Fuzzy C-Means, Gustafson-Kessel FCM, and Kernel-Based FCM: A Comparative Study

Daniel Graves and Witold Pedrycz

Department of Electrical and Computer Engineering, University of Alberta, 2nd Floor ECERF, 9107 - 116 Street, Edmonton, Alberta, Canada T6G 2V4
dgraves@ualberta.ca, pedrycz@ece.ualberta.ca

Abstract. This paper is concerned with a comparative study of the performance of fuzzy clustering algorithms Fuzzy C-Means (FCM), Gustafson-Kessel FCM (GK-FCM) and two variations of kernel-based FCM. One kernel-based FCM (KFCM) retains prototypes in the input space while the other (MKFCM) implicitly retains prototypes in the feature space. The two performance criteria used in the evaluation of the clustering algorithm deal with produced classification rate and reconstruction error. We experimentally demonstrate that the kernel-based FCM algorithms do not produce significant improvement over standard FCM for most data sets under investigation It is shown that the kernel-based FCM algorithms appear to be highly sensitive to the selection of the values of the kernel parameters.

Keywords: Fuzzy Clustering, Kernels, Fuzzy Kernel-based Clustering, FCM.

1 Introduction

In determining the structure in data, fuzzy clustering offers an important insight into data by producing gradual degrees of membership to individual patterns within clusters. A significant number of fuzzy clustering algorithms have been developed with widely known methods such as FCM [1] and GK-FCM [5,6]. Recently kernel-based fuzzy clustering has been developed and already gained some popularity, cf. [2,3,12,13,14,15,16]. Kernel-based clustering algorithms map features from the feature space to a higher dimensional kernel space and perform clustering in the kernel space. There are two major variations of kernel-based fuzzy clustering. One forms prototypes in the feature space (input space) while the other forms prototypes in the higher dimensional kernel space and completes an inverse mapping of prototypes from kernel space back to the feature space. They are given the acronyms KFCM and MKFCM, respectively. The research interest is to determine the performance of kernel-based clustering vis-à-vis some well known standards such as FCM and GK-FCM. Kernel-based fuzzy clustering has a larger computational demand than FCM thus it becomes of interest to assess to which extent they bring tangible benefits in terms of the quality of the produced results. The key objectives of the study are outlined as follows

- determine how well the two kernel-based FCM algorithms perform on a suite of synthetic and real data when compared with standard FCM and GK-FCM
- determine how sensitive the selection of the kernel parameters is to the quality of the clustering results

To offer an unbiased evaluation of the results of clustering, we introduce two criteria. The first one is a classification rate which emphasizes the role of clusters in the formation of homogeneous groups of patterns (data) belonging to the same class. The second one deals with a reconstruction error and could be sought as an internal measure of quality of the clusters. The paper is organized as follows: background presents both GK-FCM and kernel-based FCM (Section 2), evaluation criteria (Section 3), and conclusions (Section 4). Throughout the study we use the standard notation as encountered in fuzzy clustering: a finite collection of N patterns is described as $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ and collection of c cluster centers (prototypes) is denoted $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_c\}$ with $\mathbf{x}, \mathbf{v} \in \Re^d$. The fuzzy partition matrix is $\mathbf{U} = [u_{ik}]$ where $u_{ik} \in [0,1]$ and the fuzzification coefficient of the FCM is denoted as $m > 1$.

2 Background

The FCM [1] minimizes a well-known dispersion performance index. The results are provided in the form of a fuzzy partition matrix as well as a collection of "c" prototypes. Given the Euclidean distance function, FCM favors spherical shapes of the clusters. Many variations of the FCM algorithm have been developed including the Gustafson-Kessel FCM, fuzzy c-means ellipsoid version, and recently kernel-based FCM. These generalizations are aimed at the development of clusters when dealing with a non-spherical geometry of data.

