CHAPTER 2

MULTIPLE KERNELS FUZZY C-MEANS ALGORITHM
FOR SATELLITE IMAGE SEGMENTATION

2.1 PROLOGUE

Clustering, particularly Fuzzy C-Means (FCM)-based clustering and its variants have been widely used in the task of image segmentation due to their simplicity and fast convergence. By carefully selecting input features such as pixel color, intensity, texture, or a weighted combination of these data, the FCM algorithm can segment images to several regions in accordance with resulting clusters. The Kernel FCM (KFCM) algorithm is an extension of FCM, 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 easy to be separated or clustered. Recently, developments on kernel methods and their applications have emphasized the need to consider multiple kernels or composite kernels instead of a single fixed kernel. With multiple kernels, the kernel methods gain more flexibility on kernel selections. This Chapter focuses on Multiple Kernel FCM (MKFCM) algorithm for satellite image segmentation that uses a linear composite of multiple kernels and the updating rules of the linear coefficients kernel are obtained automatically.

2.2 INTRODUCTION

Clustering is the process of dividing data into groups of similar objects. Each group, called a cluster, consists of objects that are similar to one another and dissimilar to objects of other groups. A cluster is therefore a
collection of objects which are similar between them and are dissimilar to the objects belonging to other clusters. We can show this with a simple graphical example as shown in the Figure 2.1. In this case we easily identify the 4 clusters into which the data can be divided; the similarity criterion is distance, two or more objects belong to the same cluster if they are close according to a given distance (in this case geometrical distance). This is called distance-based clustering.

Another kind of clustering is conceptual clustering, two or more objects belong to the same cluster if this one defines a concept common to all that objects. In other words, objects are grouped according to their fit to descriptive concepts, not according to simple similarity measures.

![Figure 2.1 Clustering Process](image)

**2.2.1 Goals of Clustering**

The goal of clustering is to determine the intrinsic grouping in a set of unlabeled data. Consequently, it is the user which must supply this criterion, in such a way that the result of the clustering will suit their needs.
For instance, we could be interested in finding representatives for homogeneous groups (data reduction), in finding natural clusters and describe their unknown properties, in finding useful and suitable groupings or in finding unusual data objects.

2.2.2 Clustering Algorithms

Clustering algorithms may be classified as Exclusive Clustering, Overlapping Clustering, Hierarchical Clustering and Probabilistic Clustering. Selection of algorithm is based on their cluster groups.

In the first case data are grouped in an exclusive way, so that if a certain data belong to a definite cluster then it could not be included in another cluster. A simple example of that is shown in the Figure 2.2, where the separation of points is achieved by a straight line on a bi-dimensional plane. On the contrary in the second type, the overlapping clustering, uses fuzzy sets to cluster data, so that each point may belong to two or more clusters with different degrees of membership. In this case, data will be associated to an appropriate membership value.

![Figure 2.2 Cluster Separation](image)
Instead, a hierarchical clustering algorithm is based on the union between the two nearest clusters. The beginning condition is realized by setting every data as a cluster. After a few iterations it reaches the final clusters wanted. Finally, the last kind of clustering uses a completely probabilistic approach. The mostly used clustering algorithms are K-means, Fuzzy C-means, hierarchical clustering and mixture of Gaussians. Each of these algorithms is belong to one of the clustering types mentioned above. In these algorithms the k-means algorithms is an important clustering algorithm. In these the overlapped algorithm is Fuzzy c-means algorithm. Hierarchical clustering algorithm is obvious and lastly mixture of Gaussians is a probabilistic clustering algorithm (Makoto and Takenobu 1995).

2.2.3 Distance Measure

An important component of a clustering algorithm is the distance measure between data points. If the components of the data instance vectors are all in the same physical units then it is possible that the simple Euclidean distance metric is sufficient to successfully group similar data instances. However, even in this case the Euclidean distance can sometimes be misleading. Figure 2.3 illustrates this concept with an example of the width and height measurements of an object. Despite both measurements being taken in the same physical units, an informed decision has to be made as to the relative scaling. The Figure 2.3 shows, different scalings can lead to different clusterings.
Notice that this is not only a graphic issue, the problem arises from the mathematical formulae used to combine the distances between the single components of the data feature vectors into a unique distance measure that can be used for clustering purposes, different formulae leads to different clustering Minkowski Metric. For higher dimensional data, a popular measure is the Minkowski Metric given by equation (2.1),

\[ d_p(x_i, x_j) = \left( \sum_{k=1}^{d} |x_{ik} - x_{jk}|^p \right)^{\frac{1}{p}} \]  

(2.1)

Where, d is the dimensionality of the data. The Euclidean distance is a special case where p=2, while Manhattan metric has p=1. However, there are no general theoretical guidelines for selecting a measure for any given application.

It is often the case that the components of the data feature vectors are not immediately comparable. It can be that the components are not continuous variables, like length, but nominal categories, such as the days of
the week. In these cases again, domain knowledge must be used to formulate an appropriate measure.

2.3 RELATED RESEARCH

Image analysis is based on the extraction of meaningful information and can involve many steps, such as pre-processing, segmentation and characterization of identified objects (Rojas and Fernandez-Reyes 2005). Particularly, the identification of the types of object constitutes an essential issue in pattern recognition due to its practical importance, such as in the treatment of images obtained from satellite prospection.

In fact, image segmentation can be understood as the process of assigning a label to every pixel in an image, such that pixels with the same label represent the same object, or its parts. Segmentation algorithms can be classified into different categories based on segmentation techniques used such as the feature thresholding (Chuang et al. 2006), template matching (Szeliski et al. 1999), region-based technique and clustering. The techniques have their own limitations and advantages in terms of suitability, performance and computational cost. The low-level segmentation techniques are known to be fast and simple, but these methods simply analyze an image by reducing the amount of data to be processed.

