. So, overlapping films belonging to action and horror categories may geometrically appear in the boundary surface between full horror and full action films.

2.2 Problem Description

To study patterns produced by *SUM* and *AVERAGE* approaches we visualize partitioning of two existing overlapping methods, OKM and ALS methods, which are respectively based on the *AVERAGE* and the *SUM* approach. Partitionings are visualized through Voronoï cells[1] obtained for

1. To build Voronoï cells we considered a rectangle (500×500) of pixels on the screen and we considered three initial pixels $(100, 50)$, $(100, 400)$ and $(400, 400)$ as clusters' prototypes for the first case study and three other pixels $(100, 200)$, $(200, 100)$ and $(250, 200)$ as clusters' prototypes for the second case study.

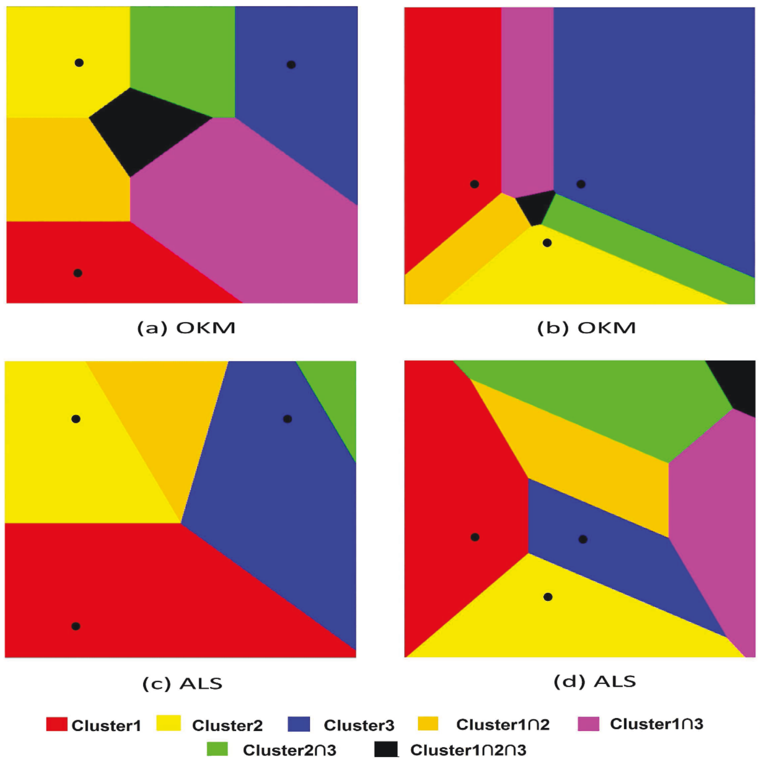

Figure 1. Voronoï cells obtained with OKM (AVERAGE based Approach) and ALS (SUM based Approach) for three clusters.

three clusters over a two dimensional space as defined by the objective criterion optimized by these methods. Figure 1 shows an example of Voronoï cells where the representation space is divided into several cells and each possible combination of clusters is associated to one cell. For OKM, Figures 1(a) and 1(b) show seven cells, with all possible combinations of clusters except the empty set, where each cell is centered on a prototype or a combination (average) of prototypes. Overlapping cells get larger as the clusters' prototypes become more distant. For ALS, Figure 1(c) shows that overlaps between clusters are not recovered when prototypes are distant, except $cluster1 \cap cluster2$ and $cluster2 \cap cluster3$. However, when prototypes become closer to each other, as shown in Figure 1(d), all combinations of cluster prototypes are given. We easily notice that overlapping cells are the resulting combination (sum) of representatives of single clusters. For example, the magenta cell results from the combination of $cluster1$ (Red cell) and $cluster3$ (Blue cell).

Hence, OKM and ALS have the ability to detect overlapping clusters even though separations between resulting clusters are linear. This fact makes these methods not well adapted to real life applications where separations between clusters are usually nonlinear with complex shapes.

Since complex and nonlinear separations between clusters are expected depending on the target application, we focus our study on the nonlinear method kernel K-means (Girolami 2002). We show how it can be generalized to detect overlapping clusters with both linear and nonlinear boundaries.

3. Kernel Machines

3.1 Mercer Kernel

To solve the problem of non-linearly-separable clusters, many methods have been modified incorporating kernels such as SVM (Cortes and Vapnik 1995) which performs better than other classification algorithms in many applications (Cristianini, Campbell, and Burges 2002). The success of SVM has extended the use of kernels to other learning algorithms (e.g., Kernel PCA (Schölkopf, Smola, and Müller 1998), kernel K-means (Girolami 2002), kernel fuzzy C-means (Wu, Xie, and Yu 2003)). These methods use positive-definite kernel, also referred to as Mercer kernel, to implicitly map data from original space called input space into a high dimensional space called feature space. Computing a partition with linear boundaries in the feature space results in a partition with nonlinear boundaries in the input space.

A function $K : X \times X \longrightarrow \mathbb{R}$ is called a Mercer kernel if and only if K is symmetric and the following condition of semi-definiteness holds:

$$\sum_{i=1}^{N}\sum_{j=1}^{N} c_i c_j K_{ij} \geq 0 \quad \forall N \geq 2 \quad and \quad \forall c_i, c_j \in \mathbb{R}, \tag{1}$$

where K_{ij} is the dot product of mapped data in the feature space defined as follows:

$$K_{ij} = K(x_i, x_j) = \phi(x_i)\phi(x_j), \tag{2}$$

where $\phi : X \longrightarrow F$ is a mapping from the input space X to a high dimensional feature space F. Table 1 gives some widely used Mercer kernels such as Linear, Polynomial, Gaussian, Exponential and Sigmoid kernels.

The advantage of kernels consists in the possibility of computing distance measures between observations in the feature space F without ex-

Table 1. Examples of Positive-definite kernel functions

Kernel function	Value
Linear Kernel	$K_{ij} = K(x_i, x_j) = x_i.x_j$
Polynomial Kernel	$K_{ij} = K(x_i, x_j) = ((x_i.x_j) + c)^d$
Gaussian RBF Kernel	$K_{ij} = K(x_i, x_j) = \exp(\frac{-\|x_i - x_j\|^2}{2\sigma^2})$
Exponential RBF Kernel	$K_{ij} = K(x_i, x_j) = \exp(\frac{-\|x_i - x_j\|}{2\sigma^2})$
Sigmoid Kernel	$K_{ij} = K(x_i, x_j) = \tanh(\theta(x_i.x_j) + c)$

plicitly knowing ϕ. The kernel induced distance measure (Schölkopf et al. 1998), also referred to as Kernel Trick, can be computed as follows:

$$\begin{aligned} \| \phi(x_i) - \phi(x_j) \|^2 &= (\phi(x_i) - \phi(x_j))((\phi(x_i) - \phi(x_j)) \\ &= \phi(x_i)\phi(x_i) - 2\phi(x_i)\phi(x_j) + \phi(x_j)\phi(x_j) \\ &= K_{ii} - 2K_{ij} + K_{jj}. \end{aligned} \tag{3}$$

The use of Mercer kernel in clustering can be divided into three categories (Filippone, Camastra, Masulli, and Rovetta 2008). First, methods based on kernelization of the metric (Wu, Xie, and Yu 2003; Zhang and Chen 2002, 2003, 2004) which look for centroids in input space while distances between patterns and centroids are computed by means of kernels. Second, methods based on clustering in the feature space (Graepel and Obermayer 1998; Inokuchi and Miyamoto 2004; Qinand and Suganthan 2004; Girolami 2002) which map data into a higher feature space and then compute centroids using the Kernel Trick. Third, methods based on support vectors (Camastra and Verri 2005; Ben-Hur, Horn, Siegelmann, and Vapnik 2001) which use One Class SVM to find a minimum enclosing sphere in the feature space able to include almost all data excluding outliers.

In this paper, Mercer kernel is used for clustering in the feature space (the second category of the use of Mercer kernel in clustering) for KOKMI and KOKMII where the whole learning process is computed in a high dimensional space.

3.2 Kernel *K*-Means

Kernel K-means is an extension of K-means algorithm to solve the problem of nonlinearly separable clusters. By an implicit mapping of the data from an input space to a higher feature space, kernel K-means looks for separations in feature space and solves the problem of clustering non-linearly-separable data. For a finite data sample X, the kernel function yields a symmetric $N \times N$ positive definite matrix K, where each K_{ij} entry is the dot product between the representations in feature space, $\phi(x_i)$ and $\phi(x_j)$, of observations x_i and x_j as measured by the kernel function (Camastra and Verri 2005).

Kernel *K*-means aims to minimize the sum of squared Euclidean errors in feature space given by:

$$J(\Pi) = \sum_{i=1}^{N} \sum_{c=1}^{C} P_{ic} \|\phi(x_i) - m_c^{\phi}\|^2, \tag{4}$$

where P_{ic} is a binary variable indicating membership of observation x_i to cluster c and m_c^{ϕ} is the prototype of cluster c in feature space. The prototype is defined in the feature space as the center of gravity of observations that belong to cluster c. This prototype cannot be computed because the mapping function ϕ is generally unknown. However, the clustering error $\| \phi(x_i) - m_c^{\phi}\|$ can be computed using the Kernel Trick as follows:

$$\begin{aligned} &\|\phi(x_i) - m_c^{\phi}\|^2 \\ &\quad = \|\phi(x_i) - \frac{1}{W_c}\sum_{j=1}^{N} P_{jc}\phi(x_j)\|^2 \\ &\quad = K_{ii} - \frac{2}{W_c}\sum_{j=1}^{N} P_{jc}K_{ij} + \frac{1}{(W_c)^2}\sum_{j=1}^{N}\sum_{g=1}^{N} P_{jc}P_{gc}K_{jg}, \end{aligned} \tag{5}$$

where $W_c = \sum_{j=1}^{N} P_{jc}$ is the number of observations that belong to cluster c, $P_{jc} \in \{0, 1\}$ and $P_{gc} \in \{0, 1\}$ denote the memberships of observations x_j and x_g to cluster c. Then, the clustering error function in kernel *K*-means can be written as follows:

$$\begin{aligned} &J(\Pi) = \\ &\sum_{i=1}^{N}\sum_{c=1}^{C} P_{ic}[K_{ii} - \frac{2}{W_c}\sum_{j=1}^{N} P_{jc}K_{ij} + \frac{1}{(W_c)^2}\sum_{j=1}^{N}\sum_{g=1}^{N} P_{jc}P_{gc}K_{jg}]. \end{aligned} \tag{6}$$

To minimize the clustering error (6), kernel *K*-means performs two principal steps: the determination of the nearest cluster from each observation in the feature space and the update of memberships matrix. The stopping rule is defined by the maximal number of iterations and the minimal improvement of the objective function between two iterations.

