



KNCM: Kernel Neutrosophic c-Means Clustering



Yaman Akbulut^a, Abdulkadir Şengür^a, Yanhui Guo^b, Kemal Polat^{c,*}

^a Firat University, Technology Faculty, Electrical and Electronics Engineering Dept., Elazig, Turkey

^b University of Illinois at Springfield, Department of Computer Science, Springfield, Illinois, USA

^c Abant İzzet Baysal University, Engineering Faculty, Electrical and Electronics Engineering Dept., Bolu, Turkey

ARTICLE INFO

Article history:

Received 11 June 2016

Received in revised form

30 September 2016

Accepted 3 October 2016

Available online 8 October 2016

Keywords:

Data clustering

Fuzzy clustering

Neutrosophic c-means

Kernel function

ABSTRACT

Data clustering is an important step in data mining and machine learning. It is especially crucial to analyze the data structures for further procedures. Recently a new clustering algorithm known as 'neutrosophic c-means' (NCM) was proposed in order to alleviate the limitations of the popular fuzzy c-means (FCM) clustering algorithm by introducing a new objective function which contains two types of rejection. The ambiguity rejection which concerned patterns lying near the cluster boundaries, and the distance rejection was dealing with patterns that are far away from the clusters. In this paper, we extend the idea of NCM for nonlinear-shaped data clustering by incorporating the kernel function into NCM. The new clustering algorithm is called Kernel Neutrosophic c-Means (KNCM), and has been evaluated through extensive experiments. Nonlinear-shaped toy datasets, real datasets and images were used in the experiments for demonstrating the efficiency of the proposed method. A comparison between Kernel FCM (KFCM) and KNCM was also accomplished in order to visualize the performance of both methods. According to the obtained results, the proposed KNCM produced better results than KFCM.

© 2016 Elsevier B.V. All rights reserved.

1. Introduction

Data clustering, or cluster analysis, is an important research area in pattern recognition and machine learning which helps the understanding of a data structure for further applications. The clustering procedure is generally handled by partitioning the data into different clusters where similarity inside clusters and the dissimilarity between different clusters are high. K-means clustering is known as a pioneering algorithm in the area with numerous applications. Until now, many variants of the K-means clustering algorithm have been proposed [1]. K-means algorithm assigns crisp memberships to all data points according to its nature. After fuzzy set theory was introduced by Zadeh, instead of using crisp memberships, the partial memberships described by membership functions sounded good in cluster analysis. Ruspini firstly adopted the fuzzy idea in data clustering [2]. Dunn proposed the popular fuzzy c-means (FCM) algorithm where a new objective function was redefined and the memberships were updated according to the distance [3]. A generalized FCM was introduced by Bezdek [4]. Although FCM has been used in many applications with successful results, it has several drawbacks. For example, FCM considers that all data points have equal importance. Noise and outlier data points are also issues that FCM is unable to handle. To alleviate these drawbacks, several

attempts have been undertaken in the past. In [5], the authors considered the Mahalanobis distance in FCM to analyze the effect of different cluster shapes. Dave et al. [6] proposed a new clustering algorithm namely 'fuzzy c-shell' which was effective on circular and elliptic-shaped datasets. On the other hand, the drawback of the FCM algorithm against the noise was investigated. The paper by [7] proposed a possibilistic c-means (PCM) algorithm, which was handled by relaxing the constraint of FCM summation to 1. Pal et al. [8] considered taking into account of both relative and absolute resemblance to cluster centers, which are considered as a combination of the PCM and FCM algorithms.

Recently, there have been numerous clustering algorithms which were developed to consider that a data point can belong to several sub-clusters at the same time [9]. These approaches were adopted based on evidential theory [10]. Masson and Denoeux proposed the evidential c-means algorithm (ECM) [10]. The authors then developed the relational-ECM (RECM) algorithm [11]. Based on neutrosophic logic [12], Guo and Sengur proposed the neutrosophic c-means (NCM) and the neutrosophic evidential c-means (NECM) clustering algorithms [13]. In NCM, a new cost function was developed to overcome the weakness of the FCM method on noise and outlier data points. In the NCM algorithm, two new types of rejection were developed for both noise and outlier rejections.

Another important drawback of the FCM algorithm is its clustering failure against the nonlinear separability clusters. This drawback can be alleviated by projecting the data points to a higher dimensional feature space in a nonlinear manner by considering the

* Corresponding author.

E-mail address: kemal.polat2003@yahoo.com (K. Polat).

Mercer Kernel in the FCM algorithm [14]. The integration of Mercer Kernel in the FCM algorithm was called Kernel FCM (KFCM) algorithm, and it achieved better clustering results, especially on the circular- and elliptic-shaped clusters.

In this paper, by inspiring the KFCM algorithm, a new Kernel NCM (KNCM) algorithm is proposed for improving the NCM method on the nonlinearly separable datasets. To do this, the NCM algorithm was re-formulated by incorporating the Mercer Kernel into NCM. Therefore, a new cost function was produced to make robust parameter estimation against noise and outliers. In addition, new membership and prototype update equations were derived from minimization of the proposed cost function. As KNCM has three memberships, T , I and F , T can be considered as the membership degree to determinant clusters, I is used to show membership degree to ambiguity cluster, and F can be used to determine outlier cluster for each data point, respectively. The membership values T , I and F are robust against the noise and outlier because they are calculated iteratively in the clustering procedure. The developed KNCM method was applied on a variety of applications such as toy dataset clustering, real dataset clustering, and noisy image segmentation. The obtained results were compared with the KFCM method, and showed that the proposed KNCM method yielded better results than KFCM.

The remainder of this paper is organized as follows. In Section 2, the NCM method is described. In Section 3, the related equations are derived and the algorithm of the proposed KNCM is described. In Section 4, experiments and the results obtained are reported. In addition, the comparison with KFCM is tabulated. Finally, several conclusions are provided in Section 5.

2. Neutrosophic c-Means Clustering (NCM)

Data clustering is an important task in data mining and machine learning. It classifies input data into different categories based on some measures of similarity. In traditional clustering algorithms such as K-means and Fuzzy c-means (FCM), samples are regarded as being of the same importance without considering noise and outlier samples. However, in some applications, datasets may contain outliers and noises that need to be determined. Recently a new clustering algorithm called neutrosophic c-means (NCM) was proposed to handle both the noise and outliers. It considers both degrees of belonging to determinate and indeterminate clusters, and a new objective function and memberships were developed as given in Eq. (1):

$$J_{NCM}(T, I, F, c) = \sum_{i=1}^N \sum_{j=1}^C (\varpi_1 T_{ij})^m \|x_i - c_j\|^2 + \sum_{i=1}^N (\varpi_2 I_i)^m \|x_i - \bar{c}_{imax}\|^2 + \delta^2 \sum_{i=1}^N (\varpi_3 F_i)^m \quad (1)$$

where T_{ij} , I_i and F_i are the membership values belonging to the determinate clusters, boundary regions and noisy data set. $0 < T_{ij}, I_i, F_i < 1$, which satisfy with the following formula:

$$\sum_{j=1}^c T_{ij} + F_i + I_i = 1 \quad (2)$$

For each point i , the \bar{c}_{imax} is computed using the clusters' centers with the largest and second largest value of T_{ij} .

$$\bar{c}_{imax} = \frac{c_{pi} + c_{qi}}{2} \quad (3)$$

$$p_i = \arg \max_{j=1,2,\dots,C} (T_{ij}) \quad (4)$$

$$q_i = \arg \max_{j \neq p_i, j=1,2,\dots,C} (T_{ij}) \quad (5)$$

where m is a constant. p_i and q_i are the cluster numbers with the biggest and second biggest value of T . When the p_i and q_i are identified, the \bar{c}_{imax} is calculated and its value is a constant number for each data point i .

