KERNEL OVERLAPPING K-MEANS FOR CLUSTERING IN FEATURE SPACE

Chiheb-Eddine Ben N'Cir, Nadia Essoussi
LARODEC, University of Tunis, Tunis, Tunisia

Patrice Bertrand
Ceremade, Université Paris-Dauphine, Paris, France

Keywords: Overlapping clustering, Overlapping K-Means, Kernel Methods, Gram matrix, Kernel induced distance measure.

Abstract: Producing overlapping schemes is a major issue in clustering. Recent overlapping methods rely on the search of optimal clusters and are based on different metrics, such as Euclidean distance and I-Divergence, used to measure closeness between observations. In this paper, we propose the use of kernel methods to look for separation between clusters in a high feature space. For detecting non linearly separable clusters, we propose a Kernel Overlapping K-Means algorithm (KOKM) that is based on a kernel induced distance measure. The number of overlapping clusters is estimated using the Gram matrix. Experiments on different datasets show the correctness of the estimation of number of clusters and show that KOKM gives better results when compared to overlapping k-means.

1 INTRODUCTION

Overlapping between clusters is a major issue in clustering. In this cluster configuration, an object can belong to one or many clusters without any membership coefficient. Based on the assumption that an object really belongs to many clusters, overlapping clustering is different from both crisp and fuzzy clustering. A number of real problems require to find overlapping clusters in order to fit the data set structure. For example, in social network analysis, community extraction algorithms should be able to detect overlapping clusters because an actor can belong to multiple communities (Banerjee *et al*, 2005); in video classification, overlapping clustering is a necessary requirement while video can potentially have multiple genres.

Several overlapping clustering models based on stratified and partitioning approaches were proposed in the literature. Examples of stratified models are pyramids (Diday, 1984), which are structures less restrictive than the trees, and k-weak hierarchies (Bertrand and Janowitz, 2003), which are a generalization of the pyramids. Since these models include specific constraints on clusters, they suffer from the major drawback that the number of feasible object assignments is reduced.

Overlapping methods based on some partitioning approach extend primarily methods of strict or fuzzy classification to produce overlapping clusters. Several clustering methods have been used such as soft k-means and Threshold Meta-clustering Algorithm (Deodhar and Ghosh, 2006). The main issue in these methods is the prior threshold which is difficult to learn. In addition, the criteria optimized successively by these partitioning methods look for an optimal partition without introducing the overlap between data in an optimization step (Cleuziou, 2009).

More recent models for overlapping clustering solve these problems and look for optimal clusters. Banerjee *et al.* proposed the Model based Overlapping Clustering (MOC) which is considered as the first algorithm looking for optimal clusters (Banerjee *et al*, 2005). This algorithm is inspired from biology and is based on the Probabilistic Relational Model (PRM). Cleuziou proposed the Overlapping K-Means algorithm (OKM) which is considered as a generalization of k-means to detect overlap between clusters (Cleuziou, 2007). These methods cannot detect non-linearly separable clusters, thus fail to produce clus-

Ben N'Cir C., Essoussi N. and Bertrand P..
KERNEL OVERLAPPING K-MEANS FOR CLUSTERING IN FEATURE SPACE.
DOI: 10.5220/0003095102500256
In *Proceedings of the International Conference on Knowledge Discovery and Information Retrieval* (KDIR-2010), pages 250-256
ISBN: 978-989-8425-28-7

ters with complex boundaries. Moreover, it assumes that the number of clusters is known.

We propose, in this paper, a kernel overlapping clustering algorithm called Kernel Overlapping *k*-Means (KOKM) to produce cluster in a high, possibly infinite, feature space. A non linear mapping of original data to a higher feature space is implicitly realized using Mercer kernels. The clusters prototypes and objects images are computed in input space and only distance between objects are computed in feature space. The proposed KOKM algorithm combines advantages of kernel *k*-means algorithm which allows detection of non linearly separable clusters and advantages of OKM algorithm which produces overlapping clusters. In addition, to deal with the number of overlapping clusters that is prior fixed in OKM, we estimate this number using the Gram matrix.

This paper is organized as follows: Section 2 describes kernel *k*-means algorithm and OKM algorithm. Section 3 presents the approximate method used to estimate the number of overlapping clusters and describes the proposed KOKM algorithm. After that, Section 4 describes experiments and show results on Iris, EachMovie, and Ionosphere datasets. Finally, Section 5 presents conclusion and future works.

