Global Fuzzy C-Means with Kernels

Gyeyongyong Heo
Hun Choi
Jihong Kim

Abstract—Fuzzy c-means (FCM) is a simple but powerful clustering method using the concept of fuzzy sets that have been proved to be useful in many areas. There are, however, several well-known problems with FCM, such as sensitivity to initialization, sensitivity to outliers, and limitation to convex clusters. In this paper, a new clustering method, which is an extension of global fuzzy c-means (G-FCM) and kernel fuzzy c-means (K-FCM), is proposed to resolve the shortcomings mentioned above. G-FCM is a variant of FCM that uses an incremental seed selection method and is effective in alleviating the sensitivity to initialization. There are several approaches to reduce the influence of noise and properly partition non-convex clusters and K-FCM is one of them. K-FCM is used in this paper because it can easily be extended with different kernels, which provide sufficient flexibility to allow for resolution of the shortcomings of FCM. By combining G-FCM and K-FCM, the proposed method, kernelized global FCM (KG-FCM), can resolve the shortcomings mentioned above. The usefulness of KG-FCM is demonstrated by experiments using artificial and real world data sets.

Keywords— Fuzzy C-Means; Global Initialization; Kernel Method; Non-Convex Cluster; Noise Robustness

I. INTRODUCTION

Clustering, also known as cluster analysis, assigns an unlabeled data set \( X = \{x_1, \ldots, x_N\} \) into \( K (1 \leq K \leq N) \) homogeneous clusters based on a similarity measure [1] and has formed an important area in pattern recognition, image processing, and, most recently, data mining. A variety of clustering algorithms has been proposed, including fuzzy c-means (FCM), which is the main concern of this paper. Although FCM is an efficient algorithm, there are several difficulties with it: (1) choosing the initial cluster seed; (2) suppressing noise; (3) accommodating non-convex clusters; and (4) determining the optimal number of clusters. In this paper, we try to solve the problems all but the last problem assuming the number of clusters is known. To address the remaining problems, global fuzzy c-means (G-FCM), which is an extension of global k-means (GKM), is further extended using kernelized fuzzy clustering. G-FCM uses a deterministic approach to select a set of initial seeds to resolve the initialization problem. Kernelized fuzzy clustering is a non-linear counterpart of fuzzy clustering and can accommodate non-convex clusters. Some kernelized clustering methods also have noise robustness according to the kernel used. Although there have been attempts to solve the shortcomings in FCM, they tried to solve one problem at a time. By combining global and kernelized clustering, however, the proposed method can resolve the shortcomings simultaneously.

It has long been known that FCM is only guaranteed to converge on a locally optimal solution, which is sensitive to initial state. Since the problem of obtaining a globally optimal initial state and a globally optimal solution has been shown to be NP-hard, the initialization method for a sub-optimal solution is more practical and of great value. There exist various initialization methods for clustering, which can be categorized into three groups [2]: random sampling methods, distance optimization methods, and density estimation methods. In random sampling methods, probably the most widely adopted ones, multiple runs of clustering with random initial seeds are conducted, and the best or aggregated one is selected as the final clustering result [3]. Although they are simple and statistically sound, they depend heavily on the number of runs, and the computational cost is very high.

Distance optimization methods are used to maximize the distance among the seeds. As many clustering methods, including FCM, try to minimize the intra-cluster variance without optimizing the inter-cluster separation, it is natural to maximize the inter-cluster distance beforehand. The MaxMin procedure [4] and its variants, in which a data point having the largest distance to the existing set of seeds is iteratively added, are the most widely used ones. Although they are simple and efficient, some of them also require multiple runs and the resulting set of seeds tends to be placed on the boundary of data.

Density estimation methods select a set of seeds based on local density, the estimation of which is crucial in these methods. The k-nearest neighbor or \( \epsilon \)-radius ball is generally used to decide neighboring points [5], and each requires one additional parameter, G-FCM [6] also belongs to this group, which iteratively adds a data point optimizing the original objective function for clustering as a seed. Due to its deterministic property, it does not require multiple runs and does not need any extra parameter, i.e. does not have the shortcomings in other methods, and demonstrated better results than other methods.