The related membership functions are calculated as follows:

$$T_{ij} = \frac{\varpi_2 \varpi_3 (x_i - c_j)^{-\left(\frac{2}{m-1}\right)}}{\sum_{j=1}^C (x_i - c_j)^{-\left(\frac{2}{m-1}\right)} + (x_i - \bar{c}_{imax})^{-\left(\frac{2}{m-1}\right)} + \delta^{-\left(\frac{2}{m-1}\right)}} \quad (6)$$

$$I_i = \frac{\varpi_1 \varpi_3 (x_i - \bar{c}_{imax})^{-\left(\frac{2}{m-1}\right)}}{\sum_{j=1}^C (x_i - c_j)^{-\left(\frac{2}{m-1}\right)} + (x_i - \bar{c}_{imax})^{-\left(\frac{2}{m-1}\right)} + \delta^{-\left(\frac{2}{m-1}\right)}} \quad (7)$$

$$F_i = \frac{\varpi_1 \varpi_2 (\delta)^{-\left(\frac{2}{m-1}\right)}}{\sum_{j=1}^C (x_i - c_j)^{-\left(\frac{2}{m-1}\right)} + (x_i - \bar{c}_{imax})^{-\left(\frac{2}{m-1}\right)} + \delta^{-\left(\frac{2}{m-1}\right)}} \quad (8)$$

$$c_j = \frac{\sum_{i=1}^N (\varpi_1 T_{ij})^m x_i}{\sum_{i=1}^N (\varpi_1 T_{ij})^m} \quad (9)$$

The partitioning is carried out through an iterative optimization of the objective function, and the membership T_{ij} , I_i , F_i and the cluster centers c_j are updated according to Eqs. (6)–(9) at each iteration. The \bar{c}_{imax} is calculated by Eqs. (3)–(5) at each iteration. The iteration will not stop until $|T_{ij}^{(k+1)} - T_{ij}^{(k)}| < \varepsilon$, where ε is a termination criterion between 0 and 1, and k is the iteration step.

3. Kernel NCM

In nonlinear data clustering, the traditional clustering methods are not able to categorize the input data into proper clusters due to the limitation on objective function. So, a mapping procedure is needed to transfer the input data to a high dimensional feature space by applying a kernel function. When the kernel function is applied to the NCM algorithm, the objective function becomes:

$$J_{KNCM}(T, I, F, c) = \sum_{i=1}^N \sum_{j=1}^C (\varpi_1 T_{ij})^m \|\Phi(x_i) - \Phi(c_j)\|^2 + \sum_{i=1}^N (\varpi_2 I_i)^m \|\Phi(x_i) - \Phi(\bar{c}_{imax})\|^2 + \delta^2 \sum_{i=1}^N (\varpi_3 F_i)^m \quad (10)$$

and

$$\|\Phi(x_i) - \Phi(c_j)\|^2 = K(x_i, x_i) - 2K(x_i, c_j) + K(c_j, c_j) \quad (11)$$

where $K(x_i, x_i) = \Phi^T(x_i) \Phi(x_i)$ shows the inner product. If Gaussian function is considered as the kernel function, then $K(x_i, x_i)$ and $K(c_j, c_j)$ becomes one. Therefore, the new objective function becomes;

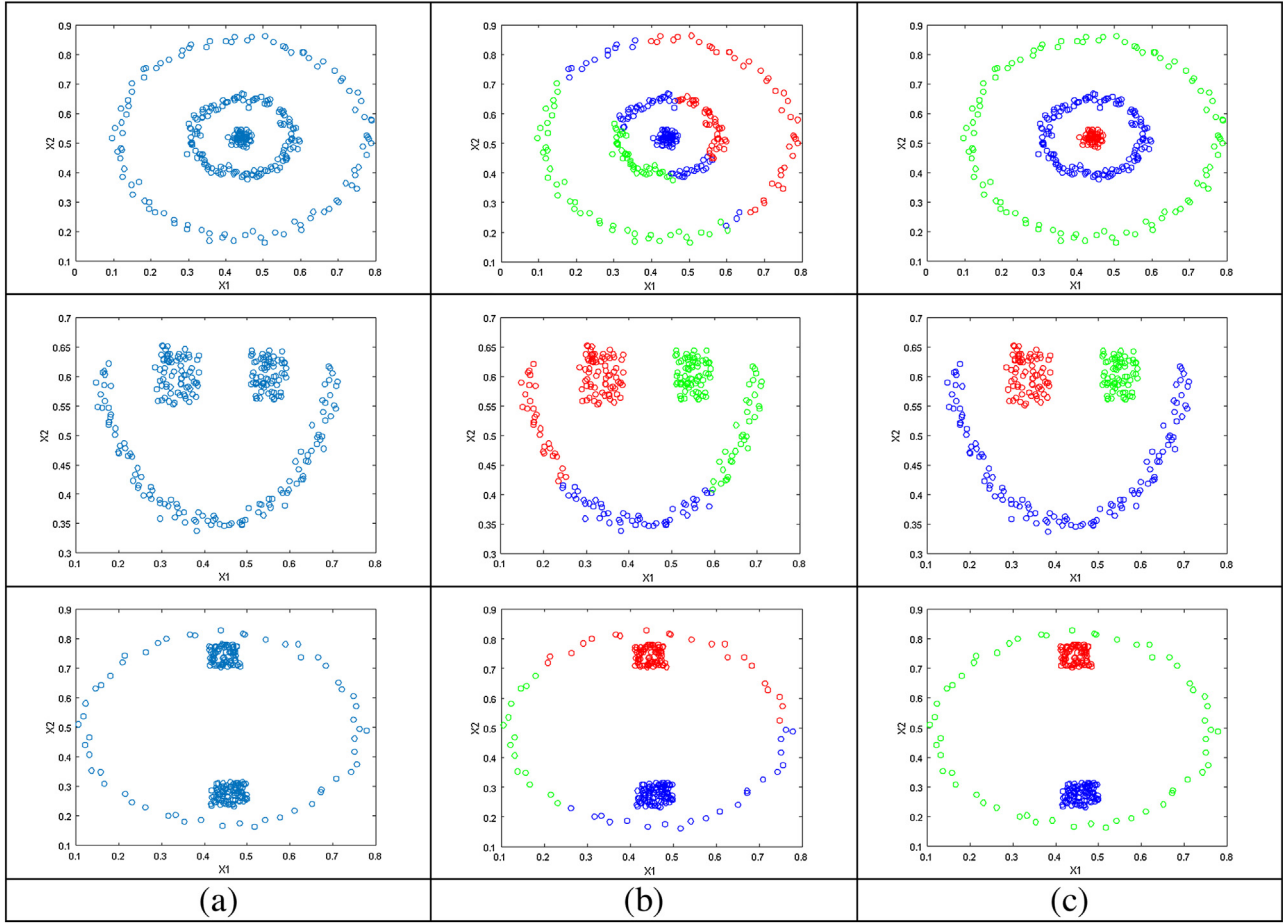


Fig. 1. Comparison of NCM and KNCM on various toy datasets a) raw data, b) NCM results, c) KNCM results.

