Multiple Kernel Interval Type-2 Fuzzy C-Means Clustering

Dzung Dinh Nguyen, Long Thanh Ngo
Department of Information Systems
Le Quy Don Technical University, No 100, Hoang Quoc Viet St., Hanoi, Vietnam
Email: dinhdung1082@gmail.com, ngotlong@gmail.com

***Abstract*—In this paper, kernel interval type-2 fuzzy c-means clustering (KIT2FCM) and multiple kernel interval type 2 fuzzy c-means clustering (MKIT2FCM) are proposed as a base for classification problems. Besides building algorithms KIT2FCM to overcome some drawback of the conventional FCM and use the advantages of fuzzy clustering technique on the interval type 2 fuzzy set in handling uncertainty, the paper also introduces combining the different kernels to construct the MKIT2FCM which provides us a new flexible vehicle to fuse different data information in the classification problems. That is, different information represented by different kernels is combined in the kernel space to produce a new kernel. The experiments are done based on well-known data-sets and application of land cover classification from multi-spectral with the statistics show that the algorithms generates good quality of classifications.**

***Index Terms*—Type-2 fuzzy sets, type-2 fuzzy c-means clustering, Kernel fuzzy c-means clustering, Multiple Kernel fuzzy c-means clustering.**

I. Introduction

Clustering is a mathematical tool used to detect any structures or patterns in the data set, in which objects within the cluster level data show certain similarities.

Clustering algorithms have different shapes from simple clustering as k-means and various improvements [2], [3], development of family of fuzzy c-mean clustering (FCM) [11].

To overcome the drawbacks of conventional FCM technique, kernel fuzzy c-means clustering (KFCM) algorithm has been proposed. The limitation of the standard FCM algorithm is based on the Euclidean norm distance in the observation space, it has been shown while it is effective for spherical clusters it does not perform well for more general clusters [12], which is eliminated in KFCM by adapting a new kernel induced metric in the data space [9], which maps the original inputs into a much higher dimensional Hilbert space by some transform function. After this reproduction in the kernel Hilbert space, the data are more easily to be separated or clustered.

However, for such kernel-based methods, a crucial step is the combination or selection of the best kernels among an extensive range of possibilities. This step is often heavily influenced by prior knowledge about the data and by the patterns we expect to discover [17]. Unfortunately, it is unclear which kernels are more suitable for a particular task [18].

The problem is aggravated for many real-world clustering applications, in which there are multiple potentially useful cues. For example, KFCM is applied in the image-segmentation problems, where the input data selected for clustering is the combination of the pixel intensity and the local spatial information of a pixel represented by the mean or the median of neighbouring pixels. Applications of kernel-based approach are used for image segmentation with spatial information [19], [20]. Chen and Zhang [21] applied the idea of kernel methods in the calculation of the distances between the examples and the cluster centroids. They compute these distances in the extended Hilbert space, and they have demonstrated that such distances are more robust to noises. To keep the merit of applying local spatial information, an additional term about the difference between the local spatial information and the cluster centroids (also computed in the extended Hilbert space) is appended to the objective function.

Thus, it is necessary to aggregate features from different sources into a single aggregated feature. However, these features are often not equally relevant to clustering; some are irrelevant, and some are less important than others [12]. As most clustering methods do not embed a feature selection capability, such feature imbalances often necessitate an additional process of feature selection, or feature fusion, before clustering. Instead of a single fixed kernel, multiple kernels may be used. Recent developments in multiple kernel learning have shown that the construction of the multiple kernel fuzzy c-means (MKFCM) algorithm simultaneously finds the the best degrees of membership and the optimal kernel weights for a non-negative combination of a set of kernels.

We also embed the feature weight computation into the clustering procedure. The incorporation of multiple kernels and the automatic adjustment of kernel weights renders MKFCM more immune to unreliable features or kernels. It also makes combining kernels more practical since appropriate weights are assigned automatically. Effective kernels or features tend to contribute more to the clustering and therefore improve results.

Recently, type-2 fuzzy sets are extension of original fuzzy sets, have the advantage of handling uncertainty, which have been developed and applied to many different problems [4], [5], [6], [8], [10] including data clustering problems. In addition, interval type-2 fuzzy c-means clustering algorithm (IT2FCM) [1] has developed a step in the clustering method in which FOU (footprint of uncertainty) is created for the fuzzier m using two parameters for handling of uncertainty, making clustering more efficiently.

Therefore, this paper proposes KIT2FCM and MKIT2FCM algorithms to handle the uncertainty better than KFCM and MKFCM, which are more appropriate when clusters have significant overlap. With MKIT2FCM, a linear composite of multiple kernels was used based on the IT2FCM algorithm. Algorithms are proposed on the basis of combination of kernel techniques and IT2FCM [1]. Experiments are implemented based various datasets of classification and land cover classification of multi-spectral satellite image to show the advantage of proposed approach.

The paper is organized as follows: Section II briefly introduces about the kernel technique, Section III proposes the Kernel Interval type-2 fuzzy C-means clustering, Section III describes the Multiple Kernel Fuzzy C-means clustering; Section IV offers some experimental results and section V is conclusion.

II. The Kernel Technique

[22]

The key idea of the kernel technique is to invert the chain of arguments, i.e., choose a kernel k rather than a mapping before applying a learning algorithm. Of course, not any symmetric function k can serve as a kernel. Suppose our input space χ has a finite number of elements, i,e., $\chi = \{x_1, .., x_r\}$. Then, the $r \times r$ kernel matrix $\mathbf{K}$ with $\mathbf{K}_{ij} = k(x_i, x_j)$ is by definition a symmetric matrix and $k(x_i, x_j)$ is kernel value between $x_i, x_j \in \chi$. The necessary and sufficient conditions of $k : \chi \times \chi \to \mathbb{R}$ be a kernel are given by Mercers theorem.

Theorem 2.1: The function $k : \chi \times \chi \to \mathbb{R}$ is a Mercer kernel if, and only if, for each $r \in \mathbb{N}$ and $x = (x_1, x_2..., x_r) \in \chi^r$ the $r \times r$ matrix $K = (k(x_i, x_j))_{i,j=1}^{r}$ is positive semi definite.