Gustafson-Kessel FCM

The Gustafson-Kessel FCM [5,6] extends the standard FCM algorithm by introducing an augmented version of the Euclidean distance to be in the form $\mathrm{d}^2_{GK} = (\mathbf{x}_k - \mathbf{v}_i)^{\mathrm{T}} \mathbf{A}_i (\mathbf{x}_k - \mathbf{v}_i)$ where $\mathbf{A}_i$ is calculated using a scaled inverse fuzzy covariance matrix from each cluster. The Gustafson-Kessel FCM minimizes the following performance criterion:

$$Q = \sum_{i=1}^{c} \sum_{k=1}^{N} u_{ik}^m \, \mathrm{d}^2_{GK} = \sum_{i=1}^{c} \sum_{k=1}^{N} u_{ik}^m (\mathbf{x}_k - \mathbf{v}_i)^{\mathrm{T}} \mathbf{A}_i (\mathbf{x}_k - \mathbf{v}_i) \tag{1}$$

The optimization of the performance criterion is subject to the standard constraints:

$$\forall i,k \;\; u_{ik} \in [0,1], \;\; \forall i \;\; 0 < \sum_{k=1}^{N} u_{ik} < N, \text{ and } \forall k \;\; \sum_{i=1}^{c} u_{ik} = 1. \tag{2}$$

The matrix $\mathbf{A}_i$ is calculated based on the inverse of fuzzy covariance matrix $\mathbf{C}_i$

$$\mathbf{A}_i = (\rho_i |\mathbf{C}_i|)^{1/d} \mathbf{C}_i^{-1}, \;\; \mathbf{C}_i = \frac{\sum_{k=1}^{N} u_{ik}^m (\mathbf{x}_k - \mathbf{v}_i)(\mathbf{x}_k - \mathbf{v}_i)^{\mathrm{T}}}{\sum_{k=1}^{N} u_{ik}^m} \tag{3}$$

Obviously, we assume that $\mathrm{C_i}$ is invertible otherwise the algorithm will not produce any solution.

Kernel-Based FCM

The kernel method involves performing an arbitrary non-linear mapping Φ from the feature space (input space) to a higher dimensional and possibly infinitely dimensional kernel space [3,4,8,11]. The rationale for going to higher dimensions is that it is possible to apply a linear classifier in the kernel space that results in non-linear classification in the input space [8,11]. Dot products in the kernel feature space can be expressed by a Mercer kernel [3,4,8,11]; thus, dot products in the feature space are not explicitly computed but rather replaced by a Mercer kernel function. A mercer kernel is defined as any positive semi-definite symmetric function [4]. Common mercer kernel functions include Gaussian $e^{-\|\mathbf{x}-\mathbf{y}\|^2/\sigma^2}$ where $\sigma^2 > 0$ and polynomial $(\mathbf{x}^{\mathrm{T}}\mathbf{y}+\theta)^p$ where $\theta \geq 0$ and $p > 0$ [4].

There are two major classes in kernel-based fuzzy clustering in the literature which will be distinguished by calling the first KFCM[12,14] and the second MKFCM [2,13,15,16] following the acronym given in [16]. KFCM is a kernel-based FCM clustering method where the prototypes are determined in the original input space and are implicitly mapped to the kernel feature space through a kernel function. The MKFCM is similar to KFCM except the prototypes are implicitly left in the kernel feature space and thus the inverse of the mapping Φ must be approximated in order to find the prototypes in the original input space. Authors in [16] provide a method of approximating the prototypes in the original feature space.

KFCM: Prototypes in Input Space

The advantage of the KFCM clustering algorithm is that the prototypes are retained in the original input space and thus could be interpreted. KFCM minimizes the following performance criterion [12,14]:

$$Q = \sum_{i=1}^{c}\sum_{k=1}^{N} u_{ik}^{m} \left\|\Phi(\mathbf{x}_k) - \Phi(\mathbf{v}_i)\right\|^2 \tag{4}$$

The distance is computed in the feature space by using a Mercer kernel such that

$$\left\|\Phi(\mathbf{x}_k) - \Phi(\mathbf{v}_i)\right\|^2 = \mathrm{K}(\mathbf{x}_k,\mathbf{x}_k) + \mathrm{K}(\mathbf{v}_i,\mathbf{v}_i) - 2\,\mathrm{K}(\mathbf{x}_k,\mathbf{v}_i) \tag{5}$$

