Fuzzy Optim Decis Making (2018) 17:159–176
https://doi.org/10.1007/s10700-017-9268-x

Fuzzy clustering using multiple Gaussian kernels with optimized-parameters

Issam Dagher[1]

Published online: 3 April 2017

Abstract In this paper, we propose a new kernel-based fuzzy clustering algorithm which tries to find the best clustering results using optimal parameters of each kernel in each cluster. It is known that data with nonlinear relationships can be separated using one of the kernel-based fuzzy clustering methods. Two common fuzzy clustering approaches are: clustering with a single kernel and clustering with multiple kernels. While clustering with a single kernel doesn't work well with "multiple-density" clusters, multiple kernel-based fuzzy clustering tries to find an optimal linear weighted combination of kernels with initial fixed (not necessarily the best) parameters. Our algorithm is an extension of the single kernel-based fuzzy c-means and the multiple kernel-based fuzzy clustering algorithms. In this algorithm, there is no need to give "good" parameters of each kernel and no need to give an initial "good" number of kernels. Every cluster will be characterized by a Gaussian kernel with optimal parameters. In order to show its effective clustering performance, we have compared it to other similar clustering algorithms using different databases and different clustering validity measures.

Keywords Fuzzy · Clustering · Kernel · FCM · Gaussian · Validity measure

1 Introduction

Clustering is the process of separating the data into different groups on the basis of some similarity measure; data belonging to the same cluster should be similar to each other while data belonging to different clusters should not. Mainly two cluster-

✉ Issam Dagher
dagheri@balamand.edu.lb

[1] Department of Computer Engineering, University of Balamand, P.O. Box 100, Elkoura, Lebanon

ing approaches exist: hard clustering and fuzzy clustering. In hard clustering, data is divided into distinct clusters, where each data point can only belong to exactly one cluster (Likas et al. 2003; Pelleg and Moore 2000; Kanungo et al. 2002). However, when clusters overlap, objects should belong to different clusters with different degrees. In this case a fuzzy clustering is more appropriate. Data points can potentially belong to multiple clusters. Membership grades are assigned to each of the data points. These membership grades indicate the degree to which data points belong to each cluster. One of the most widely used fuzzy clustering algorithms is the fuzzy c-means (FCM) algorithm (Bezdek 1981; Hoppner and Klawonn 1999; Gustafson and Kessel 1979; Babuska et al. 2002; Dunn 1973; Gath and Geva 1989). This algorithm works well for spherical-like clusters. When dealing with non-spherical shape and much overlapped data, kernel methods are best suited. They map data points from the input space to a high dimensional feature space. Thus the original non-linearly separable data structure in the input space becomes linearly separable in the feature space. There are two major variations of the FCM: Kernel-FCM (KFCM) and Multiple-Kernel-FCM (MKFCM). KFCM (Schölkopf and Smola 2002; Xie et al. 2003) transforms the data into a high-dimensional feature space and then performs clustering. It uses a **single kernel** which doesn't fit with multiple-density clusters. Instead of using a single fixed kernel, multiple kernels may be used. MKFCM (Baili and Frigui 2012, 2011) tries to find an optimal linear weighted combination of kernels with **initial fixed (not necessarily the best) parameters**. The disadvantage of that approach is twofold: Giving the number of kernels M and giving "good" parameters of each kernel.

Our **motivation** in this paper is to do an extension of the KFCM and the MKFCM algorithms. This is done by achieving the following:

- Characterizing every cluster by a different Gaussian kernel (extension of the KFCM).
- Trying to find the optimal parameters of each kernel in each cluster without the need to give "good" parameters of each kernel and no need to give an initial "good" number of kernels (extension of the MKFCM).

It should be noted that the originality and the novelty of our work stem from the fact that we achieved a new method that achieves better clustering. Our contribution and innovation can be summarized by an extended kernel-based fuzzy clustering algorithm which is more effective than previous techniques. And the benefits of this work are found in numerous applications which require clustering algorithms.

We have used the following**nomenclature**: FCM: Fuzzy c-means, KFCM: kernel-Fuzzy c-means, MKFCM: Multiple-Kernel-Fuzzy c-means, and OKFCM: our new algorithm.

The rest of the paper is organized as follows. In Sect. 2 we review the c-means, the FCM, and the MKFCM algorithms. In Sect. 3 we present our new algorithm. To validate our approach, different validation measures were shown in Sect. 4. Section 5 shows the databases used for comparisons and the results obtained. And we conclude in Sect. 6.

2 Related works

2.1 C-means clustering

Given the data $X_1, X_2, \ldots, X_N$, where each datum is a n-dimensional real vector, then c-means clustering (Likas et al. 2003; Pelleg and Moore 2000; Kanungo et al. 2002) aims to partition the N data into C clusters G_1, G_2, ..., G_C in order to minimize the following function:

$$J\left(V\right) = \sum_{i=1}^{C} \sum_{j \in G_i} \left\| X_j - v_i \right\|^2 \tag{1}$$

Here v_i is the center of the cluster G_i, which consists of N_i data. It is given by:

$$v_i = \frac{\sum_{j=1}^{N_i} X_j}{N_i}, \quad 1 \le i \le C, \quad \sum_{i=1}^{C} N_i = N \tag{2}$$

Each cluster G_i will be characterized by the prototype v_i (real center).