This problem can result in loss of important information. Moreover, the low-level segmentation techniques may incorrectly identify region or boundary of an object due to the distraction of noise in an image. Clustering, particularly Fuzzy C-Means (FCM)-based clustering and its variants, have been widely used in the task of image segmentation due to their simplicity and fast convergence (Tolias and Panas 1998) (Jafari et al. 2009). By carefully selecting input features such as pixel color, intensity, texture, or a weighted combination of these data, the FCM algorithm can segment images
to several regions in accordance with resulting clusters. Recently, the FCM and other clustering-based image-segmentation approaches are improved by including the local spatial information of pixels in classical clustering procedures. In addition to the incorporation of local spatial information, the kernelization of FCM has achieved an important performance improvement (Chen and Zhang 2004) (Liu and Wang 2006).

The kernel FCM (KFCM) algorithm is an extension of FCM, which maps the original inputs into a much higher dimensional Hilbert space by some transform function. After reproduction in the kernel Hilbert space, the data are more easily separated or clustered. 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 neighboring pixels. Cai et al. (2007) applied the idea of kernel methods in the calculation of the distances between the examples and the cluster centers. 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 centers (also computed in the extended Hilbert space) is appended to the objective function. Multiple-kernel methods provide us a great tool to fuse information from different sources (Xia et al. 2007). It is necessary to clarify that, the term “multiple kernel” in a wider sense than the one used in machine learning community.

Multiresolution segmentation is a bottom up region merging technique starting with one-pixel objects. In numerous subsequent steps, smaller image objects are merged into bigger ones. Throughout this pair wise clustering process, the underlying optimization procedure minimizes the weighted heterogeneity of resulting image objects, where $n$ is the size of a
segment and his an arbitrary definition of heterogeneity (Chen and Zhang 2004). In each step, a pair of adjacent image objects is merged which stands for the smallest growth of the defined heterogeneity. If the smallest growth exceeds the threshold defined by the scale parameter, the process stops.

Multi-resolution segmentation is a local optimization procedure. The entropy based methodology for segmentation of satellite images is performed as follows. Images are divided into square windows with a fixed size L, the entropy is calculated for each window, and then a classification methodology is applied for the identification of the category on the respective windows. The classification approach can be supervised or non-supervised. Supervised classification needs a training set composed by windows whose classes are previously known (prototypes), such as rural and urban areas.

2.4 K-MEANS CLUSTERING

K-means is one of the simplest unsupervised learning algorithms that solve the well known clustering problem (Hartigan and Wong 1979) fuzzy. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priorly. The main idea is to define k centroids, one for each cluster. These centroids should be placed in a cunning way because different location causes different result. So, the better choice is to place them as much far away as possible from each other. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When no point is pending, the first step is completed and an early group age is done. At this point we need to re-calculate k new centroids. After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new centroid. A loop has been generated. As a result of this loop we may notice that the k centroids change their location step by step until no more changes are done. In other words centroids do not move any more.
Finally, this algorithm aims at minimizing an objective function, in this case it is a squared error function. The objective function is given by

$$J = \sum_{i=1}^{k} \sum_{j=1}^{n} \|x_i - \theta_j\|^2$$  \hspace{1cm} (2.2)

where $\|x_i - \theta_j\|^2$ is a chosen distance measure between a data point $x_i$ and the cluster centre $\theta_j$ is an indicator of the distance of the $n$ data points from their respective cluster centres.

The algorithm is composed of the following steps:

1. Place $K$ points into the space represented by the objects that are being clustered. These points represent initial group centroids.
2. Assign each object to the group that has the closest centroids.
3. When all objects have been assigned, recalculate the positions of the $K$ centroids.
4. Repeat Steps 2 and 3 until the centroids no longer move.
5. This produces a separation of the objects into groups from which the metric to be minimized can be calculated.

Although it can be proved that the procedure will always terminate, the k-means algorithm does not necessarily find the most optimal configuration, corresponding to the global objective function minimum. The algorithm is also significantly sensitive to the initial randomly selected cluster centers. The k-means algorithm can be run multiple times to reduce this effect. K-means is a simple algorithm that has been adapted to many problem domains and it shows the very good results.
2.5 FUZZY CLUSTERING

In fuzzy clustering, each point has a degree of belonging to clusters, as in fuzzy logic, rather than belonging completely to just one cluster. Thus, points on the edge of a cluster may be in the cluster to a lesser degree than points in the center of cluster. An overview and comparison of different fuzzy clustering algorithms is available (Hwang and Rhee 2007).

Any point $x$ has a set of coefficients giving the degree of being in the $k_{th}$ cluster $w_k(x)$. With fuzzy c-means, the centroid of a cluster is the mean of all points, weighted by their degree of belonging to the cluster and is given by equation (3.3)

$$C_k = \frac{\sum w_k(x)x}{\sum w_k(x)} \quad (2.3)$$

The degree of belonging, $w_k(x)$, is related inversely to the distance from $x$ to the cluster center as calculated on the previous pass. It also depends on a parameter $m$ that controls how much weight is given to the closest center. The Fuzzy C-Means algorithm is very similar to the K-Means algorithm:

- Choose a number of clusters.
- Assign randomly to each point coefficients for being in the clusters.
- Repeat until the algorithm has converged (that is, the coefficients' change between two iterations is no more than $c$, the given sensitivity threshold):
  - Compute the centroid for each cluster, using the above formula.
o For each point, compute its coefficients of being in the clusters, using the formula above.

The algorithm minimizes intra-cluster variance as well, but has the same problems as k-means; the minimum is a local minimum, and the results depend on the initial choice of weights. The expectation-maximization algorithm is a more statistically formalized method which includes some of these ideas: partial membership in classes.

Fuzzy C-Means has been a very important tool for image processing in clustering objects in an image. In the 70's, mathematicians introduced the spatial term into the FCM algorithm to improve the accuracy of clustering under noise.

