NON-LINEAR SEPARATION OF CLASSES USING KERNEL BASED POSSIBILISTIC c-MEANS

NITIN KANDPAL

MARCH, 2016

NON-LINEAR SEPARATION OF CLASSES USING KERNEL BASED POSSIBILISTIC c-MEANS

NITIN KANDPAL
Dehradun, India, March, 2016

Thesis submitted to the Faculty of Geo-Information Science and Earth Observation of the University of Twente in partial fulfilment of the requirements for the degree of Master of Science in Geo-Information Science and Earth Observation.

Specialization: GeoInformatics

SUPERVISORS:

IIRS supervisor	:	Dr. Anil Kumar
ITC supervisor	:	Dr. V.A. Tolpekin

THESIS ASSESSMENT BOARD:

CHAIRPERSON (ITC)	:	Dr. A.A. Voinov
External Examiner	:	Dr. S.K. Ghosh

OBSERVORS:

ITC Observers	:	Dr. N.A.S. Hamm
IIRS Observers	:	Dr. S.K. Srivastava
		Dr. Sameer Saran

DISCLAIMER

This document describes work undertaken as part of a programme of study at the Faculty of Geo-Information Science and Earth Observation of the University of Twente. All views and opinions expressed therein remain the sole responsibility of the author, and do not necessarily represent those of the Faculty.

"Education is not the piling on of learning, information, data, facts, skills, or abilities – that's training or instruction – but is rather making visible what is hidden as a seed." **–Thomas More**

ABSTRACT

Remote sensing data have been used for recognition and classification of land use and land cover (LULC) features on Earth surface. The mixed pixel problem and non-linearity present in the image can be handled through soft classifiers. The classification was performed with Landsat-8 data, Formosat-2 data and their simulated images. The widely used FCM classifier due to its membership constraint faces limitation as it is not able to handle untrained classes and the membership value does not represent the true concept of typicality. The Possibilistic c-means classification was chosen to overcome this membership constraint with its possibilistic membership values which is a measure of belongingness and shows high resistance to untrained classes. Various measures for accuracy assessment like Pearson correlation coefficient, RMSE, FERM and entropy were used for parameter optimization and accuracy assessment. The linear-PCM classifier was not able to handle the mixed pixel problem and non-linearity in the data adequately and thus, in order to handle the mixed pixel problem and non-linearity, the kernel functions were incorporated with PCM classifier. Nine different kernel functions were incorporated with PCM classifier and the fuzzy parameter was optimized for them. The hyper tangent kernel was identified as the best performing kernel function as it showed highest overall accuracy of 98.37% and low entropy value of 0.48 as compared to linear PCM classifier, which showed low overall accuracy of 78.38% and high entropy of 0.5430. The better classification with KPCM classifier for mixed pixel was achieved with the classification of simulated image. To add the best outcome from different kernels the composite kernel was formed by fusing the best performing hyper tangent kernel and sigmoid kernel using weighted summation approach and the value of weight constant was also optimised for composite kernel. The accuracy assessment results for composite kernel were similar to the best performing hyper tangent kernel. An improved average user's accuracy of 89.90% was obtained with composite kernel, whereas the average user's accuracy with KPCM classifier was 89.17%. Hyper tangent KPCM classification was unaffected in presence of untrained classes as compared to PCM classification by showing very negligible effect in correlation values. The results revealed that the hyper tangent KPCM was consistently performing better with Landsat-8 data as well as with Formosat-2 data, in presence of non-linearity as well as in absence of non-linearity.

Keywords

Image Classification, Clustering, hard classification, pure pixel, mixed pixels, Kernels

ACKNOWLEDGEMENTS

Firstly, I would like to thank my parents for helping and motivating me to pursue my goals. I would also like to thank my family members for encouraging and holding their trust on me.

I would like to extend my heartfelt gratitude to Dr. Anil Kumar for supervising me and giving his valuable inputs. He has always been very patient in listening to my anxious queries and has guided me to find solutions for them. His support was incredible and his dedication to work has always been a motivation.

I am also very grateful to Dr. Valentyn Tolpekin for providing and sharing his in depth knowledge and expertise. I am privileged to have him as my supervisor from ITC, he made this work more interesting and helped me in broadening my view on many topics. It was a great learning experience with him.

I would like to thank Dr. S.K. Srivastava, Dr. Nicholas Hamm and Dr. Sameer Saran for their excellent support provided as course supervisors. They have been very kind and eager to sort out the problems faced during the whole academic work.