The global clustering method was first formulated for hard clustering [7,8] and extended to soft clustering [6]. These global methods, however, are still affected by noise and cannot accommodate non-convex clusters. The sensitivity to noise can be alleviated in several ways and adopting kernel methods is one of them. Kernel methods were originally proposed to convert linear methods into non-linear ones [9], but it is also well known that some kernels have outlier robustness together with their non-linear properties [10]. Another reason for adopting kernel methods is that they can be extended in several ways using different kernels. There exist other ways to accommodate non-convex clusters and clustering with non-Euclidean distance [11] and spectral clustering [12,13] are well-known and widely adopted ones.
together with kernelized clustering [14]. However, non-Euclidean distance is sensitive to noise, which is also demonstrated in the experiments, and spectral clustering has been proved to be equivalent to kernel clustering [15,16].

In this paper, G-FCM is extended to a kernel-based method, called kernelized global fuzzy c-means (KG-FCM), and realized using two different kernels. First, KG-FCM is implemented using the Cauchy kernel to improve noise robustness, called KG-FCM with Cauchy kernel (KG-FCM-C). KG-FCM-C can be used to determine initial seeds efficiently and treat outliers in a robust way. KG-FCM is also implemented in another way using the random walk kernel [17,18] to accommodate non-convex clusters effectively, called KG-FCM with random walk kernel (KG-FCM-RW). KG-FCM-C is more noise resistant than existing methods and KG-FCM-RW is the most efficient one for clustering non-convex clusters.

In the next two sections, the hard and soft global clustering algorithms – global k-means and global fuzzy c-means, respectively – are briefly described. In Section 4, global fuzzy c-means is extended to kernelized global fuzzy c-means. Experimental results are given in Section 5 and discussions and further research are given in Section 6.

II. GLOBAL K-MEANS (GKM)

The k-means algorithm [19] is an iterative algorithm that finds K crisp and hyper-spherical clusters in the data such that an objective function defined in Eq. (1) is minimized:

$$E(V|X) = \sum_{i=1}^{N} \sum_{k=1}^{K} I(x_i \in C_k)||x_i - v_k||^2$$

(1)

where \(I(x_i \in C_k)\) is an indicator function having a value of one if data point \(x_i\) belongs to the kth cluster \(C_k\) and zero otherwise. Commonly, the K cluster centers, \(V = \{v_1, \ldots, v_K\}\), are initialized randomly with data points. Although k-means is one of the simplest learning algorithms and is still in use, one of its major problems is its sensitivity to initialization. To resolve this problem, several methods for initialization have been proposed and global k-means is one of them. Global k-means [7,8] is an incremental approach to clustering that dynamically adds one cluster at a time through a deterministic search procedure. The assumption on which the method is based is that an optimal clustering solution to the k clustering problem can be obtained through N local searches using k-means starting from an initial state with

1. the (k - 1) seeds placed at the centers from the (k - 1) clustering problem and
2. the remaining kth seed placed at a data point \(x_i(1 < i < N)\).

The original version of GKM requires \(O(N)\) executions of k-means for one new seed, which results in \(O(KN)\) executions of k-means and is not feasible with a large N. Thus a fast GKM algorithm was proposed, in which only one data point maximizing an objective function is considered as a new seed [7]. The objective function is defined as

$$f_k^i = \sum_{j=1}^{N} \max \left( \left( \delta_{k-1}^j \right)^2 - ||x_i - x_j||^2, 0 \right)$$

(2)

where \(\delta_{k-1}^j\) is the distance between \(x_i\) and the center closest to \(x_j\) among the (k-1) cluster centers obtained so far. The quantity \(f_k^i\) measures the guaranteed reduction in the objective function obtained by inserting a new cluster center at \(x_i\). The modified algorithm significantly reduces the number of k-means executions from \(O(KN)\) to \(O(K)\). The fast GKM algorithm is summarized in Fig. 1, where \(d\) is the dimensionality of data.

III. GLOBAL FUZZY C-MEANS (G-FCM)

With the introduction of fuzzy memberships, global fuzzy c-means (G-FCM) is a direct extension of global k-means (GKM) as FCM is a direct extension of k-means. k-means tries to minimize the sum of squared distances in Eq. (1), which can be extended to FCM by relaxing the indicator function to have continuous values with appropriate constraints. The objective function of FCM can be written as

$$E(U,V|X) = \sum_{i=1}^{N} \sum_{k=1}^{K} u_{ki}^m ||x_i - v_k||^2$$

(3)

where \(m\) is a fuzzifier constant and \(u_{ki}\) is the membership of the ith point to the kth cluster satisfying

$$0 \leq u_{ki} \leq 1$$

(4a)

$$\sum_{k=1}^{K} u_{ki} = 1$$

(4b)

$$0 \leq \sum_{i=1}^{N} u_{ki} \leq N$$

(4c)

