Robust Heterogeneous C-means

Atieh Gharib, Hadi Sadoghi-Yazdi, Amirhossein Taherinia

Department of Computer Engineering, Ferdowsi University of Mashhad, Mashhad, Iran

Center of Excellence on Soft Computing and Intelligent Information Processing, Ferdowsi University of Mashhad, Mashhad, Iran

ARTICLE INFO

Article history:
Received 2 March 2018
Received in revised form 23 September 2019
Accepted 17 October 2019
Available online 22 October 2019

Keywords:
Robust
Loss function
Clustering
C-Means
Half-quadratic
Heterogeneous

ABSTRACT

Fuzzy c-means is one of the popular algorithms in clustering, but it has some drawbacks such as sensitivity to outliers. Although many correntropy based works have been proposed to improve the robustness of FCM, fundamentally a proper error function is required to apply to FCM. In this paper, we present a new perspective based on the expected loss (or risk) to FCM method to provide different kinds of robustness such as robustness to outliers, to the volume of clusters and robustness in noisy environments. First, we propose Robust FCM method (RCM) by defining a loss function as a least square problem and benefiting the correntropy to make FCM robust to outliers. Furthermore, we utilize the half-quadratic optimization as a problem-solving method. Second, inspiring by the Bayesian perspective, we define a new loss function based on correntropy as a distance metric to present Robust Heterogeneous C-Means (RHCM) by utilizing direct clustering (DC) method. DC helps RHCM to have robust initialization. Besides, RHCM will make some robust cluster centers in noisy environments and is capable of clustering the elliptical or spherical shaped data accurately, regardless of the volume of each cluster. The results are shown visually on some synthetic datasets including the noisy ones, the UCI repository and also on real image dataset that was gathered manually from 500px social media. Also, for evaluation of the clustering results, several validity indices are calculated. Experimental results indicate the superiority of our proposed method over the base FCM, DC, KFCM, two new methods called GPFCM and GEPFCM and a method called DC-KFCM that we created for the comparison purpose.

1. Introduction

Fuzzy C-Means (FCM) [1] has been known as one of the most powerful clustering algorithm. Since it uses Euclidean norm as similarity measurement, it is effective in clustering spherical shaped objects. Due to several drawbacks of FCM such as sensitivity to initialization, sensitivity to outliers, and limitation of equal-sized and spherical-shaped clusters, different versions of FCM have been developed to improve its performance [2–4]. Krishnapuram and Keller [4] proposed the possibilistic c-means (PCM) algorithm to improve the weakness of FCM in a noisy environment.

PCM determines a possibilistic partition in which a possibilistic type of membership function is used to illustrate the degree of belongings [4]. PCM relaxes the constraint of the sum of memberships to 1 for all data points so that it can ignore noisy and outlier data. But it depends heavily on the initial parameters and suffers from coincident clusters problem [5–8].

Pal et al. [9] also proposed fuzzy-possibilistic c-means (FPCM) model that combines the characteristics of both fuzzy and possibilistic c-means to overcome their problems but it has the constraint in which the sum over all data points of typicalities to a cluster is one. This row sum constraint may cause producing unrealistic results for large data sets. So, possibilistic-fuzzy c-means (PFCM) model introduced in [10] produces simultaneously memberships, possibilities and cluster centers for each cluster and solves the noise sensitivity defect of FCM, the coincident clusters problem of PCM and the row sum constraints problem of PFCM [5,10]. PCM was also extended by Yang and Wu [11]. Robust analysis of nonprecise data was focused by Ferraro et al. [12] on the basis of a fuzzy and possibilistic clustering method in which determining the parameter values are based on the generalized form of the Xie and Beni validity index.

In PCM, each object has equal importance in the clustering process. So the weighted PCM (WPCM) model [13,14] defines weights for each object according to their relative importance in clustering solution that can minimize the effects of noisy data.

In [15] Wu and Yung presented AFCM to replace the Euclidean norm in c-means clustering algorithm by exponential metric to make it robust to noise and outliers. It can also tolerate unequal sized clusters. However, it does not guarantee to result in homogeneous clusters. To address this problem, Wu and Yung [16]
introduced the Possibilistic version of AF CM as an unsupervised alternating clustering method called UACM. UACM can create denser regions and more accurate cluster centers. Also, a new alternative weighted FCM (AWFCM) proposed by Xiang et al. [17] suggests a new distance metric and a weight matrix based on sample dots’ density to overcome the bugs of FCM in dealing with mass shape partitions and prevents the error of cluster center for noisy samples [18].

Another development is to use kernel version of FCM proposed by Girolami [19] to map the nonseparable input data into a higher dimensional feature space via a nonlinear data transformation that can cluster more general shape of objects [2, 19, 20]. KFCM has better outlier and noise robustness than FCM, so it is used in [21, 22] to deal with incomplete data. In KFCM the values of the kernel parameters affect the final clustering results [2, 23]. In [23] Yang and Tsai proposed a Gaussian kernel-based FCM (GKFCM) with a spatial bias correction that solves this drawback. Ferreira and Carvalho [24] proposed a variable-wise kernel fuzzy c-means that uses adaptive distances. It means the relevance weight of each variable to each cluster may be different and dissimilarity measures are obtained as a sum of kernel functions applied to each variable.

In [25] Shanmugapriya and Punithavalli proposed a Kernelized Fuzzy Possibilistic CMeans (KFPCM) algorithm in which FPCM combines the advantages of both FCM and PCM and a kind of Kernel-Induced Distance Measure is used to obtain better clustering results in case of high dimensional data. It is also robust to outliers and noise. However, FCM has the ability to overcome the curse of dimensionality problem by forming information granules as used in [26]. In [6] Zhang and Chen presented a weighted kernel FCM (WkPCM) algorithm with combination of WPCM algorithm and PCM [27] to reduce the corruption caused by noisy data and to cluster big data effectively.

In addition, a generalized form of possibilistic fuzzy c-means (GFPFCM) [28] and a Generalized Entropy based Possibilistic Fuzzy C-Means algorithm (GEPPFCM) [29] have been developed for clustering noisy data. Although these methods can find accurate cluster centers in noisy environments, when the sizes of clusters are considerably different, they fail.

Since it is hard to find a classifier that simultaneously consider the uncertainty on both membership values and distance measure while the initialization is important, the data environment is noisy, and the clusters are of different shapes and sizes, we want to present some models for defining a fuzzy classifier that covers all of these issues.