I am also thankful to Dr. Krikamol from Max Planck Institute of Intelligent Systems for clearing my queries on machine learning and implementation aspects of kernels. His timely replies were an add-on in understanding the basics of kernel.

I am highly obliged to my past organization-RMSI Pvt. Ltd., from where I got interest on GIS stuffs. I want to thank my colleagues Sagan, Vipin, Kranti, Anjul and Amit for supporting my decision for continuing my studies.

Last but not the least I would like to thank all my friends, batch mates and special friends at IIRS who have always been adoring. They made stay at IIRS a fun. Thanks to Abhishek, Varsha, Richa and Fuwad for always being supportive and standing with me during tough times. Special thanks to Varsha for helping in reviewing the thesis and providing critical inputs.

This acknowledgment is incomplete without acknowledging the support staff at IIRS who have always been helpful and friendly. I would like to thank specially Ms. Sangeeta for taking care of all arrangements during the meetings and for timely informing the plans.

-Nitin Kandpal

TABLE OF CONTENTS

List of Tables

List of Figures

Abbreviations

FCM: Fuzzy c-Means

PCM: Possibilistic c-Means

KPCM: Kernel based Possibilistic c-Means

KFCM: Kernel based Fuzzy c-Means

IPCM: Improved Possibilistic c-Means

MPCM: Modified Possibilistic c-Means

FPCM: Fuzzy Possibilistic c-Means

IPCM: Improved Possibilistic c-Means

EPCM: Enhanced Possibilistic c-Means

RMSE: Root mean square error

FERM: Fuzzy error matrix

SCM: Sub-pixel uncertainty confusion matrix

KMOD: Kernel with moderate decreasing

RBF: Radial basis function

IMQ: Inverse multi Quadratic

FPCM: Fuzzy Possibilistic c-Means

CHAPTER 1

1. INTRODUCTION

1.1. Research Background

Remote sensing data have been used for recognition and classification of land use and land cover (LULC) features on Earth surface. Several type of sensors are used in remote sensing (like Radiometer, spectrometer, LIDAR, RADAR and various other types of sensors) to provide information in the form of digital images. The classification of digital images leads to the development of thematic maps which can be used for agriculture resource management, disaster management, urban planning, water resource management and in many more applications. The process of classifying each pixel within the image into categories or classes is known as *Image Classification* and the algorithms that perform classification are known as *Classifiers*. Conventionally, image classification is defined under two major categories – supervised (classification) and unsupervised (clustering) image classifications (Lillesand and Kiefer, 1979). Supervised classification algorithms are provided with sample data points and with labels for all sampled points to identify the classes or categories they belong. The sampled data are used to form decision rules (i.e. parameters for the classification algorithm) to predict the class of un-sampled data points. The unsupervised algorithms are not provided with sampled data, they group the data based on different similarity measure into homogenous groups known as clusters and the process of forming cluster is known as *clustering*.

The conventional classification techniques typically classify maps into hard, discrete categories or classes (e.g., urban, forest). This technique is known as *hard classification*, wherein each pixel has 100% belongingness to only one specific class and is known as *pure pixel*. It has been observed that in the real world rate of heterogeneity in the land cover is higher than the sampling done by the image pixels which results in the presence of pixels covering more than one land cover classes. These pixels are known as *mixed pixels*. In remotely sensed images, the digital number of these mixed pixels is cumulative sum of the different land cover classes that it covers on the ground.

The classification of a mixed pixel by hard classification technique will lead to classification of pixel to one particular class (generally to the class having higher proportion in the pixel) and in doing so the essential information about other classes present in mixed pixel is lost. As mixed pixels are dependent on image resolution, and medium and coarser resolution images have a higher proportion of mixed pixels, so hard classification of them is less favoured (Foody, 2000). As the mixed pixels cover more than one land cover class, the classification technique must present output that restore the information present in these mixed pixel. Soft or fuzzy classification approaches are commonly used to handle mixed pixel problem by assigning multiple class memberships to a pixel. Artificial neural network (ANN), linear mixture model (LMM), decision tree and fuzzy logic based classifier are some of the soft classification methods.

1.2. Fuzzy Classifier

The proposed research is focused on fuzzy logic based classifiers to handle the mixed pixels. They are based on the idea of fuzzy set logic put forwarded by Zadeh (1965). They introduce degree of vagueness or fuzziness by membership function. According to it a pixel or a sample can be assigned to more than one class with the grade of membership value ranging between 0 and 1. The value nearer to 1 resembles higher membership of the sample or pixel to the class.