For the Gaussian kernel function $\mathrm{K}(\mathbf{x},\mathbf{x}) = 1$. By applying Lagrange multipliers to optimize Q with respect to u_{ik} and $\mathbf{v}_i$ for the Gaussian kernel, one obtains [12,14]:

$$u_{ik} = \frac{1}{\sum_{j=1}^{c}\left(\frac{1-\mathrm{K}(\mathbf{x}_k,\mathbf{v}_i)}{1-\mathrm{K}(\mathbf{x}_k,\mathbf{v}_j)}\right)^{\frac{1}{m-1}}}, \quad \mathbf{v}_i = \frac{\sum_{k=1}^{N} u_{ik}^{m}\,\mathrm{K}(\mathbf{x}_k,\mathbf{v}_i)\mathbf{x}_k}{\sum_{k=1}^{N} u_{ik}^{m}\,\mathrm{K}(\mathbf{x}_k,\mathbf{v}_i)} \tag{6}$$

Other kernels can be used however the prototype equation must be re-derived.

MKFCM Prototypes in Kernel Feature Space

The advantage of leaving the prototypes in the kernel space and finding the approximate prototypes in the feature space after clustering is that any kernel function can be used in clustering and there is possibly some additional flexibility in clustering because the prototypes aren't constrained to the feature space. The disadvantage is that the prototypes need to be approximated in the feature space from the kernel space.
The MKFCM algorithm [16] minimizes Q:

$$Q=\sum_{i=1}^{c}\sum_{k=1}^{N}u_{ik}^{m}\left\|\Phi(\mathbf{x}_k)-\mathbf{v}_i\right\|^2 , \quad \mathbf{v}_i=\frac{\sum_{k=1}^{N}u_{ik}^{m}\Phi(\mathbf{x}_k)}{\sum_{k=1}^{N}u_{ik}^{m}} \tag{7}$$

Assuming the Euclidean distance and substituting $\mathbf{v}_i$ into $\mathrm{d}_{ik}^2=\left\|\Phi(\mathbf{x}_k)-\mathbf{v}_i\right\|^2$ the metric in the kernel space denoted by d_{ik} as found in terms of the kernel function K

$$\mathrm{d}_{ik}^2=\left\|\Phi(\mathbf{x}_k)-\mathbf{v}_i\right\|^2=\mathrm{K}(\mathbf{x}_k,\mathbf{x}_k)-2\frac{\sum_{j=1}^{N}u_{ij}^{m}\,\mathrm{K}(\mathbf{x}_k,\mathbf{x}_j)}{\sum_{j=1}^{N}u_{ij}^{m}}+\frac{\sum_{j=1}^{N}\sum_{l=1}^{N}u_{ij}^{m}u_{il}^{m}\,\mathrm{K}(\mathbf{x}_j,\mathbf{x}_l)}{\left(\sum_{j=1}^{N}u_{ij}^{m}\right)^2} \tag{8}$$

The same constraints apply on the membership partition as in standard FCM thus Lagrange multiplies were used to minimize Q with respect to u_{ik}

$$u_{ik}=\frac{1}{\sum_{j=1}^{c}\left(\frac{\mathrm{d}_{ik}^2}{\mathrm{d}_{jk}^2}\right)^{\frac{1}{m-1}}} \tag{9}$$

The clustering is done without explicitly calculating the prototypes in either the kernel space or the feature space thus the algorithm produces a membership partition matrix only by iteratively updating u_{ik} from some initial random membership partition matrix.