$J_{KNCM}(T, I, F, c) =$

$$\sum_{i=1}^N \sum_{j=1}^C 2(\varpi_1 T_{ij})^m (1 - K(x_i, c_j)) + \sum_{i=1}^N 2(\varpi_2 I_i)^m (1 - K(x_i, \bar{c}_{imax})) + \delta^2 \sum_{i=1}^N (\varpi_3 F_i)^m \quad (12)$$

A Lagrange objective function is constructed as:

$$L(T, I, F, c, \lambda) = \sum_{i=1}^N \sum_{j=1}^C 2(\varpi_1 T_{ij})^m (1 - K(x_i, c_j)) + \sum_{i=1}^N 2(\varpi_2 I_i)^m (1 - K(x_i, \bar{c}_{imax})) + \sum_{i=1}^N \delta^2 (\varpi_3 F_i)^m - \sum_{i=1}^N \lambda_i \left(\sum_{j=1}^C T_{ij} + I_i + F_i - 1 \right) \quad (13)$$

To minimize the Lagrange objective function, the following calculations are used:

$$\frac{\partial L}{\partial T_{ij}} = 2m(\varpi_1)^m (T_{ij})^{m-1} (1 - K(x_i, c_j)) - \lambda_i \quad (14)$$

$$\frac{\partial L}{\partial I_i} = 2m(\varpi_2)^m (I_i)^{m-1} (1 - K(x_i, \bar{c}_{imax})) - \lambda_i \quad (15)$$

$$\frac{\partial L}{\partial F_i} = \delta^2 m (\varpi_3)^m (F_i)^{m-1} - \lambda_i \quad (16)$$

$$\frac{\partial L}{\partial c_j} = -2 \sum_{i=1}^N (\varpi_1 T_{ij})^m K'(x_i, c_j) \quad (17)$$

The norm is specified as the Euclidean norm. Let $\frac{\partial L}{\partial T_{ij}} = 0$, $\frac{\partial L}{\partial I_i} = 0$, $\frac{\partial L}{\partial F_i} = 0$, and $\frac{\partial L}{\partial c_j} = 0$, the membership functions and centers of clusters are re-arranged as follows;

$$T_{ij} = \frac{\varpi_2 \varpi_3 K(x_i, c_j)^{-\left(\frac{2}{m-1}\right)}}{\sum_{j=1}^C K(x_i, c_j)^{-\left(\frac{2}{m-1}\right)} + K(x_i, \bar{c}_{imax})^{-\left(\frac{2}{m-1}\right)} + \delta^{-\left(\frac{2}{m-1}\right)}} \quad (18)$$

$$I_i = \frac{\varpi_1 \varpi_3 K(x_i, \bar{c}_{imax})^{-\left(\frac{2}{m-1}\right)}}{\sum_{j=1}^C K(x_i, c_j)^{-\left(\frac{2}{m-1}\right)} + K(x_i, \bar{c}_{imax})^{-\left(\frac{2}{m-1}\right)} + \delta^{-\left(\frac{2}{m-1}\right)}} \quad (19)$$

$$F_i = \frac{\varpi_1 \varpi_2 (\delta)^{-\left(\frac{2}{m-1}\right)}}{\sum_{j=1}^C K(x_i, c_j)^{-\left(\frac{2}{m-1}\right)} + K(x_i, \bar{c}_{imax})^{-\left(\frac{2}{m-1}\right)} + \delta^{-\left(\frac{2}{m-1}\right)}} \quad (20)$$

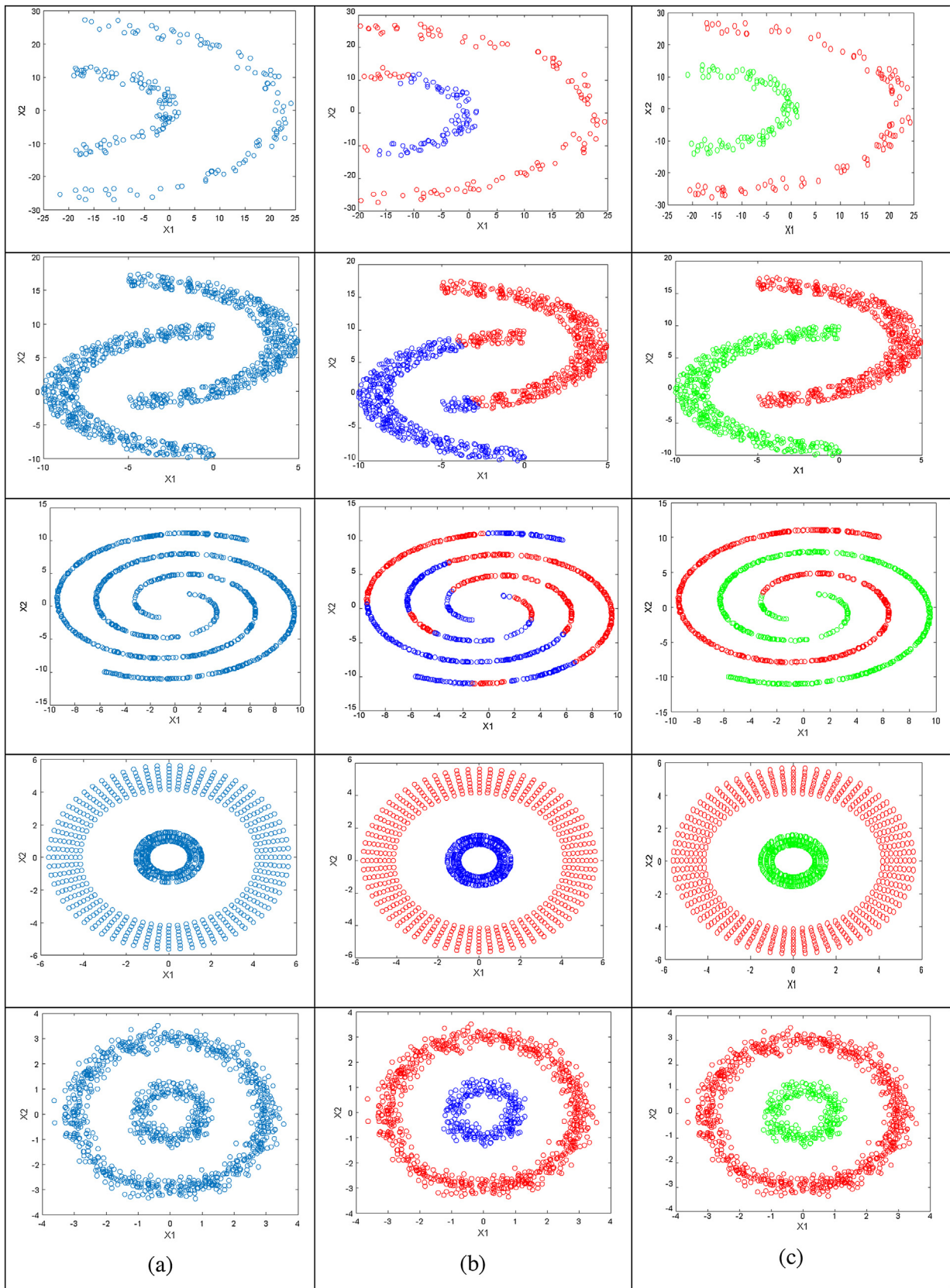


Fig. 2. Comparison of KFCM and KNKM on two-cluster toy datasets a) raw data, b) KFCM results, c) KNKM results.