FCM is a popular fuzzy classifier. Fuzzy c-Means (FCM) is based on constraint, according to which the sum of memberships of classes present in a pixel must sum to 1 (Bezdek et. al., 1984). This constraint adversely affects the performance of FCM when noise or untrained classes are present in the data (Krishnapuram and Keller, 1993). The issues in FCM due to this constraint are as follows:

a) Some pixels depending on their location in the feature space (a space where input variables are defined) will have different membership value for a class though they may be located at equal distance from that class mean value(Krishnapuram and Keller, 1993). As, the value of membership of a pixel is a relative to the number of classes defined and is not an absolute membership value.

b) Some pixel lying far away from other data points (generally known as *outliers*) are given membership value of $1/n$ where n is the number of classes (Krishnapuram and Keller, 1993).

Thus, it can be seen that the membership value in case of FCM is not a representation of the degree of belongingness rather resembles the degree of sharing. Also, during membership assignment FCM cannot discriminate between highly similar representative pixels and the highly dissimilar pixels because of its constraint membership assignment.

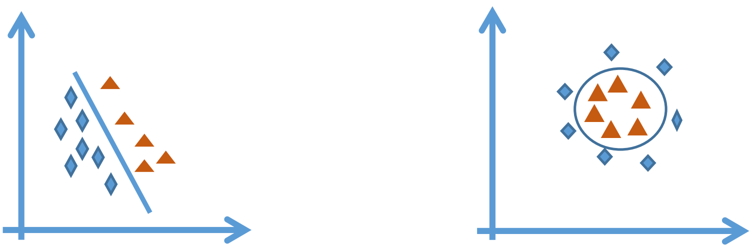
As in the case of supervised FCM classification, the membership values of each pixel are dependent on the number of classes that are defined during the training stage. So, the class membership of each pixel is divided among these defined classes. Though, it is considered that during the training stage all the classes at the site were taken into account, some of the classes are often missed unintentionally. Assigning the membership values of the pixels of the untrained, spectrally distinct classes on the basis of FCM constraint leads to the partition of full membership value of these pixels among the trained classes. As a result, these pixels are misclassified and significantly degrade the overall accuracy of the classification (Foody, 2000).

In order to overcome the limitation of FCM, Krishnapuram and Keller (1993) proposed possibilistic c-means (PCM) method which is based on possibilistic approach for clustering. Unlike as in FCM, in PCM the sum of membership of classes present in a pixel is not constrained to 1. The membership value now resembles typicality or absolute membership. Also, the membership value of a pixel to a class is not affected by the presence of other classes and thus, untrained classes present during classification have less impact on the overall accuracy (Foody, 2000). Thus, the limitations due to the FCM constraint are handled by PCM.

FCM and PCM are effective only in clustering the data by linear boundaries, and in order to extend FCM and PCM for clustering data by non-linear boundaries the kernel functions are used. A kernel function maps data from original input feature space to a higher dimensional feature space where the problem of nonlinearity can be resolved (illustrated in Figure 1.1 and Figure 1.2).

The kernel functions are classified as local kernels, global kernels and spectral kernels (Kumar et. al. , 2014). Also, a concept of composite kernels was developed which included multiple kernel functions for classification. As per the study done by Camps-Valls et. al. (2006) if an appropriate combination of kernels is chosen then a more accurate classification can be achieved by composite kernel.

Thus, in the undertaken research, mixed pixel problem and non-linearity present in the image was handled through soft classifiers. The PCM classifier was used to handle the mixed pixel problem, and kernel functions were incorporated in PCM to handle non- linearity in data. The research work aimed at developing an objective function for kernel based PCM classifier to handle non-linear class separation by selecting a suitable kernel function from nine different kernels. The chosen nine kernel functions were Gaussian kernel using Euclidean norm, radial basis kernel, kernel with the moderate decreasing (KMOD), inverse multi quadratic kernel, linear kernel, polynomial kernel, sigmoid kernel, spectral kernel, and hyper tangent kernel. Among these nine kernels, the optimal kernel function was identified based on the accuracy of classification.