In fact, in this paper we present a new perspective based on the expected loss (or risk) to FCM method. The help of this new perspective can generate different kinds of FCM including the robust versions. We first present a new robust FCM called R$_n$CM by defining an expected loss according to the error. Our definition is based on the correntropy induced loss function utilizing the half-quadratic optimization solving method to make RCM robust to outliers. Then utilizing this correntropy based method and inspired by the Bayesian perspective, we define another expected loss according to the data and parameters to show another view to robustness. The new generated robust method called R$_n$CM uses Direct clustering (DC) algorithm [30] which is an equivalent method to the weighted FCM to be robust in the initialization step. Inspired by DC, we add the concept of heterogeneity as a constraint to our problem to also make clusters robust to the volume of data in noisy environments.

The rest of the paper is structured as follows. Traditional FCM is reviewed in Section 2 and then for developing our models, FCM from the expected loss perspective is considered. In Section 3, RCM method is presented. In Section 4, direct clustering algorithm and some drawbacks of this method are reviewed. Then R$_n$CM method is introduced to solve the previous mentioned problems. Section 5 offers some experimental results both visually and in terms of cluster validity indices. Finally, conclusions are made in Section 6.

2. FCM from the expected loss perspective

Traditional FCM method is first reviewed in this section. Then, the expected loss perspective to FCM method is presented.

2.1. Traditional FCM method

Let $X = \{x_1, x_2, \ldots, x_n\}$ be a set of $n$ data samples in $d$-dimensional space. The FCM algorithm [31] tries to minimize the objective function defined in (1) to cluster the data into $c$ subsets. In FCM, each data point belongs to more than one cluster with different membership degrees ranging in $[0,1]$. So, the following objective function should satisfy some conditions as

$$J_n(u,v) = \sum_{i=1}^{n} \sum_{j=1}^{c} u_{ij} \|x_i - v_j\|^2$$

s.t.

$$\sum_{j=1}^{c} u_{ij} = 1 \quad \forall i, \quad 0 \leq \sum_{i=1}^{n} u_{ij} < n$$

Where, $v_j$ is the center of $j$-th cluster that needs to be found and $u_{ij}$ is the membership degree of $i$-th data sample to the $j$-th cluster.

2.2. FCM via the expected loss

In this section, we present a new perspective based on the expected loss (or risk) to FCM method. The help of this new perspective can generate different kinds of FCM including the robust versions. So, we briefly review the role of the loss function. To develop our models we can have two views to the loss function:

First, to define an expected loss according to the error as

$$E[l(e)] \simeq \frac{1}{c} \sum_{i=1}^{c} l(e_i)$$

if $l(e_i) = e_i^2$ then:

$$E[l(e)] = \sum_{i=1}^{n} e_i^2$$

Second, to define an expected loss according to the data and parameters as

$$E[l(x,v)] = \int_{x,v} l(x,v)f(x,v)dxdv$$

or

$$E[l(x,v)] = \frac{1}{nc} \sum_{x} \sum_{v} l(x,v)f(x,v)$$

where $c$ is the class numbers and $n$ is the number of samples. $f(x,v)$ is a distribution function and $l(x,v)$ can be any user defined loss.

Since we know the ability of Bayesian approach both to integrate the expert prior knowledge into clustering and to evaluate a noisy model with different kinds of noise in clustering, we want to define a proper classifier that handles the uncertainty on both membership values ($u$) and distances ($d$) while considering the
noisy data. So, we propose a fuzzy classifier based on the Bayesian approach by developing (4) as
\[
E[l(x, v)] = \sum_{i=1}^{n} \sum_{j=1}^{c} l(x, v) f(x_i | v_j) f(v_j)
\]  
(5)
If we let \( l(x, v) = \|x - v\|^2 \) then:
\[
E[l(x, v)] = \sum_{i=1}^{n} \sum_{j=1}^{c} \|x_i - v_j\|^2 u_i^m f(v_j)
\]  
(6)
or if \( l(x, v) = 1 - \exp(-\frac{\|x - v\|^2}{2\sigma^2}) \) then:
\[
E[l(x, v)] = \sum_{i=1}^{n} \sum_{j=1}^{c} (1 - \exp(-\frac{\|x_i - v_j\|^2}{2\sigma^2})) u_i^m f(v_j)
\]  
(7)
As it is shown in Eqs. (6), (7) we define \( u_i^m = f(x_i | v_j) \) which means that, given the center of \( j - \) th cluster, how much the degree of belonging of \( i - \) th data point is. Also, We extend the definition of the weights for each data in [32] to the weights for each cluster. So, we define the prior knowledge \( f(v_j) \) as the weight of each cluster. As a result, one can consider a lower weight for the noisy clusters as a prior knowledge.

3. Robust FCM method (RCM)

Starting from defining a proper loss function, we extend the RCM model based on the mentioned expected loss perspective in the following.

3.1. Formulation of RCM

The development of our RCM model is based on the expected loss defined in Eq. (3) of Section 2.2.

If the loss is squared error, we have the following minimization problem:
\[
\min_{u, v} J(u, v) = \sum_{i=1}^{n} e_i^2
\]  
\[s.t. \]
\[
e_i^2 = c \sum_{j=1}^{c} u_i^m d(x_i, v_j),
\]  
(8)
\[
\sum_{j=1}^{c} u_j = 1, \quad i = 1, \ldots, n
\]
where (8) can be seen as a constrained least square problem that is going to be robust by applying the correntropy loss function as \( l(e) = 1 - \exp(-\eta e^2) \). Where \( \eta \) is the scaling constant.

So the minimization of (8) is equivalent to the maximization of the following function that is applied to FCM:
\[
\max_{u, v} J(u, v) = \sum_{i=1}^{n} \exp(-\eta e_i^2)
\]  
\[s.t. \]
\[
e_i^2 = c \sum_{j=1}^{c} u_i^m d(x_i, v_j),
\]  
(9)
\[
\sum_{j=1}^{c} u_j = 1, \quad i = 1, 2, \ldots, n
\]
Using the half-quadratic problem solving method (see Appendix A), one can rewrite the cost function \( J(u, v) \) in (9) as
\[
J(u, v) = \sum_{i=1}^{n} \sup_{p < 0} \{\eta^2 e_i^2 p_i - \varphi(p_i)\}
\]  
(10)
\[
= \sup_{p < 0} \{\sum_{i=1}^{n} (\eta^2 e_i^2 p_i - \varphi(p_i))\}
\]
the second equation in (10) establishes since \( \eta^2 e_i^2 p_i - \varphi(p_i) \) are independent functions in terms of \( p_i \). So, we can derive that (9) is equivalent to
\[
\max_{u, v, p < 0} J(u, v, p) = \sum_{i=1}^{n} (\eta^2 e_i^2 p_i - \varphi(p_i))
\]  
(11)
Now, we can optimize (11) using the alternating optimization method. It means, iteratively given \( (u, v) \) we optimize over \( p \) and then given \( p \), we optimize over \( (u, v) \). If we define the subscript \( t \) as the \( t \)th iteration, then given \( (u_t, v_t) \), (11) is equivalent to
\[
\max_{p_t < 0} \sum_{i=1}^{n} (\eta^2 e_i^2 p_t - \varphi(p_t))
\]  
(12)
Whose analytic solutions are
\[
p_t = -\exp(-\eta^2 e_i^2) < 0, \quad i = 1, 2, \ldots, n
\]
Then, after obtaining \( p_t \), we can get \( (v_{t+1}, u_{t+1}) \) by solving the following problem:
\[
\max_{u_{t+1}, v_{t+1}} \sum_{i=1}^{n} \eta^2 e_i^2 p_t - \varphi(p_t)
\]  
(13)
For clarity, we omit the subscripts \( t, t+1 \). By defining \( q_i = -p_t = \exp(-\eta^2 e_i^2) \) the equivalent problem of (13) is:
\[
\min_{u, v} \sum_{i=1}^{n} \eta^2 e_i^2 q_i
\]  
\[= \sum_{i=1}^{n} \sum_{j=1}^{c} \eta^2 u_i^m d(x_i, v_j) q_i
\]  
\[= \sum_{i=1}^{n} \sum_{j=1}^{c} \eta^2 u_i^m \|u_j - x_i\|^2 q_i
\]  
(14)
\[s.t. \]
\[
\sum_{j=1}^{c} u_i = 1, \quad i = 1, 2, \ldots, n
\]
Where \( \|u_j - x_i\|^2 \) is the Euclidean distance.