2 BACKROUND

2.1 OKM: Overlapping *K*-Means

The algorithm OKM is an extension of the algorithm *k*-means to produce overlapping clusters (Cleuziou, 2008). The minimization of the objective function is performed by itering two steps: 1) computation of clusters prototypes and 2) multiassignment of objects to one or several clusters. Given a set of data vectors $X = \{x_i\}_{i=1}^{N}$ with $x_i \in \mathbb{R}^d$, the aim of OKM is to find a set $\pi = \{\pi_c\}_{c=1}^{k}$ of k overlapping clusters such that the following objective function is minimized:

$$J(\pi) = \sum_{x_i \in X} \|x_i - im(x_i)\|^2. \quad (1)$$

This objective function minimizes the sum of squared Euclidean distances between object x_i and its image $im(x_i)$ for all $x_i \in X$. Image $im(x_i)$ is defined as the gravity center of clusters prototypes to which object x_i belongs as shown by eq. (2).

$$im(x_i) = \sum_{c \in A_i} m_c / |A_i|, \quad (2)$$

where A_i is the set of clusters to which x_i belongs and m_c is the prototype of cluster c.Algorithm OKM uses a function *PROTOTYPE* to update the cluster prototypes after each assignment stage. This function takes into account objects that are assigned to several clusters. It guarantees the minimization of the objective function J and then the convergence of the algorithm using the following general criterion:

$$m_h = \frac{1}{\sum_{x_i \in \pi_h} \alpha_i} \sum_{x_i \in \pi_h} \alpha_i . m_i^h, \quad (3)$$

where m_i^h and α_i are defined respectively by:

$$m_h^i = |A_i| x_i - \sum_{m_c \in A_i \setminus \{m_h\}} m_c, \quad (4)$$

$$\alpha_i = 1/|A_i|. \quad (5)$$

For the multiple assignment step, the OKM algorithm uses the function *ASSIGN* which enables that objects are assigned to one or more clusters. This function is based on a heuristic to minimize the space of possible assignments. The heuristic consisted in sorting clusters from closest to the farthest then assigning the objects in the order defined while assignment minimizes the distance between the object and its image.

The stopping rule of OKM algorithm is characterized by two criteria as in *k*-means. The criteria are: the maximum number of iterations and the minimum improvement of the objective function between two iterations.

This algorithm is not appropriate for clusters that are non linearly separable. Like *k*-means, this method fails when clusters have a complex boundary or when they are concentric. To deal with this problem in *k*-means, many solutions are proposed based on kernel methods like the method kernel *k*-means. These solutions map data to a higher feature space and look for separation in this space.

2.2 Kernel *k*-means

Kernel *k*-means (Scholkopf *et al*, 1998) is an extension of the standard algorithm *k*-means to solve the problem of non-linearly separable clusters. By an implicit mapping of the data from an input space to a higher, possibly infinite, feature space, kernel *k*-means looks for separation in feature space and solves the problem of clustering non spherical data that *k*-means suffers from. 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 objects x_i and x_j as measured by the kernel function (Camastra and Verri, 2005): $K_{ij} = K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$.

Kernel k-means aims at minimizing the sum of squared Euclidean errors in feature space as shown by eq. (6):

$$J(\pi) = \sum_{i=1}^{N} \sum_{c=1}^{k} P_{ic} \|\phi(x_i) - m_c\|^2, \quad (6)$$

where P_{ic} is a binary variable indicating membership of object x_i to cluster c. In feature space, the prototype can't be computed because the mapping function ϕ is generally unknown. However, the clustering error $\| \phi(x_i) - m_c\|$ can be computed using kernel function. This error measure is defined as follows:

$$\|\phi(x_i) - m_c\|^2 = \|\phi(x_i) - \frac{1}{N_c} \sum_{l=1}^{N_c} \phi(x_l)\|^2,$$

which can be expressed as:

$$= K_{ii} - \frac{2}{N_c} \sum_{l=1}^{N_c} K_{il} + \frac{1}{N_c^2} \sum_{p=1}^{N_c} \sum_{l=1}^{N_c} K_{pl}, \quad (7)$$

where N_c is the number of objects that belong to cluster c. To minimize this clustering error function, kernel k-means performs two principal steps: the determination of the nearest cluster from each object in feature space and the update of membership matrix of each object. The stopping rule is defined by the maximal number of iterations and the minimal improvement of the objective function between two iterations.