### 2.6 FOUNDATIONS OF KFCM

Given a data set $X = \{x_1, x_2, \ldots, x_n\}$, where the data point $x_j \in \mathbb{R}^p (j = 1, \ldots, n)$, $n$ is the number of data, and $p$ is the input dimension of a data point, traditional FCM groups $X$ into $c$ clusters by minimizing the weighted sum of distances between the data and the cluster centers or prototypes and is defined as

$$Q = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^m \|x_j - o_i\|^2 u_{ij}$$  \hspace{1cm} (2.4)

Here, $\|\|$ is the Euclidean distance. $u_{ij}$ is the membership of data $x_j$ belonging to cluster $i$, which is represented by the prototype $o_i$. The constraint on $u_{ij}$ is $\sum_{i=1}^{c} u_{ij} = 1$ and $m$ is the fuzzification coefficient, which usually takes the value of 2. As an enhancement of classical FCM, the KFCM maps the data set $X$ from the feature space or the data space $\Xi \in \mathbb{R}^p$ into a much higher
dimensional space \( H \) (a Hilbert space usually called kernel space) by a
transform function \( \varphi : \Xi \rightarrow H \). 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
FCM algorithms. Generally, the transform function \( \varphi \) is not given out
explicitly, but the kernel function is given and it is defined as \( k: \Xi \times \Xi \rightarrow \mathbb{R} \)

\[
  k(x, y) = \langle \varphi(x), \varphi(y) \rangle \quad \forall x, y \in \Xi \quad (2.5)
\]

where \( \langle , \rangle \) is the inner product for Hilbert space \( H \). Such kernel functions are
usually called Mercer kernels. Given a Mercer kernel \( k \), we know that there is
always a transform function \( \varphi : \Xi \rightarrow H \) that satisfies \( k(x, y) = \langle \varphi(x), \varphi(y) \rangle \),
although sometimes, we do not know the specific form of \( \varphi \) widely used
Mercer kernels include the Gaussian kernel \( k(x, y) = \exp \left( -\|x - y\|^2 / \sigma^2 \right) \) and
the polynomial kernel \( k(x, y) = (x \cdot y + d)^2 \). They are both defined over \( \mathbb{R}^n \times \mathbb{R}^n \). Clearly, due to the fact that we only know the kernel functions, we need to
solve the clustering problems in the kernel space by only using kernel
functions, i.e., the inner product of the transform function. Usually this is
called kernel trick.

There are two types of KFCM. If the prototypes \( \mathbf{o}_i \) are constructed
in the kernel space, this type of KFCM is referred as KFCM-K (with K
standing for the kernel space). The objective function of KFCM-K is given by

\[
  Q = \sum_{i=1}^{n} \sum_{j=1}^{n} u_{ij}^2 \| \varphi(x_j) - \varphi(o_i) \|^2 \quad (2.6)
\]

The learning algorithm of KFCM-K iteratively updates \( u_{ij} \) as
\[ U_{ij} = \frac{1}{\sum_{k=1}^{n} \frac{d_{ij}}{d_{kk}}} \quad (2.7) \]

Where
\[ d_{ij}^F = k(x_i, x_j) - \frac{\sum_{k=1}^{n} \sum_{l=1}^{n} w_{ik} w_{jl} k(x_k, x_l)}{\sum_{k=1}^{n} w_{ik}^2} \quad (2.8) \]

Another type of KFCM confines that the prototypes in the kernel space are actually mapped from the original data space or the feature space.

That is, the objective function is defined as
\[ Q = \sum_{i=1}^{c} \sum_{j=1}^{n} U_{ij}^F \| \varphi(x_j) - \varphi(o_i) \|^2 \quad (2.9) \]

This type of KFCM is referred as KFCM-F (with F standing for feature space/data space). Usually, only the Gaussian kernel \( k(x, y) = \exp(-|x - y|^2/\sigma^2) \) is applied in KFCM-F and because \( k(x, x) = 1 \) for Gaussian kernel
\[ \| \varphi(x_j) - \varphi(o_i) \|^2 = 2 (1 - k(x_j, o_i)) \quad (2.10) \]

The objective function in the equation (3.9) is then reformulated as
\[ Q = \sum_{i=1}^{c} \sum_{j=1}^{n} U_{ij}^F (1 - k(x_j, o_i)) \quad (2.11) \]

Here, \( 1 - k(x_j, o_i) \) can be considered as a robust distance measurement derived in the kernel space, iteratively update the prototypes and memberships as given in the equation (2.12) and equation (2.13)
\[ u_{ij} = \frac{(1-K(x_j, x_j))^{(1/4)}}{\sum_{l=1}^m (1-K(x_j, x_l))^{(1/4)}} \]  
\[ \alpha_i = \frac{\sum_{j=1}^N w_{ij} r(x_j, x_i) x_j}{\sum_{j=1}^N w_{ij} r(x_j, x_i)} \]  

\[ u_{ij} = \frac{(1-K(x_j, x_j))^{(1/4)}}{\sum_{l=1}^m (1-K(x_j, x_l))^{(1/4)}} \]  
\[ \alpha_i = \frac{\sum_{j=1}^N w_{ij} r(x_j, x_i) x_j}{\sum_{j=1}^N w_{ij} r(x_j, x_i)} \]  

2.7 MKFCM

Before the introduction of the MKFCM, we first list some necessary Mercer kernels properties in the following.

Theorem 1: Let \( k_1 \) and \( k_2 \) be kernels over \( \Xi \times \Xi, \Xi \subseteq \mathbb{R}^p \), and \( k_3 \) be a kernel over \( \mathbb{R}^p \times \mathbb{R}^p \). Let function \( \psi : \Xi \rightarrow \mathbb{R}^p \)

1) \( k (x, y) = k_1(x, y) + k_2(x, y) \) is a kernel.
2) \( k (x, y) = \alpha k_1(x, y) \) is a kernel, when \( \alpha > 0 \).
3) \( k (x, y) = k_1(x, y)k_2(x, y) \) is a kernel.
4) \( k (x, y) = k_3(\psi(x), \psi(y)) \) is a kernel.