2.2 FCM clustering

Let U be the fuzzy membership matrix (Bezdek 1981; Hoppner and Klawonn 1999) where $\mu_{ik} \in [0, 1]$, $1 \le i \le c$, $1 \le k \le N$ is the degree of membership of data X_k to the cluster with prototype v_i.

$$U_{c \times n} = \begin{bmatrix} \mu_{11} & \mu_{12} & \ldots & \mu_{1N} \\ \mu_{21} & \ldots & \ldots & \mu_{2N} \\ \vdots & \vdots & \vdots & \vdots \\ \mu_{c1} & \ldots & \ldots & \mu_{cN} \end{bmatrix} \tag{3}$$

Dunn defined a fuzzy objective function:

$$J_D\left(U, V\right) = \sum_{i=1}^{c} \sum_{j=1}^{N} \mu_{ij}^2 \left\| X_j - v_i \right\|^2 \tag{4}$$

Then, (Bezdek 1981) extended it to:

$$J_m\left(U, V; X\right) = \sum_{i=1}^{c} \sum_{j=1}^{N} \mu_{ij}^m \left\| X_j - v_i \right\|^2, \quad 1 \le m < \infty \tag{5}$$

For getting the minimum (U, V) and using the following Lagrangian objective function:

$$\min\left\{J_m\left(U,V,\lambda;X\right)=\sum_{i=1}^{c}\sum_{j=1}^{N}\mu_{ij}^{m}\left\|X_j-v_i\right\|^2+\sum_{k=1}^{N}\lambda_k\sum_{i=1}^{c}\left(\mu_{ik}-1\right)\right\} \quad (6)$$

By doing $\left(\frac{\partial J_m}{\partial v_i}=0 \quad \frac{\partial J_m}{\partial \mu_{ij}}=0 \quad \frac{\partial J_m}{\partial \lambda_k}=0\right)$, we get:

$$v_i=\frac{\sum_{j=1}^{N}\left(\mu_{ij}\right)^m X_j}{\sum_{j=1}^{N}\left(\mu_{ij}\right)^m}, \quad 1\le i\le c \quad (7)$$

And

$$\mu_{ij}=\left(\sum_{k=1}^{c}\left(\left\|X_j-v_i\right\|^2\Big/\left\|X_j-v_k\right\|^2\right)^{1/m-1}\right)^{-1}, 1\le i\le c, 1\le j\le N \quad (8)$$

Each cluster i will be characterized by the prototype v_i (real center) and the fuzzy membership vector μ_{ij}.

2.3 Kernel FCM

It uses the following objective (Xie et al. 2003) function:

$$J_m\left(U,V;X\right)=\sum_{i=1}^{c}\sum_{j=1}^{N}\mu_{ij}^{m}\left\|\phi(X_j)-\phi(v_i)\right\|^2, \quad 1\le m\le\infty \quad (9)$$

where $\phi(X)$ is a transformation from the nonlinear data space to a higher dimensional space called feature space (Eschrich et al. 2003).

In that space, every dot product is replaced by a non linear kernel function K (kernel trick) (Eschrich et al. 2003).

$$\phi^T(X1)\cdot\phi(X2)=K(X1,X2) \quad (10)$$

In the feature space, the kernel function measures the similarity of vector X_1 and the vector X_2. The Gaussian kernel that we have used in this paper is given by:

$$K(X1,X2)=\exp\left(-\sum_{i=1}^{N}\frac{1}{N}\frac{(X_{1i}-X_{2i})^2}{2\sigma_i^2}\right) \quad (11)$$

where we assume we have N n-dimensional data.

It should be noted also that the Euclidian distance in the feature space is given by:

$$\left\|\phi(X_j)-\phi(v_i)\right\|^2=k(X_j,X_j)+k(v_i,v_i)-2k(X_j,v_i) \tag{12}$$

And the update formulas for the kernel FCM are given by:

$$v_i=\frac{\sum_{j=1}^{N}\mu_{ij}^{m}k(X_j,v_i)X_j}{\sum_{j=1}^{n}\mu_{ij}^{m}k(X_j,v_i)} \tag{13}$$

$$\mu_{ij}=\left(\sum_{k=1}^{c}\left(\frac{1-k(X_j,v_i)}{1-k(X_j,v_k)}\right)^{\frac{1}{m-1}}\right)^{-1}. \tag{14}$$

2.4 Multiple kernel-based clustering

In multiple kernel-based clustering (Baili and Frigui 2011), each cluster is characterized by a linear combination of M kernels with known parameters. The similarity of the data X to the center v is given by:

$$k(x,v)=\sum_{i=1}^{M}w_iK_i(X,v) \tag{15}$$

Every kernel is multiplied by a weight $w_i\in\{0,1\}$ and $\sum_{i=1}^{M}w_i=1$ reflecting the importance of that kernel. The disadvantage of that approach is twofold:

- Giving the number of kernels M.
- And giving "good" parameters of each kernel.

It uses the following objective function:

$$J_m\left(U,V;X\right)=\sum_{i=1}^{N}\sum_{c=1}^{C}\mu_{ij}^{m}(\phi(X_i)-v_c)^T\left(\phi(X_i)-v_c\right)=\sum_{i=1}^{N}\sum_{c=1}^{C}\mu_{ij}^{m}D_{ic}^{2} \tag{16}$$

where:

$$\begin{aligned}D_{ic}^{2}=&\sum_{k=1}^{M}w_k^2K_k(X_i,X_i)-2\sum_{j=1}^{N}\sum_{k=1}^{M}\mu_{jc}w_k^2K_k(X_i,X_j)\\&+\sum_{j=1}^{N}\sum_{i=1}^{N}\sum_{k=1}^{M}\mu_{jc}\mu_{ic}w_k^2K_k(X_j,X_i)\end{aligned} \tag{17}$$

And

$$\mu_{ic} = \frac{1}{\sum_{k=1}^{c} \left(\frac{D_{ic}^2}{D_{ik}}\right)^{\frac{1}{m-1}}} \tag{18}$$

The optimization problem can be solved now using:

$$\frac{dJ_m}{dw} = 0. \tag{19}$$

3 Fuzzy clustering using multiple Gaussian kernels with optimized-parameters

3.1 Problem formulation

In MKFCM, a weighted linear combination of kernels with given variances is used. This weighted linear combination will approximate the true kernel. Every cluster will be characterized by a set of weights which approximates best the true kernel.

$$\sum_{i=1}^{M} w_i k_i(X, v_i) \approx \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{(x - v_i)^2}{2\sigma_i^2}\right) \tag{20}$$

It should be noted that 4 kernels are chosen and the weight of each kernel will be the output of the MKFCM algorithm.