3 KERNEL OVERLAPPING K-MEANS

Our motivation in this paper is to improve overlapping clustering quality using advantages of kernel methods. Firstly, we use the Gram matrix to estimate the number of clusters. Secondly, we use Mercer kernel as an implicit non linear mapping of data to a higher feature space where we look for separation between overlapping clusters.

3.1 Estimating the Number of Clusters

For overlapping schemes, the overlap between clusters is an important characteristic that affects the determination of the appropriate number of clusters. However, it's difficult in real application to make a good choice between schemes with some clusters with strong overlap and schemes with many clusters with small overlap. But, if the overlap is important between two clusters, it's more suitable to create a third cluster to minimize overlapping objects. Following this approach, the overlap between data is minimized and the number of clusters is approximately equal to the number of groups in data.

Based on this assumption that the number of overlapping clusters is roughly equal to the number of clusters, and by taking advantage of the kernel trick, we can estimate an approximate number of clusters using the Gram matrix. The kernel matrix (Gram matrix) is the square matrix $K \in \mathbb{R}^{N \times N}$ such that K_{ij} is equal to $\langle \phi(x_i), \phi(x_j) \rangle$ for all $x_1, \ldots, x_N \in X$.

The Gram matrix can be used to determine the number of clusters in data. As each element of the kernel matrix is a dot-product value in the feature space, the matrix will have a block diagonal structure when there are well-separated clusters within the data sets. This diagonal structure block can be used to determine the number of clusters (Girolami, 2002). This method was first used in the c-means clustering (fuzzy clustering), and it's still interesting for overlapping clustering. Thus, through counting the number of significant eigenvalues of the kernel matrix, we can obtain the number of clusters (Zhang and Chen, 2002).

3.2 KOKM Algorithm

Mercer kernel functions map data from input space to high, possibly infinite, dimensional feature space without computing the non linear mapping function ϕ. In Feature space, the distance measure between any two patterns is given by:

$$\begin{aligned} d(\phi(x_i), \phi(x_j)) &= \|\phi(x_i) - \phi(x_j)\|^2 \\ &= (\phi(x_i) - \phi(x_j))^T (\phi(x_i) - \phi(x_j)) \\ &= \langle \phi(x_i), \phi(x_i) \rangle + \langle \phi(x_j), \phi(x_j) \rangle \\ &\quad -2\langle \phi(x_i), \phi(x_j) \rangle \\ &= K_{ii} + K_{jj} - 2K_{ij}. \end{aligned} \quad (8)$$

Since ϕ is non-linear, $d(\phi(x_i), \phi(x_j))$ is a class of kernel induced non-Euclidean distance measures (Ben-Hur *et al*, 2000). If the kernel used is an RBF kernel, eq. (8) is reduced to:

$$\|\phi(x_i) - \phi(x_j)\|^2 = 2 - 2\, exp\{\frac{-\|x_i - x_j\|^2}{\sigma^2}\}. \quad (9)$$

In the context of proposing a kernel version for overlapping k-means to deal with non spherical clusters, we propose the KOKM algorithm based on kernel induced distance measure as shown by eq. (8). In this algorithm we map initial data to a higher feature space where the objective function J is optimized in this space. The objective function in KOKM is adapted to minimize the distance between each object and its corresponding image in feature space as shown by eq.

(10). We adopt a new kernel induced distance measure to replace the original Euclidean norm metric in OKM. The images of objects are computed in the original data space so that the clustering results can be interpreted in the original space.

$$\begin{aligned} J(\pi) &= \sum_{x_i \in X} \|\phi(x_i) - \phi(im(x_i))\|^2 \\ &= \sum_{x_i \in X} K_{ii} + \sum_{x_i \in X} K_{im(x_i)\, im(x_i)} \\ &\quad -2 \sum_{x_i \in X} K_{im(x_i)\, i} \end{aligned} \tag{10}$$

In addition to the objective function, we modified also the function "*ASSIGN*" used in OKM to affect objects to their nearest clusters. The choice of the nearest center of cluster from any object is performed in feature space using eq. (11).

$$m_i^\star = \arg\min_{m_c} \|\phi(x_i) - \phi(m_c)\|^2, \tag{11}$$

in other words, $m_i^\star$ is the nearest cluster prototype from object x_i. We propose a new function called "*K_ASSIGN*" where objects are affected to one or several clusters in feature space. The function "*K_ASSIGN*" can be described as follows:

Algorithm 1: $K\_ASSIGN(x_i, \{m_1, ...m_k\}, A_i^{old}) \rightarrow A_i$.