After producing the Lagrange relation for (14) the centers and memberships are acquired as follows:
\[
v_j = \frac{\sum_{i=1}^{n} \eta^2 u_i^m q_i x_i}{\sum_{i=1}^{n} \eta^2 u_i^m q_i},
\]  
(15a)
\[
u_j = \frac{1}{\sum_{k=1}^{c} \frac{d(x_i, v_j)}{\beta(x_i, v_j)}} \frac{1}{\beta(x_i, v_j)}
\]  
(15b)
As it is shown in Eq. (15a) RCM presents a new relation for centers. It will be shown in Section 5 that it is a good solution in noisy environments and in the presence of outliers.

The proposed RCM procedure is shown in Algorithm 1.

3.2. Convergence of RCM

**Proposition.** The sequence \( (u_t, v_t, p_t) \), \( p = 1, 2, \ldots \) generated by RCM method converges.
Algorithm 1: The proposed RCM method

Input: c, n, m, ϵ, η, τ\text{max} (iteration number)

Output: u, v

1 - Initialize the cluster centers and membership values randomly;
2 - Set t=0;
3 - While t < τ\text{max} do
4 - Compute \( e_i^2 = \sum_{j=1}^{c} u_{ij}^m d(x_i, v_j) \), \( i = 1, 2, \ldots, n \);
5 - Compute \( q_{ij} = \exp(-\eta_j e_i^2) \), \( i = 1, 2, \ldots, n \);
6 - Update all cluster centers \( v_{ij} \) by Eq. (15a), \( j = 1, 2, \ldots, c \);
7 - Update membership values \( u_{ij} \) by Eq. (15b), \( i = 1, 2, \ldots, n \), \( j = 1, 2, \ldots, c \);
8 - Set t=t+1;
9 - End while

Proof. According to (10), we have: \( J(u, v, p, \rho) \leq J(u, v) \). Then resulting from (12), (13) we have:
\[
J(u_t, v_t, p_t) \leq J(u_{t+1}, v_{t+1}, p_{t+1})
\]

It means that the sequence \( J(u_t, v_t, p_t), t = 1, 2, \ldots \) generated by RCM is nondecreasing. As a result, this sequence converges.

3.3. Optimizing η parameter

Utilizing alternating optimization method, in this section we introduce adaptive η parameter, i.e., just like as Section 3.1, given \((u, v, p)\) we can also optimize over η.

We define a constraint for η parameter over the minimization problem of (14) as follows:
\[
\min_{\eta} \sum_{i=1}^{n} \eta_i e_{i}^2 q_i
\]
subject to
\[
\sum_{i=1}^{n} \eta_i = \rho, \quad \eta_i \in [0, 1]
\]
(16)
The parameter \(\rho\) is a user defined parameter. After producing the Lagrange relation by adding the penalty term we have:
\[
\eta_i = \frac{\lambda}{2e_i^2 q_i} \sum_{j=1}^{n} u_{ij}^m = \frac{\rho}{\sum_{j=1}^{n} \frac{e_j^2 q_j}{q_{ij}}}
\]
(17)
Then, for dealing with divide by zero problem we add \(\xi\) parameter to Eq. (17) as:
\[
\eta_i = \frac{\rho}{\sum_{j=1}^{n} \frac{q_j}{q_{ij}} + \xi}, \quad i = 1, 2, \ldots, n
\]
(18)
In fact, \(\eta_i\) is the weight of the error. It determines how much a sample data can affect the clustering result. For the samples with large errors (e.g., outliers), the value of \(\eta_i\) becomes small to decrease the effect of outlier data in our minimization problem.

4. Robust heterogeneous c-means method (\(R_hCM\))

In this section we define a new form of robustness over the membership function. To have more compact and homogeneous clusters for spherical or elliptical shaped data, we apply the cluster fit function based on direct clustering method. In other words, we define a new restriction based on the membership function obtained by DC algorithm. Besides, since FCM is an iterative algorithm, it depends on the initial point. Determining the initial point is another issue that leads us to search for methods that are robust to noises. DC is a method that would satisfy this goal.

4.1. Direct clustering (DC) algorithm

Given a dataset \(X = \{x_1, x_2, \ldots, x_n\}\), the DC algorithm divides the dataset into \(c\) subsets (note that \(c < n\)) by minimizing the following cluster fit function:
\[
F(v_1, v_2, \ldots, v_c) = \sum_{i=1}^{n} \prod_{j=1}^{c} \|v_j - x_i\|^2, \quad x_i, v_j \in \mathbb{R}^d
\]
(19)
Where \(d\) denotes the number of dimensions in the feature space.

DC can be viewed as an analytic solution to the 2D minimization problem. By minimizing \(\prod_{j=1}^{c} \|v_j - x_i\|^2\) each \(x_i\) will tend to associate with its nearest cluster center, and therefore will lead to compact and homogeneous clusters [30].

The definition of the above fit function matches the general definition of the weighted fuzzy c-means by rewriting the fit function as follows:
\[
F(v_1, v_2, \ldots, v_c) = \frac{1}{c} \sum_{i=1}^{n} \left\{ \frac{1}{c} \prod_{j=1}^{c} \|v_j - x_i\|^2 \right\}
= \sum_{i,r} \left\{ \|v_r - x_i\|^2 \frac{1}{c} \prod_{j \neq r} \|v_j - x_i\|^2 \right\}
= \sum_{i,r} \{u_{ir} \|v_r - x_i\|^2\}
\]
4.2. The problem of DC algorithm

Although DC algorithm results in homogeneous clusters, when the number of data points in at least one cluster is significantly more than others, it fails to result in an accurate clustering. In this case, the centers of other clusters will be attracted to larger volume cluster center. Here we provide an example to show this problem.