The general framework of MKFCM aims to minimize the same objective function as the single fixed KFCM, i.e.,

\[ Q = \sum_{i=1}^m \sum_{j=1}^N u_{ij} \left\| \phi_{\text{com}}(x_j) - \alpha_i \right\|^2 \]  
\[ \text{or} \]  
\[ Q = \sum_{i=1}^m \sum_{j=1}^N u_{ij} \left\| \phi_{\text{com}}(x_j) - \phi_{\text{com}}(\alpha_i) \right\|^2 \]  

Comparing equations (2.6), (2.9) to equations (2.14), (2.15), the only difference between them is that the transform function \( \phi \) in equation (2.6) and equation (2.9) is changed to \( \phi_{\text{com}} \), which is derived from a composite kernel \( k_{\text{com}}(x, y) = \langle \phi_{\text{com}}(x), \phi_{\text{com}}(y) \rangle \). The composite kernel \( k_{\text{com}} \)
is defined as the combination of multiple kernels using properties introduced in Theorem 1. For example, two simple composite kernels are $k_{\text{com}} = k_1 + \alpha k_2$ and $k_{\text{com}} = k_1^\alpha k_2$. Given that $k_1$ and $k_2$ are Mercer kernels, based on properties in Theorem 1, the composite kernel $k_{\text{com}}$ is also a Mercer kernel as well. In other words, it can always find some transformation $\varphi_{\text{com}}$ such that $k_{\text{com}}(x, y) = \langle \varphi_{\text{com}}(x), \varphi_{\text{com}}(y) \rangle = \varphi(x, y) \in \Xi$.

For MKFCM, updates $u_{wp}$ according to equation (2.7) and equation (2.8) or equation (2.12) and equation (2.13). The difference is that the kernel function $k$ in these equations is replaced by the combined kernel $k_{\text{com}}$. Similar to KFCM, if MKFCM assumes the prototypes in the kernel space [using objective function equation (2.14)], this type of MKFCM is referred as MKFCM-K; if MKFCM confines that the prototypes are mapped from feature space or data space [using objective function equation (2.15)], such a type of MKFCM is referred as MKFCM-F.

When the number of parameters in the combined kernel is small, the parameters can be adjusted by trial and error. For instance, the parameter $\alpha$ in the $k_{\text{com}} = k_1 + \alpha k_2$ can be selected by testing a group $\alpha$ 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, in machine learning community, a widely used composite kernel is the linear combination of several kernels, i.e., $k_{\text{com}} = w_1k_1 + w_2k_2 + \ldots + w_kk_1$. Some learning algorithms that adjust the weights $w_i$ automatically in typical kernel learning methods like multiple-kernel regressions and classifications have been studied. Here, we propose a similar algorithm for MKFCM using linearly combined kernels.
To increase the number of selections for kernel functions, a linearly combined kernel function is applied in MKFCM. The new composite kernel $K_L$ is defined as

$$K_L = w_1 \varphi K_1 + w_2 \varphi K_2 + \ldots + w_N \varphi K_N$$

(2.16)

where $b > 1$ is a coefficient similar to the fuzzy coefficient in equation (2.4) and equation (2.6). The regulation on weights, $w_1, w_2, \ldots, w_N = 1$.

The objective function of the MKFCM with the linearly combined kernel is still the weighted sum of distances between the data and prototypes in the kernel space as given in equation 2.17.

$$Q = \sum_{j=1}^{N} \sum_{i=1}^{P} U_{ij} \| \varphi(x_j) - (O_i) \|^2$$

(2.17)

where $\varphi_1$ is the transformation derived from the linearly combined kernel $k_1(x_i, x_j) = <\varphi_1(x), \varphi_1(y)>$[$k_1$ is defined in equation (2.16)]. Just the same as in the KFCM [equation (2.7) and equation (2.8)], the learning rule for membership values is

$$U_{ij} = \frac{1}{\sum_{l=1}^{N} \alpha_{l} \varphi(x_j) / \varphi(x_i)}$$

(2.18)

$$d \varphi_1^F = k_1(x_j, x_j) - \frac{\sum_{l=1}^{N} \alpha_{l} \varphi(x_l) k_1(x_j, x_l)}{\sum_{l=1}^{N} \alpha_{l} \varphi(x_l)} + \frac{\sum_{l=1}^{N} \alpha_{l} \varphi(x_l) k_1(x_j, x_l)}{\sum_{l=1}^{N} \alpha_{l} \varphi(x_l)}$$

(2.19)

Introducing the Lagrange term of the constraint on weights $w_i (i = 1, \ldots, N)$ into the objective function, we have
\[ Q = \sum_{i=1}^{N} \sum_{j=1}^{n} u_{ijm} \| \varphi_{h}(x_j) - (\mathbf{o}_i) \|^2 + \eta (1 - \sum_{i=1}^{I} w_i) \]

(2.20)

Taking the derivative of \( Q \) over \( w_i \), and setting the results to zero, we obtain the updating rule of the weights \( w_i (i = 1 \ldots I) \)

\[ \frac{\partial Q}{\partial w_i} = \mathbf{0} (i = 1, \ldots, I) \Rightarrow \frac{1}{\sum_{k=1}^{n} q_{k}^{(i)}} \]  

(2.21)

Where,

\[ q_{h} = \sum_{i=1}^{I} \sum_{j=1}^{n} u_{ijm} \| \varphi_{h}(x_j) - (\mathbf{o}_i) \|^2 \quad (h=1 \ldots I) \]  

(2.22)

Here \( \varphi_{h} \) is the transform function defined by \( k_{h} (h=1 \ldots I) \) in equation (2.16) and it is derived as

\[ \| \varphi_{h}(x_j) - (\mathbf{o}_i) \|^2 = k_{h}(x_j, x_j) - \frac{2 \sum_{h=1}^{H} q_{h}^{(i)} k_{h}(x_j, x_j) + \sum_{h=1}^{H} q_{h}^{(i)} u_{ijm} (k_{h}(x_j, x_j))}{(\sum_{h=1}^{H} q_{h}^{(i)})^2} \]