INPUT x_i: Vector in $\mathbb{R}^d$.
$\{m_1, ...m_k\}$: Set of center of k clusters.
A_i^{old}: Old affectation of object i.
OUTPUT A_i: New affectation of x_i.

1: Compute A_i using eq. (11) and compute $im(x_i)$ with affectations A_i.
2: Looking for the nearest cluster which is not included in A_i using eq. (11) and compute $im'(x_i)$ with affectations $A_i \cup \{m^\star\}$
3: **if** $\|\phi(x_i) - \phi(im'(x_i))\|^2 < \|\phi(x_i) - \phi(im(x_i))\|^2$ **then**
4: $A_i \leftarrow A_i \cup \{m^\star\}$, $im(x_i) = im'(x_i)$ and we go to step 2.
5: **else**
6: compute im^{old} with affectation A_i^{old}.
7: **if** $\|\phi(x_i) - \phi(im(x_i))\|^2 \leq \|\phi(x_i) - \phi(im^{old}(x_i))\|^2$ **then**
8: return A_i.
9: **else**
10: return A_i^{old}.
11: **end if**
12: **end if**

The prototypes are computed in input space like in OKM algorithm using the function "*PROTOTYPE*" (Cleuziou, 2008). The distances between objects, and the distances between objects and prototypes are computed only in feature space. Based on the above functions, to implement kernel overlapping k-means, we derive the following KOKM algorithm:

Algorithm 2: $KOKM(X, t_{max}, \varepsilon, k) \rightarrow \{\pi_c\}_{c=1}^k$.

INPUT X: set of vector in R^d.
t_{max}: maximum number of iterations.
ε: minimal improvement in objective function.
k: number of clusters.
OUTPUT π: set of k clusters.

1: Choose the kernel function and its corresponding parameters.
2: Initialize centers of clusters with random clusters prototypes, initialize clusters memberships using "*K_ASSIGN*" and derive value of the objective function $J_{t=0}(\pi)$ in iteration 0 using eq. (10).
3: Compute clusters prototypes using function "*PROTOTYPE*".
4: Assign objects to one or several clusters using "*K_ASSIGN*".
5: Compute objective function $J_t(\pi)$ using eq. (10).
6: **if** $(t < t_{max}$ and $J_t(\pi) - J_{t-1}(\pi) > \varepsilon)$ **then**
7: go to step 4.
8: **else**
9: return the distribution of clusters memberships.
10: **end if**

4 EXPERIMENTS

Experiments are performed on datasets including either overlapping or non overlapping clusters. We used a computer with 4-GB RAM and 2.1-GHz Intel Core 2 Duo processor and the code was implemented in C. The construction of Gram matrices and extraction of eigenvalues was realized in Matlab.

In all the experiments, the best parameter of kernel function is empirically determined. Since the appropriate kernel parameter selection is out of the scope of this work, we only give the best kernel parameter.

Numerical results obtained by OKM using Euclidean distance are compared to those obtained by KOKM using Polynomial kernel and RBF kernel. Results are compared according to three validation measures: *precision*, *recall* and *F-measure*. These validation measures attempt to estimate whether the prediction of categories was correct with respect to the underlying true categories in the data.

Precision is calculated as the fraction of objects correctly labeled as belonging to the positive class divided by the total number of objects labeled as belonging to the positive class. *Recall* is the fraction of

objects correctly labeled as belonging to the positive class divided by the total number of elements that actually belong to the positive class. The *F-measure* is the harmonic mean of *precision* and *recall*.

$$Precision = NCLO/TNLO$$
$$Recall = NCLO/TNAC$$
$$F\text{-}measure = 2 \times Precision \times Recall / Precision + Recall$$

where *NCLO*, *TNLO* and *TNAC* are respectively the number of correctly labled objects, the total number of labled objects and the total number of objects that actually belong to the positive class.