(a) (b)

Figure 1.1. Two clusters in feature space. (a) Linearly separable clusters (b) Non-linearly separable clusters

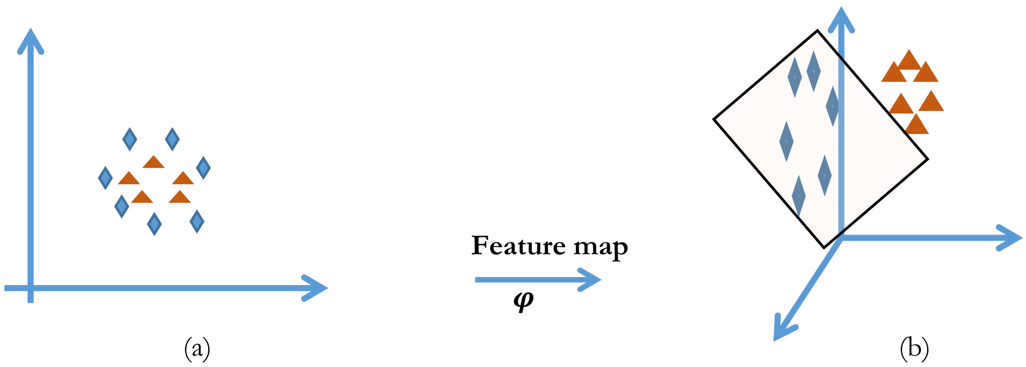


Figure 1.2. Feature space transformation using kernel function (a) Non-linearly separable clusters in input feature space (b) Linearly separated clusters in transformed kernel feature space

The results of the fuzzy classifier are the fuzzy categorical maps. The fuzzy categorical map differs from the primitive results from the hard classifier (i.e. object-based categorical map) in the sense that for object-based categorical map at any particular location "A" a single class is allowed at each location, i.e. at location "A" there is full membership for class forest and zero membership for all other classes (Figure 1.3 (a)). In contrast to this in fuzzy categorical maps the location "A" may belong to more than one class i.e. the pixel at location "A" may have non-zero membership value for more than one class (Figure 1.3 (b)).

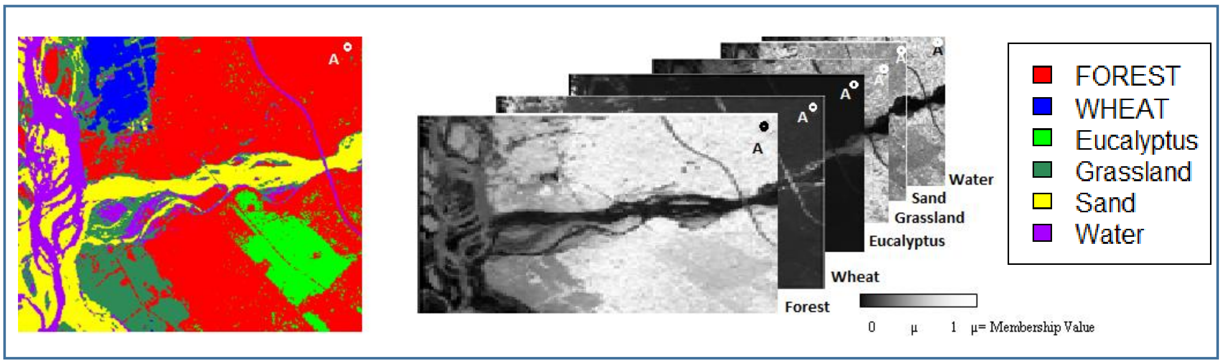


Figure 1.3. (a) object-based categorical map from hard classifiers (b) fuzzy categorical maps from soft classifier

In order to set confidence on the results from the classifier, accuracy assessment was done. As there is no universally accepted method for accuracy assessment of soft output, the result of different classifiers was compared and validated using multiple accuracy assessment techniques. To evaluate the soft classified output, various approaches have been forwarded (Foody, 1995; Binaghi et. al. , 1999;Ricotta and Avena, 2002). The image to image accuracy assessment based on fuzzy error matrix (FERM), Pearson correlation coefficient, RMSE and entropy were used to estimate the parameter value and to identify the accuracy of classification.

1.3. Problem Statement

- Hard classification of the remote sensed images that contains significant amount of mixed pixels leads to loss of information present in the mixed pixels and results in under estimation or over estimation of land cover.

- Classification of remote sensed images containing non-linearity between classes by linear classifiers (like FCM and PCM) may lead to misclassification.

- The presence of untrained classes (acting as noise in data) during classification using soft classifier like FCM significantly affects the classification accuracy.

1.4. Research Objective

The main objective of the research was to develop a method to separate the classes having non-linear boundaries using KPCM. The specific objectives were:

- To develop an objective function for kernel based PCM (KPCM) classifier.
- To derive a method for selecting parameters for optimal kernel function.
- To evaluate the performance of developed KPCM classifier in case of untrained classes.
- To study the performance of single/composite kernels with PCM classifier.
- To compare the performance of PCM with the developed KPCM classifier.