Kernel Families

So far we have seen that there are two ways of making linear classifiers nonlinear in input space:

1. Choose a mapping ϕ which explicitly gives us a (Mercer) kernel k,or

2. Choose a Mercer kernel k which implicitly corresponds to a fixed mapping ϕ.

Though mathematically equivalent, kernels are often much easier to define and have the intuitive meaning of serving as a similarity measure between objects $x, \tilde{x} \in \chi$.

Moreover, there exist simple rules for designing kernels on the basis of given kernel functions.

Theorem 2.2: Functions of kernels. Let $k_1 : \chi \times \chi \to \mathbb{R}$ and $k_2 : \chi \times \chi \to \mathbb{R}$ be any two Mercer kernels. Then, the functions $k : \chi \times \chi \to \mathbb{R}$ given by

1.$k(x, \tilde{x}) = k_1(x, \tilde{x}) + k_2(x, \tilde{x})$

2.$k(x, \tilde{x}) = c.k_1(x, \tilde{x})\ \ for\ all\ c \in \mathbb{R}^+$

3.$k(x, \tilde{x}) = k_1(x, \tilde{x}) + c\ for\ all\ c \in \mathbb{R}^+$

4. $k(x, \tilde{x}) = k_1(x, \tilde{x}).k_2(x, \tilde{x})$

5. $k(x, \tilde{x}) = f(x).f(\tilde{x})\, foranyfunctionf : \chi \to \mathbb{R}$ are also Mercer kernels

Theorem 2.3: Let $k_1 : \chi \times \chi \to \mathbb{R}$ be any Mercer kernel. Then, the functions $k : \chi \times \chi \to \mathbb{R}$ given by

1. $k(x, \tilde{x}) = (k_1(x, \tilde{x}) + \theta_1)^{\theta_2}, for\ all\ \theta_1 \in \mathbb{R}^+ and \theta_2 \in \mathbb{N}$.

2. $k(x, \tilde{x}) = \exp\left(\frac{k_1(x, \tilde{x})}{\sigma^2}\right)\ \ for\ all\ \sigma \in \mathbb{R}^+$

3.$k(x, \tilde{x}) = \exp\left(-\frac{k_1(x,x) - 2k_1(x,\tilde{x}) + k_1(\tilde{x},\tilde{x})}{2\sigma^2}\right), for\ all\ \sigma \in \mathbb{R}^+$

4.$k(x, \tilde{x}) = \frac{k_1(x,\tilde{x})}{\sqrt{k_1(x,x).k_1(\tilde{x},\tilde{x})}}$

are also Mercer kernels. it is possible to normalize data in feature space without performing the explicit mapping because, for the inner product after normalization, it holds that

$$\begin{aligned} k_{norm}(x, \tilde{x}) &\overset{def}{=} \frac{k(x,\tilde{x})}{\sqrt{k(x,x).k(\tilde{x},\tilde{x})}} \\ &= \frac{1}{\sqrt{\|x\|^2 . \|\tilde{x}\|^2}} \langle x, \tilde{x} \rangle \\ &= \left\langle \frac{x}{\|x\|}, \frac{\tilde{x}}{\|\tilde{x}\|} \right\rangle \end{aligned} \tag{1}$$

III. Kernel Interval Type-2 Fuzzy C-mean Clustering

This algorithm is based on a combination of type-2 fuzzy set with the kernel technique. Although the interval type-2 fuzzy C-mean clustering algorithm has advantages in handling uncertainty, the limitation of IT2FCM algorithm is based on the Euclidean norm distance in the observation space, it has been shown while it is effective for spherical clusters it does not perform well for more general clusters. As an enhancement of classical IT2FCM, the Kernel Interval Type 2 Fuzzy C-mean Clustering (KIT2FCM) use a non-linear map defined as $\phi : x \to \phi(x) \in H, x \in X \in R^d$.

Where X denotes the data set or the feature space and H is a Hilbert space (usually called kernel space).

In the new kernel space, the data demonstrate simpler structures or patterns. According to clustering algorithms, the data in the new space show clusters that are more spherical and therefore can be clustered more easily by IT2FCM algorithms [7], which is the main purpose of this algorithm.

Generally, the transform function ϕ is not given out explicitly, but the kernel function is given and it is defined as $k : \theta x \theta \to R$

$$k(x, y) = \phi(x)\, \phi(y)^T\ \forall x, y \in R \tag{2}$$

Here $\phi(x)\, \phi(y)^T$ is an inner product of the kernel function. Such kernel functions are usually called Mercer kernels or kernel. Given a Mercer kernel k, we know that there is always a transform function $\phi : x \to \phi(x) \in H, x \in X \in R^d$ satisfies $k(x, y) = \phi(x)\, \phi(y)^T$, although sometimes, we do not know the specific form of ϕ. Widely used Mercer kernels include the Gaussian kernel $k(x, y) = \exp\left(-\|x - y\|^2 / r^2\right)$ and the polynomial kernel $k(x, y) = \left(x^T.y + d\right)^2$.

There are two major forms of kernel interval type 2 clustering (KIT2FCM). The first one comes with prototypes constructed in the feature space, referred as KIT2FCM-F (with F standing for the feature space). In the second category, abbreviated as KIT2FCM-K (with K standing for the kernel space), the prototypes are retained in the kernel space and thus the prototypes must be approximated in the feature space by computing an inverse mapping from kernel space to feature space.