Our algorithm (OKFCM) tries to characterize every cluster by a different optimal Gaussian kernel.

3.2 Optimal Gaussian kernel and the EM approach

In our algorithm, clustering in the new high-dimensional (feature) space is based on minimizing the following objective function:

$$J_m\,(U, V; X) = \sum_{i=1}^{N}\sum_{c=1}^{C} \mu_{ij}^m (\phi(X_i) - v_c)^T (\phi(X_i) - v_c) = \sum_{i=1}^{N}\sum_{c=1}^{C} \mu_{ij}^m D_{ic}^2$$

The distance D_{ic}^2 of the data X_i to the center v_k in that space is given using by:

$$D_{ic}^2 = (\phi(X_i) - v_c)^T (\phi(X_i) - v_c)$$

$$\text{where} v_c = \frac{\sum_{i=1}^{N} \mu_{ic}^m \phi(X_i)}{\sum_{i=1}^{N} \mu_{ic}^m} \text{ is the center in the feature space.}$$

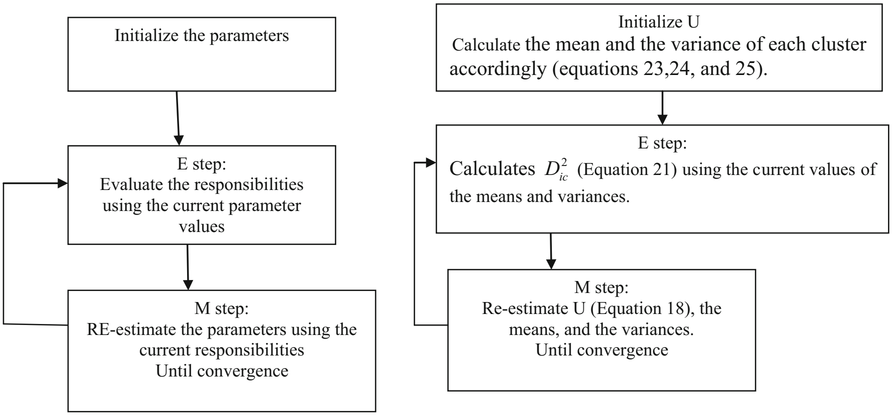
Initialize the parameters
E step:
Evaluate the responsibilities
using the current parameter
values
M step:
RE-estimate the parameters using the
current responsibilities
Until convergence
Initialize U
Calculate the mean and the variance of each cluster
accordingly (equations 23,24, and 25).
E step:
Calculates D^2_{ic} (Equation 21) using the current values of
the means and variances.
M step:
Re-estimate U (Equation 18), the
means, and the variances.
Until convergence

Fig. 1 Flow chart of our algorithm compared to the EM method

Which Can be written as:

$$D^2_{ic} = K(X_i, X_i) - 2\sum_{j=1}^{N} \mu_{jc} K(X_i, X_j) + \sum_{j=1}^{N}\sum_{i=1}^{N} \mu_{jc}\mu_{ic} K(X_j, X_i) \tag{21}$$

We have used the Gaussian kernel which is given by:

$$K(x, v_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp\left(-\frac{(x - v_k)^2}{2\sigma_k^2}\right)$$

The above equation shows that similarity in the feature space depends on the variance of the data in the original space. Our approach in determining the optimal kernel is very similar to the Expectation-Maximization (EM) method given by the following flowchart.

It should be noted that our algorithm as the other clustering algorithms which are similar to the EM method are not guaranteed to reach the global minimum (see Fig. 1). They might reach Local minimum or Saddle point (Zhou et al. 2013).

3.3 Methodology

Our method is described by the following algorithm:

- Step 1: Given the N n-dimensional data $X_1, X_2, \ldots, X_N$and small positive tolerance ε.
- Step 2: Initialize the membership function matrix which indicates in the feature space how much each datum is close to each center.

$$\text{Initialize } U_{c\times N} = \begin{bmatrix} \mu_{11} & \mu_{12} & \cdots & \mu_{1N} \\ \mu_{21} & \cdots & \cdots & \mu_{2N} \\ \vdots & \vdots & \vdots & \vdots \\ \mu_{c1} & \cdots & \cdots & \mu_{cN} \end{bmatrix} \text{ such that } \sum_{i=1}^{c} \mu_{ik} = 1 \; k = 1 \ldots N$$

- Step 3: The mean and the variance of each cluster are given by the following formulas:

$$v_i = \frac{\sum_{j=1}^{N} (\mu_{ij})^m x_j}{\sum_{j=1}^{n} (\mu_{ij})^m} \quad 1 \leq i \leq c \tag{22}$$

$$\delta_i^2 = \frac{\sum_{j=1}^{N} (\mu_{ij})^m (x_j - v_i)^2}{\sum_{j=1}^{n} (\mu_{ij})^m} \quad 1 \leq i \leq c \tag{23}$$

  Step 3: emphasizes that every cluster c is characterized by the mean v_c and the variance δ_c^2.

- In Step 4, the variance of each data is given by the following formula:

$$\sigma_i^2 = \sum_{j=1}^{c} (\mu_{ji})^m \delta_i^2 \quad i = 1, ..., N \tag{24}$$

  Now every data is characterized by the n-dimensional variance vector.

$$\{X_i\} \leftrightarrow \sigma_i^2 = \begin{bmatrix} \sigma_{i1}^2 \\ \vdots \\ \sigma_{in}^2 \end{bmatrix} \quad i = 1, ..., N$$

  It should be noted that data from the same class c will have almost similar variance which is δ_c^2.