(2.23)

The algorithm introduced here is named as LMKFCM (standing for linear combined MKFCM). LMKFCM can linearly combine more than two kernels and automatically adjust the weights of each kernel in the optimization procedure. By studying the objective function applied in this LMKFCM, we know that the prototypes are directly defined in the kernel space; therefore, this LMKFCM is indeed LMKFCM-K (with K standing for the kernel space). Here, we shorten LMKFCM-K as LMKFCM. In feature space, it is very difficult to derive a learning algorithm for feature-space LMKFCM (or LMKFCMF in short) because the linear combination of basic kernels is not a Gaussian kernel.
2.8 MKFCM-BASED IMAGE SEGMENTATION

The application of multiple or composite kernels in the FKCM has its advantages. In addition to the flexibility in selecting kernel functions, it also offers a new approach to combine different information from multiple heterogeneous or homogeneous sources in the kernel space. Specifically, in image-segmentation problems, the input data involve properties of image pixels sometimes derived from very different sources.

The intensity of a pixel is directly gained from the image itself, but the texture information of a pixel might be obtained from some wavelet filtering of the image. Therefore, we can define different kernel functions purposely for the intensity information and the texture information separately, and we then combine these kernel functions and apply the composite kernel in MKFCM (including LMKFCM) to obtain better image-segmentation results. Examples that are more visible could be found from multitemporal remote sensing images. The pixel information in these images inherit 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.

In this section, we first study some successful enhanced KFCM-based image-segmentation algorithms that consider both the pixel intensity and the local spatial information. These algorithms are proved actually as the special cases of MKFCM based methods, which mingle a kernel for the spectral information and a kernel for the local spatial information. After that, several new variants of MKFCM-based image-segmentation algorithms are developed. These new variants demonstrate the flexibility of MKFCM in kernel selections and combinations for image-segmentation problems and offer the potentials of improvement in segmentation results.
For a data point $\mathbf{x} = [x_1, x_2, \ldots, x_n] \in \mathbb{R}^n$, we also define it as $\mathbf{x} = [\mathbf{x}_p, \mathbf{x}_q]$, where $\mathbf{x}_p \in \mathbb{R}^p$ contains $p$ dimensions of the data point $\mathbf{x}$ and $\mathbf{x}_q \in \mathbb{R}^q$ contains the remaining $q$ dimensions. If $k_p: \mathbb{R}^p \times \mathbb{R}^p \rightarrow \mathbb{R}$ is a kernel over $\mathbb{R}^p \times \mathbb{R}^p$, then the function $k: \mathbb{R}^{p+q} \times \mathbb{R}^{p+q} \rightarrow \mathbb{R}$, such that $k(\mathbf{x}, \mathbf{y}) = k_p(\mathbf{x}_p, \mathbf{y}_p)$, is also a kernel over $\mathbb{R}^{p+q} \times \mathbb{R}^{p+q}$. Indeed, setting $k_\lambda = k_p$ and $\psi: \mathbb{R}^{p+q} \rightarrow \mathbb{R}^p$ such that $\psi(\mathbf{x}) = \mathbf{x}_p$.

Because the Gaussian kernel $k(\mathbf{x}_p, \mathbf{y}_p) = \exp(-||\mathbf{x}_p - \mathbf{y}_p||^2/\sigma^2)$ and the polynomial kernel $k(\mathbf{x}_p, \mathbf{y}_p) = (\mathbf{x}_p \cdot \mathbf{y}_p + d)^2$ are typical kernels defined on $\mathbb{R}^p \times \mathbb{R}^p$, based on Proposition 1, we know the Gaussian function defined as $k: \mathbb{R}^{p+q} \times \mathbb{R}^{p+q} \rightarrow \mathbb{R}$, such that $k(\mathbf{x}, \mathbf{y}) = \exp(-||\mathbf{x}_p - \mathbf{y}_p||^2/\sigma^2)$ and the polynomial function $k(\mathbf{x}, \mathbf{y}) = (\mathbf{x}_p \cdot \mathbf{y}_p + d)^2$ are both kernel functions over $\mathbb{R}^{p+q} \times \mathbb{R}^{p+q}$. Without loss of generality, these two functions are called as the Gaussian kernel and the polynomial kernel as well.

### 2.8.1 Two Enhanced KFCM Algorithms

In order to combine the local spatial information of pixels into the classical clustering-based image-segmentation algorithms, select input data $\mathbf{x}_j$ ($j = 1, 2, \ldots, n$) as $\mathbf{x}_j = [x_j, x_j] \in \mathbb{R}^2$ and directly apply the KFCM-K on these input data. Here, $x_j$ is the intensity of pixel $j$ and $x_j$ is the filtered intensity of pixel $j$, which represents the local spatial information. $x_j$ is the mean or the median filtered intensity defined in a $3 \times 3$ window centered at pixel $j$. We denote this algorithm as DKFCM (D stands for direct application of KFCM). Specifically, $\text{DKFCM}_{\text{mean}}$ is used to denote the DKFCM applying the mean filtered intensities as the spatial information and $\text{DKFCM}_{\text{median}}$ is used for DKFCM with the median filtered intensities. In DKFCM, the kernel function is the Gaussian kernel and the applied learning rules are the same as equations (2.7) and (2.8).

We now prove that DKFCM is a special case of MKFCM,
Case 1: DKFCM is a special case of MKFCM-K with $k_{com} = k_1k_2$ applied on input data $x_j = [x_{j,x},ar{x}_j] \in \mathbb{R}^2$ ($j = 1, 2 \ldots n$). Here, $k_1$ is the Gaussian kernel for pixel intensity $k_1(x_i, x_j) = \exp(-|x_i - x_j|^2/r^2)$ and $k_2$ is another Gaussian kernel for local spatial information $k_2(x_i, x_j) = \exp(-|x_i - x_j|^2/r^2)$, in which $x_j$ is the intensity of pixel $j$ and $\bar{x}_j$ is the local spatial information represented by the filtered intensity of pixel $j$.