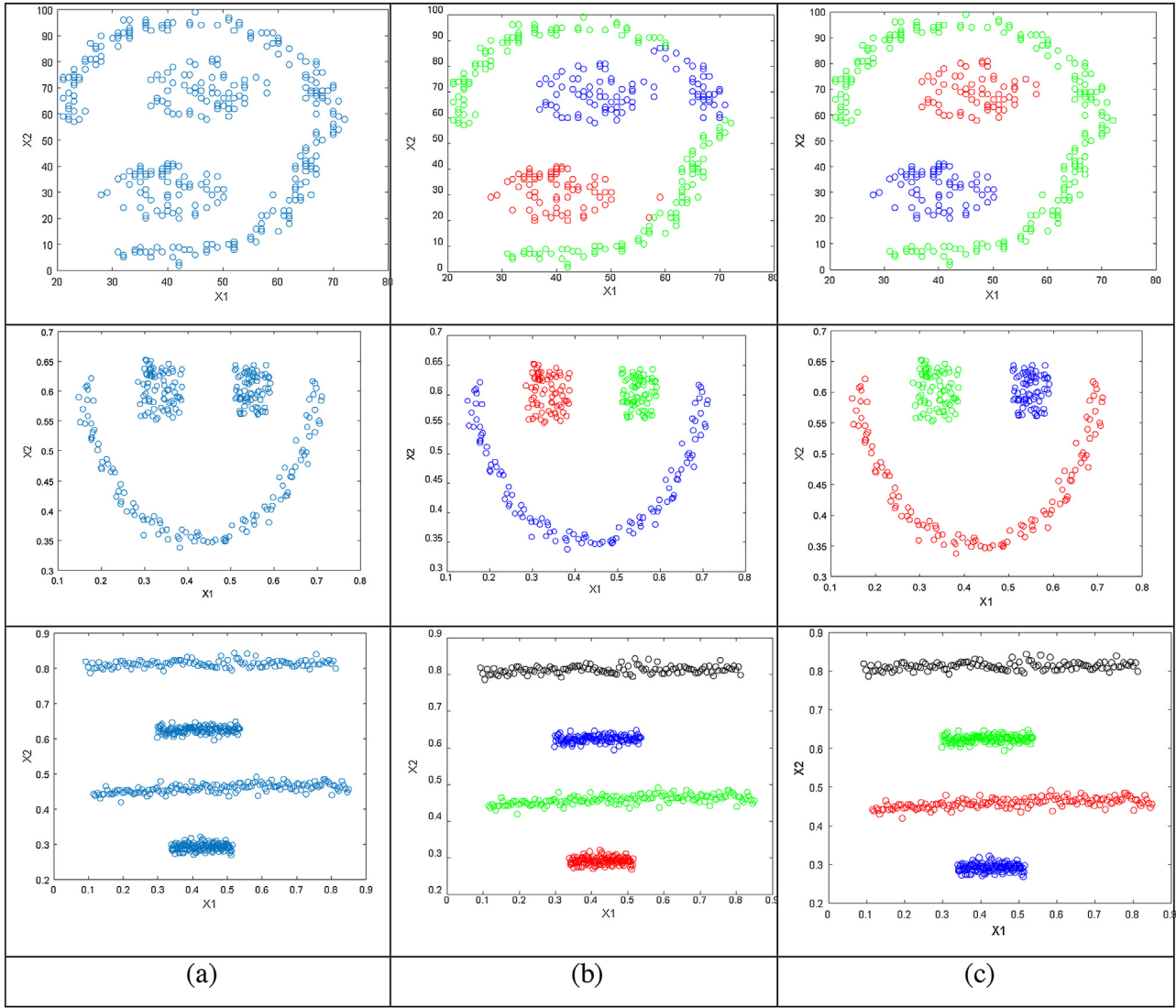


Fig. 3. Comparison of KFCM and KNCM on other toy datasets a) raw data, b) KFCM results, c) KNCM results.

4. Experiments and results

In this section, a variety of experiments were conducted in order to compare the performances of the KNCM and kernel FCM methods. The experiments were performed on toy datasets, real datasets, and images. Both KNCM and KFCM methods were run under the same initial parameters such as $\varepsilon=10^{-5}$. In addition, the weighting parameters of the KNCM was set to $\varpi_1=0.75$, $\varpi_2=0.125$, $\varpi_3=0.125$, respectively, which were obtained from trial and error. A grid search of the trade-off constant delta on $\{10\hat{-}2, 10\hat{-}1, \dots, 10\hat{2}, 10\hat{3}\}$ and w_1, w_2 and w_3 on $\{0.01, 0.02, 0.03, \dots, 0.96, 0.97, 0.98, \}$ was conducted in seek of the optimal results.

The radial basis function (RBF) was considered as the kernel function for both methods. The parameter of the RBF kernel for KFCM was obtained by an interval search method. For a given interval, the KFCM clustering algorithm was run with a proper incremental value, and the optimum RBF parameter was chosen where the clustering error was minimized. The same procedure was considered for KNCM. As KNCM has two adjustable parameters, namely RBF kernel parameter and delta, a 2D interval search was considered with proper incremental values. The platform for

$$c_j = \frac{\sum_{i=1}^N (\varpi_1 T_{ij})^m K(x_i, x_i)}{\sum_{i=1}^N (\varpi_1 T_{ij})^m} \quad (21)$$

The above equations allow the formulation of the KNCM algorithm, which can be summarized in the following steps:

- Step 1. Initialize $T^{(0)}$, $I^{(0)}$, and $F^{(0)}$;
- Step 2. Initialize the $C, m, \delta, \varepsilon, \varpi_1, \varpi_2, \varpi_3$ parameters;
- Step 3. Choose kernel function and its parameters;
- Step 4. Calculate the centers vectors $c^{(k)}$ at k step using Eq. (18);
- Step 5. Compute the $\tilde{c}_{i_{\max}}$ using the clusters' centers with the largest and second largest value of T_{ij} as Eq. (3);
- Step 6. Update $T^{(k)}$ to $T^{(k+1)}$ using Eq. (15), $I^{(k)}$ to $I^{(k+1)}$ using Eq. (16), and $F^{(k)}$ to $F^{(k+1)}$ using Eq. (17);
- Step 7. If $|T^{(k+1)} - T^{(k)}| < \varepsilon$ then stop; otherwise return to Step 4;
- Step 8. Assign each data into the class with the largest $TM = [T, I,$

$$F] \text{ value: } x(i) \in k\text{th class if } k = \arg \max_{j=1, 2, \dots, C+2} (TM_{ij})$$