4. Overlapping Clustering in a High Dimensional Feature Space

In order to look for overlapping clusters with linear and nonlinear boundaries, we propose two methods, KOKMI and KOKMII, where all steps of the clustering algorithm are computed in a high dimensional feature space. For both methods and without loss of generality, the *AVERAGE* approach is used to introduce overlaps in the objective criterion but the *SUM* approach could be used as well.

4.1 KOKMI: Kernel Overlapping *K*-Means I

The first method generalizes kernel *K*-means to produce overlapping clusters by introducing overlaps between clusters in the objective function. Given a set of observations $X = \{x_i\}_{i=1}^N$ with $x_i \in \mathbb{R}^d$ and N the number of observations and given an implicit nonlinear mapping function ϕ, the aim of KOKMI is to find a set $\Pi = \{\pi_c\}_{c=1}^C$ of C overlapping clusters such that the following objective function is minimized:

$$J(\Pi) = \sum_{i=1}^{N} \|\phi(x_i) - \overline{\phi(x_i)}\|^2, \tag{7}$$

where $\overline{\phi(x_i)}$ the representative of observation x_i defined in feature space by the *average* of clusters prototypes which x_i belongs to:

$$\overline{\phi(x_i)} = \frac{\sum_{c=1}^{C} P_{ic} m_c^{\phi}}{\sum_{c=1}^{C} P_{ic}}, \tag{8}$$

where $P_{ic} \in \{0, 1\}$ is a binary variable that indicates membership of x_i to cluster c and m_c^{ϕ} is the representative (prototype) of cluster c in the feature space. The objective function iteratively minimizes the distance between each observation and its representative in a new feature space F obtained by the nonlinear mapping function ϕ. For minimizing the objective function J in (7) KOKMI assigns, at each iteration, observations to one or several clusters and then computes the new value of J. If J shows improvement, the same steps are repeated until a stop-condition is met. The stop-condition can be either the maximum number of iterations or the minimum improvement in the objective.

4.1.1. Evaluation of the Objective Function in Feature Space

To evaluate the objective function J we need to compute at each iteration clusters prototypes in the feature space. Each prototype is defined by the average of observations that belong to the corresponding cluster weighted according to the number of assignments of each observation as follows:

$$m_c^{\phi} = \frac{\sum_{j=1}^{N} P_{jc} w_j \phi(x_j)}{W_c}, \tag{9}$$

where w_j is the weight assigned to observation x_j defined by $w_j = 1/(\sum_{c=1}^{C} P_{jc})^2$ and $W_c = \sum_{j=1}^{N} P_{jc} w_j$ is the sum of weights of the observations that belong to cluster c. The weight w_j decreases as well as assignments' cardinality of x_j increases in order to reduce the effect of overlapping observations in determining the representative of each cluster.

Since the mapping function ϕ is not explicitly known, it is impossible to compute clusters prototypes in feature space using Equation (9). Nevertheless, it is always possible to compute distances between patterns and prototypes in feature space using the Kernel Trick. Therefore, using this technique, the objective function of KOKMI becomes:

$$\begin{aligned} J(\Pi) &= \sum_{i=1}^{N} \|\phi(x_i) - \frac{1}{L_i} \sum_{c=1}^{C} P_{ic} \frac{1}{W_c} \sum_{j=1}^{N} P_{jc} w_j \phi(x_j)\|^2 \\ &= \sum_{i=1}^{N} \{\phi(x_i)\phi(x_i) - \frac{2}{L_i} \sum_{c=1}^{C} \sum_{j=1}^{N} P_{ic} \frac{1}{W_c} P_{jc} w_j \phi(x_i)\phi(x_j) + \\ &\quad \frac{1}{L_i^2} \sum_{c=1}^{C} \sum_{j=1}^{N} \sum_{t=1}^{C} \sum_{g=1}^{N} P_{ic} \frac{1}{W_c} P_{jc} P_{it} \frac{1}{W_t} P_{gt} w_j w_g \phi(x_j)\phi(x_g)\}, \end{aligned} \tag{10}$$

where $L_i = \sum_{c=1}^{C} P_{ic}$. If each dot product in feature space is replaced by any Mercer Kernel, the objective function J can be computed in feature space as follows:

$$J(\Pi) = \sum_{i=1}^{N} d[\phi(x_i), \overline{\phi(x_i)}], \tag{11}$$

where:

$$\begin{aligned} & d[\phi(x_i), \overline{\phi(x_i)}] \\ &= K_{ii} - \frac{2}{L_i}\sum_{c=1}^{C}\sum_{j=1}^{N} P_{ic}\frac{1}{W_c}P_{jc}w_jK_{ij} + \\ & \frac{1}{L_i^2}\sum_{c=1}^{C}\sum_{j=1}^{N}\sum_{t=1}^{C}\sum_{g=1}^{N} P_{ic}\frac{1}{W_c}P_{jc}P_{it}\frac{1}{W_t}P_{gt}w_jw_gK_{jg}. \end{aligned} \tag{12}$$

If observations are assigned to only one cluster, as in hard clustering methods, the objective function in KOKMI is reduced to:

$$\begin{aligned} J(\Pi) &= \sum_{i=1}^{N}[K_{ii} - \frac{2}{1}\sum_{c=1}^{C}\sum_{j=1}^{N} P_{ic}\frac{1}{W_c}P_{jc}K_{ij} + \\ & \frac{1}{1^2}\sum_{c=1}^{C}\sum_{j=1}^{N}\sum_{t=1}^{C}\sum_{g=1}^{N} P_{ic}\frac{1}{W_c}P_{jc}P_{it}\frac{1}{W_t}P_{gt}K_{jg}] \\ &= \sum_{i=1}^{N}\sum_{c=t=1}^{C} P_{ic}[K_{ii} - \frac{2}{W_c}\sum_{j=1}^{N} P_{jc}K_{ij} + \\ & \frac{1}{(W_c)^2}\sum_{j=1}^{N}\sum_{g=1}^{N} P_{jc}P_{gc}K_{jg}]. \end{aligned} \tag{13}$$

This reduced objective function exactly matches with the objective function of kernel K-means.

4.1.2. Multi-Assignments of Observations in Feature Space

The assignment step looks for optimal assignments which minimize the objective function. The latter can be solved through searching for optimal cluster memberships by evaluating all possible 2^C combinations of clusters for each observation. However, it becomes computationally infeasible in real life applications. To overcome this problem, a heuristic solution is introduced to minimize the objective function and to explore a subspace of possible assignments. Given an observation x_i, a set of C clusters and possibly an old assignments of x_i, new assignments of x_i are determined using the function ASSIGN. Firstly, this function assigns x_i to the closest cluster, updates representative $\overline{\phi(x_i)}$ and then evaluates the distance between

```
Algorithm 1 $ASSIGN(x_i, \{c = 1, ..., c = C\}, A_i^{old}) \to A_i$
INPUT  $x_i$: Observation in $\mathbb{R}^d$.
    $\{c = 1, ..., c = C\}$: Set of $C$ clusters.
    $A_i^{old}$: Old assignment of observation $x_i$.
OUTPUT  $A_i$: New assignment of $x_i$.
 1: set $A_i = \{c^\star\}$ using Equation 14 , evaluate the distance $d[\phi(x_i), \overline{\phi(x_i)}]$
    with assignments $A_i$ using Equation 12.
 2: Look for the next nearest cluster $c^\star$ which is not included in $A_i$ such that
    $c^\star = \min_{\{c=1,\ldots,c=C\}/A_i} \|\phi(x_i) - m_c^\phi\|^2$ using Equation 14
 3: Evaluate the distance $d[\phi(x_i), \overline{\phi(x_i)'}]$ with assignments $A_i^{'} = A_i \cup c^\star$
 4: if $d[\phi(x_i), \overline{\phi(x_i)'}] \leq d[\phi(x_i), \overline{\phi(x_i)}]$ then
 5:   $A_i \leftarrow A_i \cup \{c^\star\}$, and go to step 2.
 6: else
 7:   if $d[\phi(x_i), \overline{\phi(x_i)}] \leq d[\phi(x_i), \overline{\phi(x_i)^{old}}]$ then
 8:     return $A_i$.
 9:   else
10:     return $A_i^{old}$.
11:   end if
12: end if
```

the observation and its representative using Equation (12). Next, this function looks for the next nearest cluster which is not considered in the set of assignments. This cluster is added to the assignments of observation x_i, then the representative $\overline{\phi(x_i)}$ is updated and the distance between x_i and its new representative is reevaluated. If the distance is minimized, the function continues the assignment of x_i to the next nearest cluster. Otherwise, the function compares obtained assignments in the last step with those given before performing the function ASSIGN and returns the optimal assignments which minimize the distance between x_i and its representative. The old assignments are evaluated to ensure the minimization of the objective function after each assignment step. The pseudo code of the function ASSIGN is described by Algorithm 1.