The advantage of the KIT2FCM-F clustering algorithm is that the prototypes reside in the feature space and are implicitly mapped to the kernel space through the use of the kernel function. Similar to IT2FCM, KIT2FCM-F use two fuzzifiers m_1 and m_2 to handle the uncertainty, it minimized the following objective function as

$$\begin{aligned} J_{m_1}(U,V) &= \sum_{i=1}^{c}\sum_{j=1}^{n} {u_{ij}}^{m_1} \parallel \phi(x_j) - \phi(v_i) \parallel^2 \\ J_{m_2}(U,V) &= \sum_{i=1}^{c}\sum_{j=1}^{n} {u_{ij}}^{m_2} \parallel \phi(x_j) - \phi(v_i) \parallel^2 \end{aligned} \tag{3}$$

Where c is the number of cluster, n the number of data points and $\parallel \phi(x_j) - \phi(v_i) \parallel$ is the Euclidean distance between the pattern x_j and the prototype v_i in the kernel space. By using the Euclidean distance $\parallel \phi(x_k) - \phi(v_i) \parallel$, the squared distance is computed in the kernel space using a kernel function

$$\left\| \phi\left(x_j\right) - \phi\left(v_i\right) \right\|^2 = k\left(x_j, x_j\right) + k\left(v_i, v_i\right) - 2k\left(x_j, v_i\right) \tag{4}$$

Upper/lower degrees of membership, $\overline{u}_{ij}$ and $\underline{u}_{ij}$ are determined as follows:

$$\overline{u}_{ij} = \begin{cases} \dfrac{1}{\sum\limits_{l=1}^{c}\left(\frac{d_{\phi ij}}{d_{\phi lj}}\right)^{2/(m_1-1)}} & \mathit{if}\ \dfrac{1}{\sum\limits_{l=1}^{c}\left(\frac{d_{\phi ij}}{d_{\phi lj}}\right)} < \dfrac{1}{c} \\ \dfrac{1}{\sum\limits_{l=1}^{c}\left(\frac{d_{\phi ij}}{d_{\phi lj}}\right)^{2/(m_2-1)}} & otherwise \end{cases} \tag{5}$$

$$\underline{u}_{ij} = \begin{cases} \dfrac{1}{\sum\limits_{l=1}^{c}\left(\frac{d_{\phi ij}}{d_{\phi lj}}\right)^{2/(m_1-1)}} & \mathit{if}\ \dfrac{1}{\sum\limits_{l=1}^{c}\left(\frac{d_{\phi ij}}{d_{\phi lj}}\right)} \geq \dfrac{1}{c} \\ \dfrac{1}{\sum\limits_{l=1}^{c}\left(\frac{d_{\phi ij}}{d_{\phi lj}}\right)^{2/(m_2-1)}} & otherwise \end{cases} \tag{6}$$

in which $i = \overline{1,c}, j = \overline{1,n}, d_{\phi ij} = \left\|\phi\left(x_j\right) - \phi\left(v_i\right)\right\|$. If we use the Gaussian kernel then $k(x,x) = 1$ and $\left\|\phi\left(x_j\right) - \phi\left(v_i\right)\right\|^2 = 2(1 - k(x_j, v_i))$. The derivation of the prototypes depends on the specific selection of the kernel function. The calculation of the prototypes v_i for i =1,2,...,c with the Gaussian kernel and degree of membership $u_{ij} = \frac{\overline{u}_{ij} + \underline{u}_{ij}}{2}$ and m is a type 1 fuzzifier (m usually is 2) as follows

$$\begin{gathered} \nabla_{v_i} J = 0 \\ \sum_{j=1}^{n} u_{ij}^m \nabla_{v_i}\left(\left\|\phi\left(x_j\right) - \phi\left(v_i\right)\right\|^2\right) = 0 \\ \sum_{j=1}^{n} u_{ij}^m \nabla_{v_i}\left(2 - 2k\left(x_j, v_i\right)\right) = 0 \\ \sum_{j=1}^{n} u_{ij}^m \nabla_{v_i}\left(e^{-\|x_j - v_i\|^2/\sigma^2}\right) = 0 \\ \sum_{j=1}^{n} u_{ij}^m \nabla_{v_i}\left(\left\|\phi\left(x_j\right) - \phi\left(v_i\right)\right\|^2\right) = 0 \\ \sum_{j=1}^{n} u_{ij}^m \left(e^{-\|x_j - v_i\|^2/\sigma^2}\right)\left(\tfrac{2}{\sigma^2}\left(x_j - v_i\right)\right) = 0 \\ v_i \sum_{j=1}^{n} u_{ij}^m k\left(x_j, v_i\right) = \sum_{j=1}^{n} u_{ij}^m k\left(x_j, v_i\right) x_j \\ v_i = \frac{\sum\limits_{j=1}^{n} u_{ij}^m k(x_j, v_i) x_j}{\sum\limits_{j=1}^{N} u_{ij}^m k(x_j, v_i)} \end{gathered} \tag{7}$$

KIT2FCM-F Algorithm

In general, fuzzy memberships in interval type-2 fuzzy C means algorithm is achieved by computing the relative distance among the patterns and cluster centroids. Hence, to define the interval of primary membership for a pattern, we define the lower and upper interval memberships using two different values of m . In (5)-(6), m_1 and m_2 are fuzzifiers which represent different fuzzy degrees. We define the interval of a primary membership for a pattern, as the highest and lowest primary membership for a pattern. These values are denoted by upper and lower membership for a pattern, respectively.

Because each pattern has membership interval as the upper $\overline{u}$ and the lower $\underline{u}$, each centroid of cluster is represented by the interval between v^L and v^R.

The iterative algorithm for finding centroids

- Step 1: Find $\overline{u}_{ij}, \underline{u}_{ij}$, by the equations (5)-(6).
- Step 2: Set m = arbitrary and $m \geq 1$;
  Compute $v'_j = (v'_{j1}, ..., v'_{jM})$ by (7) with $u_{ij} = \frac{(\overline{u}_{ij} + \underline{u}_{ij})}{2}$.
  Sort N patterns on each of M features in ascending order.
- Step 3: Find index k such that: $x_{kl} \leq v'_{jl} \leq x_{(k+1)l}$ with $k = 1, .., N$ and $l = 1, .., M$.
  Update u_{ij}:
  o If $i \leq k$ then $u_{ij} = \underline{u}_{ij}$.
  o If $i > k$ then $u_{ij} = \overline{u}_{ij}$.
  Define v_L or v_R
- Step 4: Define v_L or v_R Compute v''_j by (7).
  Compare v'_{jl} with v''_{jl}
  o If $v'_{jl} = v''_{jl}$ then $v_R = v'_j$.
  o Otherwise:
  Set $v'_{jl} = v''_{jl}$.
  Back to Step 3.