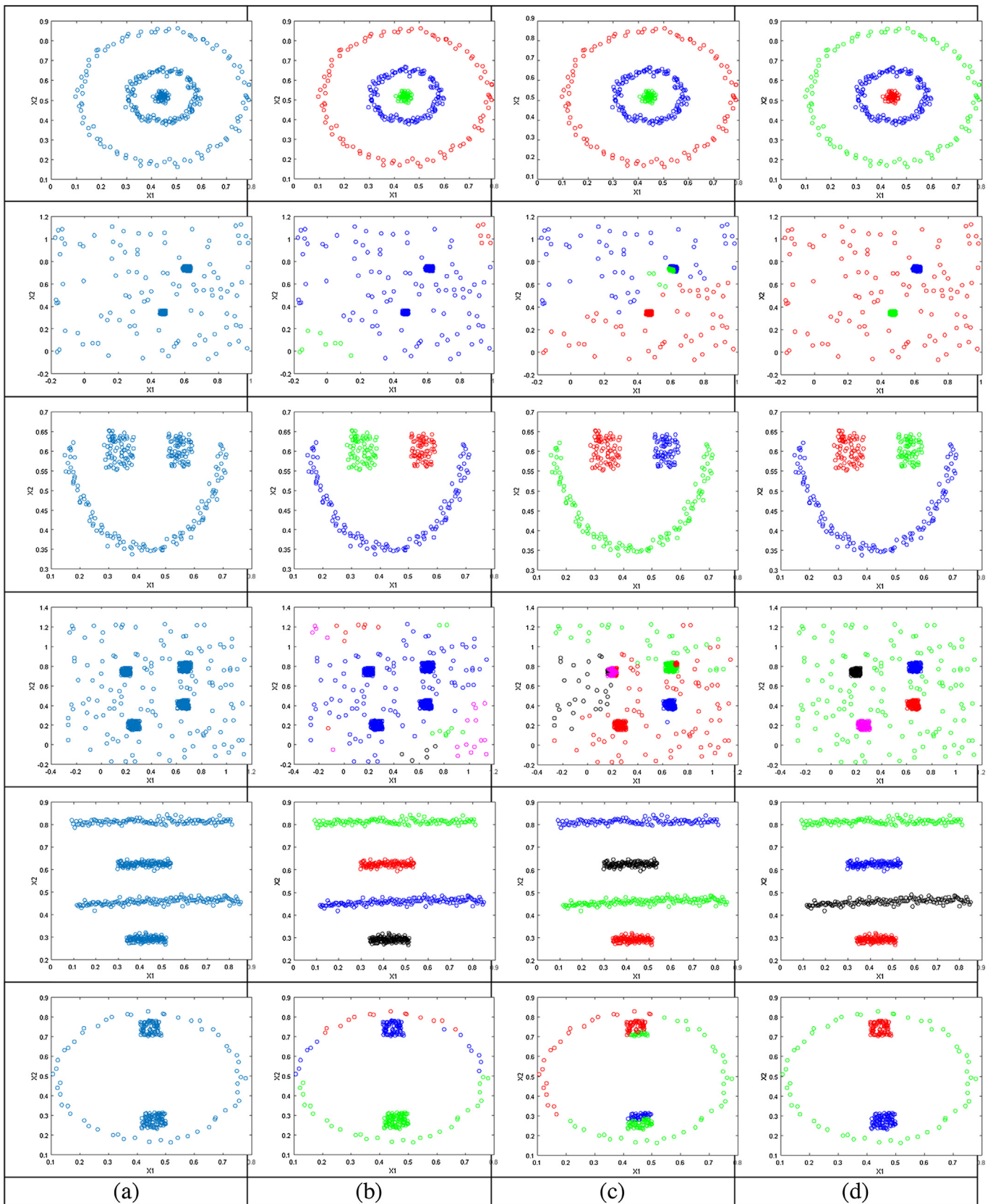


Fig. 4. Comparison of SC, SMMC, and KNCM methods on six toy datasets a) raw data, b) SC results, c) SMMC results, d) KNCM results.

experiment is under MATLAB 2014b software on a computer having an Intel Core i7-4810 CPU and 32 GB memory.

Experimental works started with a comparison between NCM and KNCM, in order to show the effect of the kernel idea on NCM clustering. As it is obvious K-means, FCM, and NCM type clustering

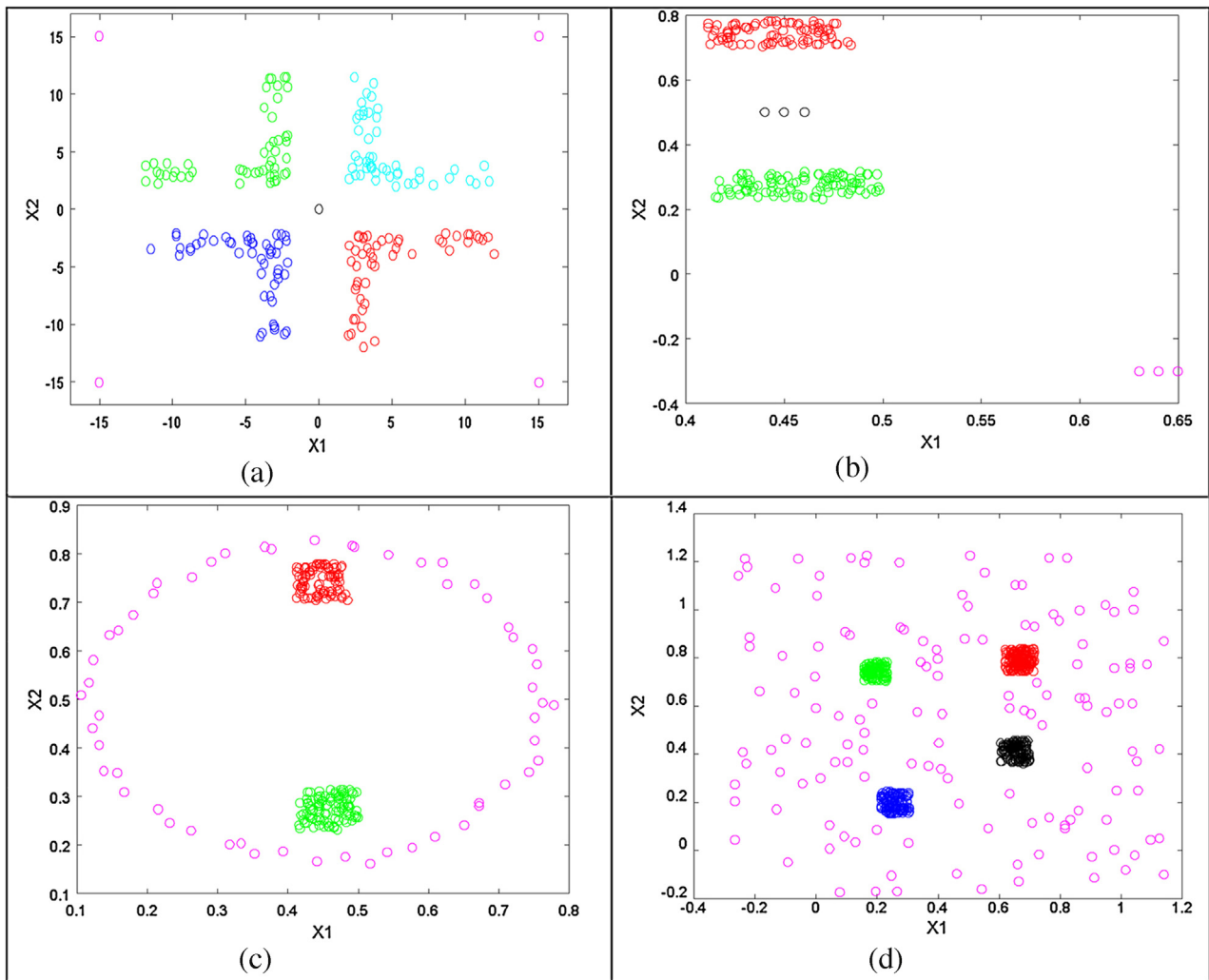


Fig. 5. KNCM clustering results on noisy and outlier data points.

algorithms cannot properly cluster nonlinear datasets. To show this effect, several nonlinear toy datasets were used, as shown in Fig. 1, which contains three different shaped datasets. In Fig. 1, column (a) shows the raw data, where columns (b) and (c) show the NCM and KNCM results respectively. According to the obtained results, while KNCM obtained the ground-truth clustering results, NCM failed to obtain the exact clusters.

4.1. Toy-data example 1

In the first type of experiment, the two-cluster datasets were considered as shown in Fig. 2. In the first column of Fig. 2, the raw datasets are illustrated. In the second column, the clustering results of kernel FCM are given and, in the third column, the KNCM results are shown. In the first row of Fig. 2, the 'two kernel' dataset is shown, and is composed of 200 samples. Each cluster has 100 samples. The search procedure automatically selected 223 for the RBF parameter for kernel FCM and similarly 280 was found for the RBF parameter and 100 was assigned to the value of delta parameter. The obtained clusters are shown with different colors. As can be seen in Figs. 2 and 3, the KFCM method did not produce valid clustering. Some of the samples were wrongly clustered; especially, some of the inner samples were mis-clustered. On the other hand, KNCM classified both clusters error-free. It is worth mentioning that when KNCM was run several times, each time KNCM always produced

the correct clusters. Another example is shown in the second row of Fig. 2.