The kernel function $k(x_i, x_j)$ is used in DKFCM, where $x_j = [x_{j,x},\bar{x}_j]$ and $k$ is the Gaussian kernel. By its definition,

$$k(x_i, x_j) = \exp\left(\frac{||x_i - x_j||}{r^2}\right)$$

$$k(x_i, x_j) = k_1(x_i, x_j)k_2(x_i, x_j)$$

$$k(x_i, x_j) = k_{com}(x_i, x_j)$$  \hspace{1cm} (2.24)

That is, the DKFCM uses the same kernel function as MKFCM-K. Considering that both DKFCM and MKFCM-K, we know that DKFCM is a special case of MKFCM-K that uses the composite kernel $k_{com} = k_1k_2$. To enhance the Gaussian kernel based KFCM-F by adding a local information term in the objective function, i.e., the new objective function becomes

$$Q = \sum_{i=1}^{n} \sum_{j=1}^{n} u_{ij} Q_i \left(1 - k(x_i, Q_j)\right) + \alpha \sum_{i=1}^{n} \sum_{j=1}^{n} u_{ij} \left(1 - k(x_i, Q_j)\right)$$

(2.25)

where $x_j$ is the intensity of pixel $j$. In the new objective function, the additional term is the weighted sum of differences between the filtered intensity $\bar{x}_j$ (the local spatial information) and the clustering prototypes. The differences are also measured using the kernel induced distances. Such kind
of enhanced KFCM based algorithm is denoted as AKFCM (with a standing for additional term). Like DKFCM meanf and DKFCM medianf, we use AKFCM meanf to represent the AKFCM applying the mean filtered intensities as the local spatial information, and AKFCM medianf denotes the AKFCM using the median filtered intensities. In AKFCM, the kernel function is the Gaussian kernel. Next, we prove that AKFCM is a special case of MKFCM.

Case 2: AKFCM is a special case of MKFCM-F with \( k_{\text{com}} = k_1 + \alpha k_2 \) on the input data \( x_j = [x_j, \tilde{x}_j] \) \( (j = 1, 2, \ldots, n) \), where \( k_1, k_2, x_j \), and \( \tilde{x}_j \) are the same as the ones defined in Case 1. In AKFCM, the goal is to minimize the following objective function:

\[
Q_1 = \sum_{i=1}^{C} \sum_{j=1}^{N} U_{ij}^w (1 - k(x_j, O_i)) + \alpha \sum_{i=1}^{C} \sum_{j=1}^{N} U_{ij}^w (1 - k(\tilde{x}_j, O_i))
\]

\[
= \sum_{i=1}^{C} \sum_{j=1}^{N} U_{ij}^w (1 + \alpha - k(x_j, O_i)) + \alpha k(\tilde{x}_j, O_i))
\]

\[
= \sum_{i=1}^{C} \sum_{j=1}^{N} U_{ij}^w \left(1 + \alpha - \exp\left(-\frac{||x_j - \hat{O}_i||^2}{\sigma^2}\right) + \alpha \exp\left(-\frac{||\tilde{x}_j - \hat{O}_i||^2}{\sigma^2}\right)\right)
\]

(2.26)

On the other hand, if \( k_{\text{com}} = k_1 + \alpha k_2 \) is the composite kernel for MKFCM-F, the objective function of the MKFCM is given by

\[
Q = \sum_{i=1}^{C} \sum_{j=1}^{N} U_{ij}^w \left\|\phi_{\text{com}}(x_j) - \phi_{\text{com}}(O_i)\right\|^2
\]
\[ Q = \sum_{j=1}^{C} U_{ij} \left( k_{\text{com}}(x_j, x_j) + k_{\text{com}}(o_i, o_i) - 2k_{\text{com}}(x_j, o_i) \right) \]

\[ Q = 2 \sum_{i=1}^{n} \sum_{j=1}^{n} U_{ij} \left( 1 + \alpha - (k_1(x_j, o_i) + \alpha k_2(x_j, o_i)) \right) \]

By the definitions of \( k_1 \) and \( k_2 \)

\[ Q = 2Q_1 \quad \text{(2.27)} \]

Comparing equation (2.26) to equation (2.27), we know that AKFCM is actually the MKFCM-F using \( k_{\text{com}} = k_1 + \alpha k_2 \) (minimizations of \( Q_1 \) and \( 2Q_1 \) are the same problem). In other words, Adaptive kernel fuzzy c-means algorithm is a special case of multiple kernel fuzzy c-means with the fuzzy function KFCM-F with the function \( k_{\text{com}} \).

2.8.2 Variants of MKFCM

As shown in the previous section, AKFCM is a special case of MKFCM-F with \( k_{\text{com}} = k_1 + \alpha k_2 \) (\( k_1 \) is the kernel for intensity; and \( k_2 \) is the kernel for local spatial information). Therefore, similar to MKFCM-F, AKFCM confines the prototypes as the points mapped from the original data space or the feature space. This limits the search space of the prototypes. The natural choice to fix this shortcoming is applying MKFCM-K, which searches prototypes in the total kernel space, directly to image-segmentation problems. Indeed, we have demonstrated that DKFCM is an MKFCM-K with a composite kernel \( k_{\text{com}} = k_1 k_2 \). More variants of MKFCM-K- based image-segmentation algorithms can be proposed. For instance, we propose that the first variant of MKFCM-K that uses the composite kernel on the input data is