G-FCM uses the same algorithm as GKM, except for two things. First, FCM is used instead of k-means. Equations (5) and (6) show the update equations for FCM [20]:

$$u_{ki} = \frac{||x_i - v_k||^{-\frac{m}{m-1}}}{\sum_{j=1}^{N} ||x_i - v_j||^{-\frac{m}{m-1}}}$$

(5)

$$v_k = \frac{\sum_{i=1}^{N} u_{ki}^m x_i}{\sum_{i=1}^{N} u_{ki}^m}$$

(6)

Another difference lies in the objective function for seed selection. Using Eq. (2), GKM tries to find a data point minimizing Eq. (1). Likewise, in G-FCM, one should find a data point minimizing Eq. (3). Equation (3) can be reformulated using Eq. (5) as [21]

$$E_{FCM}(V|X) = \sum_{i=1}^{N} \left( \sum_{k=1}^{K} ||x_i - v_k||^{-\frac{m}{m-1}} \right)^{1-m}$$

(7)

Fig. 1: GKM: Fast global k-means

Require: K : number of clusters, X : data set (N x d)
1: \(V_1 = \frac{1}{N} \sum_{i=1}^{N} x_i\)
2: for \(k = 2\) to \(K\) do
3: for \(l = 1\) to \(N\) do
4: \(f_k^l = \sum_{j=1}^{N} \max \left( \left( \delta_{k-1}^j \right)^2 - ||x_l - x_j||^2, 0 \right)\)
5: end for
6: \(a = \arg \max_{1 \leq i \leq N} f_k^l\)
7: \(V_k = V_{k-1} \cup \{x_a\}\)
8: \(\{V_k\} \leftarrow \text{k-means}(X, V_k)\)
9: end for
10: return \(V_K\)
The objective function for seed selection then becomes

\[ J_k^* = E_{FCM}(V_{k-1} \cup \{x_i\} | X). \]  

(8)

Therefore, Equation (8) should be divided into two according to the type of a center: one from the previous \((k-1)\) clustering problem and the other from a data point.

\[ J_k = \sum_{l=1}^{L} \left( \sum_{j=1}^{k-1} ||x_i - v_j||^{2/m} + ||x_i - x_l||^{2/m} \right)^{1-m}. \]  

(9)

The data point \(x_\alpha\) minimizing Eq. (9) is selected as the initial position of a new cluster center. The G-FCM algorithm is summarized in Fig. 2. One thing that should be noted is that memberships do not appear in the algorithm explicitly. However, all the algorithms proposed satisfy the constraints in Eq. (4) due to the re-formulation using Eq. (5).

**Require:** \(K\): number of clusters, \(X\): data set \((N \times d)\)

1: \(V_1 = \frac{1}{N} \sum_{i=1}^{N} x_i\)

2: for \(k = 2\) to \(K\) do

3: for \(l = 1\) to \(N\) do

4: \(J_k^* = \sum_{l=1}^{L} \left( \sum_{j=1}^{k-1} ||x_i - v_j||^{2/m} + ||x_i - x_l||^{2/m} \right)^{1-m}\)

5: end for

6: \(\alpha = \text{argmax}_{x \in ISN} J_k^* \)

7: \(V_k = V_{k-1} \cup \{x_\alpha\}\)

8: \([V_k] \leftarrow \text{FCM}(X, V_k)\)

9: end for

10: return \(V_K\)

Fig. 2. G-FCM: Global fuzzy c-means

**IV. KERNELIZED GLOBAL FUZZY C-MEANS (KG-FCM)**

Although global fuzzy c-means can be used to efficiently decide initial seeds, it still has drawbacks in that it is sensitive to outliers and it is limited to convex clusters. Several approaches have been proposed to alleviate these problems and kernel-based clustering was used in this paper.

Kernel methods were first introduced to clustering by Girolami [14], who proposed kernel k-means. Other kernel-based clustering algorithms were proposed after that [15]. The direct extension of FCM using kernels, kernel FCM (K-FCM), tries to minimize the following objective function:

\[ E(U, V | X) = \sum_{l=1}^{N} \sum_{k=1}^{K} u_{lk}^m ||\phi(x_i) - \phi(v_k)||^2, \]  

(10)

where \(\phi()\) is a mapping function. The update equations for K-FCM can be written as

\[ u_{lk} = \frac{||\phi(x_i) - \phi(v_k)||^{2-m}}{\sum_{j=1}^{K} ||\phi(x_i) - \phi(v_j)||^{2-m}}. \]  

(11)

\[ \phi(v_k) = \frac{\sum_{l=1}^{N} u_{lk}^{m+1} \phi(x_i)}{\sum_{l=1}^{N} u_{lk}^{m+1}}. \]  