This dataset called 'half-moons'. Two half-moon-shaped datasets, which is nonlinearly separable, contain 1000 samples. For KFCM, the RBF kernel parameter was obtained as 9.3, and for kernel NCM, the RBF kernel parameter and delta value was assigned as 2.5 and 10, respectively. KFCM did not classify some of the data samples correctly. Especially, the data points in the inner tail part of the half-moons were misclassified. Similar to the previous dataset, kernel NCM performed the clustering error free. All data points assigned to the correct clusters.

Another popular non-linear dataset, the 'two-spirals', was also considered in the experiments. The raw dataset can be seen in the third row of Fig. 2. A RBF kernel parameter of 1.7 was generated by the search algorithm for both kernel FCM and NCM, and the delta value was 7.0. As the clustering results are shown in the third row of Fig. 2, both methods did not produce the correct clusters. Especially, the KFCM method produced meaningless clusters. There were misclassified points in both clusters. On the other hand, KNCM produced more reasonable results. One of the clusters was classified correctly (the points labeled in green). Moreover, only a part of the other cluster (the points labeled in red) was misclassified. In the last two rows of Fig. 2, similar datasets were experimented with. In both cases, there were circular clusters where both kernel FCM

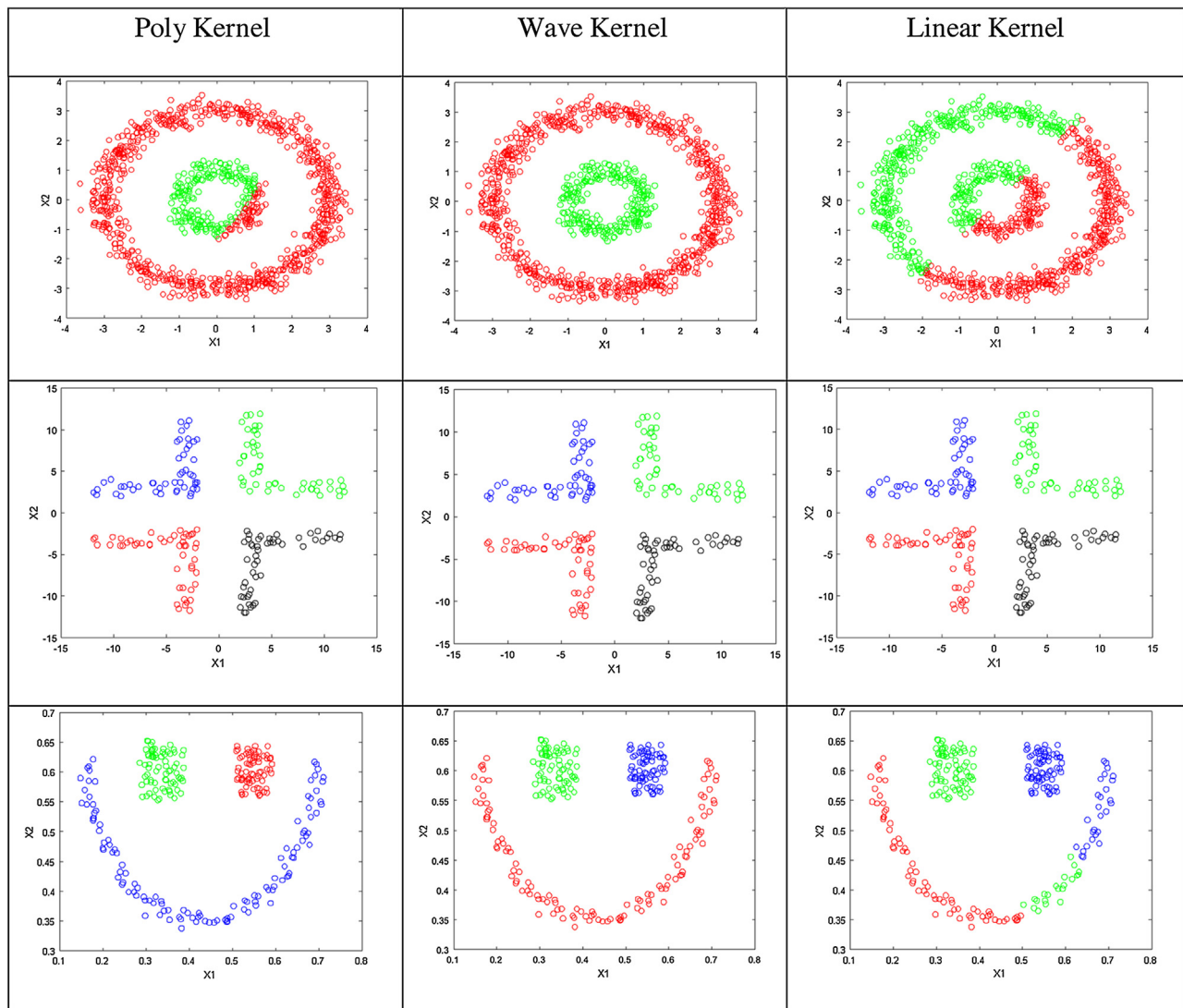


Fig. 6. KNCM clustering results with different kernel functions.

and NCM produced clear clusters. The clustering accuracies for both methods were 100%.

4.2. Toy-data example 2

Further experiments were performed on toy-datasets where the number of clusters was greater than two. The related results are shown in Fig. 3. Similar to Fig. 2, the first column of Fig. 3 shows the raw toy-datasets. The second column of Fig. 3 shows the KFCM clustering results, and the third column shows the clustering results of the proposed kernel NCM. The toy-datasets, which contained three and four clusters, were considered in the experiments. When the obtained results were evaluated, it was seen that except ‘ear’ data, both methods produced 100% correct clustering results for all toy-datasets. For ‘ear’ dataset, KNCM produced more accurate clustering than kernel FCM method. The kernel parameter and the delta value were set similar to the first experiments.

4.3. Comparison with spectral clustering methods

The KNCM method was also compared with two eigenvalue-based clustering methods, namely “Spectral Clustering” (SC) [15] and “Spectral Multi-Manifold Clustering” (SMMC) [16]. SC is a pop-

ular clustering algorithm which makes use of the eigenvalues of the similarity matrix of the input data. The main purpose of using eigenvalues is to perform dimensionality reduction before clustering. Finally, k-means algorithm is used for clustering the reduced dataset. SMMC algorithm is also a SC-based clustering method which improves the SC performance by integrating the multiple smooth low-dimensional manifolds into SC algorithm. SMMC then uses the local geometric information of the sampled data to construct a suitable affinity matrix. Finally, SC is used with this affinity matrix to group the data. The comparisons were made on six toy datasets which are shown in Fig. 4. While the first column shows the raw toy datasets, the second, third, and fourth columns show the SC, SMMC, and KNCM results, respectively. As can be seen, all methods produced ground-truth clustering results for the toy datasets, as can be seen in the first, third, and fifth rows of Fig. 4. On the other hand, only the KNCM method yielded ground-truth clustering results for the toy datasets, as seen in the second, fourth, and sixth rows of Fig. 4. These results show that both SC and SMMC methods are not able to cluster the datasets which contains noise cluster. In other words, KNCM produces similar performance on the nonlinear datasets. In addition, KNCM outperforms on data which contains noise clusters.