\[ k_{\text{com}} = k_1 + \alpha k_2 \quad \text{(2.28)} \]
\( x_j = [x_j, \overline{x}_j] \in \mathbb{R}^2 \) (\( j = 1, 2, \ldots, n \)), in which \( x_j \in \mathbb{R} \) is the intensity of pixel \( j \) and \( x_j \in \mathbb{R} \) is the filtered intensity of pixel \( j \) that stands for the local spatial information. Identical to DKFCM, in the composite kernel, \( k_1 \) is the Gaussian kernel for pixel intensities, i.e., \( k_1(x_i, x_j) = \exp(-|x_i - x_j|^2 \theta^2) \), and \( k_2 \) is the Gaussian kernel for the local spatial information, i.e., \( k_2(x_i, x_j) = \exp(-|x_i - \overline{x}_j|^2 \theta^2) \). As a variant of the general MKFCM-K introduced in Section II, this algorithm still updates \( u_i \) following the rules (4) and (5), in which the kernel function \( k \) is replaced by \( k_{\text{com}} \). It is worth pointing out that \( k_1 \) or \( k_2 \) in the first variant of MKFCM-K-based image segmentation can be changed to any other Mercer kernel function for the information related to image pixels. This empowers the flexibility to the segmentation algorithm in kernel function selections and combinations.

For example, a composite kernel that joins different shaped kernels can be defined as

\[
k_{\text{com}} = k_1 + \alpha k_2
\]  

(2.29)

where \( k_1 \) is still the Gaussian kernel for pixel intensities

\[
k_1(x_i, x_j) = \exp(-|x_i - x_j|^2 \theta^2)
\]

(2.30)

\( k_1(x_i, x_j) = \exp(-|x_i - x_j|^2 \theta^2) \), but \( k_2 \) is a polynomial kernel for the spatial information

\[
k_2(x_i, x_j) = (x_i \overline{x}_j + d)^2
\]

(2.31)

where \( \overline{x}_j \) is the filtered intensity of pixel \( j \). We denote this MKFCM-K-based algorithm as the second variant of MKFCM-K.
Apart from the flexibility of selecting different shaped kernel functions for the intensity and the spatial information, MKFCM-K allows us to apply kernel functions for other information derived from the image. Take the texture information as the example, we can set the input data $x_j$ as $x_j = [x_j, \overline{x}_j, s/j] \in R^3$, in which $x_j \in R$ is the intensity of pixel $j$. The two-tuple $[\overline{x}_j, s/j] \in R^2$ is a simple descriptor of the texture information at pixel $x_j$, where $\overline{x}_j$ is the filtered intensity of pixel $j$ and $s_j$ is the standard variance of the intensities of the pixels in the neighborhood of pixel $j$. Then, we define the combined kernel as

$$k_{comb} = k_1 + \alpha k_2$$

(2.32)

where $k_1$ is the Gaussian kernel for pixel intensities and $k_2$ is the Gaussian kernel for the texture information $k_2(x_i, x_j) = \exp(-[\overline{x}_i - \overline{x}_j]^2 / r^2)$. Just like what we did in AKFCM and DKFCM, the notations “meanf” and “medianf” can be attached to the names of algorithms to refer different applied spatial information. Therefore, MKFCM-Kmeanf is designated to variants of MKFCM-K using the mean filtered intensities as the local spatial information, and the MKFCM-K medianf is used when the median filtered intensities are selected as the local spatial information in MKFCM-K.

To increase the information diversity of an image, LMKFCM can be applied in image-segmentation problems as well. Specifically, we define different kernels for different image information, linearly ensemble them into a new kernel, and then apply equation (2.18), equation (2.19) and equation (2.21), equation (2.22) to update the membership values and weighting coefficients.

For example, the input image data $x_j$ is set to be $x_j = [x_j, \overline{x}_j, s_j] \in R^3$, the same as the third variant of MKFCM-K. Then, the composite kernel is designed as
\[ \kappa_L = w_1k_1 + w_2k_2 + w_3k_3 \]  

(2.33a)

where \( k_1 \) is the Gaussian kernel for pixel intensities

\[ k_1(x_i, x_j) = \exp(-|x_i - x_j|^2 / r^2) \]  

(2.33b)

\( k_2 \) is the Gaussian kernel for spatial information

\[ k_2(x_i, x_j) = \exp(-|\vec{\nabla}_i - \vec{\nabla}_j|^2 / r^2) \]  

(2.33c)

and \( k_3 \) is the Gaussian kernel for texture information

\[ k_3(x_i, x_j) = \exp(-||\vec{\alpha}_i - \vec{\alpha}_j||^2 / r^2) \]  

(2.33d)

In the next section will apply this composite kernel for LMKFCM in simulations. For simplicity, we also use LMKFCM to refer the specific segmentation algorithm utilizing the composite kernel defined in equation (2.33).

The MKFCM algorithm evaluates the centroids so as to minimize the influence of outliers. Unlike FCM, it does not attempt fuzzification for elements having membership values above the calculated threshold. This reduces the computational burden compared to FCM; also there is an absence of external user-defined parameters. The removal of this initial trial and error factor makes MKFCM more robust, as well as insensitive to the fluctuations in the incoming data.
Figure 2.4  a) Synthetic texture image  b) sample satellite image c) low resolution satellite image1 d) low resolution satellite image2

To further improve the performance of segmentation, MKFCM that linearly combines three kernels, i.e., the first two kernels are the kernels for intensities and another one for the local spatial information. To sum up, the merit of MKFCM-based image-segmentation algorithms is the flexibility in selections and combinations of the kernel functions in different shapes and for different pieces of information. After combining the different kernels in the kernel space (building the composite kernel), there is no need to change the computation procedures of MKFCM. This is another advantage to reflect and fuse the image information from multiple heterogeneous or homogeneous sources.

MKFCM-based image-segmentation algorithms are inherently better than other KFCM-based image segmentation methods. We can demonstrate the MKFCM significant flexibility in kernel selections and
combinations and the great potential of this flexibility could bring to image segmentation problems. In the MKFCM framework, we can easily fuse the texture information into segmentation algorithm by just adding a kernel designed for the texture information in the composite kernel. As in the satellite image-segmentation and two-texture image-segmentation problems, simply add a Gaussian kernel function of the texture descriptor in the composite kernel of MKFCM leads to better segmentation results.