(12)

Due to the dimensionality of the mapped data, however, Eqs. (11) and (12) cannot be evaluated directly and one has to use a kernel trick to evaluate the values indirectly using kernels. To make KG-FCM noise-robust, instead of the commonly used Gaussian kernel, the Cauchy kernel was used, which results in kernel FCM with Cauchy kernel (K-FCM-C). The Cauchy kernel has been shown to be outlier-robust and able to effectively accommodate clusters with different densities compared to the Gaussian kernel [10]. The Cauchy kernel can be defined as

\[ k_C(x, y) = \frac{1}{1 + \beta ||x-y||^2}, \]  

(13)

where \(\beta\) is a kernel parameter. Using Eq. (13), the update equations can be written as [10]

\[ u_{kl} = \frac{(1-k_C(x_l, v_k))^{m-1}}{\sum_{l=1}^{L}(1-k_C(x_l, v_j))^{m-1}}, \]  

(14)

\[ v_k = \frac{\sum_{l=1}^{L} u_{lk}^m k_C(x_l, v_k)^2 x_l}{\sum_{l=1}^{L} u_{lk}^m k_C(x_l, v_k)^2}. \]  

(15)

Although the coordinates of cluster centers in the feature space (Eq. (12)) cannot be evaluated due to dimensionality, the corresponding ones in the input space (Eq. (15)) can be calculated when the Cauchy kernel is used. In Eq. (15), the kernel function \(\kappa(x_i, v_l)\) works as a weighting function. The function \(\kappa(x_i, v_l)\) weights a data point based on the similarity between \(x_i\) and \(v_l\), which results in noise robustness. To do a global search, Eq. (9) also should be kernelized as

\[ J_k^* = \sum_{l=1}^{L} \left( \sum_{j=1}^{k-1} ||\phi(x_i) - \phi(v_j)||^{2/m} + ||\phi(x_i) - \phi(v_l)||^{2/m} \right)^{1-m}. \]  

(16)

which can be simplified as

\[ J_k^* = \sum_{l=1}^{L} \left( 2 - 2k_C(x_l, v_l) \right)^{2/m}. \]  

(17)

The final consideration is the evaluation order of the update equations. At the beginning of a \(k\) clustering problem, the membership in Eq. (14) must be calculated first to incorporate the current problem into the prior information – the optimal clustering result from the \((k-1)\) clustering problem and the data point minimizing Eq. (17). Figure 3 summarizes K-FCM-C. Kernelized global FCM with Cauchy kernel (KG-FCM-C) is summarized in Fig. 4. Together with its non-linear property, the proposed algorithm can be used to decide initial seeds efficiently and is more robust to outliers than existing methods. However, it still cannot efficiently accommodate non-convex clusters due to its noise suppressing property. This problem can be tackled by the introduction of a random walk kernel.
The random walk distance is a graph theoretic distance, in which each data point is considered as a node and the distance between two points is defined using the Markov property. The random walk distance decreases. The average commute time can be calculated in terms of the pseudo-inverse of Laplacian matrix, \( L^+ \), as in Fig. 5.

\[
\text{Fig. 5. Average commute time}
\]

The cluster center cannot be evaluated in the feature space nor in the input space when the random walk kernel is used, however, \( \kappa_{RW}(x, y) \) can be evaluated as a weighted sum of dot-products between data points,

\[
\kappa_{RW}(x, y) = \frac{\sum_{i=1}^{N} u_i^{m} \kappa_{RW}(x_i, x_j)}{\sum_{i=1}^{N} u_i^{m}}.
\]  

Equation (20) can be used to update membership values in Eq. (19) and to calculate the objective function for seed selection in Eq. (16). The algorithm in Fig. 3 also should be modified, because the coordinates of centers cannot be evaluated in K-FCM-RW and the membership values describe the centers indirectly. Thus, the set of centers, \( V \), in Fig. 3 should be replaced with the membership matrix \( U \), whose \( k \)th column represents the membership vector to the \( k \)th cluster.