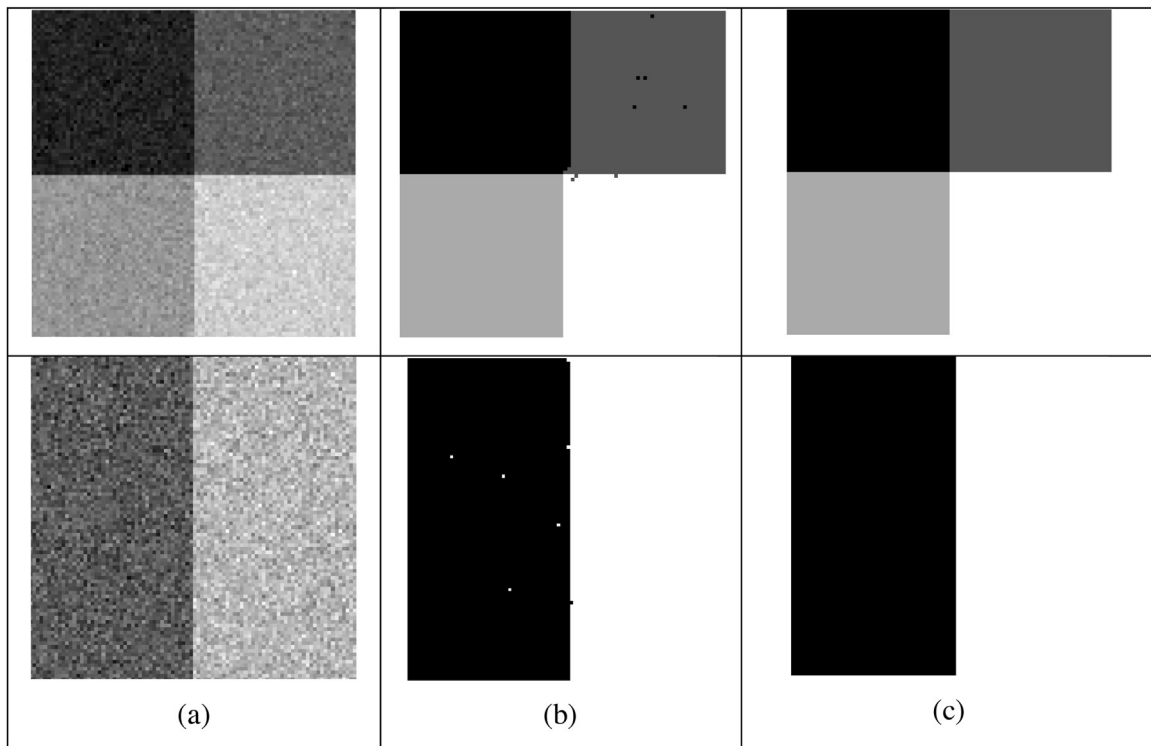


Fig. 7. Comparison of KFCM and KNCM on image segmentation application a) original image, b) KFCM results, c) KNCM results.

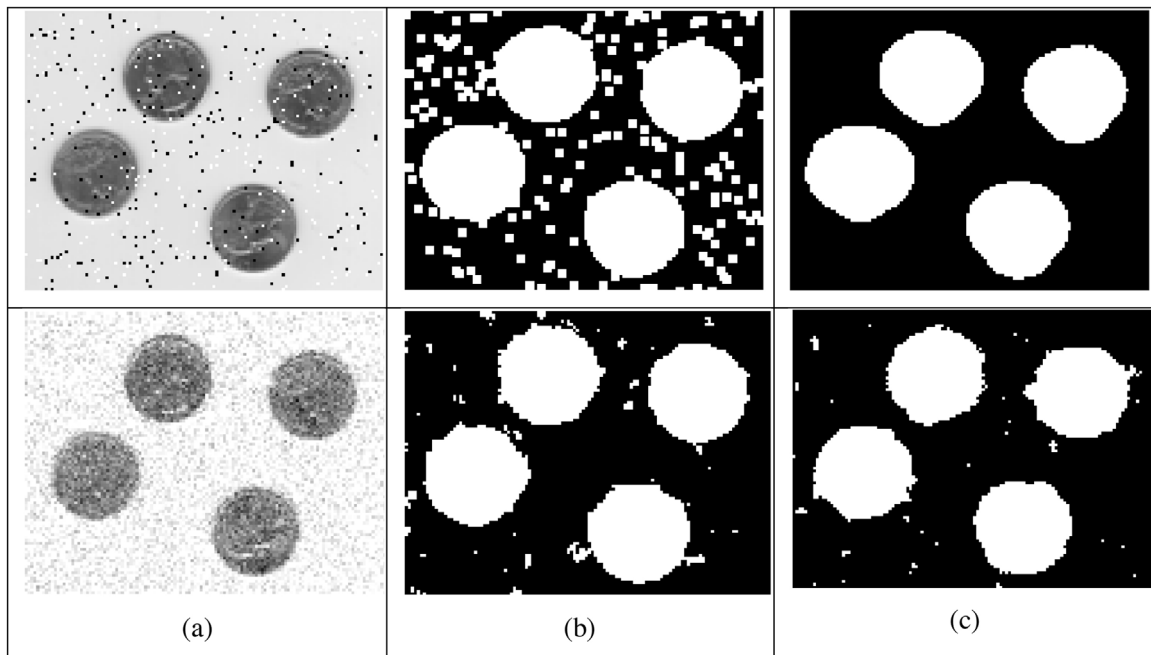


Fig. 8. Comparison of kernel FCM and NCM on image segmentation application a) original image, b) Kernel FCM results, c) Kernel NCM results.

4.4. Toy datasets with noise and outlier

As mentioned earlier, NCM algorithm showed better performance in the clustering of noisy and outlier data points. In order to demonstrate that KNCM algorithm works well with datasets which contain noisy and outlier data points, several experiments were conducted on various toy datasets. The obtained results are tabulated in Fig. 5, which shows (a) the 'corner' dataset where four linearly-separable clusters are located. A data point in the middle

of the four clusters was artificially located. Moreover, we located four more data points as illustrated in Fig. 5(a). For the 'corner' dataset, KNCM algorithm not only clustered correctly all four clusters, but also detected the noise and outlier data points. The black and magenta colors represent the noise and outlier data points respectively. A similar scenario was established for a two-clusters case, as shown in Fig. 5(b), and similar successful clustering results were also obtained for this scenario. In Fig. 5(c), circular outlier data points are considered which surrounds two linearly-separable clus-

Table 1
KNCM clustering results of the IRIS dataset.

		Actual	versicolor	virginica
Clustered	setosa	50	0	0
	versicolor	0	49	4
	virginica	0	1	46

ters. In addition, in Fig. 5(d), another toy dataset with many outliers was illustrated. For these two toy datasets (Fig. 5(c) and (d)), the proposed KNCM algorithm found reasonable clusters.

4.5. Using different kernels on various toy datasets

Although RBF kernel-based KNCM yielded better results; in this section, it is demonstrated that KNCM produces better results with various kernels for various toy datasets. To this end, “poly”, “wave”, and “linear” kernels are used, respectively. The definition of the kernel functions (RBF, Linear, Poly, and Wave) that were used in each experiment are given as;

$$K_{RBF}(x, x_i) = e^{-\frac{\|x-x_i\|_2^2}{\sigma^2}} \tag{22}$$

$$K_{lin}(x, x_i) = x_i^T x \tag{23}$$

$$K_{poly}(x, x_i) = (x_i^T x + \tau)^d \tag{24}$$

$$K_{Wave}(x, x_i) = \cos\left(\frac{\alpha(x-x_i)}{\beta}\right) e^{-\frac{\|x-x_i\|_2^2}{\sigma^2}} \tag{25}$$

As can be seen in Eq. (22), “RBF” kernel has only one adjustable parameter. Eq. (23) shows that “linear” kernel does not have any adjustable parameter, whereas, “poly” and “wave” kernels have two and three adjustable parameters respectively.