Since the prototypes are retained in the kernel space, there needs to be a mechanism to determine the prototypes in the feature space. The method outlined in [16] minimizes

$$V=\left\|\Phi(\mathbf{v}_i)-\mathbf{v}_i\right\|^2=\frac{\sum_{j=1}^{N}\sum_{l=1}^{N}u_{ij}^{m}u_{il}^{m}\,\mathrm{K}(\mathbf{x}_j,\mathbf{x}_l)}{\left(\sum_{j=1}^{N}u_{ij}^{m}\right)^2}-2\frac{\sum_{j=1}^{N}u_{ij}^{m}\,\mathrm{K}(\mathbf{x}_j,\mathbf{v}_i)}{\sum_{j=1}^{N}u_{ij}^{m}}+\mathrm{K}(\mathbf{v}_i,\mathbf{v}_i) \tag{10}$$

Solving $\partial V / \partial \mathbf{v}_i = \mathbf{0}$ requires knowledge of the kernel function K . The formulae for Gaussian (left) and polynomial (right) kernels are

$$\mathbf{v}_i = \frac{\sum_{k=1}^{N} u_{ik}^m e^{-\|\mathbf{x}_k - \mathbf{v}_i\|^2/\sigma^2} \mathbf{x}_i}{\sum_{k=1}^{N} u_{ik}^m e^{-\|\mathbf{x}_k - \mathbf{v}_i\|^2/\sigma^2}}, \quad \mathbf{v}_i = \frac{\sum_{k=1}^{N} u_{ik}^m (\mathbf{x}_k^{\mathrm{T}} \mathbf{v}_i + \theta)^{p-1} \mathbf{x}_k}{\sum_{k=1}^{N} u_{ik}^m (\mathbf{x}_k^{\mathrm{T}} \mathbf{v}_i + \theta)^{p-1}} \tag{11}$$

The prototypes are approximated iteratively using a Kernel-dependant formula such as the ones given for Gaussian kernels or polynomial kernels after the evaluation of the membership partition matrix.

3 Experiments

A number of experiments were run for a series of data sets on the standard FCM algorithm, Gustafson-Kessel FCM and variations of the Kernel-based FCM algorithm for a thorough comparison of the performance of the algorithms. Three synthetic data sets were used and a number of data sets from UCI Machine Learning database were used.

The classification rate and reconstruction error are the two performance criteria that are used to evaluate the performance of the clustering algorithms in this paper. The first evaluates performance by using the correct class as in the context of a classifier. The second is a performance measure that measures the cohesiveness of patterns within a cluster and thus does not require knowledge of the correct classes.

A value of $\rho_i = 1$ was used for the GK-FCM algorithm and to ensure a non-zero determinate for the fuzzy covariance matrix, if $|\mathbf{C}_i| < 10^{-10}$ the covariance matrix was replaced by the identity matrix. FCM, Gustafson-Kessel FCM and KFCM algorithms were averaged over 100 runs while MKFCM was averaged over 20 runs due to increased overhead of the kernel method and varying the kernel parameters. The mean and standard deviation of the classification rates and reconstruction errors are reported. Some of the standard deviations are quite small in the results and thus appear as 0 in the tables. This is due to highly consistent results during the runs and the relatively large number of runs that were averaged.

3.1 Evaluation Criteria

The structure of the data discovered by a clustering algorithm can be compared with a known structure of classes to produce the classification rate. This is accomplished by transforming the fuzzy partition matrix into a partition matrix and using the maximum rule for each pattern to select the cluster with the largest membership value. Class labels are assigned to each cluster according to the class that dominates that cluster. The percentage of patterns that belong to a correctly labeled cluster is called the classification rate. The classification rate can be determined by building a contingency matrix [7]. Higher classification rates indicate better clustering results.

The reconstruction error is a fuzzy-based measure of fuzzy clustering algorithms [10]. The reconstruction error is a measure of spread of clusters from the prototypes by finding the sum of square differences between the original patterns and the decoded patterns. The decoded pattern $\tilde{\mathbf{x}}_k$ and reconstruction error r are calculated by the following formulae:

$$r = \sum_{k=1}^{N} \left\| \mathbf{x}_k - \tilde{\mathbf{x}}_k \right\|^2 , \ \tilde{\mathbf{x}}_k = \frac{\sum_{i=1}^{c} u_{ik}^{m} \mathbf{v}_i}{\sum_{i=1}^{c} u_{ik}^{m}} \tag{12}$$

Smaller values for the reconstruction error indicate better clustering results as the clusters are denser and the prototypes are further apart.