We note that looking for the index of the closest cluster $c_m in$ from an observation is determined by the minimum of distances between the observation and the clusters' prototypes which are computed in feature space using the Kernel Trick as follows:

$$c_{min} = \min_{\{c=1,\ldots,c=C\}} \|\phi(x_i) - m_c^\phi\|^2$$

Algorithm 2 $KOKMI(X, t_{max}, \varepsilon, C) \rightarrow \{\pi_c\}_{c=1}^{C}$

```
INPUT $X$: set of observations in $\mathbb{R}^d$.
OUTPUT $\pi$: Set of $C$ clusters.
1: Initialize representatives of clusters with random clusters prototypes.
2: t=0.
3: Assign each observation $x_i$ to one or several clusters using function"$ASSIGN$".
4: Compute objective function $J_t(\Pi)$ using Equation 11.
5: if ($t < t_{max}$ and $J_{t-1}(\Pi) - J_t(\Pi) > \varepsilon$) or ($t = 0$) then
6:     set $t = t + 1$ and go to step 3.
7: else
8:     Return the clusters' memberships.
9: end if
```

$$= \min_{\{c=1,\ldots,c=C\}} \{K_{ii} - \frac{2}{W_c} \sum_{j=1}^{N} P_{jc} w_j K_{ij} + \frac{1}{(W_c)^2} \sum_{j=1}^{N} \sum_{g=1}^{N} P_{jc} P_{gc} w_j w_g K_{jg}\}. \quad (14)$$

A pseudo code of KOKMI is described by Algorithm 2. The computational complexity[2] of KOKMI is in the order $O(N^3C^2)$ evaluated over the complexity of assignments which is $O(N^3C^2)$ and the complexity of the objective function which is $O(N^3C^2)$.

4.2 KOKMII: Kernel Overlapping *K*-means II

KOKMI proposed in the previous section computes distances, representatives and clusters prototypes in the feature space. This proposed method is characterized by a high computational complexity induced by the evaluation of its objective function which is in the order of $O(N^3C^2)$ making it not suitable to large data sets. In order to improve KOKMI in terms of efficiency, computational complexity and to make it well adapted to large data sets, we propose KOKMII where cluster centroids are replaced by clusters medoids.

2. All reported computational complexities are computed without evaluating the computational complexity of Mercer Kernel $K(x_i, x_j)$ between each pair of observations because we use an online implementation of the Kernel function. Moreover, the complexity of computing $K(x_i, x_j)$ depends of the type of kernel.

4.2.1 Optimized Cluster Prototypes Computation in Feature Space

For KOKMII, the definition of prototypes in the feature space is modified and clusters centroids are replaced by clusters medoids. Each cluster prototype is defined as the observation that minimizes the weighted sum of distances over all observations belonging to the responding cluster. Using Mercer Kernel, the prototype is defined as follows:

$$\begin{aligned} m_c = & \min_{i \in N_c}(x_i) \frac{\sum_{j=1, j\neq i}^{N_c} \frac{1}{w_j}\|\phi(x_i) - \phi(x_j)\|^2}{N_c \sum_{j=1, j\neq i}^{N_c} \frac{1}{w_j}} \\ = & \min_{i \in N_c}(x_i) \frac{\sum_{j=1, j\neq i}^{N_c} \frac{1}{w_j}[K_{ii} - 2K_{ij} + K_{jj}]}{N_c \sum_{j=1, j\neq i}^{N_c} \frac{1}{w_j}}, \end{aligned} \tag{15}$$

where N_c is the number of observations in cluster c. In this way, the prototype is determined in the feature space F and is a member of the initial set of observations.

4.2.2 Clustering Algorithm of KOKMII

Given the new way of determining prototypes in the feature space, the computational complexity of the objective function J is reduced to $O(NC^2)$. The objective function is given in Equation 16.

$$\begin{aligned} J(\Pi) &= \sum_{i=1}^{N} \|\phi(x_i) - \overline{\phi(x_i)}\|^2 \\ &= \sum_{i=1}^{N} \|\phi(x_i) - \frac{\sum_{c=1}^{C} P_{ic}\phi(m_c)}{L_i}\|^2 \\ &= \sum_{i=1}^{N} d'[\phi(x_i), \overline{\phi(x_i)}], \end{aligned} \tag{16}$$

where $d'[\phi(x_i), \overline{\phi(x_i)}]$ is defined by:

Algorithm 3 $KOKMII(X, t_{max}, \varepsilon, C) \rightarrow \{\pi_c\}_{c=1}^{C}$

```
INPUT  $X$: set of observations in $\mathbb{R}^d$.
    $t_{max}$: maximum number of iterations.
    $\varepsilon$: minimal improvement in the objective function.
    $C$: number of clusters.
OUTPUT  $\Pi$: set of C clusters.
 1: Initialize representatives of clusters with random cluster prototypes
 2: t= $0$.
 3: Compute cluster prototypes using Equation 15.
 4: Assign each observation $x_i$ to one or several clusters using "$ASSIGN$".
 5: Compute objective function $J_t(\Pi)$ using Equation 16.
 6: if ($t < t_{max}$ and $J_{t-1}(\Pi) - J_t(\Pi) > \varepsilon$) or (t= $0$) then
 7:     go to step 3.
 8: else
 9:     return the distribution of clusters' memberships.
10: end if
```

$$\begin{aligned}
&= \phi(x_i).\phi(x_i) - \frac{2}{L_i}\sum_{c=1}^{C} P_{ic}\phi(m_c)\phi(x_i) + \\
&\quad \frac{1}{(L_i)^2}\sum_{c=1}^{C}\sum_{l=1}^{C} P_{ic}P_{il}\phi(m_c)\phi(m_l) \\
&= K_{ii} - \frac{2}{L_i}\sum_{c=1}^{C} P_{ic}K_{im_c} + \frac{1}{(L_i)^2}\sum_{c=1}^{C}\sum_{l=1}^{C} P_{ic}P_{il}K_{m_c m_l}. \qquad (17)
\end{aligned}$$

Differently to KOKMI, KOKMII adds a new step for the determination of cluster prototypes independently from the evaluation of the objective function. At each iteration, cluster prototypes are computed, then observations are assigned to one or several clusters, and finally the objective function J is evaluated. These steps are repeated until improvement of J is no longer significant or the maximum number of iterations is reached. The computational complexity of KOKMII is in the order of $O(N_c^2C)$ where N_c is the maximum number of observations per cluster. This complexity is evaluated over the computational complexity of prototypes ($O(N_c^2C)$), the computational complexity of assignments (reduced to $O(NC^2)$) and the computational complexity of the objective function (reduced to $O(NC^2)$). KOKMII is given in Algorithm 3.

Assignments of each observation to one or several clusters in KOKMII are computed using the same function ASSIGN. The new definition of cluster prototypes has made computation of the distance between each observation and its representative easier as described in Equation 17. Also, finding the closest cluster from an observation x_i in the feature space is computationally easier and is given by:

$$c^\star = \min_{\{m_c\}_{c=1}^{C}} \|\phi(x_i) - \phi(m_c)\|^2 = \min_{\{m_c\}_{c=1}^{C}} K_{ii} - 2K_{im_c} + K_{m_c m_c}. \quad (18)$$

5. Experiments and Discussions

In this section, through an experimental study we evaluate the performance of the two proposed methods in detecting overlapping groups with complex and nonlinear separations between clusters.

5.1 Methodology

We begin by studying patterns induced by the proposed objective criterion through the report of Voronoï cells for the first example described in Figure 1. After that, we perform two experimental studies. First, experiments are conducted on data sets having non-linearly-separable clusters with complex shapes to check the effectiveness of proposed methods in detecting these clusters. Second, experiments are conducted on real multi-labeled data sets to check the effectiveness in identifying overlapping groups. Experiments are performed on a computer, with 4 GB RAM and 2.1 GHZ Intel Core 2 duo processor, and all proposed methods are implemented in C. For ALS method, we use ADPROCLUS software proposed by Wilderjans, Ceulemans, Van Mechelen, and Depril (2011). Since proposed methods require defining the number of clusters and the parameter of the kernel function in prior, we have considered in all experiments the following values:

- Number of clusters: for each data set, the number of clusters is set by the number of underlying true labels. However, in practice the number of clusters is usually unknown. In that case, one could use different model heuristics for determining the optimal number of clusters (Ceulemans and Kiers 2006; Depril, Van Mechelen, and Wilderjans 2012; Wilderjans, Depril, and Van Mechelen 2012). For example, the user can test different clusterings with increasingly number of clusters and then, selects the clustering which has the best balance between the minimization of the objective function and the number of clusters (Wilderjans, Ceulemans, Van Mechelen, and Depril 2011; Wilderjans, Ceulemans, and Meers 2012).

- parameter of the kernel function: although many techniques were proposed to select the best parameter of the kernel function, it is still a challenging issue. Since designing the best parameter of a kernel function is not the objective of this work, it is determined empirically over different executions of proposed methods. However, we report results using different kernels with different initialization of parameters to show the sensitivity of proposed methods to the selection of a kernel.

Obtained results are compared in terms of four validation measures: Precision, Recall, F-measure and Overlap size. The first three validation measures estimate whether the prediction of categories is correct with respect to the underlying true categories in the data. Precision is calculated as the fraction of observations correctly labeled as belonging to class c_i divided by the total number of observations labeled as belonging to class c_i. Recall is the fraction of observations correctly labeled as belonging to class c_i divided by the total number of observations that really belong to class c_i. The F-measure is the harmonic mean of Precision and Recall. They are defined as follows:

$$\text{Precision}(c_i) = NCLO/TNLO$$
$$\text{Recall}(c_i) = NCLO/TNAC$$
$$\text{F-measure}(c_i) = 2{*}\text{Precision}(c_i){*}\text{Recall}(c_i) \;/\; (\text{Precision}(c_i){+}\text{Recall}(c_i)),$$

where NCLO, TNLO and TNAC are respectively the number of correctly labeled observations, the total number of labeled observations and the total number of observations that really belong to the correct class. All these measures are computed separately on each cluster, then the average value of all clusters is reported. The fourth measure, Overlap size, evaluates the size of overlaps yielded by the learning method. This measure is defined by the average number of groups per observation in the data set:

$$Overlap = \frac{\sum_{x_i \in X} |c_i|}{|X|}, \tag{19}$$

where $|X|$ is the total number of observations and $|c_i|$ is the number of clusters to which x_i belongs to. The built overlap is compared with the true rate of overlaps in the labeled data set. The rate of overlaps is an important characteristic that affects the performance of different overlapping clustering methods.