- Step 5: Calculates the similarity between all the data using the kernel matrix K:

```
$for\ i = 1 : N$
$for\ j = i : N$
$K(i, j) = \exp\left(-\sum_{i=1}^{n} \frac{1}{N} \frac{(X_{1i} - X_{2i})^2}{2\sigma_i^2}\right)$
$K(j, i) = \exp\left(-\sum_{i=1}^{n} \frac{1}{N} \frac{(X_{1i} - X_{2i})^2}{2\sigma_j^2}\right)$
$end$
$end$
$K = \frac{1}{2}(K + K^T);$ % Make the matrix K symmetric
```

  In order for the matrix K to be a similarity matrix, it should be a symmetric matrix $K(X_i, X_j) = K(X_j, X_i)$ for all $i, j = 1, \ldots, N$.

  There are 2 cases:
  Case1: X_i and X_j belong to the same cluster and eventually have approximately the same variances:

$$\sigma_i^2 \approx \sigma_j^2 \Rightarrow K(X_i, X_j) \approx K(X_j, X_i)$$
$$\Rightarrow \frac{1}{2}(K(X_i, X_j) + K(X_j, X_i)) \approx K(X_j, X_i)$$

Case2: X_i and X_j belong to different clusters and eventually have different variances:

$$\sigma_i^2 \neq \sigma_j^2 \Rightarrow K(X_i, X_j) \neq K(X_j, X_i) \Rightarrow \text{do averaging}$$
$$K(X_i, X_j) = K(X_j, X_i) = \frac{1}{2}(K(X_i, X_j) + K(X_j, X_i))$$

- Step 6: Calculates **in the feature space** the distance of each data to each center using the similarity matrix found in Step 5 and updates the membership matrix according to those distances.

$$\begin{aligned}&\textit{for all i and c}\\ &D_{ic}^2 = K(X_i, X_i) - 2\sum_{j=1}^{N} \mu_{jc} K(X_i, X_j) + \sum_{j=1}^{N}\sum_{i=1}^{N} \mu_{jc}\mu_{ic} K(X_j, X_i)\\ &\mu_{ic} = \frac{1}{\sum_{k=1}^{c}\left(\frac{D_{ic}^2}{D_{ik}}\right)^{\frac{1}{m-1}}}\end{aligned}$$

- Step7: Stopping Criterion. If the membership matrix doesn't change (all the data are remaining in their assigned clusters) then stop else go to Step 3 for another epoch.

$$||U_{new} - U_{old}|| < \varepsilon$$

4 Validation measures

In order to show the effectiveness of our clustering algorithm, we have used different validity measures (Bensaid et al. 1996; Xie and Beni 1991; Eschrich et al. 2003; Balasko et al.). The Partition Coefficient (PC) measures the amount of "overlapping" between clusters. The Classification Entropy (CE) measures the fuzziness of the cluster partitions. The Partition Index (SC) denotes the ratio of the sum of compactness and separation of the clusters. And the Separation Index (S) uses a minimum-distance separation for partition validity.

The validity measures used have the following formulas:

- *Partition coefficient (PC)*: $PC(c) = \frac{1}{N}\sum_{i=1}^{c}\sum_{j=1}^{N} \mu_{ij}^2$
- *Classification entropy (CE)*: $CE(c) = \frac{-1}{N}\sum_{i=1}^{N}\sum_{j=1}^{c} \mu_{ij}\log(\mu_{ij})$
- *Partition index (SC)*: $SC(c) = \sum_{i=1}^{c} \frac{\sum_{j=1}^{N} \mu_{ij}^2 (x_i - v_j)^2}{N_i \sum_{k=1}^{c} (v_k - v_i)^2}$
- *Separation index (S)*: $S(c) = \frac{\sum_{i=1}^{c}\sum_{j=1}^{N} \mu_{ij}^2 (x_j - v_i)^2}{N(\min_{i,k}(v_k - v_i)^2)}$

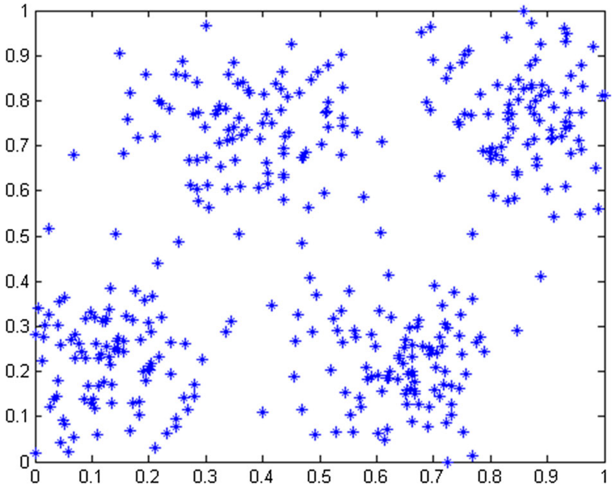

Fig. 2 The 4cluster dataset

It should be noted that the above measures can be applied to fuzzy clustering algorithms due to the presence of the membership values μ_{ij}.

In all the experiments for the MKFCM, we have followed (Baili and Frigui 2011) in choosing the 4 kernels. This is done by calculating:

$$D = \left(\sum_{k=1}^{N} (\max(X_k) - \min(X_k))^2 \right)^{0.5}$$

$$Then \quad \sigma_1 = 0.1{*}D, \sigma_2 = 0.2{*}D, \sigma_3 = 0.3{*}D, \text{ and } \sigma_4 = 0.4{*}D$$

It should be noted that the weight of each kernel will be the output of the MKFCM algorithm.

For the KFCM we have repeated the algorithm 4 times. Each time the KFCM is initialized with one of the 4 variances. We have shown the results of the KFCM which gave the best results.

Figures 2, 3 show the 2 artificial datasets (4clusters dataset and the Spiral dataset) that we have used in our simulations.

It should be noted that **higher values of the PC and lowest values of SC, S, and CE** imply better clustering performance.