**Algorithm**

Require: \( K \) : number of clusters, \( X \) : data set \( (N \times d) \)

1. \( V_1 = \frac{1}{N} \sum_{i=1}^{N} x_i \)
2. for \( k = 2 \) to \( K \) do
3. for \( l = 1 \) to \( N \) do
4. \( f_{kl} = \sum_{i=1}^{N} \left( \left( -2 \kappa_c(x_i, x_j) \right) \left( x_i \right) \right)^{2} \)
5. \( \alpha = \arg \min_{\alpha \leq 0} f_{kl} \)
6. \( V_k = V_k \cup \{ x_i \} \)
7. \([V_k, U_k] \leftarrow K-FCM-C(X, V_{k-1}, \alpha, k)\)
8. end for
9. end for
10. return \( V_k, U_k \)

**Fig. 3. K-FCM-C: Kernel FCM with the Cauchy kernel**

where \( \sigma \) is a kernel parameter. The update equation for K-FCM with random walk kernel (K-FCM-RW) can be written in a way similar to Eq. (14) as

\[
u_{kl} = \frac{1}{\sum_{i=1}^{N} (1-\kappa_{RW}(x_i, x_k))^{-m}} (1-\kappa_{RW}(x_i, x_k))^{-m}.
\]  

Equation (19) can be used to update membership values in K-FCM-RW and the membership values describe the center indirectly. Thus, the set of centers, \( V \), in Fig. 3 should be replaced with the membership matrix \( U \), whose \( k \)th column represents the membership vector to the \( k \)th cluster.

**Algorithm**

Require: \( K \) : number of clusters, \( X \) : data set \( (N \times d) \)

1. \( V_1 = V_{k-1} \cup \{ x_i \} \)
2. repeat
3. for \( k = 1 \) to \( K \) do
4. for \( i = 1 \) to \( N \) do
5. \( u_{ki} = \frac{1}{\sum_{j=1}^{N} (1-\kappa_{RW}(x_i, x_j))^{-m}} (1-\kappa_{RW}(x_i, x_j))^{-m} \)
6. end for
7. \( K-FCM-RW(X, V_{k-1}) \)
8. until \( U \) satisfies the convergence criterion
9. return \( U_k \)

**Fig. 6. K-FCM-RW: Kernel FCM with the random walk kernel**
Require: \( K \): number of clusters, \( X \): data set \((N \times d)\)
1: \( U_i = [1 \, 1 \, \ldots \, 1] \)
2: for \( k = 2 \) to \( K \) do
3: \( I_k \) = \( \sum_{l=1}^{N} \left( \frac{1}{\sqrt{1 - \phi(x_l) - \phi(v_l)}} \right)^2 \)
4: \( \alpha = \arg\min_{1\leq l \leq N} I_k \)
5: \( [U_i] \leftarrow K\text{-FCM-RW}(X, U_{k-1}, \sigma, k) \)
6: end for
7: return \( U_K \)

Fig. 7. KG-FCM-RW: Kernelized global FCM with the random walk kernel

In this section, four kernelized algorithms were developed, two using explicit initialization (K-FCM-*) and two using global initialization (KG-FCM-*). A global method is a wrapper method of the corresponding explicit initialization method and decides initial seeds incrementally. Table 1 compares the methods described in this paper.

### TABLE I. COMPARISON OF CLUSTERING METHODS

<table>
<thead>
<tr>
<th>Method</th>
<th>Cluster boundary</th>
<th>Initialization</th>
<th>Kernel</th>
</tr>
</thead>
<tbody>
<tr>
<td>FCM</td>
<td>convex</td>
<td>random</td>
<td>-</td>
</tr>
<tr>
<td>G-FCM</td>
<td>convex</td>
<td>global</td>
<td>-</td>
</tr>
<tr>
<td>K-FCM-C</td>
<td>non-convex</td>
<td>random</td>
<td>Cauchy</td>
</tr>
<tr>
<td>KG-FCM-C</td>
<td>non-convex</td>
<td>global</td>
<td>Cauchy</td>
</tr>
<tr>
<td>K-FCM-RW</td>
<td>non-convex</td>
<td>random</td>
<td>random walk</td>
</tr>
<tr>
<td>KG-FCM-RW</td>
<td>non-convex</td>
<td>global</td>
<td>random walk</td>
</tr>
</tbody>
</table>

V. EXPERIMENTAL RESULTS

To investigate the effectiveness of the proposed method, the existing and proposed methods were implemented and tested using Matlab. We tested the methods in Table 1 using artificial and UCI data sets.

A. Experiments using artificial data sets

In the first set of experiments, the usefulness of the global search technique is shown.

![Figure 8](image8.jpg)

Figure 8 shows a clustering result using FCM. The test data \((D_7)\) consists of seven well-separated clusters generated from Gaussian distributions with means \( \mu = [0 \, 7 \, 7 \, 7 \, 14 \, 14; \, 0 \, 7 \, 0 \, 7 \, 14 \, 0 \, 7] \) and equal covariance matrix \( \Sigma = I \). The large dots represent the random initial seeds used in FCM. Although FCM usually finds the correct structure, it sometimes fails due to the convergence to a local optimum even in a data set with well-separated clusters. For example, only three clusters out of seven are correctly identified in Fig. 8. Figure 9 shows the initial seeds in G-FCM, which are decided in a deterministic way. Conceptually, KG-FCM works exactly the same way as in Fig. 9, but is difficult to visualize due to the dimensionality of the mapped data.