3.2 Synthetic Data Sets

Three synthetic data sets of two dimensions are used. The fuzzy XOR data set has 1000 patterns while the parallel lines and the ring data sets have 200 patterns. The fuzzy XOR data set is two strip clusters that follow an XOR pattern. The lines data set consists of two parallel lines. The ring data consists of two rings – one inside the other. There are two clusters in each synthetic data set and both clusters have an equal number of patterns.

The results for the clustering of the Fuzzy XOR data set show the Gustafson-Kessel (GK) FCM clustering algorithm is the clear winner in terms of a classification rate of 93.4% when $c = 2$ however as the number of clusters increase to 4, all clustering algorithms perform very similarly. The prototypes produced by GK FCM are at the center thus the reconstruction error is relatively large. Interestingly the polynomial MKFCM algorithm performs fairly well for $c = 2$ with an average classification rate of 70.2% by capturing three sides of the X in one cluster. The reconstruction error for MKFCM kernel-methods contends very closely with standard FCM.

The kernel methods perform significantly better for the ring data set in terms of classification rate requiring fewer clusters than GK and standard FCM. The kernel-based algorithms appear to be providing some non- spherical cluster shapes for this particular example. As figures 1 and 2 indicate, the performance of the kernel-based method is very sensitive to the choice of the numeric values of the kernels. MKFCM performs better than the other kernel method KFCM. For the Polynomial kernel a larger power of p gives better classification rates but poorer reconstruction errors.

As for the parallel line data set, the GK-FCM was able to classify the data set with a classification rate close to 100%. FCM performed fairly poorly on the line data in terms of classification rate however FCM and Polynomial MKFCM had the smallest reconstruction error. The kernel algorithms performed rather poorly except the Gaussian MKFCM algorithm was able to classify around 94% of the data however the results were highly sensitive to the kernel parameters.

Table 1. Fuzzy XOR Data Set Results

c	Clustering	Classification Rate	Reconstruction Error
2	FCM	$51.1 \pm 0.1\%$ (m=2.5)	0.1047 ± 0.0005 (m=1.4)
	GK-FCM	$93.4 \pm 0.0\%$ (m=3)	0.1400 ± 0.0000 (m=1.4)
	KFCM (G)	$51.9 \pm 3.9\%$ (m=1.4, $\sigma^2 = 0.5$)	0.1096 ± 0.0005 (m=1.4, $\sigma^2 = 2$)
	MKFCM (G)	$51.2 \pm 0.1\%$ (m=2.5, $\sigma^2 = 8$)	0.1047 ± 0.0005 (m=1.2, $\sigma^2 = 20$)
	MKFCM (P)	$70.2 \pm 7.6\%$ (m=2.5, $\theta = 50$, $p = 16$)	0.1045 ± 0.0005 (m=1.4, $\theta = 15$, $p = 2$)
3	FCM	$77.5 \pm 8.1\%$ (m=3)	0.0676 ± 0.0003 (m=1.4)
	GK-FCM	$93.2 \pm 0.4\%$ (m=1.2)	0.0973 ± 0.0000 (m=3)
	KFCM (G)	$79.9 \pm 9.1\%$ (m=1.4, $\sigma^2 = 0.5$)	0.0699 ± 0.0003 (m=1.2, $\sigma^2 = 2$)
	MKFCM (G)	$78.3 \pm 8.2\%$ (m=2.5, $\sigma^2 = 4$)	0.0676 ± 0.0002 (m=1.4, $\sigma^2 = 30$)
	MKFCM (P)	$80.0 \pm 8.3\%$ (m=2.5, $\theta = 15$, $p = 4$)	0.0676 ± 0.0002 (m=1.4, $\theta = 35$, $p = 2$)
4	FCM	$93.8 \pm 0.0\%$ (m=1.4)	0.0251 ± 0.0000 (m=2)
	GK-FCM	$94.0 \pm 0.0\%$ (m=3)	0.0221 ± 0.0000 (m=3)
	KFCM (G)	$93.9 \pm 0.1\%$ (m=1.4, $\sigma^2 = 2$)	0.0306 ± 0.0002 (m=1.2, $\sigma^2 = 2$)
	MKFCM (G)	$93.8 \pm 0.1\%$ (m=1.4, $\sigma^2 = 12$)	0.0252 ± 0.0000 (m=2, $\sigma^2 = 30$)
	MKFCM (P)	$93.4 \pm 0.1\%$ (m=1.4, $\theta = 50$, $p = 2$)	0.0251 ± 0.0000 (m=2, $\theta = 25$, $p = 2$)