Figures 4, 5, 6, and 7 show the PC, S, CE, and SC validation results vs. the number of clusters for the 4cluster dataset. We have applied the FCM, KFCM, MKFCM, and the OKFCM with different number of clusters (2, 3, … 10). And we have plotted the obtained validation results.

We have repeated the same procedure for the Spiral dataset. The validation results vs. The number of clusters was plotted in Figs. 8, 9, 10, and 11.

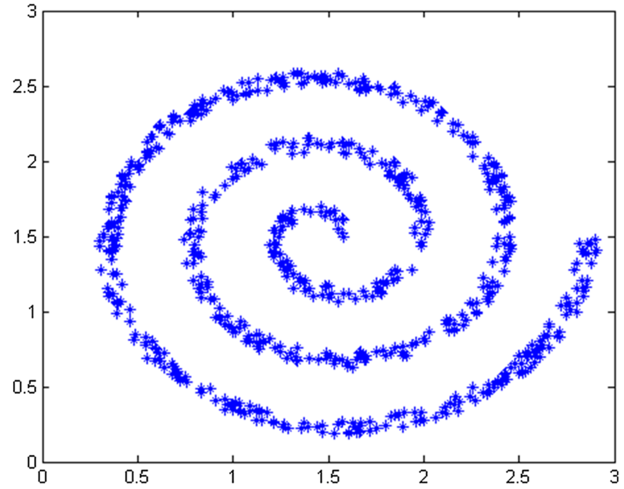

Fig. 3 The 2 Spiral dataset

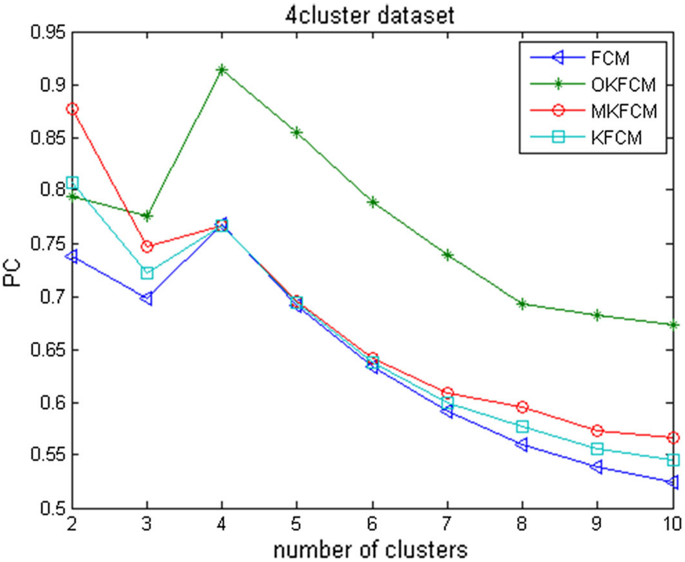

Fig. 4 The PC validation results for different number of clusters for the 4cluster dataset

5 UCI-datasets simulation results

We have also compared the performance of the 4 algorithms on real datasets. These datasets were chosen from the UCI repository [19]. They are summarized as follows: *Iris database* The dataset contains 3 classes of 50 instances each. The number of features for each instance are the sepal length, the sepal width, the petal length, and the petal width.

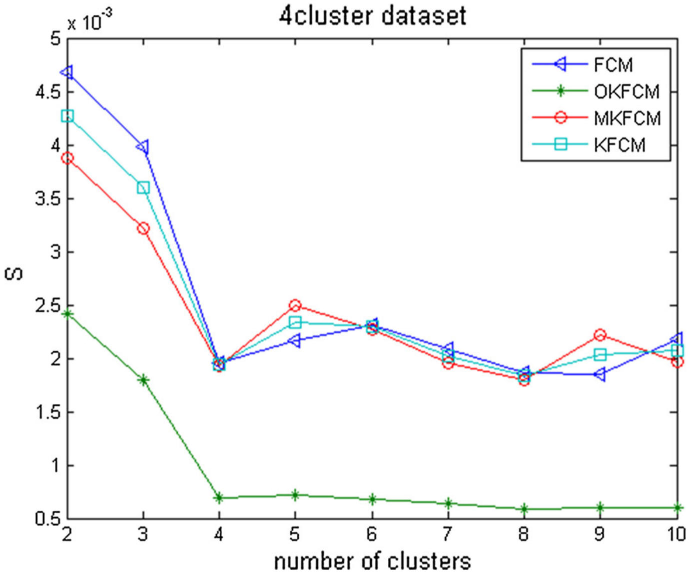

Fig. 5 The S validation results for different number of clusters for the 4cluster dataset

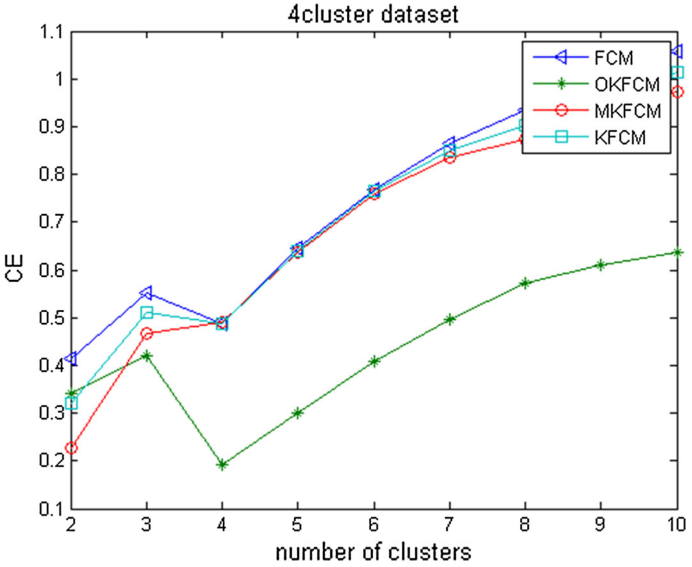

Fig. 6 The CE validation results for different number of clusters for the 4cluster dataset

Wine database The data contains 3 classes. The number of features is 13. The number of instances is 178 belonging to class1, 59 belonging to class2, and 48 belonging to class3.