1.5. Research Questions

- How well non-linearity between classes in the input feature space will be handled by KPCM?
- How can mixed pixels be handled using KPCM?
- How well KPCM performs in case of untrained classes (considering one or more than one classes at a time)?
- How can we evaluate the performance in terms of accuracy and robustness during classification with single/composite kernel in KPCM?

1.6. Innovation aimed at

- To incorporate nine kernels with PCM classifier as PCM classifier has been studied only with Euclidean/ Mahalanobis norms and with limited kernels like Gaussian or RBF kernel.
- To study behaviour of single as well as composite kernels with PCM classifier.

1.7. Research approach

The research was started with literature review about different fuzzy classification algorithms. Specifically, c-means algorithm was chosen because of its potentiality to be extended to produce other methods of classification. The objective was to develop a fuzzy classifier that can handle non-linearity in the data, for which kernel methods were selected to be incorporated with the possibilistic c-means algorithm to handle the non-linearity.

Firstly, the presence of non-linearity was verified in the data using SVM. The non-linearity is simulated in the real data by removing some features and merging some classes. Further, a synthetic image was generated for the available Formosat-2 real image, parameters were optimized and the best kernel was selected. The parameter estimation and accuracy assessment was done using various accuracy assessment measures like FERM, SCM, Entropy, RMSE and correlation for the image to be classified. Further the strengths of kernel based PCM algorithm were identified by comparing the results of selected kernel with the primitive PCM algorithm. The general thesis approach is explained in Figure 1.4.

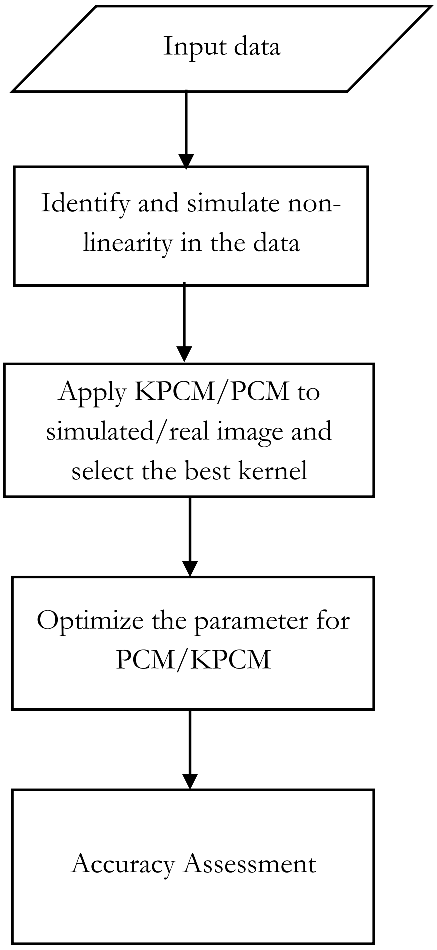


Figure 1.4. General approach of the thesis

1.8. Thesis Structure

The thesis has been organized into six different chapters. The **first chapter** gives a brief introduction about the basic and background knowledge on the aspects of this research, the objectives to be accomplished, research questions formulated from the research objectives and research approach followed. The **second chapter** describes about the previous work that was done related to the research work. The **third chapter** describes classification approach adopted to complete the objective. The **fourth chapter** elaborates the study area and methodology adopted in this research work. The **fifth chapter** includes the classification results obtained. The **sixth chapter** explains the classification results obtained. The **seventh chapter** deals with the recommendation on further work that can be taken forward from this study.

CHAPTER 2

2. LITERATURE REVIEW

This chapter summarizes the literature survey done for identifying the possible solutions for achieving the desired objective. There are existing researches in the field of machine learning with kernels where the selection of kernel is emphasised to achieve the best classification. The first section describes the existing research on PCM classifier in context with supervised PCM which has been used in this study. The second section includes the existing researches on KPCM classifiers and the significant results obtained in these researches. The third section describes the literature survey related to accuracy assessment.