5.2 Nonlinear Separations

To evaluate the performance of proposed methods in detecting complex and nonlinear separations between clusters, we visualize Voronoï cells (for 3 clusters) obtained with KOKMII using different types of kernels with different parameters. Figure 2 shows the ability of KOKMII to detect overlapping clusters with both linear and nonlinear boundaries depending of the type of the kernel function. For example, the Polynomial Kernel performs linear separations, the Gaussian performs nonlinear separations while Sigmoid performs a more complex and nonlinear separations. In fact, the choice of the kernel function and its parameters affect the structure of obtained patterns and affect the type of separations. For example, using Gaussian kernel, as the parameter σ increases overlapping regions between clusters are reduced. This result is also confirmed in real data sets where obtained overlap decreases as σ becomes larger.

Some kernels perform only linear separations between clusters, such as the Linear and Polynomial Kernels. However, other kernels can perform both linear and nonlinear separations depending on the value of their parameters. Example of these kernels is the Sigmoid which can build linear boundaries when $\theta = -10^{-9}$ and $c = 10$ as shown in Figure 2.

Moreover, Voronoï cells show that some kernels have a similar behavior and can detect the same patterns. For example, the Gaussian Kernel with $\sigma = 1000$ and the Sigmoid Kernel with $\theta = -10^{-9}$ and $c = 10$ build identical clusters shapes.

Compared to Voronoï cells obtained with OKM and ALS, those obtained with KOKMII using different kernels show the ability of the proposed methods to detect overlapping clusters with both linear and nonlinear separations. Next, these results are checked over non-linearly separable and real multi-labeled data sets.

5.3 Convolution Data Sets Having Non-Overlapping Classes With Nonlinear Separations

5.3.1. Data Sets Description

Experiments are performed on three convolution data sets which are Iris data set, Test data set and Ionosphere data set. Iris[3] data set is traditionally used as a basis test for evaluation. It is composed of 150 data in $\mathbb{R}^4$ tagged according to three non-overlapping clusters with 50 observations per class. One of these classes, "Setosa", is clearly separated while the two others "Versicolour" and "Virginica" are difficult to learn by linear separations.

3. cf. http://archive.ics.uci.edu/ml/datasets/Iris.

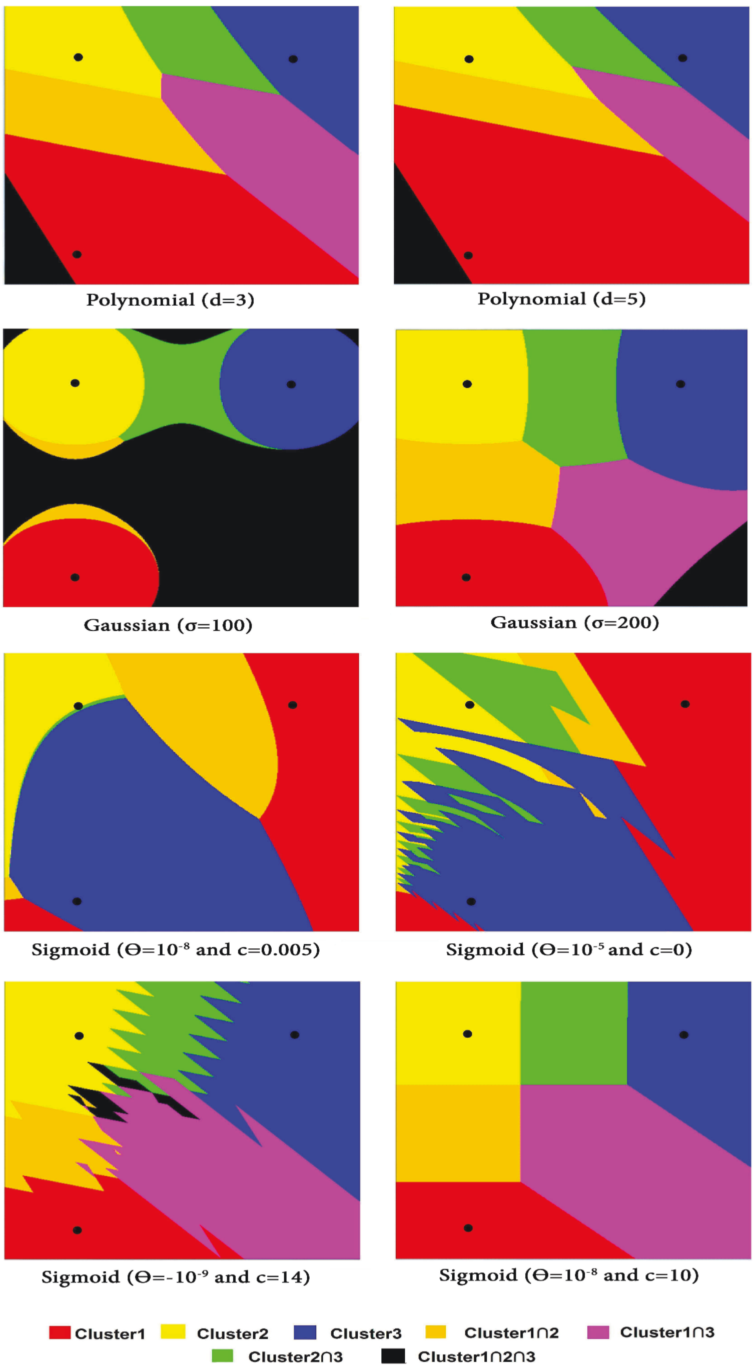

Figure 2. Voronoï cells obtained with KOKMII using Polynomial, Gaussian and Sigmoid Kernels

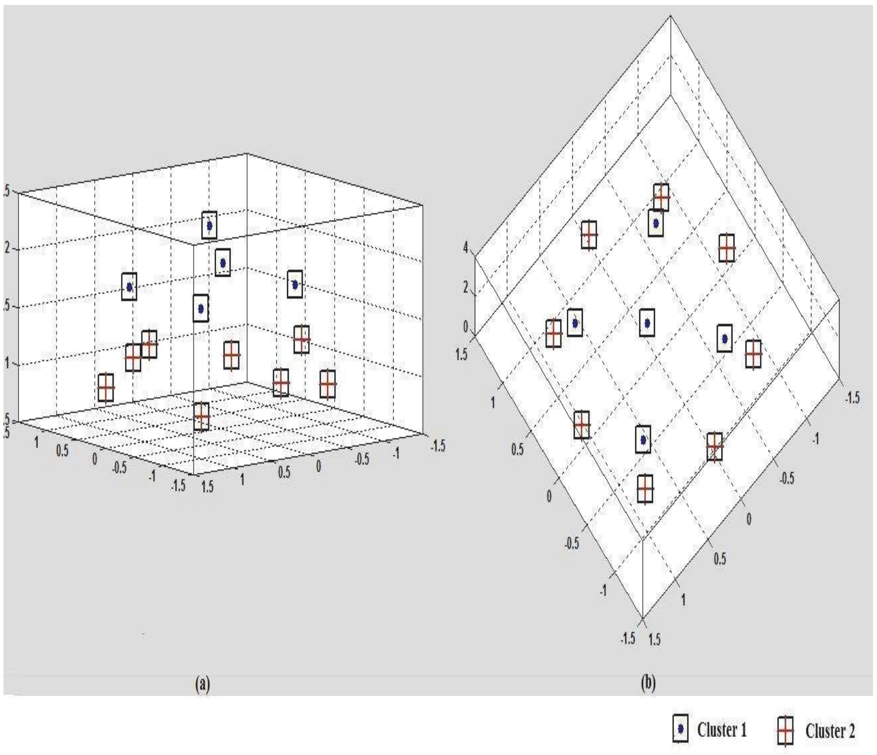

Figure 3. 3D plot of Test data set using first three principal axes obtained with PCA method in the mapped data using RBF kernel: (a) data shown from a first angle (b) data shown from a second angle

The second data set is Test data set which has a simple structure with two non-linearly-separable clusters. All samples are randomly chosen from unit square $[0, 1] \times [0, 1]$ in a two dimensional space with the uniform distribution. Points that fall in the kernel area with center $(0.5, 0.5)$ and radius 0.4 are assigned label $+1$. Those in the complement are assigned label -1. The particularity of this data set is that corresponding classes have two ring shapes where algorithms with linear separations fail to separate them. However, when these data are mapped to a higher feature space using RBF kernel, the clusters lose their ring shapes and become easier to separate as shown in Figure 3(a). If data are shown from another angle, the ring shapes are there as reported in Figure 3(b).

The third data set is Ionosphere[4], built by a radar system in Goose Bay Labrador. This system analyzes the electrons in the ionosphere, where some

4. cf. http://archive.ics.uci.edu/ml/datasets/Ionosphere.

electrons show a certain type of structure. These electrons determine the first class in the data set that labeled "good". Other electrons, with no structure in ionosphere, define the second class in the data set that labeled "bad". Electrons are transmitted from antennas via a signal, which is described by 34 attributes that will constitute the size of the data set Ionosphere. The total number of signals in the data set is 351 signals. The characteristic of this data set is that the two classes have a two ring shapes that are difficult to separate by linear clustering algorithms.