In Case, to define v_L:

- In step 3 we modify
  Update u_{ij}: o If $i \leq k$ then $u_{ij} = \overline{u}_{ij}$. o If $i > k$ then $u_{ij} = \underline{u}_{ij}$. and
- In step 4 replace V_R with v_L.

Finish the iterative algorithm for finding centroid

In this section, we represent KIT2FCM-F algorithm as the following diagram.

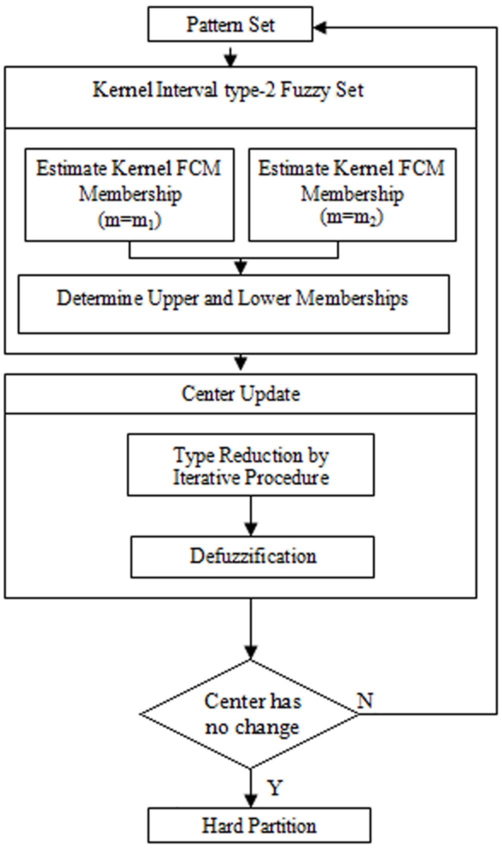


Fig. 1. Type-2 FCM Diagram

In the Figure 1, we can't do clustering algorithm with centroids V_R,v_L and memberships $\overline{u}$, $\underline{u}$. We need do type reduction for them.

After obtaining v_i^R, v_i^L, type-reduction is applied to get centroid of clusters as follows:

*** Compute the mean of centroid , v_j as:** Compute the mean of centroid , v_j as:

$$v_i = (v_i^R + v_i^L)/2 \tag{8}$$

For membership grades:

$$u_i(x_k) = (u_i^R(x_k) + u_i^L(x_k))/2, j = 1, ..., C \tag{9}$$

in which

$$u_i^L = \sum_{l=1}^{M} u_{il}/M, u_{il} = \begin{cases} \overline{u}_i(x_k) & \text{if } x_{il} \text{ uses } \overline{u}_i(x_k) \text{ for } v_i^L \\ \underline{u}_i(x_k) & otherwise \end{cases} \tag{10}$$

$$u_i^R = \sum_{l=1}^{M} u_{il}/M, u_{il} = \begin{cases} \overline{u}_i(x_k) & \text{if } x_{il} \text{ uses } \overline{u}_i(x_k) \text{ for } v_i^R \\ \underline{u}_i(x_k) & otherwise \end{cases} \tag{11}$$

Next, defuzzification for KIT2FCM is made as if $u_i(x_k) > u_j(x_k)$ for $j = 1, ..., C$ and $i \neq j$ then x_k is assigned to cluster i.

If the prototypes v_i are constructed in the kernel space, this type of KIT2FCM is referred as KIT2FCM-K. The objective function of KIT2FCM-K is

$$\begin{aligned} J_{m_1}(U,V) &= \sum_{i=1}^{c}\sum_{j=1}^{n} {u_{ij}}^{m_1} \| \phi(x_j) - v_i \|^2 \\ J_{m_2}(U,V) &= \sum_{i=1}^{c}\sum_{j=1}^{n} {u_{ij}}^{m_2} \| \phi(x_j) - v_i \|^2 \end{aligned} \tag{12}$$

Using the Euclidean distance and optimizing J_{m1} and J_{m2} with respect to v_i located in the kernel space such that $\nabla v_i J_{m_1} = 0$ or $\nabla v_i J_{m_2} = 0$, we obtain prototype v_i as follow:

$$v_i = \frac{\sum_{j=1}^{n} u_{ij}^m k(x_j)}{\sum_{j=1}^{n} u_{ij}^m} \tag{13}$$

Where m is a type 1 fuzzifier (m usually is 2). Upper/lower degrees of membership, $\overline{u}_{ij}$ and $\underline{u}_{ij}$ are calculated by 5 and 6 and with

$$d_{\phi ij}^2 = \|\phi(x_j) - v_i\|^2 = \phi(x_j)^T\phi(x_j) - 2\phi(x_j)^T v_i + {v_i}^T v_i \tag{14}$$

Inserting v_i from Eq(13) into Eq(14) gives:

$$\begin{aligned} \|\phi(x_j) - v_i\|^2 &= k(x_j, x_j) - \frac{2\sum_{h=1}^{n} u_{ih}^m k(x_j, x_h)}{\sum_{h=1}^{n} u_{ih}^m} \\ &+ \frac{\sum_{h=1}^{n}\sum_{l=1}^{n} u_{ih}^m u_{il}^m k(x_h, x_l)}{\left(\sum_{h=1}^{n} u_{ih}^m\right)^2} \end{aligned} \tag{15}$$

The advantage of KIT2FCM-K is that the prototypes are not constrained to the feature space; however, the disadvantage is that the prototypes are implicitly located in the kernel space and thus need to be approximated by an inverse mapping to the feature space. The method outlined in [13] iteratively determines approximate prototypes $\tilde{v}_i$ in the feature space by inverse mapping ϕ. The objective function to be minimized as:

$$\begin{aligned} V &= \sum_{i=1}^{c} \|\phi(\tilde{v}_i) - v_i\|^2 \\ &= \sum_{i=1}^{c}\left(k(\tilde{v}_i, \tilde{v}_i) - 2\frac{\sum_{t=1}^{N} u_{it}^m k(x_t, \tilde{v}_i)}{\sum_{t=1}^{N} u_{it}^m} + \frac{\sum_{t=1}^{N}\sum_{l=1}^{N} u_{it}^m u_{il}^m k(x_t, x_l)}{\left(\sum_{t=1}^{N} u_{it}^m\right)^2} \right) \end{aligned} \tag{16}$$