2.1. Fuzzy based classifiers

The fuzzy classification is useful when the input data contains overlapping cluster, outliers or noise because the membership value assigned by soft classification may then be an appropriate measure to explain the degree to which a pixel belongs to a class(Filippone et. al., 2010). The possibilistic c-means (PCM) algorithm is based on theory of possibility forwarded by Zadeh (1978). According to theory, the possibilistic analysis doesn't provide any measure on the data but provides the appropriate meaning or information about the data. So, during classification the possibilistic membership values for a pixel defines the degree of belongingness for a class and it doesn't measure the degree of sharing among the classes (Krishnapuram and Keller, 1993). The probabilistic membership value in Fuzzy c-Means(FCM) algorithm follows the probability constraint according to which the sum of membership value for a pixel must sum to one. Here the membership value of a pixel to a particular cluster depends on the distance of the pixel from all the cluster centres (Bezdek et al., 1984). This leads to certain problems- 1) pixels located at two distinct locations but equidistant from a cluster may have different membership value for that cluster 2) the noise or outlier may have high membership value and can affect the classification accuracy. In order to overcome these problems, Krishnapuram and Keller (1993) proposed the Possibilistic c-Means (PCM) algorithm. Unlike FCM, in case of PCM, the membership value of data point for a cluster is not affected by the presence of other neighbouring clusters, thus, providing the degree of belongingness for a cluster (Krishnapuram and Keller, 1993). Later on, Krishnapuram and Keller (1996) further clarified the implementation principle for PCM and mentioned the need of good initialization and estimation of parameters for effectively functioning of PCM. It has also been observed that the PCM algorithm was not affected by the presence of untrained classes and showed no effect in classification accuracy due to the presence of noise or untrained classes in the data set (Foody, 2000).

The weighting component m determines the fuzziness in classification. This fuzzy parameter is known as *fuzzifier* or *fuzzification constant*. When fuzzy parameter tends to 1, the classification becomes hard and with m tending to infinity, the classification becomes maximally fuzzy. In PCM, with increasing the value of m, the possibility of a pixel belonging to a given class increases. The value of η_i is a non-negative constant that determines the zone of influence and shape of the cluster. FCM algorithm can be used to estimate the value for η_i and for initializing the cluster centre (Krishnapuram and Keller, 1996). Foody (2000) has given a supervised version of PCM where the cluster centres are determined from the labelled data. Kumar et. al. (2006) investigated that the supervised PCM gave best result with Euclidean distance norm and showed higher accuracy than FCM classifier.

2.2. Kernel methods

The primitive classification algorithm can be combined with kernel functions to generate a non-linear hypersurface between the clusters. Rhee et. al. (2012) proposed a kernel based possibilistic clustering technique, in which fuzzy kernel c-means (FKCM) algorithm for initialization of PCM was used and PCM was modified using kernel induced metric replacing Euclidean distance measure and showed better results than FCM, PCM, and FKCM. It has been shown that KPCM assign least membership value to the noise/outliers than PCM or FCM, thus provides more accurate results. Hu et al. (2012) showed that integrating kernel with possibilistic c-means, the classifier not only inherits the capability of PCM of handling noise/outliers but also adds the capability of detecting clusters with different shape and thus, handling non-linearity in the data. Wu (2006) and Ganesan and Rajini (2010) introduced the KPCM classifier by modifying PCM objective function by replacing Euclidean norm metric by kernel induced metric (Gaussian kernel) which is more robust to noise than PCM and FPCM. Ganesan and Rajini (2010) also observed that the approach using kernel method is much faster (in terms of time elapsed and number of iterations) than FCM. Mittal and Tripathy (2015) studied that the Gaussian kernel produced more accurate clustering than radial and hyper tangent kernel for small-sized dataset though hyper tangent kernel out performed other kernel for considerably large data set. Camps-Valls et al. (2006) and Kumar et. al. (2005) have shown that the properties of different kernel can be added up by forming composite kernels. The composite kernel can be formed by adding up the kernel functions based on the weight factor.

2.3. Accuracy assessment

The accuracy assessment is the most important part of any classification process as the classifiers are evaluated and compared on the basis of results of accuracy assessment. It also provides confidence on the results of classifiers.

The primitive confusion matrix was modified by Binaghi et al. (1999) to incorporate the vagueness in classification and several fuzzy indices like user's accuracy, producer's accuracy, and overall accuracy were defined for evaluating the accuracy of soft classification. The degree of uncertainty in classification of different classes is measured through entropy, lower the value of entropy higher is the confidence in classification of class. This technique is favoured when the classified data is soft and reference data is hard classified (Ricotta and Avena, 2002; Dehghan and Ghassemian, 2006). The cross-entropy technique is used for accuracy assessment when the classified data and reference data are soft (Foody, 1995). The correlation and RMSE value defines the correspondence of the classified output with the referenced data set. The RMSE and correlation coefficient have also been used for accuracy assessment of soft classification (Foody, 2000). Silvan-Cardenas and Wang (2008) introduced the concept of sub-pixel confusion uncertainty that looks into the uncertainty for the class distribution within the pixel. The fuzzy operators like min, prod and their composite operators were used for sub pixel accuracy assessment. But the SCM follows probabilistic constraint for membership and not suitable for pixel level accuracy assessment for Possibilistic c-Means algorithm (Upadhyay et. al., 2014).