4.1 Non Overlapping Datasets

Experiments are performed on two artificial non overlapping datasets which are Iris [1] dataset and Ionosphere[2] dataset. Iris dataset is traditionally used as a base's test for evaluation. It is composed of 150 data in $\mathbb{R}^4$ tagged according to three non-overlapping clusters (50 objects per class). One of these clusters "setosa" is known to be clearly separated from the two others.

Figure 1 shows the most significant eigenvalues of the Gram matrix. The kernel used is an RBF kernel with $\sigma^2 = 150$. There are at least two clusters, another clusters looks less important. If we choose to add this cluster then we obtain less overlap between data. So, known that Iris doesn't contain overlaps between data, it's more suitable to add this third cluster. The optimal choice is then three clusters.

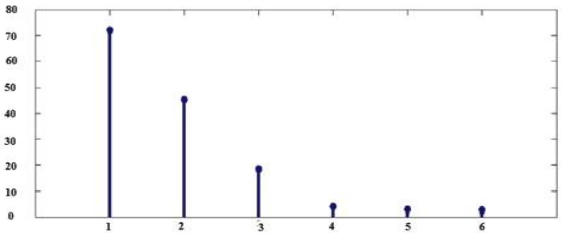

Figure 1: Most significant Eigenvalues on Iris dataset.

Then using OKM with Euclidean distance, KOKM with both RBF and Polynomial kernel and estimating the number of clusters as $k = 3$, we run each algorithm twenty times (with similar initializations). The mean of results obtained are reported in Table 1. We note that KOKM with both Polynomial and RBF kernel gives better classification results than OKM. The improvement is achieved both in terms of precision and in terms of recall.

[1] cf. http://archive.ics.uci.edu/ml/datasets/Iris.

[2] cf. http://archive.ics.uci.edu/ml/datasets/ Ionosphere.

The second artificial dataset is Ionosphere which is built by a radar system in Goose Bay Labrador. This radar system has 16 antennas with a total transmitted power of 6.4 kilowatts. This system analyzes the electrons in the ionosphere. Some electrons show a certain type of structure in ionosphere. These electrons determine the first class in the dataset that is the class "good". Other electrons show no structure in ionosphere. They define the second class in the dataset that is the class "bad".

Table 1: Comparison between OKM and KOKM on Iris dataset.

Method	Precesion	Recall	F-measure
OKM based Euclidean distance	0.707	0.900	0.815
KOKM with RBF kernel	0.771	0.906	0.834
KOKM with Polynomial kernel	**0.808**	**0.993**	**0.892**

Table 2: Comparison between OKM and KOKM on Ionosphere dataset.

Method	Precesion	Recall	F-measure
OKM based Euclidean distance	0.532	0.689	0.597
KOKM with RBF kernel	0.560	**0.737**	0.636
KOKM with Polynomial kernel	**0.580**	0.721	**0.644**

Electrons are transmitted via a signal from the antennas. This signal is described by 34 attributes that will constitute the size of the database Ionosphere. The total number of signal in the database is 351 signals. The characteristic of this dataset is that the two classes have a concentric shape that will be difficult to separate by some linear clustering algorithm as shown in Figure 2.

When we map this data to higher dimensional space using RBF kernel, it will be easier to find separation between good and bad electrons. The two classes lose their concentric shape and become linearly separable. Figure 3 shows the most significant eigenvalues on Ionosphere dataset. There are two important eigenvalues that we consider as the number of clusters. Then, we run OKM and KOKM twenty times (with similar initializations) with $k = 2$. The mean of results obtained are presented in Table 2. Similarly to experiments on Iris data set, this table shows the usefulness of KOKM with both Polynomial and RBF kernel where improvement is achieved in terms of both precision and recall.