Solving $\nabla_{\tilde{v}_i} = 0$ requires knowledge of the kernel function k. If we use Gaussian kernel then $k(x_j, x_l)$ is independent of $\tilde{v}_i$, $k(\tilde{v}_i, \tilde{v}_i) = 1$ is independent of $\tilde{v}_i$ and $\nabla_{\tilde{v}_i} k(x_j, x_l) = 0$

The prototype expression for the Gaussian kernel for i =1,2,...,c is then given as [13]:

$$\tilde{v}_i = \frac{\sum_{t=1}^{n} u_{it}^m k(x_t, \tilde{v}_i) x_t}{\sum_{t=1}^{n} u_{it}^m k(x_t, \tilde{v}_i)} \tag{17}$$

Considering the polynomial kernel, we obtain the prototype expression for the polynomial kernel for i =1,2,...,c is [13]

$$\tilde{v}_i = \frac{\sum_{t=1}^{N} u_{it}^m \left(x_t^T \tilde{v}_i + \theta \right)^{p-1} x_t}{\left(\tilde{v}_i^T \tilde{v}_i + \theta \right)^{p-1} \sum_{t=1}^{N} u_{it}^m} \tag{18}$$

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

KIT2FCM-K algorithm

KIT2FCM-F algorithm contains two main steps.

In step 1: We also performs similarly to KIT2FCM-F algorithm above with equations 5,6 are calculated with the equations 13,14.

In step 2: We calculate $\tilde{v}_i$ following equation 17for Gaussian kernel or equation 18 for polynomial kernels until termination criteria satisfied or maximum iterations reached.

IV. Multiple kernel Interval Type 2 Fuzzy C-mean Clustering

For KIT2FCM, a crucial step is the selection of the best kernels among an extensive range of possibilities. This step is often heavily influenced by prior knowledge about the data and it is unclear which kernels are more suitable. Many real-world clustering applications, in which there are multiple potentially useful cues. Thus,it is necessary to aggregate features from different sources into a single aggregated feature. A multiple kernel interval type 2 fuzzy c-means (MKIT2FCM) algorithm extends the KIT2FCM by combining different kernels to obtain better results. The general framework of MKIT2FCM aims to minimize the objective function as the KIT2FCM, i.e.,

$$\begin{aligned} J_{m_1}(U,V) &= \sum_{i=1}^{c} \sum_{j=1}^{n} {u_{ij}}^{m_1} \| \phi_{com}(x_j) - \phi_{com}(v_i) \|^2 \\ J_{m_2}(U,V) &= \sum_{i=1}^{c} \sum_{j=1}^{n} {u_{ij}}^{m_2} \| \phi_{com}(x_j) - \phi_{com}(v_i) \|^2 \end{aligned} \tag{19}$$

or

$$\begin{aligned} J_{m_1}(U,V) &= \sum_{i=1}^{c} \sum_{j=1}^{n} {u_{ij}}^{m_1} \| \phi_{com}(x_i) - v_i \|^2 \\ J_{m_2}(U,V) &= \sum_{i=1}^{c} \sum_{j=1}^{n} {u_{ij}}^{m_2} \| \phi_{com}(x_i) - v_i \|^2 \end{aligned} \tag{20}$$

Where ϕ_{com} is the transformation defined by the combined kernels.

$$k_{com}(x,y) =< \phi_{com}(x), \phi_{com}(y) > \tag{21}$$

The composite kernel k_{com} is defined as a combination of multiple kernels using properties introduced in Theorem 2.2 For example, two simple composite kernels are $k = k_1 + \alpha * k_2$ or $k = k_1 * k_2$ A linearly combined kernel function is

$$k_{com} = {w_1}^b k_1 + {w_2}^b k_2 + ... + {w_l}^b k_l \tag{22}$$

Where $b > 1$ is a coefficient kernel. The regulation on weights, $w_1, w_2, ..., w_l$,is $\sum_{i=1}^{l} w_i = 1$. When the number of parameters in the combined kernel is small, the parameters can be adjusted by trial and error. For instance, the parameter α in the $k_{com} = k_1 + \alpha k_2$ can be selected by testing a group α in a predefined range or set. While the number of parameters in the combined kernel is large, the more feasible method is automatically adjusting these parameters in the learning algorithms. For example, $k_{com} = {w_1}^b k_1 + {w_2}^b k_2 + ... + {w_l}^b k_l$. Some learning algorithms that adjust the weights w_i automatically in typical kernel learning methods like multiple-kernel regressions and classifications [14] have been studied. Here, we propose a similar algorithm for MKIT2FCM using linearly combined kernels. The typical kernels are defined on $\mathbb{R}^p \times \mathbb{R}^p$ are: Gaussian kernel $k(x_i, x_j) = exp(-|x_i - x_j|^2 / r^2)$ And Polynomial kernel $k(x_i, x_j) = (x_i * x_j + d)^2$ Upper/lower degrees of membership, $\overline{u}_{ij}$ and $\underline{u}_{ij}$ are determined as follows:

$$\overline{u}_{ij} = \begin{cases} \dfrac{1}{\sum_{l=1}^{c} \left(\frac{d_{\phi_{com}}(x_j,v_i)}{d_{\phi_{com}}(x_j,v_l)} \right)^{2/(m_1-1)}} & if\ \dfrac{1}{\sum_{l=1}^{c} \left(\frac{d_{\phi_{com}}(x_j,v_i)}{d_{\phi_{com}}(x_j,v_l)} \right)} < \dfrac{1}{c} \\ \dfrac{1}{\sum_{l=1}^{c} \left(\frac{d_{\phi_{com}}(x_j,v_i)}{d_{\phi_{com}}(x_j,v_l)} \right)^{2/(m_2-1)}} & otherwise \end{cases} \tag{23}$$