Sonar database These data contain 111 instances of patterns obtained by bouncing off sonar signals off a metal cylinder at various angles and under various conditions.

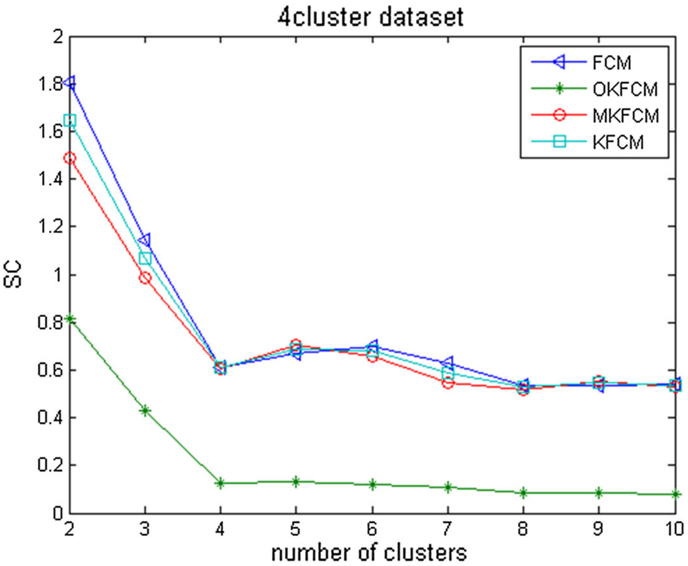

Fig. 7 The SC validation results for different number of clusters for the 4cluster dataset

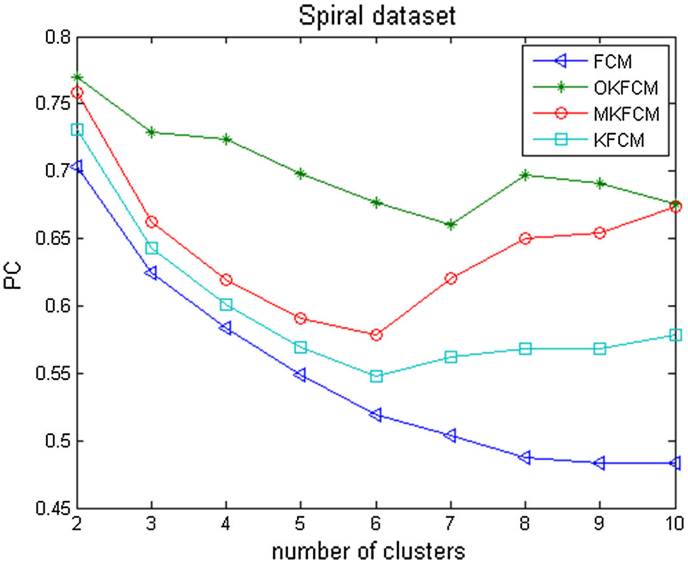

Fig. 8 The PC validation results for different number of clusters for the Spiral dataset

It also contains 97 instances of patterns obtained from rocks under similar conditions. Each input pattern is a set of 60 features.

Diabetes database The database consists of 500 instances of patients that did not have the disease and 268 instances of patients that have the disease. The number of features is 8.

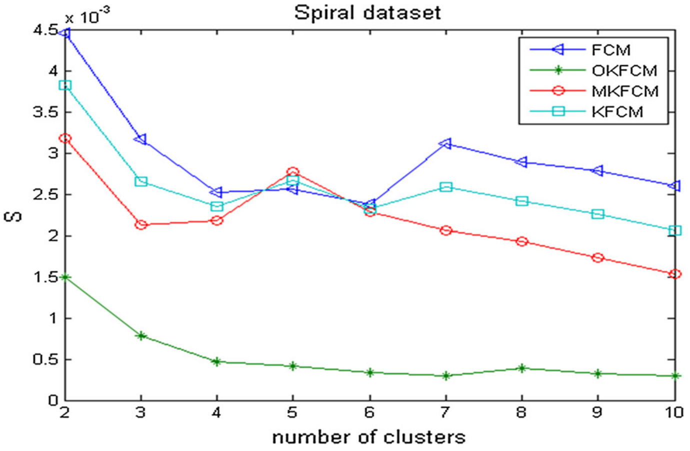

Fig. 9 The S validation results for different number of clusters for the Spiral dataset

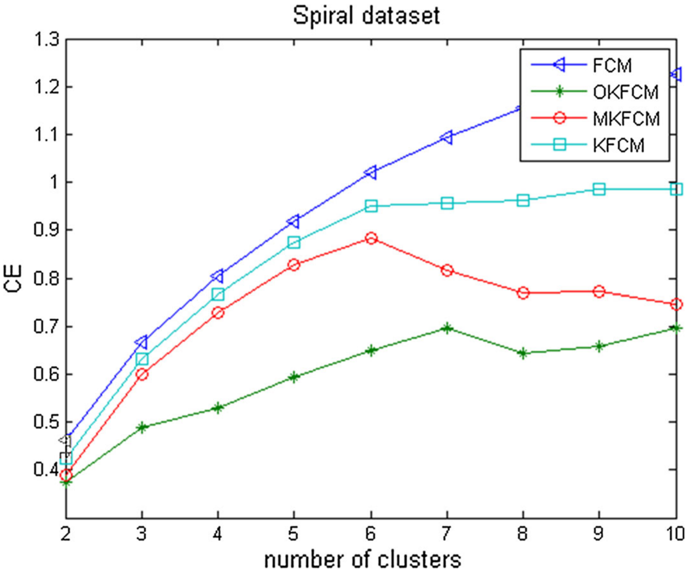

Fig. 10 The CE validation results for different number of clusters for the Spiral dataset

Breast database The database contains 458 instances of patients that had benign cancer, and 241 instances of patients that had malignant cancer. The number of features is 9. Table 1 shows the validation results for the FCM, the KFCM, the MKFCM, and the OKFCM. The number of clusters C is the number of classes of each database.