5.3.2 Results on Non-Linearly-Separable Data Sets

We compare the proposed methods with 5 existing methods: K-means, kernel K-means, fuzzy C-means, OKM and ALS. Table 2 reports Precision (P), Recall (R) and F-measure (F) of the best run for each method, which gives the minimal value of the objective criterion, among twenty runs while Table 3 reports the size of overlaps relative to the best run. A different initialization of prototypes have been used over the twenty runs, whereas within each run the same initialization of prototypes has been used for the different methods.

Compared to existing overlapping methods, and for all the data sets F-measures obtained with KOKMI and KOKMII outperform F-measures obtained with OKM and ALS using Euclidean distance. The improvement in classification results is achieved in terms of Precision and Recall. The improvement is important in Test data set where OKM and ALS fail to determine clusters with ring shapes. Figure 4 shows structures of patterns obtained by K-means, fuzzy C-means, OKM, kernel K-means, KOKMI and KOKMII in Test data set. Figures 4 (b,e,f) show that methods incorporating kernel can detect clusters with concentric shapes. On the other side, K-means and OKM completely fail to detect these clusters as shown in Figures 4 (a,d). These results show the ability of KOKMI and KOKMII to achieve nonlinear separations between clusters and then their performance in detecting clusters with complex shapes.

In fact, methods based on the kernel approach yield better results than traditional non kernel methods. For hard methods, Kernel K-means outperforms K-means and fuzzy C-means. For overlapping methods, KOKMI and KOKMII outperform OKM and ALS. These empirical results demonstrate that looking for separations between clusters in a high dimensional space is better than looking for separations in an Euclidean input space either for hard or for overlapping methods.

Table 2 shows that hard methods, especially kernel K-means, give better classification results than overlapping methods. This usefulness of hard methods is explained by the fact that all the data sets are non overlapping

Table 2. Comparison of the performance of KOKMI and KOKMII versus other existing methods for data sets with non-linearly-separable categories

Dataset Label	**Iris data set**			**Test data set**			**Ionosphere data set**		
	P	R	F	P	R	F	P	R	F
K-means	0.891	0.885	**0.890**	0.540	0.580	0.559	0.702	0.710	0.705
Kernel K-means	0.925	0.926	**0.925**	1	1	**1**	0.887	0.885	**0.886**
Fuzzy C-means	0.891	0.911	**0.898**	0.499	0.550	0.523	0.701	0.722	0.713
OKM	0.701	0.905	0.814	0.307	0.500	0.381	0.531	0.691	0.598
ALS	0.672	0.943	0.789	0.562	0.692	0.620	0.545	0.900	0.681
KOKMI with RBF kernel	0.871	0.928	**0.896**	0.940	1	**0.969**	0.662	0.745	0.701
KOKMII with RBF kernel	0.831	0.975	**0.898**	0.940	1	**0.969**	0.610	0.877	0.720

Table 3. Size of overlaps obtained with KOKMI, KOKMII and other methods for data sets with non-linearly-separable categories

	Size of Overlap		
	Iris data set	Test data set	Ionosphere data set
Real overlap size	(1)	(1)	(1)
K-means	1	1	1
Kernel K-means	1	1	1
Fuzzy C-means	1	1	1
ALS	1.37	1.23	1.45
OKM	1.34	1	1.39
KOKMI with RBF kernel	1.15	1.07	1.28
KOKMII with RBF kernel	1.22	1.07	1.25

(overlap size$=1$). Classification results obtained with overlapping methods are characterized by a low Precision because observations are assigned to more than one cluster. Although all data sets do not contain overlapping observations, all overlapping methods build an overlap size greater than 1 as reported in Table 3. For example in Iris data set, observations which are assigned to multiple clusters come from the two last classes, "Versicolour" and "Virginica", which are known to be strongly overlapping. In fact, the size of overlaps affects the value of the F-measure. As the obtained size of overlaps increases, the value of Precision decreases inducing a decrease of F-measure.

In the next section, we study the effectiveness of these methods over real data sets having different degrees of natural overlaps.

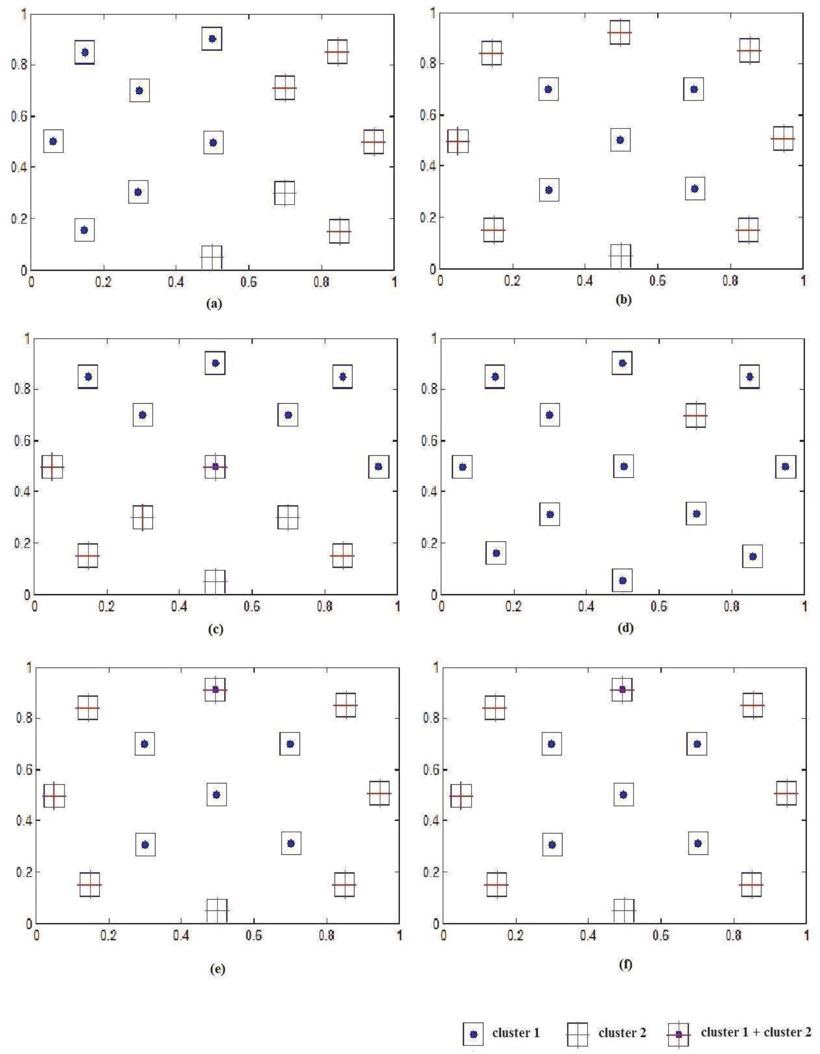

Figure 4. Clusters obtained with different methods on Test data set : (a) Clusters obtained with *K*-means (Euclidean distance), (b) Clusters obtained with kernel *K*-means (RBF kernel), (c) Clusters obtained with fuzzy *C*-means (threshold=0.4), (d) Clusters obtained with OKM (Euclidean distance), (e) Clusters obtained with KOKMI (RBF kernel), (f) Clusters obtained with KOKMII (RBF kernel)

5.4 Real Multi-Labeled Data Sets

We conducted experiments on real multi-labeled data sets from three domains that strongly motivate overlapping clustering researches: video classification, detection of emotions in music and image classification.

5.4.1 Multi-Labeled Data Sets Description

The first data set is EachMovie[5] containing user ratings for each movie in the collection. Users give ratings on a scale of 1 to 5, with 1 indicating extreme dislike and 5 indicating strong approval. For each movie, the corresponding genre information is extracted from the Internet Movie Database (IMDB) collection. If each genre is considered as a separate category or cluster, then this data set has naturally overlapping clusters since many movies are annotated in IMDB as belonging to multiple genres (Banerjee et al. 2005). For example, Aliens movie belongs to three genres: action, horror and science fiction.

From the EachMovie data set, we extracted a subset of 75 movies scattered over three overlapping clusters as follows: "action" = 21 movies; "comedy" = 26 movies; "crime"= 17 movies and "action+crime"= 11 movies. A description of the subset can be found in Appendix A. Based on age, sex and rate of users we try to find categories for each video. Figure 5 shows the initial distribution of these movies where overlapping movies belong to both action and crime genres. When mapping the same movies into a higher feature space, using a Gaussian RBF kernel with $\sigma = 1000$, overlapping movies are easily detected while they are geometrically laying in the extremity surface between action and crime movies. In addition, separations between "crime" and "comedy" movies become easier to detect compared to their first distribution.