The parameters of these kernels were also tuned. Several clustering results are shown in Fig. 6. The results, which were obtained with “Wave” kernel, were deemed to be quite successful. The “poly” kernels also yielded better results, with only a few data points mis-clustered. The “Linear” kernel obtained the worst clustering; with only the linearly-separable dataset (“corners”) clustered correctly. The “Linear” kernel could not separate the clusters correctly which have a circular data structure.

4.6. Real dataset example 1

Various real datasets were also used in order to evaluate and compare the obtained clustering results with both methods. To this end, first of all the famous ‘iris’ dataset was considered. The IRIS dataset contains three classes, i.e., three varieties of Iris flowers, namely, Iris Setosa, Iris Versicolor, and Iris Virginica, consisting of 50 samples each. Each sample has four features, namely, sepal length (SL), sepal width (SW), petal length (PL), and petal width (PW). One of the three clusters is clearly separated from the other two, while these two classes admit some overlap. The RBF kernel parameter was 0.001 and delta was 3. As can be seen from Table 1, the Setosa is clustered with a 100% correct clustering rate, but other clusters such as Versicolor and Virginica are not clustered exactly. Four Virginica samples were wrongly clustered as Versicolor and one Versicolor sample was clustered as Virginica. The total accuracy was 96.67%.

4.7. Real dataset example 2

Experiments on the second real dataset were conducted our on the ‘wine’ dataset. The ‘wine’ dataset was constructed based on the results of chemical analysis of wines grown in the same region of

Table 2
KNCM clustering results of the wine dataset.

		Actual	Class 2	Class 3
Clustered	Class 1	57	7	0
	Class 2	2	58	1
	Class 3	0	6	47

Table 3
KNCM clustering results of the Parkinson dataset.

		Actual	Healthy
Clustered	PD	118	4
	Healthy	29	44

Table 4
Performance comparisons of both methods on real datasets.

	KFCM	KNCM
Iris	95.33	96.67
Wine	89.89	91.01
Parkinson	80.00	83.08

Italy, but derived from three different cultivars. The dataset contains 13 attributes and three clusters. The total number of the sample is 178 and each cluster has 59, 71, and 48 samples, respectively. The RBF kernel parameter was 0.0095 and delta was 100. The obtained results are given in Table 2, where none of the clusters were classified 100% correctly, and the total number of misclassified samples was 16. Thus, the overall accuracy was 91.01%.

4.8. Real dataset example 3

The third experiment on a real dataset was performed on the ‘Parkinson’ dataset. The dataset contains 22 attributes, and the total number of samples is 195; of which, 147 are Parkinson and the rest 48 are healthy. The data is used to discriminate healthy people from those with Parkinson disease (PD). In other words, there are two clusters.

The RBF kernel parameter was 0.005 and delta was 10. The obtained results are given in Table 3, in which 29 PD are clustered as healthy and 4 healthy samples were clustered as PD. Thus, the total accuracy was 83.08%.

The same experiments were also performed with KFCM and the accuracy comparisons of both methods are given in Table 4. It is evident that for all real datasets, the proposed KNCM produced better results than KFCM.

4.9. Image segmentation example 1

Before describing the image segmentation experiments, it should be mentioned that spatial information was used for both kernel methods. The spatial information [17] was considered for both KNCM and KFCM experiments.

Two different synthesized images were used. The first has four classes of size 100 × 100, and the corresponding gray values are 50 (upper left, UL), 100 (upper right, UR), 150 (low left, LL) and 200 (low right, LR), respectively. The image is degraded by the Gaussian noise with μ = 0, σ = 25. The second image has two clusters. The corresponding gray values are 30 for the left column and 80 for the right column. The image is degraded by the Gaussian noise with μ = 0, σ = 15.

The obtained results are given in Fig. 7. The original images are illustrated (a) in the first column, with the second column (b) showing the KFCM results, and column three (c) for the KNCM results. With visual inspection, the proposed method yielded exact seg-

mentations and KFCM produced several misclassified pixels for both cases.

4.10. Image segmentation example 2

In the second image segmentation experiment, the 'eight' image was used with different noise types and levels. The noise types were 'salt and pepper' and 'Gaussian', the noise density for 'salt and pepper' was 0.04, and the noise parameters for 'Gaussian' were 0 mean and 0.1 variance, respectively.

The obtained results for real image are also given in Fig. 8. Upon visual inspection, the proposed KNKM yielded better results for both noise types and levels than KFCM.

5. Conclusions

In this paper, a new data clustering algorithm KNKM has been proposed and its efficiency tested through extensive experimentation. Incorporating kernel information to NCM made it available in nonlinear-shaped data clustering. The proposed scheme was quite successful on clustering a variety of toy and real datasets. In addition, image segmentation applications of the proposed method were also promising. Besides its efficiency, the proposed method can handle noise and outlier data points due to its new objective and membership functions. KNKM will find more applications in data mining and machine learning with its ability to handle indeterminacy information efficiently.

References

- [1] M.R. Andenberg, *Cluster Analysis for Applications*, Academic Press, New York, 1973.
- [2] E. Ruspini, A new approach to clustering, *Inf. Control* 15 (1969) 22–32.

- [3] J.C. Dunn, A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters, *J. Cybernet.* 3 (1974) 32–57.
- [4] J.C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, New York, 1987.
- [5] D.E. Gustafson, W.C. Kessel, Fuzzy clustering with a fuzzy covariance matrix, San Diego, CA, in: *Proceedings of IEEE CDC*, 10, 1979, pp. 761–766 (12).
- [6] R.N. Dave, Clustering of relational data containing noise and outliers In: *FUZZ'IEEE* 98, vol. 2, pp. 1411–1416 (1998).
- [7] R. Krishnapuram, J. Keller, A possibilistic approach to clustering, *IEEE Trans. Fuzzy Syst.* 1 (2) (1993) 98–110.
- [8] N.R. Pal, K. Pal, J.C. Bezdek, A mixed c-means clustering model, in: *Proceedings of the Sixth IEEE International Conference on Fuzzy Systems*, Barcelona, 1997, pp. 11–21.
- [9] M. Ménard, C. Demko, P. Loonis, The fuzzy c + 2 means: solving the ambiguity rejection in clustering, *Pattern Recognit.* 33 (2000) 1219–1237.
- [10] M.H. Masson, T. Denoeux, ECM: an evidential version of the fuzzy c-means algorithm, *Pattern Recognit.* 41 (2008) 1384–1397.
- [11] M.H. Masson, T. Denoeux, RECM: Relational Evidential c-means Algorithm, *Pattern Recognit. Lett.* 30 (2009) 1015–1026.
- [12] F. Smarandache, *A Unifying Field in Logics Neutrosophic Logic. Neutrosophy, Neutrosophic Set, Neutrosophic Probability*, third ed., American Research Press, 2003.
- [13] Y. Guo, A. Sengur, NCM: Neutrosophic c-means Clustering Algorithm, *Pattern Recognit.* 48 (8) (2015) 2710–2724.
- [14] Dao-Qiang Zhang, Song-Can Chen, Clustering incomplete data using kernel-based fuzzy c-means algorithm, *Neural Process. Lett.* 18 (3) (2003) 155–162.
- [15] Andrew Y. Ng, Michael I. Jordan, Yair Weiss, On spectral clustering: analysis and an algorithm, *Adv Neural Inf. Process. Syst.* 2 (2002) 849–856.
- [16] Yong Wang, et al., Spectral clustering on multiple manifolds, *IEEE Trans. Neural Networks* 22 (7) (2011) 1149–1161.
- [17] Miin-Shen Yang, Hsu-Shen Tsai, A Gaussian kernel-based fuzzy c-means algorithm with a spatial bias correction, *Pattern Recognit. Lett.* 29 (12) (2008) 1713–1725.