Example. Without loss of generality, suppose that the clusters are spherical, \(c = 2\) (the number of clusters) and the points are in 2D space \((x_1, x_2)\). The distribution of data points in Fig. 1 shows that cluster 1 has larger volume (more data). The data points are symmetric with respect to line \(L1\) that is parallel to \(x_1\)-axis.

First, we prove that the center of each cluster is located on this line.
Proof. we use the method of proof by contradiction. Consider the situation in which the cluster centers are not on line L1.

Since the data distribution in each cluster is symmetric with respect to L1, so there should be another center with the same cost function that is located in a symmetric position with respect to L1. This is a contradiction, because DC cost function has only one global optimum (this was proved in [30]). Therefore, the initial assumption that the cluster centers are not located on line L1 must be false.

Then, under this condition we must show that how DC algorithm fails to result in an accurate clustering. So, we consider two cases:

Case I: Suppose that there are two clusters. The total number of data points is n and the number of data points in one cluster is k times the other one. First, the cluster centers are located on the centroid part of each cluster that is illustrated in Fig. 1 as v1 and v2. One can rewrite the DC cost function as follows:

\[ F(v_1, v_2) = \sum_{i=1}^{n} \prod_{j=1}^{2} \| v_j - x_i \|^2, \quad v_j, x_i \in R^2 \]  

(20)

\[ F(v_1, v_2) = \sum_{i=1}^{n} \| v_1 - x_i \|^2 \| v_2 - x_i \|^2 \]

In this case, the distance of ith data from jth cluster center is denoted as \( d_{ij} \).

Case II: If the center of the cluster with less data points (cluster 2) approaches near to the center of the other cluster (cluster 1) on distance of h (h > 0) from the previous position \( (v_2) \)-lets call this point \( v_2' \)- one can rewrite the DC cost function as follows:

\[ F(v_1, v_2) = \sum_{i=1}^{n} \prod_{j=1}^{2} \| v_j - x_i \|^2, \quad v_j, x_i \in R^2 \]

(21)

\[ F(v_1, v_2) = \sum_{i=1}^{n} \| v_1 - x_i \|^2 \| v_2' - x_i \|^2 \]

It is obvious that the distance of each data point in cluster 1 from \( v_2' \) is less than the previous case and it is at most \( k_{1,2} - h \). The distance of most of the data points in the second cluster from \( v_2' \) is more than the previous case. Since the number of data points in the first cluster is significantly more, so the value of objective function in Eq. (21) is less than the previous case (Eq. (20)). As we know, the objective function should be minimized. So, point \( v_2' \) will be chosen as the center of the cluster. Since the DC algorithm is independent of the shape of the clusters, this proof can be modified for general cases.

4.3. Formulation of R0lCM

The goal of the proposed method is to cluster unequal spherical or elliptical shaped data accurately and efficiently. So, we use a cluster fit function inspired by KFCM method. Also, we need a formula for \( u_{ij} \) that expresses how well each data point is approximated by at least one cluster center. To reach this goal, we use the previous mentioned direct clustering formula to evaluate \( u_{ij} \) in each iteration as

\[ u_{ij} = \frac{1}{c} \cdot \left( \frac{\prod_{i=1}^{c} \| v_j - x_i \|^2}{\| v_j - x_i \|^2} \right) \]  

(22)

The development of our R0lCM method is also based on the expected loss defined in Eq. (7) of Section 2.2. For simplicity we put \( f(v_j) = 1 \).

By using Eq. (22) the following constrained optimization problem is produced:

\[
\begin{align*}
\min_{u,v} J_0(u,v) &= \sum_{i=1}^{n} \sum_{j=1}^{c} u_{ij}^m (1 - k(x_i, v_j)) \\
\text{s.t.} \quad &\prod_{j=1}^{c} u_{ij} = \left( \frac{\prod_{i=1}^{c} \| v_j - x_i \|^2}{\| v_j - x_i \|^2} \right)^{c-1} \frac{1}{c^c} \\
&k(x_i, v_j) = \exp \left( -\frac{\| x_i - v_j \|^2}{2\sigma_j^2} \right)
\end{align*}
\]

(23)

(24)

Now, by using Lagrange multipliers \( \lambda_i \) we have:

\[
\begin{align*}
L(u,v,\lambda) &= \sum_{i=1}^{n} \sum_{j=1}^{c} u_{ij}^m (1 - k(x_i, v_j)) - \sum_{i=1}^{n} \lambda_i \left( \prod_{j=1}^{c} u_{ij} - \left( \prod_{i=1}^{c} \| v_j - x_i \|^2 \right)^{c-1} \right) \frac{1}{c^c}
\end{align*}
\]

(25)

First, we obtain \( \lambda_i \) as

\[
\frac{\partial L}{\partial u_{ij}} = 0 \Rightarrow \lambda_i = \frac{m u_{ij}^m (1 - k(x_i, v_j))}{\prod_{k=1}^{c} u_{ik}}, \quad i = 1, 2, \ldots, n
\]

And then by substituting \( \lambda_i \) in Eq. (25) and taking the partial derivatives with respect to \( v_j \) we get the centers as follows:

\[
v_j = \frac{\sum_{i=1}^{n} u_{ij}^m d_2(v_j, v_j) x_i}{\sum_{i=1}^{n} u_{ij}^m d_2(v_j, v_j)}, \quad j = 1, 2, \ldots, c
\]

(26)

\[
d_2(v_j, v_j) = \left( \frac{m (1 - k(x_i, v_j))(c-1)}{\| v_j - x_i \|^2} \right) + k(x_i, v_j)
\]
The proposed $R_H$CM procedure is shown in Algorithm 2. Utilizing DC as a method of initialization makes $R_H$CM to have robust initial points in noisy environments.

**Algorithm 2: The proposed $R_H$CM method**

**Input:** c, n, m, $\epsilon$, $\sigma$ ($\sigma$ is the kernel parameter)

**Output:** $u$, $v$

1. - Initialize the cluster centers by DC algorithm;
2. - Set $t=0$ and initialize the membership values randomly;
3. - **do**
4. - Update memberships $u_{ij,t+1}$ by Eq. (22),
   \[ i = 1, 2, \ldots, n; j = 1, 2, \ldots, c; \]
5. - Update all cluster centers $v_{j,t+1}$ by Eq. (26),
   \[ j = 1, 2, \ldots, c; \]
6. - Set $t = t+1$;
7. - **while** $|J(u_{ij,t}, v_{j,t}) - J(u_{ij,(t-1)}, v_{j,(t-1)})| < \epsilon$;

Since the constraint of our cost function in (23) is the multiplication of all membership values of a data point over all clusters, each cluster center is used to approximate the data points that are not being approximated closely enough to the other cluster centers. Therefore, this algorithm results in compact clusters having inter-cluster heterogeneity. Clearly, this definition is reasonable.