$$\underline{u}_{ij} = \begin{cases} \dfrac{1}{\sum_{l=1}^{c} \left(\frac{d_{\phi_{com}}(x_j,v_i)}{d_{\phi_{com}}(x_j,v_l)} \right)^{2/(m_1-1)}} & if\ \dfrac{1}{\sum_{l=1}^{c} \left(\frac{d_{\phi_{com}}(x_j,v_i)}{d_{\phi_{com}}(x_j,v_l)} \right)} \geq \dfrac{1}{c} \\ \dfrac{1}{\sum_{l=1}^{c} \left(\frac{d_{\phi_{com}}(x_j,v_i)}{d_{\phi_{com}}(x_j,v_l)} \right)^{2/(m_2-1)}} & otherwise \end{cases} \tag{24}$$

(25)

$$\begin{aligned} d_{\phi_{com}}(x_j, v_i)^2 &= \| \phi_{com}(x_j) - v_i \|^2 \\ &= k_{com}(x_j, x_j) + \frac{2\sum_{h=1}^{n} (u_{ih})^m k_{com}(x_j, x_h)}{\sum_{h=1}^{n} (u_{ih})^m} + \\ &\frac{\sum_{h=1}^{n} \sum_{l=1}^{n} (u_{ih})^m (u_{il})^m k_{com}(x_h, x_l)}{\left(\sum_{h=1}^{n} (u_{ih})^m \right)^2} \end{aligned} \tag{26}$$

By introducing the Lagrange term of the constraint of weights into the objective function, defined as

$$J = \sum\nolimits_{i=1}^{c} \sum\nolimits_{j=1}^{n} u_{ij}^m \| \phi_{com}(x_j) - v_i \|^2 + \eta \left(1 - \sum\nolimits_{i=1}^{l} w_i \right) \tag{27}$$

By taking derivative of J over w_i and assuming the result zero, we will obtain updating rule of the total weights.

$$\frac{\partial J}{\partial w_i} = 0\,(i = 1...l) \Rightarrow w_i = 1 / \left(\sum\nolimits_{h=1}^{n} J_i / J_h^{1/(b-1)} \right)^{1/(b-1)} \tag{28}$$

Where

$$J_h(u,v) = \sum\nolimits_{i=1}^{c} \sum\nolimits_{j=1}^{n} u_{ij}^m \| \phi_h(x_j) - v_i \|^2 \tag{29}$$

Here ϕ_h is the transformation function defined by k_h (h=1,2,...l) in Eq(22) and

$$\begin{aligned} \| \phi_h(x_j) - v_i \|^2 = k_h(x_j, x_j) - \frac{2\sum_{h=1}^{n} (u_{ih})^m k_h(x_h, x_j)}{\sum_{h=1}^{n} (u_{ih})^m} \\ + \frac{\sum_{h=1}^{n} \sum_{l=1}^{n} (u_{ih})^m (u_{il})^m k_h(x_h, x_l)}{\left(\sum_{h=1}^{n} (u_{ih})^m \right)^2} \end{aligned} \tag{30}$$

Multiple kernel interval type 2 fuzzy c-means algorithm

Given a set of n data points $X = \{x_i\}_{i=1}^n$, a set of kernel functions $\{k_i\}_{i=1}^l$, parameters m_1, m_2 and the desired number of clusters c. Output a membership matrix $U = \{u_{ic}\}_{i,c=1}^n, c$ and weights $\{w_i\}_{i=1}^l$ for the kernels.

Step 1: Initialize centroid matrix $V^0 = \{v_i\}_{i=1}^c$ by choosing random from dataset and the membership matrix U^0 follow the equation:

$$u_{ij} = \frac{1}{\sum_{l=1}^{c}\left(\frac{d_{ij}}{d_{lj}}\right)^{2/(m-1)}} \tag{31}$$

Where m is a constant, $m > 1$ and $d_{ij} = d(x_j - v_i) = \|x_j - v_i\|$

Step 2: Repeat: + Calculate the weights w_i for the kernels following the Eq(28)

+ Calculate Interval membership values $\overline{u_{ij}}$ and $\underline{u_{ij}}$ followed Eq(23,24) and Eq(26).

+ Update the centroid matrix followed the iterative algorithm for finding centroids in KIT2FCM and Eq(8).

+ Update the membership matrix by Eq(9)

+ Assign data x_j to cluster c_i if data $(u_j(x_i) > u_k(x_i))$, $k = 1, .., c$ and $j \neq k$.

Step 3: Termination criteria satisfied or maximum iterations reached Return U and V else back to step 2.

V. Experiments

The first experiment, the well-known datasets are IRIS, Wisconsin Diagnostic Breast Cancer (WDBC), Wine [23] is considered. These datasets was classified by the various algorithms such as KFCM, IT2FCM and KIT2FCM (the proposed algorithm). The performance of the classification was evaluated with the True Positive Rate (TPR) and False Positive Rate (FPR) which are defined by the following equations:

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{32}$$

where TP is the number of correctly classified data and FN is the number of incorrectly misclassified data.

$$\text{FPR} = \frac{\text{FP}}{\text{TN} + \text{FP}} \tag{33}$$

where FP is the number of incorrectly classified data and TN is the number of correctly misclassified data

This experiment was carried out as follows: Every experiment dataset is randomly divided into K sets of approximately equal size by 2/3 the size of experiment dataset and K is set to 20. Experimental work will be conducted K times on K different data sets. We randomly initialized centroids with the given number of clusters from the experiment data and defined the stop criterion: the number of Iterations G= 20 and the error $\sigma < 0.00001$. With kernel methods, the Gaussian kernel is used for all datasets. The results of the clustering or the quality of classification is shown in the indicators TPR and FPR.

Table I shows the evaluation results of the different algorithms. The efficient algorithms have larger TPR value and smaller FTR value.