![Figure 9](image9.jpg)

Table 2 summarizes the clustering results using \( D_7 \). The numbers are averaged over 100 runs. FCM occasionally falls into a local optimum. K-FCM-C is a little better than FCM, but sometimes it also falls into a local optimum. K-FCM-RW is the least satisfactory because the random walk distance is based on the path connectivity between points, which is sensitive to outliers. The large variance for K-FCM-RW also corroborates this. The two global methods, G-FCM and KG-FCM-C, always find the correct structure of \( D_7 \). KG-FCM-RW is much better than K-FCM-RW, but not as good as other global methods.

### TABLE II. EXPERIMENTAL RESULTS USING \( D_7 \) (\( M = \# \) OF CLUSTERS CORRECTLY IDENTIFIED)

<table>
<thead>
<tr>
<th>Method</th>
<th>( M )</th>
<th>( \text{var}(M) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>FCM</td>
<td>6.82</td>
<td>0.4925</td>
</tr>
<tr>
<td>G-FCM</td>
<td>7.00</td>
<td>0.0000</td>
</tr>
<tr>
<td>K-FCM-C</td>
<td>6.92</td>
<td>0.3168</td>
</tr>
<tr>
<td>KG-FCM-C</td>
<td>7.00</td>
<td>0.0000</td>
</tr>
<tr>
<td>K-FCM-RW</td>
<td>6.42</td>
<td>3.1164</td>
</tr>
<tr>
<td>KG-FCM-RW</td>
<td>6.96</td>
<td>0.0788</td>
</tr>
</tbody>
</table>
To test noise robustness, some noise points were added to $D_7$ and the previous experiments were repeated. Noise points were sampled from a uniform distribution and the noise ratio is given as the ratio of the number of noise points to the number of data points. Figure 10 shows the number of correctly identified clusters with respect to noise ratio. For a clear comparison, two subsets out of six methods are plotted on different scales and the numbers are averaged over 100 runs. From Fig. 10(a), we can conclude that the global methods are more robust to noise than the random initialization methods, and KG-FCM-C is best. As the noise ratio increases, the random initialization methods tend to fall into a local optimum, while the global methods do not suffer noise problems and find the right structure. In Fig. 10(b), we can see that K-FCM-RW collapses as the noise ratio increases. Although the global initialization helps, KG-FCM-RW also fails to find the right structure in noisy conditions.

![Fig. 10. Number of clusters correctly identified with respect to noise ratio using $D_7$](image)

In this experiment, the methods using global initialization showed almost the same result as the corresponding methods with random initialization, so the results of the global methods are not plotted. As is clear from Fig. 12, FCM with Mahalanobis distance achieves the best result when $d_{between}$ is small because the distance is specialized for Gaussian distributions. However, it sometimes fails to separate the two clusters even when the $d_{between}$ is sufficiently large, because the number of data points in $D_{parallel}$ is not large enough to estimate parameters, including the covariance matrix. The overall large variance also supports that. Another interesting point in Fig. 13 is that the variance for K-FCM-C decreases as $d_{between}$ increases. On the other hand, FCM and K-FCM-RW have peaks, which can be considered as a threshold that determines marginal performance. FCM with Mahalanobis distance shows relatively large variances for all values of $d_{between}$ and does not show a clear relationship between $d_{between}$ and the variance of error rate.

![Fig. 11. Clustering results using (a) FCM and (b) K-FCM-C with random initialization](image)

![Fig. 12. Error rate with respect to the distance between two cluster centers in $D_{parallel}$](image)

![Fig. 13. Variance of error rate with respect to the distance between two cluster centers in $D_{parallel}$](image)
Although the random walk kernel is noise-sensitive, it is suitable for non-convex clusters. Figure 14 shows a sample data set that is not convex ($D_{\text{non-convex}}$). FCM and K-FCM-C cannot separate the two clusters in Fig. 14 for two different reasons. First, FCM fails to separate the clusters mainly due to the mean squares measure, so FCM tends to separate $D_{\text{non-convex}}$ in a vertical direction (Fig. 15(a)). Second, K-FCM-C fails to separate the clusters due to its noise suppressing property. As a result, K-FCM-C considers the points at the end of semi-circles as outliers and tends to separate $D_{\text{non-convex}}$ in a horizontal direction (Fig. 15(b)). Figure 16 shows the error rate with respect to the distance depicted in Fig. 14. As the distance decreases, the error rate of FCM increases. K-FCM-C is slightly better than FCM, but also fails to separate two clusters although the cluster boundaries in K-FCM-C are not convex. Only K-FCM-RW can correctly separate the non-convex clusters, but it also fails when two clusters are too close.