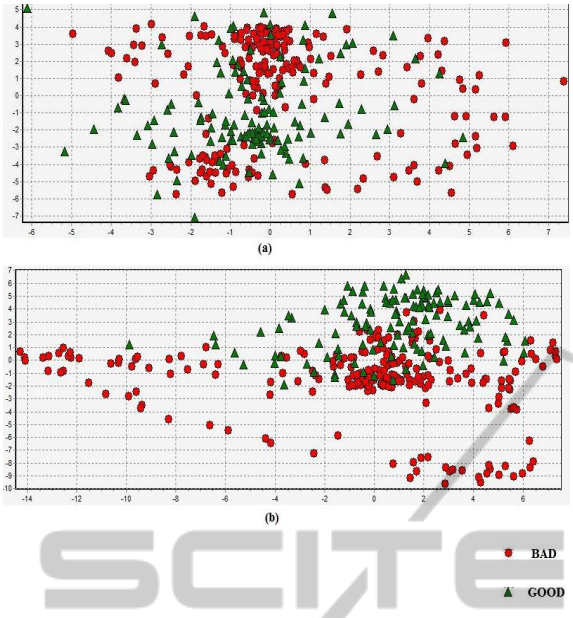

Figure 2: 2D plot of Ionosphere dataset using the first and second principal axis obtained with PCA: (a) data in input space (b) data in feature space.

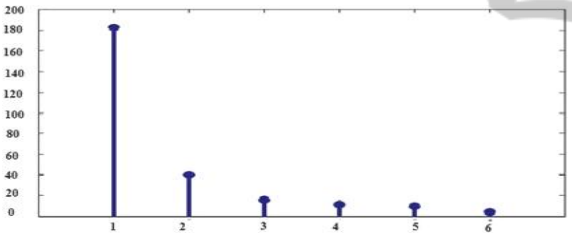

Figure 3: Most significant Eigenvalues on Ionosphere dataset.

We note that Polynomial kernel give better results than RBF kernel on both IRIS and Ionosphere datasets.

4.2 Overlapping Datasets

The EachMovie[3] dataset contains user ratings for each movie in the collection. Users give ratings on a scale of 1-5, with 1 indicating extreme dislike and 5 indicating strong approval. There are 74,424 users in this dataset, but the mean and the median number of users voting on any movie are 1732 and 379 respectively. As a result, if each movie in this dataset is represented as a vector of ratings over all the users, the vector is high-dimensional but typically very spars (Banerjee and al, 2005). For every movie in this dataset, 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 dataset has naturally overlapping clusters since many movies are annotated in IMDB as belonging to multiple genres. For example, Aliens movie belongs to three genres: action, horror and science fiction.

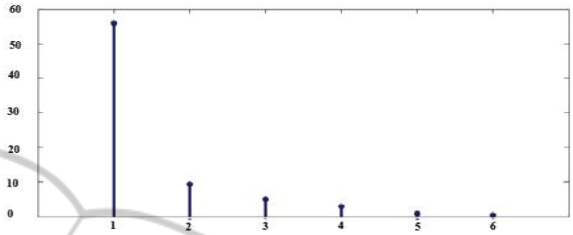

Figure 4: Most significant Eigenvalues on EachMovie dataset.

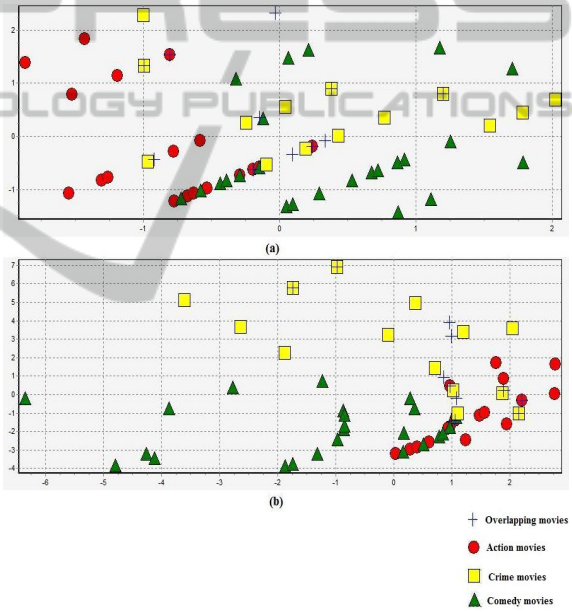

Figure 5: 2D plot of EachMovie dataset using the first and second principal axis obtained with PCA: (a) data in input space (b) data in feature space.

We extracted a subset from the EachMovie dataset: 75 movies scattered on three overlapping clusters as follows: action=21 movies; comedy=26 movies; crime=17 movies; action+crime=11 movies; and based on age, sex and rate of users we try to find a category of video. Figure 5 shows the initial distribution of these movies where overlapping movies belong to both action and crime genres. When we map the same movies in a higher feature space, we don't perceive a good improvement on movies distribution because the selected attributes (age, sex and user rate) can't easily predict the genre of movie. But,

[3]cf. http://www.grouplens.org/node/76.

we remark that overlapping movies are more easily detected in feature space and it lays in the surface between action and crime movies.