By using an RBF kernel as a similarity measure, this algorithm can cluster spherical or elliptical shaped data as well as more general ones effectively.

As we proved in this section, the DC algorithm does not guarantee to result in accurate cluster centers when the size of clusters are unequal. So, as the test results indicate, $R_H$CM algorithm overcomes this drawback.

5. Experimental results

In this section, first we show the results of our RCM method on both synthetic and real datasets from the UCI machine learning repository in compare with basic FCM. Then we will show the superiority of our $R_H$CM method in clustering unequal sized spherical or elliptical shaped data on synthetic datasets, synthetic noisy datasets and real datasets. Finally, the clustering quality of the algorithms are compared utilizing several cluster validity indices.

5.1. Evaluation of RCM algorithm

To show the differences between FCM and RCM method, consider the distribution of datapoints in Fig. 2. If we have two clusters, the outlier point in FCM method attracts one of the cluster centers, but RCM method resists to the effect of the outlier data.

In Fig. 3, the result of the movement of the centers are shown on a synthetic dataset as well as the UCI repository. For visualizing high dimensional data, Andrews plot method [33] is used.

As it can be seen, RCM algorithm has much fewer changes of the cluster centers in the presence of outliers in compare with FCM. So, the robustness of our proposed method is properly shown.

5.2. Evaluation of $R_H$CM algorithm

In this section, the experimental results for $R_H$CM algorithm compared with DC algorithm, KFCM (that initialized randomly), DC-KFCM (which is a KFCM method that we initialized with DC results), GPFCM and GEPFCM methods are shown.

Since DC algorithm has an exact solution (i.e. it does not have any iterations), it is very fast. So, instead of initializing randomly, our $R_H$CM cluster centers are initialized by DC results. Since DC works on 2D space, we will show the results on 2D space.

Fig. 4 shows the clustering results on synthetic datasets having different volume of clusters. Different values of kernel parameter ($\sigma$) were tested and the best ones were selected for each method. Since KFCM, GPFCM and GEPFCM methods are initialized randomly, we have repeated each of these methods 50 times and then the best results (the results with the minimum entropy) are reported. Red arrows in Fig. 4 show the wrong movement of the centers. $R_H$CM centers are shown with the red circles.

In Figs. 4(a) to 4(d), KFCM, DC-KFCM, GPFCM, GEPFCM and $R_H$CM methods have almost similar behaviors; but as shown in Fig. 4(e) when the volume of a few clusters dramatically increases and the clusters have overlaps, the centers resulted by KFCM, GPFCM and GEPFCM cannot find their positions correctly; but in DC-KFCM, initializing by DC helps the algorithm find the more reasonable centers. However, DC-KFCM can also be sensitive to the volume of clusters as shown in Fig. 4(f). Since the shape and the volume of clusters have a lot of differences with each other and the clusters have overlaps, the larger volume cluster pulls the center of the smaller one towards itself in all KFCM, DC-KFCM, GPFCM and GEPFCM methods. But the results show the superiority of $R_H$CM in robustness to the shape and the volume of clusters in all the figures. The $R_H$CM algorithm not only addresses the drawbacks of DC algorithm but also results in compact clusters, heterogeneous from each other.

Fig. 5 shows six synthetic noisy datasets having various number and different volume of clusters. The uniform noise has been added to the data. The number of noise and outlier points increases from noisyData1 to noisyData6 to study the impact of noise on the performances of the algorithms.

Since GPFCM and GEPFCM methods were designed to have good clustering results especially in noisy environments, their overall performances on nearly equal-sized clusters are reasonable (Figs. 5(d) and 5(e)). But when the clusters are of different sizes, similar misbehavior that were stated in Fig. 4 can be seen here again. To be more clear, as shown in Figs. 5(a)–5(c) and 5(f), larger volume clusters pull the centers of smaller ones for all DC, DC-KFCM, GPFCM and GEPFCM methods. But in all of these datasets, our $R_H$CM method is robust both to the noise and to the volume of clusters and hence it has a lot more reasonable clustering results than others.

In Fig. 6 the effect of $\sigma$ parameter on the mentioned methods is shown. As it can be seen, KFCM, DC-KFCM, GPFCM and GEPFCM are very dependent to the value of $\sigma$ parameter. It means, by changing the value of $\sigma$, the position of centers in these methods changes a lot; but due to the definition of membership function ($u$), the role of $\sigma$ in $R_H$CM is lower than the other methods. So, $R_H$CM shows almost the same behavior while changing the value of $\sigma$. In other words, $R_H$CM is nearly robust to the changes of $\sigma$ parameter.

To generalize our method on nonspherical and non-elliptical shaped data, we also tested our $R_H$CM method on a real dataset that contains 80,000 images from 500 px social media. We have used the 500 px api to search and gather the uploaded images on different concepts. We have grouped the data images into four categories: ‘sport’, ‘food’, ‘cold’ and ‘war’. The data contain images and other information such as the tags users created for each uploaded images, the latitude and longitude, the rating and etc.

We use the latitude and longitude information of each image for clustering purpose and then we show the results of our clustering method on a desired number of clusters (Fig. 7(a)). According to the shape of the overlapping clusters and a noisy environment, $R_H$CM and DC-KFCM cluster centers are more satis- fiable than the other methods (it is shown in the right picture of Fig. 7(a)). After obtaining the centers, the nearest real data to the
center of each cluster are found (according to their latitude and longitude) to be shown on the map (Fig. 7(b)). The images found at the center of clusters are shown in Fig. 7(c).

The results of clustering of the uploaded images by 500 px social media users on the concepts of food, sport and war are also shown in Fig. 8.

As mentioned before, R_CCM tends to create compact clusters with heterogeneity among them. So, it will tend to find the centers of clusters far from each other and in denser regions. When the volume of clusters are dramatically different as in Fig. 8(b), the centers obtained by DC, DC-KFCM, GPFCM and GEPFCM in smaller cluster (red circle) tend to move towards larger volume clusters, but R_CCM shows robustness. Even when different clusters have overlaps, R_CCM has the ability to find reasonable cluster centers as in Fig. 8(c).

Fig. 9 shows the cluster centers for the last 3 concepts (food, sport and war) on a real map along with their corresponding centers as in Fig. 8(c).