TABLE I
Classification results of KFCM,IT2FCM, KIT2FCM-F, KIT2FCM-K

Methods	KFCM	IT2FCM	KIT2FCM-F	KIT2FCM-K
(WDBC,c=2) TPR(%)	91.5 ± 3.5	92.8 ± 5.5	95.3 ± 3.3	94.1 ± 3.2
FPR(%)	3.4 ± 1.2	3.4 ± 1.5	1.9 ± 0.9	1.5 ± 0.6
(IRIS,c=3) TPR(%)	91.4 ± 3.5	93.6 ± 5.7	96.1 ± 4.2	93.2 ± 4.1
FPR(%)	3.1 ± 1.5	4.3 ± 1.6	1.7 ± 0.6	1.4 ± 0.5
(Wine,c=3) TPR(%)	95.5 ± 2.5	94.8 ± 2.1	97.9 ± 1.2	96.8 ± 1.2
FPR(%)	2.0 ± 0.8	2.3 ± 1.4	0.7 ± 0.4	0.9 ±0.4

The second experiments are more visible could be found from multi spectral remote sensing images. The pixel information in these images inherits from different temporal sensors. As a result, we can define different kernels for different temperature channels and apply the combined kernel in a multiple-kernel learning algorithm. With the Multiple Kernel algorithms, the data inputs contain Gaussian kernel k_1 for pixel intensities and Gaussian kernel k_2 for spatial information.

To exploit the spatial information, a spatial function is defined as

$$h_{ij} = \sum_{k \in NB(x_j)} u_{ik} \tag{34}$$

where $NB(x_j)$ represents a square window centered on pixel x_j in the spatial domain which was a window 5×5 . Just like the membership function,the spatial function h_{ij} represents the probability that pixel x_j belongs to i^{th} cluster.

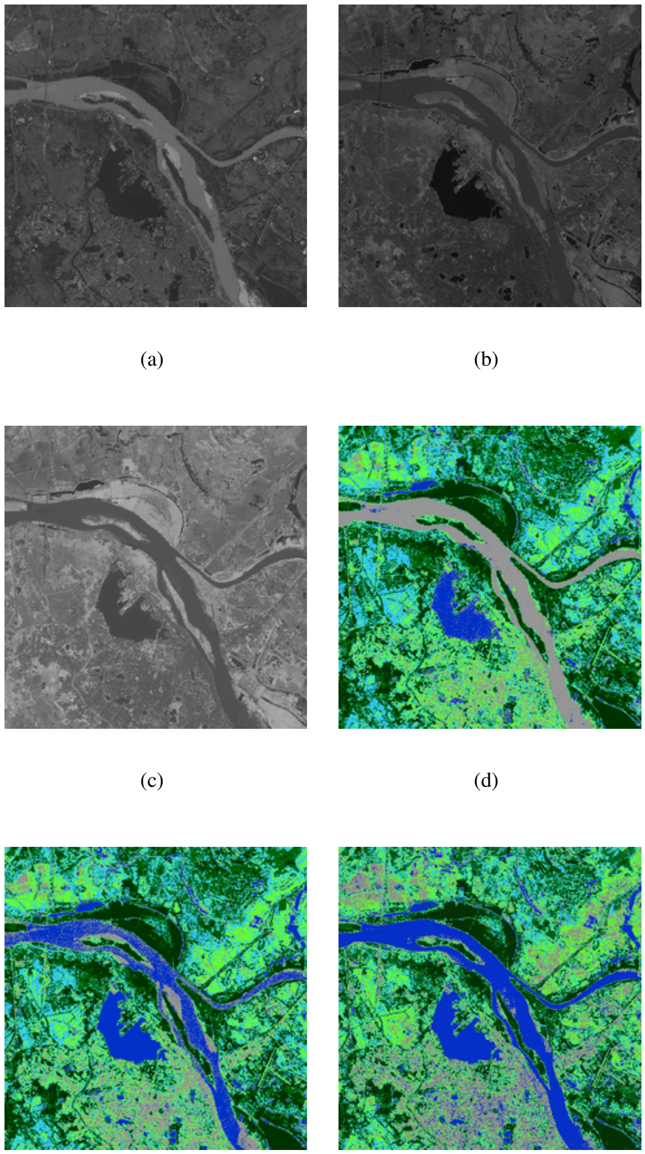
While kernel algorithm only gets Gaussian kernel k for pixel intensities as data inputs.

Test Data from LANDSAT-7 image is Ha Noi region(Vietnam) with square of area:20306.25 *hectares*.

In Figure (2), colors of classes are denoted as follows: Class 1. Rivers, ponds, lakes. Class 2. Rocks, bare soil. Class 3. Fields, sparse tree. Class 4. Planted forests, low woods. Class 5. Perennial tree crops. Class 6. Jungles.

The experimental results are shown in (2) in which a) for NIR channel, b) for VR channel, c) for NDVI image generating from NIR and VR channels, d), e), f) for result image of the classification of MKFCM and KIT2FCM-F, MKIT2FCM algorithm, respectively. Figure (3) is the comparing results between MKFCM, KIT2FCM-F, MKIT2FCM and the the result of The Vietnamese Center of Remote Sensing Technology (VCRST) on each class (in percentage %). The significant difference between the algorithms MKFCM, KIT2FCM-F, MKIT2FCM in determining the area of regions, the largest difference between the algorithms up to 10%. Compare these experimental results with the result of VCRST, with the result of MKFCM algorithm, the largest difference is 11% and KIT2FCM-F algorithm is 8%. Meanwhile, the result of MKIT2FCM algorithm does not exceed 5% difference.