Table 2. Ring Data Set Results

c	Clustering	Classification Rate	Reconstruction Error
2	FCM	$51.5 \pm 0.6\%$ (m=1.2)	1.3474 ± 0.0043 (m=1.4)
	GK-FCM	$52.5 \pm 1.9\%$ (m=1.2)	1.3355 ± 0.0035 (m=1.4)
	KFCM (G)	$61.4 \pm 7.4\%$ (m=1.2, $\sigma^2 = 1$)	1.6868 ± 0.1058 (m=1.2, $\sigma^2 = 2$)
	MKFCM (G)	$100 \pm 0.0\%$ (m=1.4, $\sigma^2 = 0.5$)	1.3490 ± 0.0039 (m=1.4, $\sigma^2 = 100$)
	MKFCM (P)	$100 \pm 0.0\%$ (m=2, $\theta = 2$, $d = 4$)	1.3454 ± 0.0039 (m=1.4, $\theta = 15$, $d = 2$)
3	FCM	$62.5 \pm 2.7\%$ (m=2)	0.8664 ± 0.0037 (m=1.4)
	GK-FCM	$87.4 \pm 0.9\%$ (m=2)	1.0670 ± 0.0030 (m=2)
	KFCM (G)	$73.5 \pm 10.6\%$ (m=1.2 $\sigma^2 = 2$)	1.2520 ± 0.0948 (m=1.2, $\sigma^2 = 2$)
	MKFCM (G)	$100 \pm 0.0\%$ (m=1.4, $\sigma^2 = 0.5$)	0.8656 ± 0.0031 (m=1.4, $\sigma^2 = 100$)
	MKFCM (P)	$100 \pm 0.0\%$ (m=2.5, $\theta = 7$, $d = 8$)	0.8548 ± 0.0040 (m=1.4, $\theta = 25$, $d = 2$)
4	FCM	$99.9 \pm 0.1\%$ (m=1.7)	0.5866 ± 0.0018 (m=1.7)
	GK-FCM	$87.4 \pm 7.3\%$ (m=1.2)	0.9453 ± 0.1613 (m=1.7)
	KFCM (G)	$89.7 \pm 9.8\%$ (m=1.7, $\sigma^2 = 2$)	0.7690 ± 0.1204 (m=1.2, $\sigma^2 = 2$)
	MKFCM (G)	$100 \pm 0.0\%$ (m=1.2, $\sigma^2 = 0.5$)	0.5730 ± 0.0025 (m=1.4, $\sigma^2 = 4$)
	MKFCM (P)	$100 \pm 0.0\%$ (m=2, $\theta = 2$, $d = 8$)	0.5991 ± 0.0013 (m=1.4, $\theta = 50$, $d = 12$)

The average computation time on the Fuzzy XOR data set over 20 runs is given in table 3. The MKFCM algorithms require significantly more computation than FCM. The KFCM requires much less computation than the MFKCM algorithms.