Fig. 14. Non-convex clusters and the distance between two clusters

(a) (b)

Fig. 15. Clustering results using (a) FCM and (b) K-FCM-C

Fig. 16. Error rate with respect to the distance between two clusters $D_{\text{non-convex}}$

B. Experiments using real world data sets

The proposed methods were also applied to some UCI data sets available from the UCI Machine Learning Repository [23]. To compare the results using different clustering methods, an information-based distance, a simplified variation of information (VI) was used. As VI is a true metric on the space of clusterings and retains the comparability over different experimental conditions, it has strength over other measures. As we compared a clustering to ground truth, a modified VI was used. To obtain more detailed information about VI measure, refer to [24] and the references therein.
Suppose two different clusterings, $C = \{C_1, \ldots, C_K\}$ and $C' = \{C_1', \ldots, C_K'\}$ (corresponds to the set of known labels), which cluster the data into $K$ and $K'$ clusters respectively. The probability of a point being in cluster $C_i$ equals $p(k) = N_k / N$, where $N_k$ is the number of points belonging to the $k$th cluster and $N$ is the number of data points. The numbers satisfy

$$N = \sum_{k=1}^{K} N_k = \sum_{k=1}^{K'} N'_k. \quad (21)$$

The information contained in clustering $C$ can be represented as the entropy of the random variable $p(k)$,

$$H(C) = -\sum_{k=1}^{K} p(k) \log p(k), \quad (22)$$

and the amount of information common to both clusterings can be represented as the mutual information of the two random variables, $p(k)$ and $p(k')$,

$$I(C, C') = \sum_{k=1}^{K} \sum_{k'=1}^{K'} p(k, k') \log \frac{p(k, k')}{p(k)p(k')}, \quad (23)$$

where $p(k, k') = |C_k \cap C_{k'}| / N$ is the probability that a point belongs to $C_k$ in clustering $C$ and to $C_{k'}$ in clustering $C'$. Then the difference between the two terms represents the amount of information that is not described by a clustering $C$:

$$D_I(C, C') = H(C) - I(C, C'), \quad (24)$$

Equation (24) is zero if and only if the two clusterings are identical when $K = K'$, and can be considered as a distance of a clustering ($C$) from another clustering or ground truth ($C'$). As the number of clusters in $C$ increases, the mutual information increases and $D_I$ decreases. Experimentally, clustering with large number of clusters, $K > K'$, showed a smaller $D_I$ than that with the same number of clusters, $K = K'$, but the relative performance did not change. Therefore, the number of clusters ($K$) is set to be equal to the number of classes ($K'$) in this paper. Table 3 summarizes the experimental results using UCI data sets. DI values for random methods in Table 3 are averaged over 100 runs.

- Iris: On average, the global method is better than the corresponding random method, and the kernel-based methods are better than the input space method. KG-FCM-RW is best for the iris data because the iris data is relatively noise-free and one of the three clusters is clearly separated from the others. One interesting thing is that K-FCM-RW is likely to fall into a local optimum. Conversely, FCM and K-FCM-C rarely fall into a local optimum.

- Wine: As before, the kernel-based methods show better results, and KG-FCM-C shows the best result. One thing that should be noted is that K-FCM-RW with random initialization hits the minimum and, on average, K-FCM-RW is better than KG-FCM-RW.

- Breast cancer: Although the number of classes is only two, the dimensionality is higher than those in previous data sets. Again, the global methods are better than the random methods, but K-FCM-RW performs worst. This is mostly due to the small size of the data, which distributes the data sparsely in high dimensional space and makes it difficult to estimate the connectivity information. In this experiment, KG-FCM-C also fails to achieve the minimum of K-FCM-C.

- Yeast: Only 5 of 10 classes were used in this experiment. The results are almost the same as the previous ones using the breast cancer data. As the number of clusters increases, K-FCM-RW becomes more unstable and K-FCM-RW performs worst in this experiment.