To estimate the number of clusters, we built a Gram matrix using an RBF kernel with $\sigma = 2$. Figure 4 shows the most significant eigenvalues of the Gram matrix. We get between 3 and 4 significant eigenvalues. Known that EachMovie subset is an overlapping subset, the suitable choice is three clusters.

Table 3: Comparison between OKM and KOKM on EachMovie dataset.

Method	Precesion	Recall	F-measure
OKM based Euclidean distance	0.582	0.827	0.687
KOKM with RBF kernel	0.594	0.827	0.692
KOKM with Polynomial kernel	**0.628**	**0.851**	**0.722**

Then, using OKM with Euclidean distance and KOKM based on an RBF and on a Polynomial kernel, and by fixing the number of cluster to $k = 3$, we run each algorithm twenty times (with similar initializations). Table 3 shows the results obtained where KOKM algorithm with Polynomial kernel gives the best results. These results confirm the first results obtained on Iris and Ionosphere datasets. KOKM improves overlapping clustering quality and Polynomial kernel gives the best results on all tested datasets.

5 CONCLUSIONS

We have proposed in this paper the kernel overlapping *k*-means clustering algorithm. This algorithm maps data from input space to a higher dimensional feature space through the use of a kernel Mercer function and optimizes an objective function that looks for optimal clusters in feature space. The main advantages of this algorithm are its ability to identify nonlinearly separable clusters in input space and its ability to separate clusters with complex boundary. Moreover, we propose an estimation of the number of clusters using the Gram matrix. This estimation is based on the assumption that we must add more clusters when the overlap between clusters becomes larger. Empirical results show that KOKM using both Polynomial and RBF kernels outperforms OKM in terms of precision recall and F-measure for overlapping clusters and for non overlapping clusters.

As a future work, we plan to improve this kernel overlapping k-means algorithm by proposing another version of KOKM where prototypes and objects images are computed in feature space. In this way, kernel overlapping clustering can be applied to structured data, such as trees, strings, histograms and graphs.

REFERENCES

Banerjee, A., Krumpelman, C., Basu, S., Mooney, R., and Ghosh, J. (2005). Model based overlapping clustering. In *International Conference on Knowledge Discovery and Data Mining*, Chicago, USA. SciTePress.

Ben-Hur, A., Horn, D., Siegelmann, H. T., and Vapnik, V. (2000). Support vector clustering. In *International Conference on Pattern Recognition*, pages 724–727, Barcelona, Spain.

Bertrand, P. and Janowitz, M. F. (2003). The k-weak hierarchical representations: an extension of the indexed closed weak hierarchies. *Discrete Applied Mathematics*, 127(2):199–220.

Camastra, F. and Verri, A. (2005). A novel kernel method for clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:801–804.

Cleuziou, G. (2007). Okm : une extension des k-moyennes pour la recherche de classes recouvrantes. *Revue des Nouvelles Technologies de l'Information, Cpadus-Edition RNTI-E*, 2:691–702.

Cleuziou, G. (2008). An extended version of the k-means method for overlapping clustering. In *International Conference on Pattern Recognition ICPR*, pages 1–4, Florida, USA. IEEE.

Cleuziou, G. (2009). Okmed et wokm : deux variantes de okm pour la classification recouvrante. *Revue des Nouvelles Technologies de l'Information, Cpadus-Edition*, 1:31–42.

Deodhar, M. and Ghosh, J. (2006). Consensus clustering for detection of overlapping clusters in microarray data.workshop on data mining in bioinformatics. In *International Conference on data mining*, pages 104–108, Los Alamitos, CA, USA. IEEE Computer Society.

Diday, E. (1984). Orders and overlapping clusters by pyramids. Technical Report 730, INRIA, France.

Girolami, M. (2002). Mercer kernel-based clustering in feature space. *IEEE Transactions on Neural Networks*, 13(13):780–784.

Schölkopf, B., Smola, A., and Müller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319.

Zhang, D. and Chen, S. (2002). Fuzzy clustering using kernel method. In *International Conference on Control and Automation*, pages 123–127, Xiamen, China.