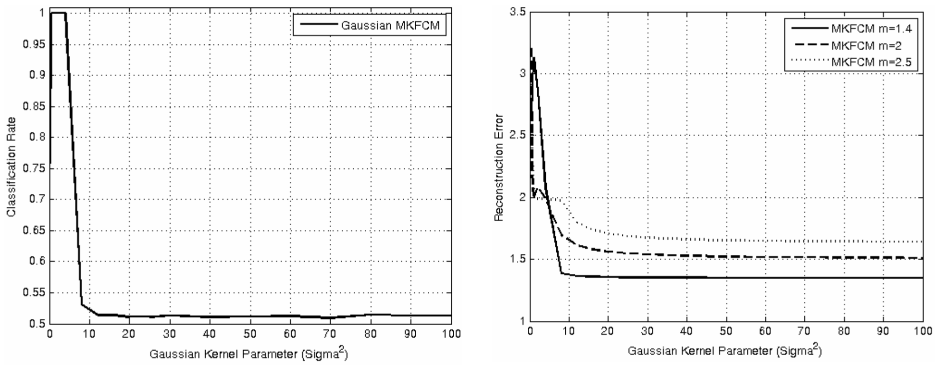

Fig. 1. Classification Rate (left) and Reconstruction Error (right) versus σ^2 for Gaussian MKFCM on Ring (c=2)

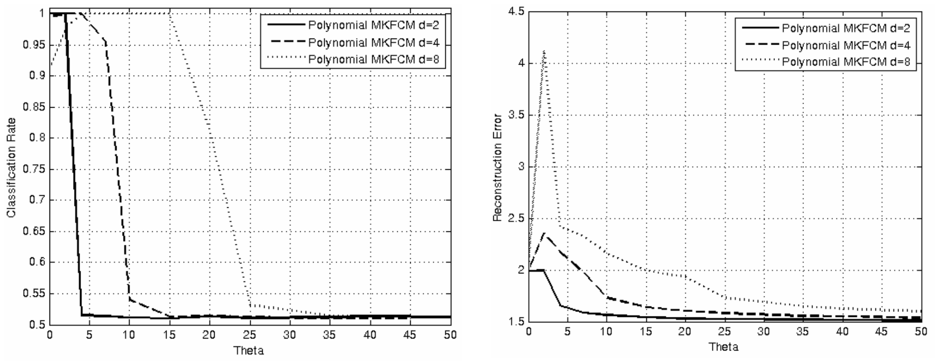

Fig. 2. Classification Rate (left) and Reconstruction Error (right) versus Polynomial Kernel Parameters for Polynomial MKFCM on Ring (c=2)

Table 3. Average Computation Time for Clustering Algorithms

Algorithm	FCM	GK-FCM	KFCM	MKFCM (G)	MKFCM (P)
Time (s)	1.36	3.43	3.21	32.93	31.68

The kernel-based algorithms do not appear to provide a magical solution to the problem of clustering and in particular with non-spherical shaped clusters. Although the kernel-based algorithms tend to perform much better for the ring cluster, there is a trade-off with the sensitivity of the kernel parameters to the clustering results.

3.3 UCI Machine Learning Data Sets

The iris, wine, liver and breast cancer UCI Machine Learning data sets [9] were included in the study. The attributes of the UCI data sets were normalized by subtracting the mean and dividing by the standard deviation. Overall standard FCM had the smallest reconstruction error and the MKFCM algorithm followed closely behind FCM. As for classification rate, the MKFCM algorithm out-performed only slightly better in most cases.

The classification rate for all clustering algorithms was fairly similar for the Iris data with the GK-FCM algorithm achieving the highest classification rate at 95.3%. The MKFCM algorithm fell slightly behind the rest with 85.6% and 88.7% for Gaussian and polynomial kernels respectively. The MKFCM algorithm performs fairly well for wine and breast cancer however there is only marginal improvement in the classification rate and the reconstruction error was similar to FCM. The GK-FCM algorithm performs fairly poorly for the wine data set with an average classification rate at 71.4% which is about 25% lower than the others. All the algorithms perform fairly similarly for the liver data set with classification rates around 60% and the highest classification rate belonging to MKFCM with the Gaussian kernel at 61.3%. The reconstruction error was the lowest for FCM at around 3.1 with MKFCM closely behind while KFCM was at 5.7.

The DELVE ring data set was used as a test data set for the MKFCM algorithm outlined in [16] in which the authors obtained 98.7% classification accuracy. Our implementation of the MKFCM algorithm based on [13,15,16] was run on the same DELVE ring data set achieving 98.6% accuracy using same parameters in [16] (m=2, $c=2, \sigma^2 = 42.25$). A kernel-based weighted FCM method in [12] obtained 96% classification on the Iris data set.