From the experiments using UCI data sets, we can conclude that:

- The global initialization is useful for deciding initial seeds, although the global methods sometimes fail to achieve the minimum of the corresponding random methods. This failure is mainly due to the different sets of possible initial values, which originates from the fact that the random methods can only have data points as their initial positions but the global methods can have non-occupied positions.

<table>
<thead>
<tr>
<th>Data set</th>
<th>N</th>
<th>d</th>
<th>K'</th>
<th>Global method</th>
<th>DI</th>
<th>Random method</th>
<th>DI</th>
<th>D_I</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris</td>
<td>150</td>
<td>4</td>
<td>3</td>
<td>G-FCM</td>
<td>0.4041</td>
<td>FCM</td>
<td>0.4041</td>
<td>0.4042</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>KG-FCM-C</td>
<td>0.3898</td>
<td>K-FCM-C</td>
<td>0.3898</td>
<td>0.3898</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>KG-FCM-RW</td>
<td>0.2663</td>
<td>K-FCM-RW</td>
<td>0.6057</td>
<td>0.6164</td>
</tr>
<tr>
<td>Wine</td>
<td>178</td>
<td>13</td>
<td>3</td>
<td>G-FCM</td>
<td>1.2944</td>
<td>FCM</td>
<td>1.2944</td>
<td>1.3088</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>KG-FCM-C</td>
<td>1.2284</td>
<td>K-FCM-C</td>
<td>1.2284</td>
<td>1.2368</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>KG-FCM-RW</td>
<td>1.2683</td>
<td>K-FCM-RW</td>
<td>1.1542</td>
<td>1.2351</td>
</tr>
<tr>
<td>Breast</td>
<td>569</td>
<td>30</td>
<td>2</td>
<td>G-FCM</td>
<td>0.9329</td>
<td>FCM</td>
<td>0.9329</td>
<td>0.9339</td>
</tr>
<tr>
<td>cancer</td>
<td></td>
<td></td>
<td></td>
<td>KG-FCM-C</td>
<td>0.8712</td>
<td>K-FCM-C</td>
<td>0.8243</td>
<td>0.9224</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>KG-FCM-RW</td>
<td>0.9381</td>
<td>K-FCM-RW</td>
<td>0.9202</td>
<td>0.9393</td>
</tr>
<tr>
<td>Yeast</td>
<td>180</td>
<td>8</td>
<td>5</td>
<td>G-FCM</td>
<td>1.6129</td>
<td>FCM</td>
<td>1.6043</td>
<td>1.6139</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>KG-FCM-C</td>
<td>1.5855</td>
<td>K-FCM-C</td>
<td>1.8001</td>
<td>1.8001</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>KG-FCM-RW</td>
<td>1.8021</td>
<td>K-FCM-RW</td>
<td>1.6755</td>
<td>1.8430</td>
</tr>
</tbody>
</table>

(N : # of data points, d : data dimension, K' : # of classes)
• KG-FCM-C is more robust to outliers than other methods and shows better and more stable results on test data sets.

• KG-FCM-RW shows better result on the iris data, which is clean and has a relatively well separated cluster structure. However, as the dimensionality of data or the number of clusters increases, it is more likely to fall into a local optimum.

VI. DISCUSSION

Fuzzy c-means (FCM) is a simple and efficient method for clustering. However there are also several well-known shortcomings with FCM. In this paper, we developed kernelized global fuzzy c-means (KG-FCM) to solve the problems in FCM: sensitivity to initialization, sensitivity to noise, and inability to accommodate non-convex clusters. KG-FCM is based on global fuzzy c-means (G-FCM) then extended with the help of kernel methods. KG-FCM is implemented using two different kernels: the Cauchy kernel to suppress noise and the random walk kernel to accommodate non-convex clusters. Experimental results show that, on average, the global methods are superior to the random methods, and KG-FCM-C gave the best results for test data sets most of the time. KG-FCM-RW shows good results only when data are clean and have well-separated clusters, for example, the iris data.

Although the proposed algorithm, KG-FCM, is better than existing methods, it can still be further improved. Initially, kernel methods were adopted because of the various effects obtained using different kernels. As is clear from the experiments, the Cauchy kernel is good at suppressing outliers and the random walk kernel is efficient at separating non-convex clusters. However, each kernel has the other’s strength as its weakness. Still another kernel may combine the convex clusters. Nevertheless, each kernel has its other’s strength and the random walk kernel is efficient at separating non-convex clusters. Experimental results show that, on average, the global methods are superior to the random methods, and KG-FCM-C gave the best results for test data sets most of the time. KG-FCM-RW shows good results only when data are clean and have well-separated clusters, for example, the iris data.

REFERENCES