Another typical validation measure is the **Normalized Mutual Information (NMI)** [20]. Given the knowledge of the ground truth class assignments and the clustering algorithm assignments of the same samples, the **NMI** is a function that measures the

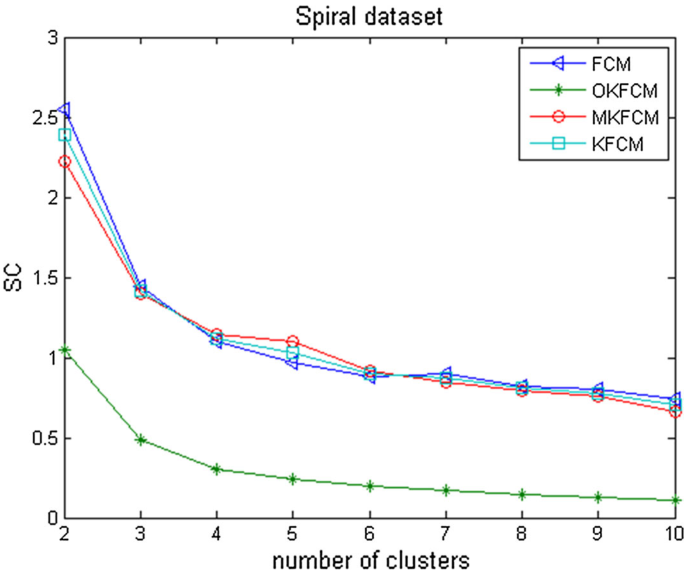

Fig. 11 The SC validation results for different number of clusters for the Spiral dataset

agreement of the two assignments. **Bigger values of the NMI** imply better agreements. The following table shows the **NMI** values for different databases, and for different percentages of training data. We have compared the performances of the c-means, the FCM, the KFCM, the MKFCM, and the OKFCM on the chosen datasets.

5.1 Discussion

Table 1 shows that the OKFCM algorithm gave the highest values (better performance) compared to the other algorithms for the PC and the lowest values (better performance) for the SC, S and CE validity measures.

Table 2 shows the**NMI** validation results for different databases and different percentages of training data. We have used 90 %, 80%, 70%, 60%, 50%, and 40% training data. The OKFCM gave the highest value (better performance) for the **NMI** validation results.

6 Conclusions

In this paper, fuzzy clustering using Multiple-Gaussian-Kernels-With-Optimized-Parameters (OKFCM) is introduced. The c-means is best fitted for non-overlapped clusters. The FCM works well for spherical-like clusters. The KFCM uses a single kernel which doesn't well work with multiple-density clusters. The MKFCM tries to find an optimal linear weighted combination of kernels with initial fixed (not necessarily the best) parameters. Our algorithm is an extension of the KFCM and the

Table 1 UCI database validity results

Databases	PC	CE	SC	S	#Clusters
Iris					
FCM	0.78	0.57	0.67	0.0089	3
KFCM	0.82	0.44	0.55	0.0067	
MKFCM	0.86	0.41	0.53	0.0033	
OKFCM	0.9	0.38	0.51	0.002	
Wine					
FCM	0.76	0.55	0.66	0.0085	3
KFCM	0.81	0.44	0.56	0.0076	
MKFCM	0.87	0.39	0.53	0.0039	
OKFCM	0.89	0.36	0.5	0.0034	
Diabetes					
FCM	0.82	0.44	0.58	0.0077	2
KFCM	0.84	0.31	0.48	0.0065	
MKFCM	0.86	0.35	0.45	0.004	
OKFCM	0.88	0.32	0.41	0.0039	
Breast					
FCM	0.85	0.39	0.55	0.0073	2
KFCM	0.86	0.35	0.43	0.0048	
MKFCM	0.88	0.3	0.4	0.0039	
OKFCM	0.9	0.27	0.37	0.0036	
Sonar					
FCM	0.81	0.45	0.57	0.0075	2
KFCM	0.83	0.37	0.48	0.0042	
MKFCM	0.87	0.34	0.44	0.0039	
OKFCM	0.89	0.31	0.4	0.0035	

MKFCM. It uses multiple kernels. There is no need to give "good" initial parameters of each kernel and initial "good" number of kernels. It is *robust and flexible* (Vasant 2013). Every cluster will be characterized by a different optimal Gaussian kernel. Its *effectiveness and efficiency* can be shown by comparing our algorithm to the c-means, FCM, KFCM, and MKFCM algorithms. It is well known that better clustering outputs, the better *stable and reliable* it is. We have used the following validity measures: The Partition Coeffcient (PC), the Classification Entropy (CE), the Partition Index (SC), the Separation Index (S), and the Normalized Mutual Information (NMI). Two artificial datasets (4clusters dataset and the Spiral dataset) and 5 databases obtained from the UCI repository (Iris, Wine, Diabetes, Breast, and Sonar) were used in our simulations.