5.3. Clustering quality

In this section, we first mention a number of cluster validity indices and then we use them to measure the clustering quality of the algorithms.

5.3.1. Validity indices

Partition Coefficient (PC): The PC index \[ PC = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{c} u_{ij}^2 \] (27)
The index values range in \([1/c, 1]\) where \(c\) is the number of clusters. If all membership grades are equal, the PC has its lower value; that is, a PC value close to \(1/c\) indicates the failure of the clustering algorithm. The closer the value to 1 the crisper the clustering is. Hence, the larger the PC value the better.

Partition Entropy (PE): The PE index \[ PE = -\frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{c} u_{ij} \log a u_{ij} \] (28)
The index values range in \([0, \log_a c]\) where \(a\) is the base of the logarithm. A PE value close to \(\log_a c\) shows the inability of the algorithm to extract the clustering structure in the dataset. The closer to zero the entropy the better.

Xie and Beni’s Index (XB): The XB index \[ XB = \frac{\sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^2 \| x_i - v_j \|^2}{\min_{i,j} \| v_i - v_j \|^2} \] (29)

Where the compactness uses the variance of data within a cluster to indicate the degree of closeness of cluster members; and the separation of cluster \(i\) is defined as the sum of distances of its center to the other cluster centers. The lower the value of SC the better the clustering result.

Dunn’s Index (DI): DI \[ DI = \min_{i,j \neq k} \left\{ \frac{d(c_i, c_j)}{\max_{k \neq i, j} \{d(x, y)\}} \right\} \] (31)

For a good partition, the value of compactness is small and for well-separated clusters, the value of separation is high. Therefore, for compact and well-separated clusters the value of XB is expected to be small.

For a good partitioning, the distances among clusters are expected to be large and the diameters of the clusters are small. So, larger value of DI indicates better clustering results.
(a) A synthetic dataset containing 600 datapoints and 6 attributes. 20 outliers have been added to the samples. Data are clustered in 3 classes.

(b) Iris dataset: movement of two of the cluster centers is shown. 10% of the 3rd class have been added as outliers.

(c) Wine dataset: movement of one of the cluster centers is shown. 10% of the 3rd class have been added as outliers.

(d) Contraceptive dataset: movement of one of the cluster centers is shown. 10% of the 3rd class have been added as outliers.

**Fig. 3.** Movement of the cluster centers in the presence of outliers is shown. The blue curve shows the real center of clusters. The green curve shows the centers achieved by RCM and FCM method. The red curve shows the centers obtained by the two algorithm after adding outliers. To avoid the interference of the curves, in (c) and (d) the results for one of the classes are shown. The centers are presented in 2D space using the andrewsplot function in matlab. \( t \) is a continuous dummy variable that was created by Andrews plot method and \( f(t) \) represents each observation.

**Rand Index (RI):** This index [41] measures the degree of similarity between the set of discovered clusters \( C \) and the predefined partition \( P \) of data [36,42]. It is given by:

\[
RI = \frac{a + d}{a + b + c + d}
\]

where

- \( a \) is the number of pairs \((x, x')\) of data that belongs to the same cluster of \( C \) and to the same group of partition \( P \).
- \( b \) is the number of pairs \((x, x')\) of data that belongs to the same cluster of \( C \) and to different groups of \( P \).
- \( c \) is the number of pairs \((x, x')\) of data that belongs to different clusters of \( C \) and to the same group of \( P \).
Fig. 4. Movement of the class centers while having different volume of clusters (in almost all cases $\sigma_{\text{HC}} = 2$, but different parameter values are used for the other methods in each figure). In KFCM, DC-KFCM, GPECM and GEPFCM the centers in smaller volume clusters move towards the much larger volume clusters; but in $R_{\text{HC}}$CM the cluster volumes hardly affect the position of other cluster centers.

Fig. 5. Movement of the class centers while having different volume of clusters in a noisy environment. In KFCM, DC-KFCM, GPECM and GEPFCM the centers in smaller volume clusters move towards the much larger volume clusters; but in $R_{\text{HC}}$CM the cluster volumes hardly affect the position of other cluster centers.
Fig. 6. The effect of different $\sigma$ parameter.

(a) $\sigma = 2$

(b) $\sigma = 10$

(c) $\sigma = 100$

Fig. 7. Clustering result of uploaded images by 500 px social media users on the ‘cold’ concept. We desire to create 3 clusters of data (a) Two clusters have overlaps in the right picture, just like the previous shown synthetic datasets, $R_0$CM tends to result in compact clusters having inter-cluster heterogeneity (b) the centers of each cluster obtained by $R_0$CM are shown on the map to specify the regions (c) the images related to the ‘cold’ concept as the center of clusters are shown.
Fig. 8. (a), (b), (c) Clustering result of uploaded images by 500 px social media users on 3 concepts: ‘food’, ‘sport’ and ‘war’.

<table>
<thead>
<tr>
<th>Category</th>
<th>PC</th>
<th>PE</th>
<th>SC</th>
<th>XB</th>
<th>DI</th>
<th>RI</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC</td>
<td>0.31</td>
<td>1.27</td>
<td>0.179</td>
<td>0.014</td>
<td>0.0046</td>
<td>0.59</td>
</tr>
<tr>
<td>KFCM</td>
<td>0.64</td>
<td>0.65</td>
<td>0.081</td>
<td>0.011</td>
<td>0.0032</td>
<td>0.81</td>
</tr>
<tr>
<td>DC-KFCM</td>
<td>0.31</td>
<td>1.27</td>
<td>0.129</td>
<td>0.0088</td>
<td>0.0064</td>
<td>0.92</td>
</tr>
<tr>
<td>GPFCM</td>
<td>0.27</td>
<td>1.33</td>
<td>0.572</td>
<td>0.043</td>
<td>0.0024</td>
<td>0.73</td>
</tr>
<tr>
<td>GEPFCM</td>
<td>0.41</td>
<td>1.05</td>
<td>0.412</td>
<td>0.038</td>
<td>0.0019</td>
<td>0.73</td>
</tr>
<tr>
<td>RHCM</td>
<td>0.75</td>
<td>0.49</td>
<td>0.059</td>
<td>0.0064</td>
<td>0.0064</td>
<td>0.93</td>
</tr>
</tbody>
</table>

Table 2

The mean value of cluster validity indices on synthetic datasets for data1-data6.