Besides, Figure 2 also clearly show that MKIT2FCM clas-



(a) (b)

(c) (d)

(e) (f)

Fig. 2. Study data 2: Result of land cover classification. a) NIR channel image; b) VR channel image; c) NDVI image; d) MKFCM classification; e) KIT2FCM-F classification; f) MKIT2FCM classification

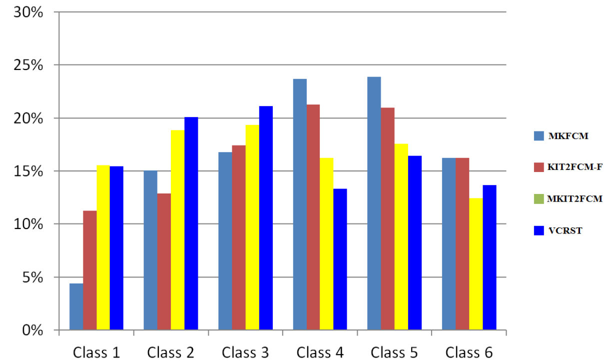


Fig. 3. Study data 2: Comparisons between the result of MKFCM, KIT2FCM-F, MKIT2FCM and the result of VCRST

sifier gives clusters better. Low accuracy of classification for class 1 may see in Figure 2 d) (MKFCM) and e) (KIT2FCM-F), especially in river region (center of image).

VI. Conclusions

This paper presented a fuzzy clustering algorithm based on kernel technique which improved the clustering results and overcome the drawbacks of the conventional clustering algorithms which depend on the spherical distances. The proposed approach have solved the problem of combining between kernel technique and type 2 fuzzy sets to handle the uncertainty better in kernel space.The experiments are done based on well-known dataset with the statistics show that the algorithm generates good quality clusters.

The next goal is some researches related to use the genetic algorithm to automatic update the parameters of MKIT2FCM.

References

[1] C. Hwang, F. C. H. Rhee, "Uncertain Fuzzy clustering: Interval Type-2 Fuzzy Approach to C-Means", *IEEE Tran. on Fuzzy Systems*, Vol. 15, no.1, pp. 107-120, 2007.

[2] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, A. Y. Wu, "An Efficient k-Means Clustering Algorithm: Analysis and Implementation", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol 24, no.7, pp. 881-893, 2002.

[3] K.R. Zalik, "An efcient k-means clustering algorithm", *Pattern Recognition Letters*, Vol. 29, pp. 1385 - 1391, 2008.

[4] J.M. Mendel and R.I. John, "Type-2 fuzzy set made simple", *IEEE Trans. on Fuzzy Systems*, vol. 10, no.2, pp.117 - 127, 2002.

[5] N. Karnik, J.M. Mendel, "Operations on Type-2 Fuzzy Sets", *Fuzzy Sets and Systems*, vol. 122, pp.327-348, 2001.

[6] Q. Liang , J.M. Mendel, "Interval Type-2 Fuzzy Logic Systems: Theory and Design", *IEEE Trans. on Fuzzy Systems*, Vol.8, no.5, 535-550, 2000.

[7] L. Chen, C. L. Philip Chen, "A Multiple-Kernel Fuzzy C-Means Algorithm for Image Segmentation", *IEEE Trans. on System, Man, and Cybernetics*, Vol.41, no.5, pp.1263-74, 2011.

[8] J.M. Mendel, R.I. John, F. Liu, "Interval Type-2 Fuzzy Logic Systems Made Simple", *IEEE Trans. on Fuzzy Systems*, Vol.14, no.6, 808-821, 2006.

[9] D. Graves, W. Pedrycz, "Fuzzy C-Means, GustafsonKessel FCM, and Kernel-based FCM: a comparative study", *Advances in Soft Computing*, Vol.41, pp.140-149, 2007.

[10] Long Thanh Ngo, Binh Huy Pham, "Approach to Image Segmentation Based on Interval Type-2 Fuzzy Subtractive Clustering", in *Springer LNCS 7197, ACIIDS (2)* , pp. 1-10, 2012.

[11] J. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York: Plenum, 1981.

[12] H. Shen, J. Yang, S. Wang, and X. Liu, "Attribute weighted mercer kernel based fuzzy clustering algorithm for general non-spherical datasets", *Soft Computing*, vol. 10, no.11, pp. 1061-1073, 2006.

[13] S. Zhou, J. Gan, "Mercer kernel fuzzy c-means algorithm and prototypes of clusters", Proceedings of *Conference on Data Engineering and Automated Learning*, Vol. 3177, pp. 613-618, 2004.

[14] F. R. Bach, G. Lanckriet, M. I. Jordan, "Multiple kernel learning, conic duality and the SMO algorithm", *Proc. 21st International Conference on Machine Learning*, pp.41-48, 2004.

[15] M. Girolami, "Mercer kernel-based clustering in feature space", *IEEE Trans. on Neural Networks*, vol. 13, no.3, pp. 780-784, 2002.

[16] D.-Q. Zhang and S.-C. Chen, "Clustering incomplete data using kernel-based fuzzy c-means algorithm", *Neural Processing Letters*, vol. 18, no.3, pp.155-162, 2003.

[17] J. Shawe-Taylor and N. Cristianini, "Kernel Methods for Pattern Analysis", *Cambridge University Press*, 2004.

[18] D. Graves and W. Pedrycz, "Kernel-based fuzzy clustering and fuzzy clustering: A comparative experimental study", *Fuzzy Sets and Systems*, vol. 161, no.4, pp.522-543, 2010.
[19] G. Camps-Valls., L. Gomez-Chova., J. Munoz-Mari., J. Vila-Frances., and J. Calpe-Maravilla., "Composite Kernels for Hyperspectral Image Classification", IEEE Geoscience and Remote Sensing Letters, vol. 3, no.1, pp.93-97, 2006.
[20] S. Chen., and D. Zhang., "Robust Image Segmentation Using FCM With Spatial Constraints Based on new Kernel-Induced Distance Measure", *IEEE Trans. on System, Man, and Cybernetics, Part B*, Vol.34, no.4, pp.1907-1916, 2004.
[21] W. L. Cai., S. C. Chen., and D. Q. Zhang., "Fast and Robust Fuzzy C-Means Clustering Algorithms Incorporating Local Information for image Segmentation", *Pattern Recognition*, vol. 40, no.3, pp.825-838, 2007.
[22] R. Herbrich, *Learning Kernel Classifiers*, MIT Press, 2002.
[23] Datasets: http://www.ics.uci.edu/ mlearn/mlrepository.html