Till date various techniques have been proposed by researchers for accuracy assessment of soft classification but none of them have been universally accepted as a standard to evaluate accuracy. So, in the current research work multiple accuracy assessment techniques were used for accuracy assessment of the soft classified output from KPCM classifier. Here RMSE, correlation, FERM and Entropy measure were used for accuracy assessment.

CHAPTER 3

3. CLASSIFICATION APPROACHES

This chapter discusses about different classification approaches that have been used to obtain the objective for developing the kernel based PCM classifier.

Section 3.1 discusses about the PCM classifier and its advantage over FCM. Section 3.2 discusses different kernel methods that are used in this study. Section 3.3 provides introduction to the kernel based possibilistic c-means (KPCM) algorithm. Here what and how of the classification approaches used in this research work have been explained. The accuracy assessment is an integral part of classification to rely on the output of the classifier, Section 3.4 deals with different accuracy assessment techniques.

3.1. Possibilistic c-Means (PCM) classifier

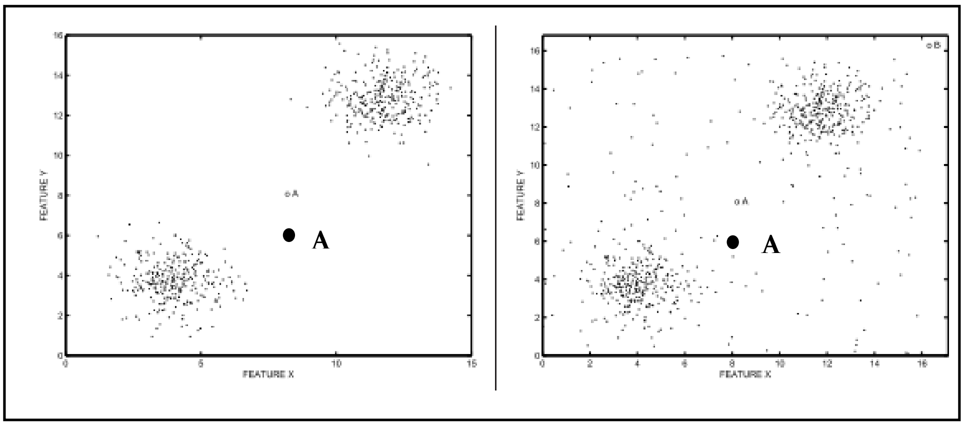
The PCM algorithm was introduced by Krishnapuram and Keller (1993) as solution for the shortcoming of FCM. The classification based on FCM algorithm were not able to handle the situations like:

Strong Outliers or Anomaly

Strong outlier is a feature vector that deviates from other observations so much that it seems to be indifferent in classification. Due to their large deviating values, they may significantly affects the classification result (Aggarwal, 2015). In the rest of this study report, strong outliers will be referred simply as an outlier.

Weak Outliers or Noise

Noise are those pixels which do not belong to any defined class. They are a kind of weak outliers, which do not strongly meet the criteria necessary for a data point to be considered as different enough (Aggarwal, 2015).



(a) (b)

Figure 3.1. Difference between noise and outlier (Aggarwal, 2015) (a) Noise and (b) Outlier

In Figure 3.1 (a), the feature vector **'A'** lies far from the class or cluster boundaries present in the feature space and is very different from the remaining feature vectors. Whereas in Figure 3.1 (b), for the feature vector **'A'** it is much harder to state that the feature vector represents a deviation from the remaining feature vectors. The feature vector **'A'** in Figure 3.1 (a) represents an outlier in the dataset whereas the feature vector **'A'** in Figure 3.1 (b) seems to fit the pattern represented by other randomly distributed point and is considered as a noise in the dataset. Remotely sensed data is effected by noise and outlier due to various reasons, some of them are - atmospheric interference (aerosols, clouds etc.) and instrument malfunction.

Untrained classes

In supervised classification, it is impossible to get sample for all classes present in the study area or some of the classes may not be considered for classification. As a result, some of the classes are left untrained during classification.