<table>
<thead>
<tr>
<th>Category</th>
<th>PC</th>
<th>PE</th>
<th>SC</th>
<th>XB</th>
<th>DI</th>
<th>RI</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC</td>
<td>0.34</td>
<td>1.27</td>
<td>0.181</td>
<td>0.0164</td>
<td>0.039</td>
<td>0.79</td>
</tr>
<tr>
<td>KFCM</td>
<td>0.76</td>
<td>0.47</td>
<td>0.058</td>
<td>0.0080</td>
<td>0.044</td>
<td>0.90</td>
</tr>
<tr>
<td>DC-KFCM</td>
<td>0.70</td>
<td>0.58</td>
<td>0.066</td>
<td>0.0075</td>
<td>0.044</td>
<td>0.93</td>
</tr>
<tr>
<td>GPFCM</td>
<td>0.44</td>
<td>0.98</td>
<td>0.180</td>
<td>0.0131</td>
<td>0.042</td>
<td>0.88</td>
</tr>
<tr>
<td>GEPFCM</td>
<td>0.70</td>
<td>0.57</td>
<td>0.117</td>
<td>0.0125</td>
<td>0.040</td>
<td>0.89</td>
</tr>
<tr>
<td>RHCM</td>
<td>0.75</td>
<td>0.49</td>
<td>0.059</td>
<td>0.0064</td>
<td>0.0064</td>
<td>0.93</td>
</tr>
</tbody>
</table>

Table 3

Clustering validity indices on synthetic noisy datasets for noisyData3.

<table>
<thead>
<tr>
<th>Category</th>
<th>PC</th>
<th>PE</th>
<th>SC</th>
<th>XB</th>
<th>DI</th>
<th>RI</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC</td>
<td>0.32</td>
<td>1.24</td>
<td>0.15</td>
<td>0.0109</td>
<td>0.001</td>
<td>0.89</td>
</tr>
<tr>
<td>KFCM</td>
<td>0.64</td>
<td>0.68</td>
<td>0.068</td>
<td>0.0053</td>
<td>0.002</td>
<td>0.68</td>
</tr>
<tr>
<td>DC-KFCM</td>
<td>0.71</td>
<td>0.53</td>
<td>0.057</td>
<td>0.0060</td>
<td>0.004</td>
<td>0.71</td>
</tr>
<tr>
<td>GPFCM</td>
<td>0.34</td>
<td>1.07</td>
<td>0.108</td>
<td>0.0056</td>
<td>0.002</td>
<td>0.68</td>
</tr>
<tr>
<td>GEPFCM</td>
<td>0.69</td>
<td>0.58</td>
<td>0.059</td>
<td>0.0056</td>
<td>0.002</td>
<td>0.69</td>
</tr>
<tr>
<td>RHCM</td>
<td>0.75</td>
<td>0.48</td>
<td>0.059</td>
<td>0.0056</td>
<td>0.002</td>
<td>0.69</td>
</tr>
</tbody>
</table>

Table 4

The mean value of cluster validity indices on synthetic noisy datasets for noisyData1-noisyData6.

<table>
<thead>
<tr>
<th>Category</th>
<th>PC</th>
<th>PE</th>
<th>SC</th>
<th>XB</th>
<th>DI</th>
<th>RI</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC</td>
<td>0.52</td>
<td>0.82</td>
<td>0.006</td>
<td>0.0011</td>
<td>0.005</td>
<td>0.005</td>
</tr>
<tr>
<td>KFCM</td>
<td>0.79</td>
<td>0.39</td>
<td>0.006</td>
<td>0.0010</td>
<td>7.99×10^{-4}</td>
<td>0.005</td>
</tr>
<tr>
<td>DC-KFCM</td>
<td>0.56</td>
<td>0.75</td>
<td>0.005</td>
<td>8×10^{-4}</td>
<td>0.00025</td>
<td>0.005</td>
</tr>
<tr>
<td>GPFCM</td>
<td>0.51</td>
<td>0.75</td>
<td>0.012</td>
<td>0.0014</td>
<td>0.0014</td>
<td>0.0014</td>
</tr>
<tr>
<td>GEPFCM</td>
<td>0.82</td>
<td>0.34</td>
<td>0.008</td>
<td>0.0014</td>
<td>0.0015</td>
<td>0.0015</td>
</tr>
<tr>
<td>RHCM</td>
<td>0.83</td>
<td>0.32</td>
<td>0.002</td>
<td>6×10^{-4}</td>
<td>0.0019</td>
<td>0.0019</td>
</tr>
</tbody>
</table>

Table 5

Clustering validity indices on real datasets for the concept of “cold”.

<table>
<thead>
<tr>
<th>Category</th>
<th>PC</th>
<th>PE</th>
<th>SC</th>
<th>XB</th>
<th>DI</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC</td>
<td>0.51</td>
<td>0.83</td>
<td>0.006</td>
<td>0.0011</td>
<td>0.050</td>
</tr>
<tr>
<td>KFCM</td>
<td>0.79</td>
<td>0.40</td>
<td>0.003</td>
<td>7.6×10^{-4}</td>
<td>0.034</td>
</tr>
<tr>
<td>DC-KFCM</td>
<td>0.57</td>
<td>0.75</td>
<td>0.005</td>
<td>8.9×10^{-4}</td>
<td>0.034</td>
</tr>
<tr>
<td>GPFCM</td>
<td>0.43</td>
<td>0.86</td>
<td>0.049</td>
<td>0.0065</td>
<td>0.001</td>
</tr>
<tr>
<td>GEPFCM</td>
<td>0.58</td>
<td>0.69</td>
<td>0.044</td>
<td>0.0074</td>
<td>0.002</td>
</tr>
<tr>
<td>RHCM</td>
<td>0.85</td>
<td>0.29</td>
<td>0.002</td>
<td>6.5×10^{-4}</td>
<td>0.014</td>
</tr>
</tbody>
</table>

Tables 5–8 show the clustering quality of the algorithms on the real dataset. Since the true labels of the real data are not available, RI values are not reported. It can be inferred that the overall performance of RHCM is better than that of the other methods.
6. Conclusion

In this paper, we presented a new perspective based on the expected loss (or risk) to the FCM method. As it was mentioned, FCM is a sensitive algorithm to initialization, noise and outliers. This new perspective will help to generate different kinds of FCM including the robust versions.

So, first we proposed the expected loss based on the error to present a robust FCM called RCM. By using correntropy loss function and utilizing half-quadratic (HQ) problem-solving method, we showed in the experimental results that RCM is a lot more robust to outliers than FCM.

Then, we proposed another kind of expected loss based on the data and parameters to present some other kinds of robustness. Besides, to find some initial points in noisy environments, we needed a fast robust method like DC, which does not need any
initial center guessing and iterations. However, DC does not result in accurate cluster centers when at least one of the clusters in the dataset has a larger volume than the others. To address this drawback, we introduced the iterative RUCM utilizing direct clustering (DC) method and using the correntropy induced metric similar to RCM. The goal is to have a robust method to the noise, outliers and different volume clusters while the initialization is also robust. We have tested our method on three groups of datasets: synthetic datasets, synthetic noisy datasets and a real dataset consisting of 4 groups of data gathered from 500 px social media. The results were reported both visually and in terms of cluster validity indices.