2.9 RESULTS AND DISCUSSION

In this section, we present the texture based segmentation and MKFCM-based segmentation for synthetic images (texture images) as shown in the Figure 2.4. Since the proposed MKFCM-based image segmentation algorithm provides better segmentation results, it is applied for various satellite images. The performance of FCM-type algorithms depends on the initialization. The initialization and iterations depend upon the input images and choose the one with the best objective function value. This increases the reliability of comparison results acquired in the simulations. In MKFCM, the different kernels, balances the importance of pixel intensities and the local spatial information. The main goals of an image segmentation algorithm are optimization of segmentation accuracy and its efficiency. Considering accuracy, the proposed method is concentrated on obtaining a robust segmentation for noisy images) and a correct detection of small regions.

To demonstrate the flexibility and the advantages of MKFCM, a two-texture image is tested in this simulation. Figure 2.5 shows the segmentation performance of texture segmentation algorithm for synthetic image. The segmented image is as shown in Figure 2.5, in which the left half of the image and the right half are of great difference because the left half is coarse and the right half is smooth, i.e., their textures are visibly different.
Traditional enhanced KFCM-based algorithms like DKFCM and AKFCM cannot deal with this kind of image very well, because they only consider the local spatial information. While considering the problem in the MKFCM framework, we can simply apply the combined kernel like the Gaussian kernel for the texture information.

![Figure 2.5 Texture based segmentation in synthetic image](image)

In the Figure 2.5, we observed that the texture segmentation yields the segmented image with overlapping the adjacent pattern. In addition, it requires more number of image processing stages like histogram equalization, image interpolation and edge detection based rough segmentation. Figure 2.6 shows the segmentation result of the MKFCM for the same texture image. Due to the consideration of the texture information, it has a better result for texture segmentation than other FCM based methods.
In order to evaluate the performance of the proposed method in detecting small regions, the algorithm was applied on a satellite image by considering different number of clusters. In these images, the clusters appear with widely varying sizes, and thus, one can evaluate the ability of an algorithm in detecting small regions. As can be seen from these results, the performance of MKFCM and KFCM algorithm provides useful detailed information before defuzzification that can be utilized in algorithms such as the proposed method. Also note that the segmentation result of the proposed method is the best with respect to robustness against noise and preserving the details of the object borders.

Figure 2.6 MKFCM based segmentation in synthetic image

Although the proposed method utilizes some processing units that are not used in the KFCM method, its overall elapsed time for the multispectral data sets is less than that of the KFCM method. It is mainly obtained by reduction of spatial redundancy in the image tessellation step of
the watershed transform block. Figure 2.7 and Figure 2.8 shows the simulation results obtained for a sample satellite image and low resolution satellite image.

![Figure 2.7 MKFCM based segmentation in sample satellite image](image)

The Figure 2.7 requires less number of iterations and segmentation has been achieved for less threshold level but the low resolution image in Figure 2.8 requires more number of iterations and hence requires high threshold level compared to the sample satellite image in Figure 2.7.
Figure 2.8 MKFCM based segmentation in low resolution satellite image

Figure 2.9 KFCM based segmentation in low resolution satellite image
The MKFCM algorithm performs well for normal images and low resolution images. The performance of MKFCM algorithm is compared with the KFCM and FCM algorithm as shown in the Figure 2.9, 2.10 and 2.11. The number of iterations and threshold level as shown below Tables 2.1 and 2.2 respectively.

Figure 2.10 FCM based segmentation in low resolution satellite image

Now apply the different fuzzy clustering methods like fuzzy c-means, kernel fuzzy c-means and multiple kernel fuzzy c-means algorithm to different images and derived the threshold values between the texture image, sample satellite image and low resolution satellite image. Similarly calculate the number of iterations to segment the different images as shown in the Table 2.1 and Table 2.2. Finally we conclude that our proposed method had very less number of iterations and threshold level value for the texture, satellite image and low resolution satellite images.
Figure 2.11 FCM based segmentation in sample satellite image

Table 2.1. Threshold level Comparison

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>FCM</th>
<th>KFCM</th>
<th>MKFCM (Proposed method)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Texture image</td>
<td>0.974</td>
<td>0.854</td>
<td>0.758</td>
</tr>
<tr>
<td>Sample satellite image</td>
<td>0.616</td>
<td>0.594</td>
<td>0.498</td>
</tr>
<tr>
<td>Low resolution satellite image</td>
<td>0.634</td>
<td>0.645</td>
<td>0.567</td>
</tr>
</tbody>
</table>

Table 2.2 Comparison using number of iterations

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>FCM</th>
<th>KFCM</th>
<th>MKFCM (Proposed Method)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Texture image</td>
<td>180</td>
<td>116</td>
<td>95</td>
</tr>
<tr>
<td>Sample satellite image</td>
<td>102</td>
<td>75</td>
<td>43</td>
</tr>
<tr>
<td>Low resolution satellite image</td>
<td>98</td>
<td>78</td>
<td>73</td>
</tr>
</tbody>
</table>
2.10 CONCLUSION

Satellite images often require segmentation in the presence of uncertainty, caused due to factors like environmental conditions, poor resolution and poor illumination. The multiple-kernel fuzzy C-means clustering (MKFCM) methodology for satellite image segmentation has been presented in this work. The proposed MKFCM algorithm provides us a new flexible methodology for image segmentation problems. That is, different pixel information represented by different kernels are combined in the kernel space to produce a new kernel. It is shown that this algorithm performs better than the kernel fuzzy C-means and fuzzy C-means clustering algorithm. Simulations are performed on the synthetic texture and satellite images to demonstrate the flexibility and advantages of MKFCM-based approaches.