The second multi-labeled data set is the Music[6] data set. Emotion detection in music can be realized by analyzing music signals. The emotion labels are not usually disjoint in the sense that a single music sound may be classified simultaneously into multiple emotional categories e.g. both "happy" and "relaxing" (Trohidis, Tsoumakas, Kalliris, and Vlahavas 2008). This stipulation seems to make the Music data set naturally overlapping. The Music data set contains sound clips described by 72 real attributes and annotated by three male experts. This process led to a final annotated data set of 548 songs with 6 main emotional clusters as described in Table 4. The overlap between clusters is important in Music data set. For example, all

5. cf. http://www.grouplens.org/node/76.
6. cf. http://mlkd.csd.auth.gr/multilabel.html

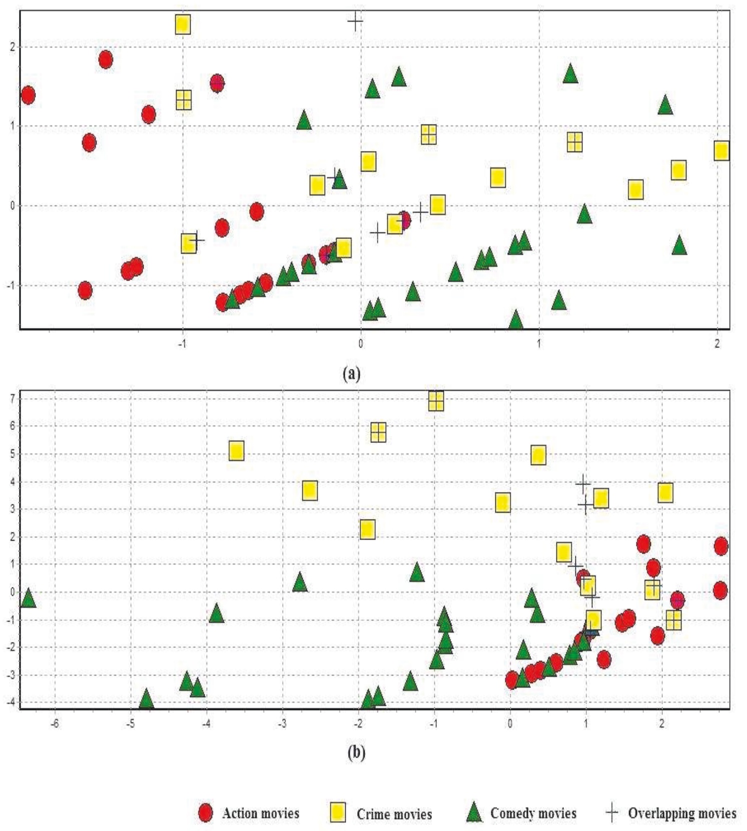

Figure 5. 2D plot of EachMovie data set using the first and second principal axes obtained with PCA: (a) data in input space (b) data in feature space

music sounds that evoke "Quite Still" emotion also evoke "Relaxing Calm" or/and "Sad Lonely" emotions.

The third multi-labeled data set is the Scene[7] data set that contains 2407 natural scene images. Each image is transformed into a $49 \times 3 \times 2 = 294$ dimensional features vectors. Table 5 gives the detailed description of the number of images associated with different label sets, where all the possible class labels are Beach, Sunset, Fall foliage, Field, Mountain and Urban. Over 8% of images belong simultaneously to multiple classes such as images belonging to beach and mountain as illustrated in Figure 6. On average, each image is associated with 1.08 class labels.

7. cf. http://mulan.sourceforge.net/datasets.html

Table 4. Distribution of songs by six principal class labels in Music data set: Amazed Surprised, Happy Pleased, Relaxing Calm, Quite Still, Sad Lonely and Angry Aggressive

Label	**number of songs**
Amazed Surprised	24
Happy Pleased	23
Relaxing Calm	42
Quite Still	0
Sad Lonely	12
Angry Aggressive	72
Happy Pleased+Relaxing Calm	74
Happy Pleased+Amazed Surprised	38
Amazed Surprised+Angry Aggressive	81
Angry Aggressive+Sad Lonely	12
Sad Lonely+Quite Still	37
Quite Still+Relaxing Calm	30
Sad Lonely+Relaxing Calm	25
Happy Pleased+Amazed Surprised+Relaxing Calm	11
Sad Lonely+Quite Still+Relaxing Calm	67
Total	**548**

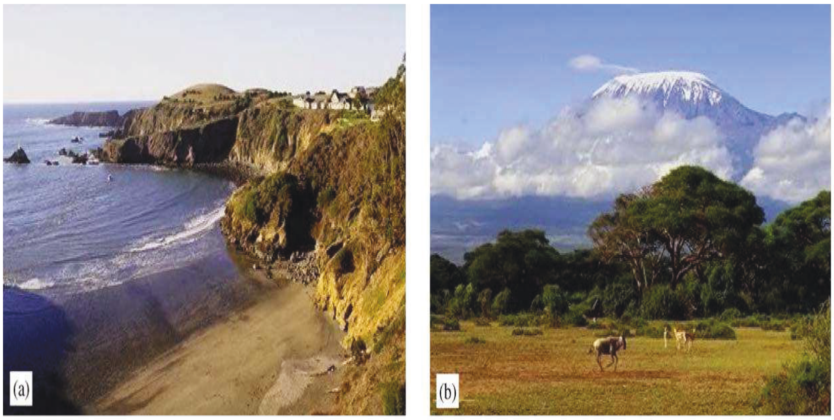

Figure 6. Example of overlapping images in Scene data set: **(a)** image associated to Beach+Mountain **(b)** image associated to Field+Mountain

Table 5. Distribution of images by six principal class labels in Scene data set: Beach, Sunset, Fall foliage, Field, Mountain and Urban

Label	**number of images**
Beach	369
Sunset	364
Fall foliage	360
Field	327
Beach+Field	1
Fall foliage+Field	23
Mountain	405
Beach+Mountain	38
Fall foliage+Mountain	13
Field+Mountain	75
Field+Fall foliage+Mountain	1
Urban	405
Beach+Urban	19
Field+Urban	6
Mountain+Urban	1
Total	**2407**

5.4.2 Results on Multi-Labeled Data Sets

For data sets described in Section 5.4.1, Table 6 presents results obtained with KOKMI and KOKMII versus K-means, kernel K-means, OKM and ALS methods in terms of Precision, Recall and F-measure. Each reported measure results from the evaluation of the best run over twenty runs of each algorithm. Results of KOKMI for Music and Scene data sets are not reported due to time execution exceeding 24 hours however, it is easily solved with KOKMII. The improvement with KOKMII is shown in terms of computational complexity and classification results.

Table 6 shows that, for all data sets, F-measures of the proposed methods are better than those of existing ones. Compared to overlapping methods, the improvement of F-measure is induced by the improvement in Precision. Also, we note that there is no significant difference between overlapping methods in terms of Recall since all of them assign observations to many clusters. But, in terms of Precision KOKMII has the best values with a large margin compared to other overlapping methods. These results show that our proposed methods can detect more relevant and complex separations between clusters leading to the improvement of the classification precision.

In addition, reported results show that the size of actual overlaps in each data set affects the performance of overlapping methods. As the size of overlaps increases the performance of overlapping methods compared to hard methods becomes more noticeable. For example, in Music data set,

Table 6. Comparison of the performance of KOKMI and KOKMII versus other existing methods for multi-labeled data sets

Data set Label	**Eachmovie**			**Music**			**Scene**		
	P	R	F	P	R	F	P	R	F
K-means	0.735	0.541	0.624	0.500	0.203	0.287	0.501	0.516	0.509
Kernel K-means	0.771	0.603	0.679	0.576	0.214	0.311	0.532	0.564	**0.546**
OKM	0.590	0.829	0.691	0.395	0.336	0.364	0.330	0.889	0.491
ALS	0.510	0.770	0.618	0.291	0.558	0.387	-	-	-
KOKMI with RBF kernel	0.715	0.820	**0.761**	-	-	-	-	-	-
KOKMII with RBF kernel	0.785	0.819	**0.800**	0.475	0.365	**0.415**	0.422	0.755	**0.549**

where actual overlaps is 1.81, all F-measures obtained with OKM, ALS and KOKMII outperform those of K-means and kernel K-means. However, when the size of overlaps nearly reaches 1, such as in Scene data set, F-measures obtained with hard methods outperform, or at least equal, those of overlapping methods. For example, F-measure obtained with kernel K-means in Scene data set is 0.546 which outperforms F-measures obtained with OKM and ALS. In fact, high values of F-measure obtained with hard methods are induced by high values of Precision, while for overlapping methods, high values of F-measure are induced by high values of Recall.

Therefore, knowing the actual overlaps in each data set, the sizes of overlaps built by each method are discussed. Table 7 summarizes overlaps obtained with KOKMI and KOKMII compared to overlaps obtained with existing overlapping and non overlapping methods. For hard methods, all sizes of overlaps are equal to 1 since these methods build non disjoint clusters and ignore the possibility that an observation belongs to more than one cluster. Fuzzy C-means builds acceptable overlaps if the threshold is well determined, elsewhere we can obtain an overlap size less than 1. For overlapping methods, we notice that OKM and ALS build large overlaps. For example, in Music data set, the size of overlaps obtained with OKM and ALS are 2.35 and 3.46 respectively, while the actual overlaps is 1.81. However, for the same data set, KOKMII builds an acceptable overlap size of 1.98. These results show the capacity of the proposed methods to build acceptable sizes of overlaps given the actual overlaps in each data set.

As described in Section 5.1, the structure of separations between overlapping clusters depends of the choice of the kernel function and its parameters. Therefore, we study the sensitivity of proposed methods to these parameters by analyzing the performance of KOKMII in real multi-labeled data sets using Gaussian, Exponential and Polynomial kernels with different

Table 7. Size of overlaps obtained with KOKMI, KOKMII and other methods for multi-labeled data sets

	Size of Overlap		
	Eachmovie data set	Music data set	Scene data set
Real overlap size	(1.14)	(1.81)	(1.08)
K-means	1	1	**1**
Kernel K-means	1	1	**1**
Hard Fuzzy C-means	1	1	**1**
OKM	1.40	2.35	2.85
ALS	1.73	3.46	-
Fuzzy C-means (threshold=0.3)	**1.26**	1.22	0.00
Fuzzy C-means (threshold=0.4)	0.93	0.97	0.00
KOKMI with RBF kernel	**1.26**	-	-
KOKMII with RBF kernel	1.36	**1.98**	**1.99**

values of kernel parameters. Tables 8, 9 and 10 report average results and standard deviations, on ten runs, obtained with KOKMII using different kernels in Eachmovie, Music and Scene data sets. Reported results show many interesting facts about our proposed methods:

- For all data sets, kernels performing nonlinear separations, such as Exponential and Gaussian, give better results than those performing linear separations (Polynomial). These results state that in real life applications, where separations between clusters may be complex, it would be better to perform a learning process with nonlinear separations.
- KOKMII is highly sensitive to the choice of the kernel function. For example, in Scene data set, the best obtained F-measure with Exponential kernel is 0.548; whereas this value does not exceed 0.513 using Gaussian kernel.
- Choice of parameters of the kernel function can considerably affect the performance of classification. Some kernel parameters have identical behaviors in all data sets. For example, we notice for Gaussian and Exponential kernels that the size of overlaps increases, Precision decreases and Recall increases as well as σ becomes large.
- Results of standards deviations on ten runs with different initializations of clusters representatives show that KOKMII is not very sensitive to the these initializations.