Based on the experimental results, RUCM algorithm, in contrast with KFCM, DC, DC-KFCM, GPPCM and GEPFCM methods, can provide competitive clustering results and produce homogeneous clusters that have heterogeneity among themselves.

Declaration of competing interest

No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.asoc.2019.105885.

Appendix A. Half-quadratic (HQ) optimization

The description of HQ modeling is based on the conjugate function theory [43,44]. So, for a function $\phi(p)$, the conjugate function is $\phi^*(s)$ that is defined as follows:

$$\phi^*(s) = \sup_{p \in dom_{\phi}} (s^T p - \phi(p)) \quad (A.1)$$

By deriving the function $\phi$ with respect to $p$, the supremum is achieved at $s = \phi'(p)$.

If we define $\phi(p) = -p \log(-p) + p$, where $p < 0$ as a convex function [45], then using the definition of conjugate function one can find supremum at $p = -\exp(-s) < 0$. So, by replacing $p$ in (A.1) the conjugate function of $\phi(p)$ is $\phi^*(s) = \exp(-s)$.

Appendix B. The proof of Eq. (23) (constraint of RUCM optimization problem)

As it was mentioned in (19) for DC method we had:

$$U_{ij} = \frac{1}{c} \prod_{r \neq j}^c \|v_r - x_i\|^2$$

Then, one can rewrite $U_{ij}$ as

$$U_{ij} = \frac{1}{c} \prod_{r = 1}^c \|v_r - x_i\|^2$$

Consequently, the following constraint can be achieved:

$$\prod_{j=1}^n U_{ij} = \left( \frac{1}{c} \prod_{r = 1}^c \|v_r - x_i\|^2 \right)^{\frac{1}{c}} \prod_{j=1}^n \frac{1}{c} \prod_{r = 1}^c \|v_r - x_i\|^2$$

Appendix C. The proof of Eq. (26) (RUCM optimization problem)

According to (25) we have:

$$L(u, v, \lambda) = \sum_{i=1}^n \sum_{j=1}^c n_i^m (1 - k(x_i, v_j)) - \sum_{i=1}^n \lambda_i \left( \prod_{i=1}^c u_{ij} - \frac{\left( \prod_{i=1}^c \|v_r - x_i\|^2 \right)}{c^c} \right)$$ \hfill (C.1)

By taking the partial derivatives with respect to $u_{ij}$ and $v_{ij}$ we have:

$$\frac{\partial L}{\partial u_{ij}} = 0 \Rightarrow \lambda_i \prod_{r=1}^c u_{ij} = \frac{1}{c^c} \left( \prod_{r=1}^c \|v_r - x_i\|^2 \right)$$

Then, by substituting $\lambda_i$ as:

$$\sum_{i} 2n_i^m (v_j - x_i)k(x_i, v_j) + \sum_{i} \lambda_i \left( 2c - 1 \right) (v_j - x_i)$$

$$\times \frac{\left( \prod_{r \neq j} \|v_r - x_i\|^2 \right)}{c^c} \left( \prod_{r \neq j} \|v_r - x_i\|^2 \right) = 0$$

and by substituting $\prod_{i=1}^c U_{ij} = \frac{\left( \prod_{i=1}^c \|v_r - x_i\|^2 \right)}{c^c}$ we have:

$$\Rightarrow \sum_{i} n_i^m (v_j - x_i)k(x_i, v_j)$$

Table 7

<table>
<thead>
<tr>
<th>Category</th>
<th>PC</th>
<th>PE</th>
<th>SC</th>
<th>XE</th>
<th>DI</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC</td>
<td>0.53</td>
<td>0.79</td>
<td>0.005</td>
<td>8.2 * 10^-4</td>
<td>0.01</td>
</tr>
<tr>
<td>KFCM</td>
<td>0.81</td>
<td>0.35</td>
<td>0.0032</td>
<td>6.75 * 10^-4</td>
<td>0.013</td>
</tr>
<tr>
<td>DC-KFCM</td>
<td>0.61</td>
<td>0.68</td>
<td>0.005</td>
<td>8.4 * 10^-4</td>
<td>0.019</td>
</tr>
<tr>
<td>GPPCM</td>
<td>0.40</td>
<td>0.99</td>
<td>1.52</td>
<td>0.21</td>
<td>3.6 * 10^-4</td>
</tr>
<tr>
<td>GEPFCM</td>
<td>0.39</td>
<td>1.01</td>
<td>2.69</td>
<td>0.45</td>
<td>6.4 * 10^-4</td>
</tr>
<tr>
<td>RUCM</td>
<td>0.87</td>
<td>0.24</td>
<td>0.001</td>
<td>4.08 * 10^-4</td>
<td>0.004</td>
</tr>
</tbody>
</table>

Table 8

<table>
<thead>
<tr>
<th>Category</th>
<th>PC</th>
<th>PE</th>
<th>SC</th>
<th>XE</th>
<th>DI</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC</td>
<td>0.50</td>
<td>0.8</td>
<td>0.005</td>
<td>0.001</td>
<td>0.001</td>
</tr>
<tr>
<td>KFCM</td>
<td>0.79</td>
<td>0.4</td>
<td>0.006</td>
<td>0.001</td>
<td>0.002</td>
</tr>
<tr>
<td>DC-KFCM</td>
<td>0.56</td>
<td>0.8</td>
<td>0.005</td>
<td>9.35 * 10^-4</td>
<td>0.007</td>
</tr>
<tr>
<td>GPPCM</td>
<td>0.59</td>
<td>0.7</td>
<td>0.033</td>
<td>0.005</td>
<td>6.09 * 10^-5</td>
</tr>
<tr>
<td>GEPFCM</td>
<td>0.83</td>
<td>0.3</td>
<td>0.007</td>
<td>0.001</td>
<td>0.002</td>
</tr>
<tr>
<td>RUCM</td>
<td>0.82</td>
<td>0.3</td>
<td>0.003</td>
<td>6.9 * 10^-4</td>
<td>0.002</td>
</tr>
</tbody>
</table>
\[ + \sum_{i} \hat{u}_{ij}^{m} \left(1 - \frac{1}{\|x_i - v_j\|^2} \right)^{(c-1)} = 0 \]
\[ \Rightarrow \hat{v}_j = \frac{\sum_{i=1}^{n} \hat{u}_{ij}^{m} d_j(x_i, v_j)}{\sum_{i=1}^{n} \hat{u}_{ij}^{m}} \quad , \quad j = 1, 2, \ldots, c, \]
where \[ d_j(x_i, v_j) = \left( \frac{1}{\|x_i - v_j\|^2} - 1 \right) + k(x_i, v_j) \]

References