Table 2 NMI validity results for different databases, and for different training percentages of data

% Data	90%	80%	70%	60%	50%	40%
Iris						
Kmean	0.7420	0.7329	0.721932	0.721932	0.760877	0.7194
FCM	0.7420	0.7329	0.728344	0.728344	0.760877	0.7194
KFCM	0.81325	0.88654	0.76432	0.76555	0.74543	0.72321
MKFCM	0.8534	0.9041	0.805277	0.805277	0.777473	0.7565
OKFCM	0.8796	0.9237	0.8675	0.8654	0.864186	0.8567
Wine						
Kmean	0.853666	0.787503	0.838802	0.809670	0.877179	0.797017
FCM	0.857443	0.787503	0.838802	0.809670	0.877179	0.763332
KFCM	0.812116	0.720754	0.813421	0.824311	0.854234	0.724315
MKFCM	0.852508	0.760174	0.833144	0.841316	0.867406	0.787435
OKFCM	0.865877	0.782211	0.835431	0.860145	0.865189	0.791774
Diabetes						
Kmean	0.050703	0.025024	0.0463	0.052077	0.067244	0.047269
FCM	0.048843	0.026885	0.050766	0.056876	0.0732	0.042926
KFCM	0.009876	0.015431	0.016541	0.019876	0.02431	0.004591
OKFCM	0.016865	0.020307	0.020404	0.020369	0.0388	0.007618
MKFCM	0.060185	0.042111	0.027863	0.057657	0.0597	0.023043
Breast						
Kmean	0.763804	0.753012	0.751245	0.714159	0.727149	0.760439
FCM	0.763804	0.753012	0.751245	0.703572	0.727149	0.760439
KFCM	0.709881	0.726709	0.688811	0.629889	0.724319	0.706511
MKFCM	0.722747	0.761021	0.702268	0.654843	0.744905	0.736875
OKFCM	0.787766	0.783211	0.769811	0.768785	0.755432	0.780134
Sonar						
Kmean	0.007451	0.012186	0.022278	0.000034	0.033213	0.007473
FCM	0.007708	0.010107	0.019145	0.003523	0.033213	0.007473
KFCM	0.007911	0.009011	0.027651	0.007321	0.032131	0.015432
MKFCM	0.016585	0.009516	0.032515	0.008936	0.043647	0.016046
OKFCM	0.043221	0.027651	0.041279	0.051234	0.051213	0.032215

The OKFCM algorithm gave the highest values (better performance) for the PC and the NMI and the lowest values (better performance) for **SC, S,** and **CE** validity measures.

Future work could include the application of this new algorithm on different real world problems like image segmentation and image compression problems.

Finally, I thank the reviewers for comments that greatly improved the manuscript.

References

Babuska, R., van der Veen, P. J., Kaymak, U. (2002). Improved covariance estimation for Gustafson–Kessel clustering. In *IEEE International Conference on Fuzzy Systems* (pp. 1081–1085).

Baili, N., & Frigui, H. (2011). Fuzzy clustering with multiple kernels. In *IEEE International Conference on Fuzzy Systems* (pp. 490–496).

Baili, N., & Frigui, H. (2012). Fuzzy clustering with multiple kernels in feature space. In *IEEE International Conference on Fuzzy Systems* (pp. 1–8).

Balasko, B., Abonyi, J., & Feil, B. *Fuzzy clustering and data analysis toolbox*. http://www.fmt.vein.hu/softcomp/fclusttoolbox/.

Bensaid, A. M., Hall, L. O., Bezdek, J. C., Clarke, L. P., Silbiger, M. L., Arrington, J. A., et al. (1996). Validity-guided (Re)Clustering with applications to image segmentation. *IEEE Transactions on Fuzzy Systems*, *4*, 112–123.

Bezdek, J. C. (1981). *Pattern recognition with fuzzy objective function algorithms*. Berlin: plenum press.

Dunn, J. C. (1973). A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, *3*, 32–57.

Eschrich, S., Ke, J., Hall, L., & Goldgof, D. (2003). Fast accurate fuzzy clustering through data reduction. *IEEE Transactions on Fuzzy Systems*, *11*(2), 262–270.

Evaluation of clustering. http://nlp.stanford.edu/IRbook/html/htmledition/evaluation-of-clustering-1.html.

Gath, I., & Geva, A. B. (1989). Unsupervised optimal fuzzy clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *7*, 773–781.

Gustafson, D. E., & Kessel, W. C. (1979). Fuzzy clustering with fuzzy covariance matrix. In *Proceedings of the IEEE CDC, San Diego* (pp. 761–766).

Hoppner, F., Klawonn, R., Kruse, R., & Runkler, T. (1999). *Fuzzy cluster analysis*. Chichester: Wiley.

Kanungo, T., Mount, D. M., Netanyahu, N. S., Piatko, C., Silverman, R., & Wu, A. Y. (2002). An Efficient k-means clustering algorithm: Analysis and implementation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*(7), 881–892.

Likas, A., Vlassis, N., & Verbeek, J. J. (2003). The global k-means clustering algorithm. *Pattern Recognition*, *36*, 451–461.

Pelleg, D., & Moore, A. (2000). X-means: Extending k-means with efficient estimation of the number of clusters. In *Proceedings of the Seventeenth International Conference on Machine Learning, Palo Alto, CA*.

Pelleg, D., & Moore, A.: *Accelerating exact k-means algorithms withgeometric reasoning*.(Technical Report CMU-CS-00105), Carnegie Mellon University, Pittsburgh, PA.

Schölkopf, B., & Smola, A. J. (2002). *Learning with Kernels*. Cambridge, MA: MIT Press.

UCI Machine Learning Repository. http://archive.ics.uci.edu/ml/.

Vasant, P. M. (2013). *Handbook of research on novel soft computing intelligent algorithms: Theory and practical applications*. Hershey: IGI Global.

Vasant, P. M. (2014). *Handbook of research on artificial intelligence techniques and algorithms*. Hershey: IGI Global.

Vasant, P. M., Weber, G.-W., & Dieu, V. N. (2015). Classical and hybrid optimization approaches and their applications in engineering and economics. *Mathematicals Problems in Engineering*, Hindawi, Vol. (2015).

Xie, X. L., & Beni, G. A. (1991). Validity measure for fuzzy clustering. *EEE Transactions on Pattern Analysis and Machine Intelligence*, *3*(8), 841–846.

Xie, W. X., Wu, Z. D., & Yu, J. P. (2003) Fuzzy c-means clustering algorithm based on kernel method. *Computational Intelligence and Multimedia Applications*.

Zhou, J., Cao, L., & Yang, N. (2013). On the convergence of some possibilistic clustering algorithms. *Fuzzy Optimization and Decision Making*, *12*, 415–432.