Table 8. Validation measures obtained with KOKMII using different types of kernels in Eachmovie Data set

Kernel	Value	Precision	Recall	F-measure	Overlap
Gaussian	$\sigma = 1$	0.610 ± 0.06	0.840 ± 0.15	0.719 ± 0.11	1.58 ± 0.10
	$\sigma = 15$	0.715 ± 0.05	0.816 ± 0.15	0.754 ± 0.08	1.36 ± 0.05
RBF kernel	$\sigma = 100$	0.715 ± 0.05	0.816 ± 0.15	0.754 ± 0.08	1.36 ± 0.05
	$\sigma = 10000$	0.715 ± 0.05	0.816 ± 0.15	0.754 ± 0.08	1.36 ± 0.05
Exponential	$\sigma = 10$	0.554 ± 0.03	0.771 ± 0.11	0.674 ± 0.09	1.74 ± 0.12
	$\sigma = 15$	0.629 ± 0.06	0.817 ± 0.21	0.711 ± 0.15	1.52 ± 0.15
RBF kernel	$\sigma = 100$	0.629 ± 0.06	0.817 ± 0.21	0.711 ± 0.15	1.52 ± 0.15
	$\sigma = 10000$	0.629 ± 0.06	0.817 ± 0.21	0.711 ± 0.15	1.52 ± 0.15
	$d = 2$	0.665 ± 0.09	0.739 ± 0.11	0.700 ± 0.10	1.33 ± 0.08
Polynomial kernel	$d = 3$	0.665 ± 0.08	0.774 ± 0.14	0.715 ± 0.10	1.42 ± 0.02
	$d = 4$	0.674 ± 0.02	0.710 ± 0.13	0.689 ± 0.07	1.32 ± 0.11

Table 9. Validation measures obtained with KOKMII using different types of kernels in Music Data set

Kernel	Value	Precision	Recall	F-measure	Overlap
Gaussian	$\sigma = 1$	0.539 ± 0.03	0.367 ± 0.00	0.436 ± 0.00	1.98 ± 0.00
	$\sigma = 5$	0.471 ± 0.00	0.335 ± 0.02	0.392 ± 0.02	1.81 ± 0.03
RBF kernel	$\sigma = 100$	0.480 ± 0.02	0.263 ± 0.02	0.354 ± 0.03	1.42 ± 0.01
	$\sigma = 10000$	0.477 ± 0.08	0.282 ± 0.02	0.354 ± 0.02	1.44 ± 0.02
Exponential	$\sigma = 1$	0.479 ± 0.02	0.362 ± 0.02	0.412 ± 0.02	1.98 ± 0.00
	$\sigma = 5$	0.463 ± 0.00	0.313 ± 0.02	0.373 ± 0.02	1.64 ± 0.01
RBF kernel	$\sigma = 100$	0.466 ± 0.01	0.300 ± 0.02	0.365 ± 0.02	1.57 ± 0.03
	$\sigma = 10000$	0.466 ± 0.01	0.300 ± 0.02	0.365 ± 0.02	1.57 ± 0.03
	$d = 2$	0.485 ± 0.01	0.327 ± 0.02	0.365 ± 0.01	1.44 ± 0.01
Plynomial kernel	$d = 3$	0.493 ± 0.00	0.290 ± 0.01	0.365 ± 0.01	1.40 ± 0.02
	$d = 4$	0.492 ± 0.00	0.295 ± 0.01	0.369 ± 0.01	1.41 ± 0.01

6. Conclusion

In order to better look for overlapping clusters with linear and non-linear separations two clustering methods, based on the Mercer Kernel, are proposed. The first method, based on centroids, generalizes kernel K-means for overlapping clustering taking into account that an observation can belong to more than one cluster. The second method, based on medoids, improves the clustering accuracy of the previous method and adapts it to large data sets. Experiments on artificial and real multi-labeled data sets show the ability and the efficiency of these methods to detect clusters with complex linear and nonlinear separations.

Table 10. Validation measures obtained with KOKMII using different types of kernels in Scene Data set

Kernel	Value	Precision	Recall	F-measure	Overlap
Gaussian RBF kernel	$\sigma = 10$	0.419 ± 0.04	0.663 ± 0.05	0.513 ± 0.04	1.75 ± 0.04
	$\sigma = 15$	0.418 ± 0.04	0.655 ± 0.05	0.510 ± 0.05	1.75 ± 0.04
	$\sigma = 100$	0.423 ± 0.05	0.657 ± 0.05	0.511 ± 0.05	1.75 ± 0.04
	$\sigma = 10000$	0.423 ± 0.05	0.657 ± 0.05	0.511 ± 0.05	1.75 ± 0.04
Exponential Gaussian	$\sigma = 10$	0.425 ± 0.01	0.750 ± 0.04	0.548 ± 0.02	1.99 ± 0.00
	$\sigma = 15$	0.426 ± 0.01	0.753 ± 0.05	0.544 ± 0.02	1.99 ± 0.00
	$\sigma = 100$	0.425 ± 0.01	0.753 ± 0.04	0.544 ± 0.02	1.99 ± 0.00
	$\sigma = 10000$	0.426 ± 0.01	0.753 ± 0.05	0.544 ± 0.02	1.99 ± 0.00
Polynomial kernel	$d = 2$	0.435 ± 0.04	0.661 ± 0.06	0.525 ± 0.05	1.75 ± 0.05
	$d = 3$	0.455 ± 0.05	0.656 ± 0.05	0.537 ± 0.05	1.74 ± 0.05
	$d = 4$	0.481 ± 0.03	0.593 ± 0.02	0.530 ± 0.02	1.63 ± 0.14

To easily interpret results after classification we consider a hard assignment of an observation to one or several groups. Each observation would be a member or not of one or several groups. However, we can add a probabilistic or fuzzy assignments of the observation to model its membership to each cluster.

In addition, there are many real cases where the input data cannot be described by explicit feature vectors but described by strings, trees or histograms such as for images, graphs and text documents. These types of applications are referred to as symbolic clustering. For such data, it should be interesting to investigate the application of an overlapping clustering process with specific kernels such as String (Lodhi, Cristianini, Shawe-Taylor, and Watkins 2001) and Histogram (Barla, Odone, and Verri 2003) kernels.

Appendix A. Descriptions of the Subset Extracted From EachMovie

Table 11 shows the descriptions and the labels of the subset of movies extracted from Eachmovie and used in Section 5.4.2. The "age of user" is normalized in the interval $[1..5]$, the "sex of user" takes (1) for male and takes (2) for female, the "movie's rate" is a scale from (1) to (5) with (1) indicates extreme dislike and the remaining columns indicate the categories of each movie in the collection.

Table 11. Dataset extracted from EachMovie database.

age of user	sex of user	movie's rate	action movie	comedy movie	crime movie
2.77	2	4	1	0	0
3.22	1	4	1	0	0
1.44	2	4	1	0	0
1.53	1	4	1	0	0
2.42	1	4	1	0	0
2.86	1	5	1	0	0
2.24	1	4	1	0	0
1.44	1	5	1	0	0
3.22	1	5	1	0	0
3.13	2	5	1	0	0
1.88	1	5	1	0	0
2.51	1	5	1	0	0
1.97	1	5	1	0	0
2.06	2	4	1	0	0
1.80	1	4	1	0	0
1.62	1	4	1	0	0
2.51	1	4	1	0	0
2.60	1	4	1	0	0
1.35	1	4	1	0	0
3.57	1	4	1	0	0
2.33	2	5	1	0	0
1.80	1	3	0	1	0
2.86	2	1	0	1	0
2.24	1	4	0	1	0
2.15	2	3	0	1	0
1.80	1	2	0	1	0
2.24	1	3	0	1	0
1.97	1	4	0	1	0
2.51	1	3	0	1	0
1.35	1	3	0	1	0
2.60	1	3	0	1	0
1.71	1	4	0	1	0
3.40	2	2	0	1	0
2.95	1	3	0	1	0
3.04	1	2	0	1	0
1.35	1	3	0	1	0
2.86	2	3	0	1	0
3.57	1	3	0	1	0
1.35	1	2	0	1	0
3.13	2	3	0	1	0
2.51	1	4	0	1	0
1.44	1	4	0	1	0
2.24	1	3	0	1	0
1.00	2	2	0	1	0
1.44	1	3	0	1	0

(continued on next page)

Table 11. Dataset extracted from EachMovie database (continued)

age of user	sex of user	movie's rate	action movie	comedy movie	crime movie
2.06	1	4	0	1	0
2.86	1	3	0	1	0
4.55	1	3	0	0	1
4.11	1	3	0	0	1
4.37	1	5	0	0	1
3.13	1	4	0	0	1
3.93	2	5	0	0	1
4.20	1	4	0	0	1
3.84	1	5	0	0	1
2.60	1	4	0	0	1
2.51	1	5	0	0	1
4.37	1	5	0	0	1
3.84	1	5	0	0	1
2.42	2	4	0	0	1
5.00	1	3	0	0	1
5.00	1	3	0	0	1
5.00	1	4	0	0	1
5.00	1	5	0	0	1
3.57	1	4	0	0	1
2.77	2	4	1	0	1
3.22	1	4	1	0	1
2.95	1	4	1	0	1
5.00	1	4	1	0	1
5.00	1	5	1	0	1
4.20	2	4	1	0	1
2.42	2	4	1	0	1
2.60	1	5	1	0	1
4.02	1	5	1	0	1
3.40	1	4	1	0	1
2.42	1	4	1	0	1

References

ALIGULIYEV, R.M. (2009), "Clustering of Document Collection - A Weighting Approach," *Expert Systems with Applications, 36*, 7904–7916.