4 Conclusions

The kernel-based clustering algorithms – especially MKFCM – can cluster specific non-spherical clusters such as the ring cluster quite well outperforming FCM and GK-FCM; however overall the performance of the kernel-based methods is not very impressive due to similar or only slight increases in clustering classification rates compared to FCM. From the perspective of the reconstruction error, MKFCM often performed similar to that of FCM and KFCM. A major disadvantage of kernel-based algorithms is their sensitivity to the kernel parameters. In fact in some cases a change in the kernel parameters could reduce the classification rate by one-half and multiply the reconstruction error. Thus kernel-based fuzzy clustering requires tuning in order to achieve optimal performance. Nevertheless in most of the artificial and UCI machine learning test data sets run, the optimal performance of the kernel-based clustering algorithms is not that much of an improvement over FCM and GK-FCM.

Acknowledgments. Support from the Natural Sciences and Engineering Research Council of Canada (NSERC) and Canada Research Chair Program (W. Pedrycz) is gratefully acknowledged.

References

1. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum, New York (1981)
2. Chiang, J.H., Hao, P.Y.: A new kernel-based fuzzy clustering approach: support vector clustering with cell growing, IEEE Transactions of Fuzzy Systems, **11**(4) (2003) 518-527
3. Girolami, M.: Mercer kernel-based clustering in feature space. IEEE Transactions on Neural Networks. **10**(5) (1999) 1000-1017
4. Herbrich, R.: Learning Kernel Classifiers, MIT Press, Cambridge Massachusetts (2002).
5. Klawonn, F., Kruse, R.: Constructing a fuzzy controller from data. Fuzzy Sets and Systems, **85** (1997) 177-193
6. Krishnapuram, R., Kim, J.: A note on the Gustafson-Kessel and adaptive fuzzy clustering algorithms, IEEE Transactions on Fuzzy Systems, **7**(4) (1999) 453-461
7. Li, T., Ma, S., Ogihara, M.: Entropy-based criterion in categorical clustering. ACM Int. Conf. Proc. Series; Vol. 69 Proc. 21st int. conf. on Machine Learning, **69** (2004) 68-75
8. Muller, K.R., Mika, S., Ratsch, G., Tsuda, K., Scholkopf, B.: An introduction to kernel-based learning algorithms, IEEE Transactions on Neural Networks, **12**(2) (2001) 181-201
9. Newman, D.J., Hettich, S., Blake, C.L., Merz, C.J.: UCI Repository of machine learning databases (1998) http://www.ics.uci.edu/~mlearn/MLRepository.html (Accessed Feb 2007)
10. Pedrycz, W.: Knowledge-Based Clustering, J. Wiley, Hoboken, NJ (2005)
11. Scholkopf, B., Smola, A., Muller, K.R.: Nonlinear component analysis as a kernel eigenvalue problem, Neural Computation, **10** (1998) 1299-1319
12. Shen, H., Yang, J., Wang, S., Liu, X.: Attribute weighted mercer kernel based fuzzy clustering algorithm for general non-spherical datasets, Soft Computing, **10**(11) (2006) 1061-1073
13. Zeyu, L., Shiwei, T., Jing, X., Jun, J.: Modified FCM clustering based on kernel mapping, Proc. of Int. Society of Optical Engineering, Vol. 4554 (2001) 241-245
14. Zhang, D.Q., Chen, S.C.: Clustering incomplete data using Kernel-based Fuzzy c-means Algorithm, Neural Processing Letters, **18**(3) (2003) 155-162
15. Zhang, D., Chen, S.: Fuzzy Clustering using Kernel Method, Proc. Int. Conf. Control and Automation (2002) 123-127
16. Zhou, S., Gan, J.: Mercer kernel fuzzy c-means algorithm and prototypes of clusters, Proc. Conf. on Int. Data Engineering and Automated Learning (2004) 613-618