BANERJEE, A., KRUMPELMAN, C., BASU, S., MOONEY, R.J., and GHOSH, J. (2005), "Model Based Overlapping Clustering," in *International Conference on Knowledge Discovery and Data Mining*, Chicago, USA, pp. 532–537.

BARLA, A., ODONE, F., and VERRI, A. (2003), "Histogram Intersection Kernel for Image Classification," in *2003 International Conference on Image Processing (ICIP)*, pp. 513–516.

BEN-HUR, A., HORN, D., SIEGELMANN, H.T., and VAPNIK, V. (2001), "Support Vector Clustering," *Journal Of Machine Learning Research, 2*, 125–137.

CAMASTRA, F., and VERRI, A. (2005), "A Novel Kernel Method for Clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence, 27*, 801–804.

CEULEMANS, E., and KIERS, H.A. (2006), “Selecting Among Three-Mode Principal Component Models of Different Types and Complexities: A Numerical Convex Hull Based Method,” *British Journal of Mathematical and Statistical Psychology, 59*, 133–150.

CHAO-LIU, Y., WU, C., and LIU, M. (2011), “Research of Fast SOM Clustering for Text Information,” *Expert Systems with Applications, 38*, 9325–9333.

CLEUZIOU, G. (2008), “An Extended Version of the K-means Method for Overlapping Clustering,” in *International Conference on Pattern Recognition (ICPR)*, Florida, USA: IEEE, pp. 1–4.

CLEUZIOU, G. (2010), “Two Variants of the OKM for Overlapping Clustering,” in *Advances in Knowledge Discovery and Management*, eds. F. Guillet et al., Heidelberg: Springer, pp.149–166.

CLEUZIOU, G. (2013), “Osom: A Method for Building Overlapping Topological Maps,” *Pattern Recognition Letters, 34*, 239–246.

CORTES, C., and VAPNIK, V. (1995), “Support Vector Networks,” *Machine Learning, 20*, 273–297.

CRISTIANINI, N., CAMPBELL, C., and BURGES, C. (2002), “Editorial: Kernel Methods: Current Research and Future Directions,” *Machine Learning, 46*, 5–9.

DEODHAR, M., and GHOSH, J. (2006), “Consensus Clustering for Detection of Overlapping Clusters in Microarray Data,” in *International Conference on Data Mining*, Los Alamitos, CA, USA: IEEE Computer Society, pp. 104–108.

DEPRIL, D., VAN MECHELEN, I., and WILDERJANS, T.F. (2012), “Lowdimensional Additive Overlapping Clustering,” *Journal of Classification, 29*, 297–320.

DEPRIL, D., VAN MECHELEN, I., and MIRKIN, B. (2008), “Algorithms for Additive Clustering of Rectangular Data Tables,” *Computational Statistics and Data Analysis, 52*, 4923–4938.

DESARBO, W., and CRON, W. (1988), “A Maximum Likelihood Methodology for Clusterwise Linear Regression,” *Journal of Classification, 5*, 249–282.

FELLOWS, M.R., GUO, J., KOMUSIEWICZ, C., NIEDERMEIER, R., and UHLMANN, J. (2011), “Graph-based Data Clustering with Overlaps,” *Discrete Optimization, 8*, 2–17.

FILIPPONE, M., CAMASTRA, F., MASULLI, F., and ROVETTA, S. (2008), “A Survey of Kernel and Spectral Methods for Clustering,” *Pattern Recognition, 41*, 176 – 190.

GIROLAMI, M. (2002), “Mercer Kernel-Based Clustering in Feature Space,” *IEEE Transactions on Neural Networks, 13*, 780–784.

GRAEPEL, T., and OBERMAYER, K. (1998), “Fuzzy Topographic Kernel Clustering,” in *Proceedings of the Fifth GI Workshop Fuzzy Neuro Systems*, pp. 90–97.

INOKUCHI, R., and MIYAMOTO, S. (2004), “LVQ Clustering and SOM Using a Kernel Function,” in *Proceedings of IEEE International Conference on Fuzzy Systems*, pp. 1497–1500.

LINGRAS, P., and WEST, C. (2004), “Interval Set Clustering of Web Users with Rough K-Means,” *Journal of Intelligent Information Systems, 23*, 5–16.

LODHI, H., CRISTIANINI, N., SHAWE-TAYLOR, J., and WATKINS, C. (2001), “Text Classication Using String Kernel,” *Journal of Machine Learning Research, 2*, 419–444.

LU, H., HONG, Y., STREET, W., WANG, F., and TONG, H. (2012), "Overlapping Clustering with Sparseness Constraints," in *IEEE 12th International Conference on Data Mining Workshops (ICDMW)*, pp. 486–494.

MASSON, M.-H. and DENUX, T. (2008), "ECM: An Evidential Version of the fuzzy c-means Algorithm," *Pattern Recognition, 41*, 1384 – 1397.

MIRKIN, B.G. (1987a), "Additive Clustering and Qualitative Factor Analysis Methods for Similarity Matrices," *Journal of Classification, 4*, 7–31.

MIRKIN, B.G. (1987b), "Method of Principal Cluster Analysis," *Automation and Remote Control, 48*, 1379–1386.

MIRKIN, B.G. (1990), "A Sequential Fitting Procedure for Linear Data Analysis Models," *Journal of Classification, 7*, 167–195.

PÉREZ-SUÁREZ, A., MARTÍNEZ-TRINIDAD, J.F., CARRASCO-OCHOA, J.A., and MEDINA-PAGOLA, J.E. (2013), "OClustR: A New Graph-Based Algorithm for Overlapping Clustering," *Neurocomputing, 109*, 1–14.

QINAND, A.K., and SUGANTHAN, P.N. (2004), "Kernel Neural Gas Algorithms with Application to Cluster Analysis," *International Conference on Pattern Recognition, 4*, 617–620.

SCHÖLKOPF, B., SMOLA, A., and MÜLLER, K.-R. (1998), "Nonlinear Component Analysis as a Kernel Eigenvalue Problem," *Neural Computation, 10*, 1299–1319.

SNOEK, C. G.M., WORRING, M., VAN GEMERT, J.C., GEUSEBROEK, J.-M., and SMEULDERS, A.W.M. (2006), "The Challenge Problem for Automated Detection of 101 Semantic Concepts in Multimedia," in *Proceedings of the 14th annual ACM international conference on Multimedia*, New York, USA: ACM, MULTIMEDIA '06, pp. 421–430.

TANG, L.,and LIU, H. (2009), "Scalable Learning of Collective Behavior Based on Sparse Social Dimensions," in *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, pp. 1107–1116.

TROHIDIS, K., TSOUMAKAS, G., KALLIRIS, G., and VLAHAVAS, I.P. (2008), "Multi-Label Classification of Music into Emotions," in *International Conference on Music Information Retrieval (ISMIR)*, pp. 325–330.

VAN HATTUM, P., and HOIJTINK, H. (2009), "Market Segmentation Using Brand Strategy Research: Bayesian Inference with Respect to Mixtures of Log-Linear Models," *Journal of Classification, 26*, 297–328.

WANG, Q., and FLEURY, E. (2011), "Uncovering Overlapping Community Structure," in *Complex Networks*, Vol. 116 of *Communications in Computer and Information Science*, pp. 176–186.

WANG, X., TANG, L., GAO, H., and LIU, H. (2010), "Discovering Overlapping Groups in Social Media," in *Proceedings of the 2010 IEEE International Conference on Data Mining*, pp. 569–578.

WIECZORKOWSKA, A., SYNAK, P., and RAS, Z. (2006), "Multi-Label Classification of Emotions in Music," in *Intelligent Information Processing and Web Mining*, Vol. 35 of *Advances in Soft Computing*, pp. 307–315.

WILDERJANS, T., CEULEMANS, E., VAN MECHELEN, I., and DEPRIL, D. (2011), "ADPROCLUS: A Graphical User Interface For Fitting Additive Profile Clustering Models to Object by Variable Data Matrices," *Behavior Research Methods, 43*, 56–65.

WILDERJANS, T.F., DEPRIL, D., and VAN MECHELEN, I. (2012), “Additive Biclustering: A Comparison of One New and Two Existing ALS Algorithms,” *Journal of Classification, 30*, 56–74.

WILDERJANS, T.F., CEULEMANS, E., and MEERS, K. (2013), “CHull: A Generic Convex Hull Based Model Selection Method,” *Behavior Research Methods, 45*, 1–15.

WU, Z., XIE, W., and YU, J. (2003), “Fuzzy C-Means Clustering Algorithm Based on Kernel Method,” in *Proceedings of the 5th International Conference on Computational Intelligence and Multimedia Applications (ICCIMA)*, Washington, DC, USA: IEEE Computer Society.

ZHANG, D., and CHEN, S. (2002), “Fuzzy Clustering Using Kernel Method,” in *International Conference on Control and Automation*, Xiamen, China, pp. 123–127.

ZHANG, D., and CHEN, S. (2003), “Kernel-based Fuzzy and Possibilistic C-means Clustering,” in *International Conference on Artificial Neural Networks (ICANN)*, Istanbul, Turkey, pp. 122–125.

ZHANG, D., and CHEN, S. (2004), “A Novel Kernelized Fuzzy C-means Algorithm with Application in Medical Image Segmentation,” *Artificial Intelligence in Medicine, 32*, 37–50.

ZHANG, S., WANG, R.-S., and ZHANG, X.-S. (2007), “Identification of Overlapping Community Structure in Complex Networks Using Fuzzy C-means Clustering,” *Physica A: Statistical Mechanics and its Applications, 374*, 483–490.