The assignment of membership value by FCM leads to partition of membership value of outlier, noise and untrained pixels among the trained classes due to its probabilistic constraint on membership value. Here, the noise, outlier or untrained class pixels usually get relatively higher membership value than PCM due to its membership constraint. As a result, these pixels will be misclassified and will significantly degrade the overall accuracy of the classification in case of the FCM (Krishnapuram and Keller, 1993). As, membership value in FCM measures the relative degree of sharing of the pixel among the classes/clusters. This membership value does not represent the real world concept of degree of belongingness or typicality.

The PCM algorithm assigns the membership value to each feature vector (pixel) based on its distance from the mean value of each cluster. Unlike FCM, the membership value is not partitioned between the classes but is assigned to a pixel as a degree of belongingness i.e. the compatibility of a pixel with each independent cluster. So, the outlier and anomalous feature vector are assigned very small value as compared to FCM.

$$\mu_{\mathrm{ik}} \in [0,1] \quad \text{for all } i \text{ and } k, \tag{3.1}$$

$$0 \leq \sum_{i=1}^{c} \mu_{ik} \leq c \tag{3.2}$$

The membership value μ_{ik} is the membership value of feature vector k in class i, the value μ_{ik} varies between 0 and 1 as mentioned in equation (3.1). Unlike FCM, in case of PCM the sum of membership value of a pixel in different classes need not be sum to 1 (Pal et.al., 1997) as shown in equation (3.2). Due to this constraint lower membership value can be assigned to noise and outliers making PCM classifier robust in the presence of untrained classes.

$$J_{PCM} = \sum_{i=1}^{c} \sum_{k=1}^{n} (\mu_{ik}{}^{m}) \, ||x_k - v_i||^2 + \sum_{i=1}^{c} \eta_i \sum_{k=1}^{n} (1 - \mu_{ik})^m \tag{3.3}$$

$$\eta_i = \frac{\sum_{k=1}^{n} \mu_{ik}^{m} d_{ik}^{2}}{\sum_{k=1}^{N} \mu_{ik}^{m}} \tag{3.4}$$

$$\mu_{ik} = \left[1 + \left(\frac{d_{ik}}{\eta_i}\right)^{1/(m-1)}\right]^{-1} \tag{3.5}$$

The equation (3.3) shows the objective function for PCM where n is the number of feature vectors (pixels); c is the total number of classes present in the site. μ_{ik} is the membership value of feature vector k in class i. From equation (3.5), it can be seen that in PCM the membership value of a feature vector k in class i is computed with respect to a single class (i) and is independent of all other classes present. Here, $\boldsymbol{d_{ik}^2}$ is the Euclidean distance of an unknown feature vector $\boldsymbol{x_k}$ from the mean vector of the class $\boldsymbol{v_i}$. For computing distance different distance norms can be used, like Euclidean distance, Manhattan distance and Mahalanobis distance. The parameter η_i is the scale parameter or bandwidth parameter, it defines the shape and size of clusters. The value of η_i needs to be known prior for each class and FCM can be used for initialization of η_i as shown in equation (3.4) (Krishnapuram and Keller, 1996). It also works as the weighting factor for the second term in the objective function for PCM. The fuzzy parameter m is used to define the fuzziness in the possibilistic c-means partition. The value of fuzzy parameter varies between [1,∞], when it approaches to 1 the classification becomes hard and with it approaches to infinity the classification becomes maximally fuzzy (Krishnapuram and Keller, 1993). The objective function can be minimized by decreasing the distance of data points from the cluster centres and by increasing the membership values.

In this study the supervised PCM algorithm is implemented. Unlike unsupervised PCM, the supervised PCM algorithm requires specification of class centroid using the sample data and a single pass of the data from the algorithm (Foody, 2000). The general form of PCM algorithm is as follows:

Supervised Possibilistic c-Means (PCM) clustering algorithm:

1- Identify the number of classes, and calculate the class centroid based on the sample data.
2- Fix the value of m, such that $1< m <\infty$.
3- Compute the distance vector for unknown feature vector from the centroid of each class (using Euclidean norm).
4- Calculate the value of η_i for each class (using equation (3.4)).
5- Compute the membership value for unknown feature vector (using equation (3.5)).

3.2. Kernel methods

The machine learning algorithms are generally divided into two basic categories: linear and non-linear, based on the type of data. The linear classification algorithms are not able to separate the non-linearly separable classes present in the data set. The kernel method adds capability to linear algorithms to separate the non-linearly separable classes.

The kernel method projects the data from the input feature space to higher dimensional feature space. Each coordinate in the input feature space corresponds to one feature. In this higher dimensional feature space, the non-linearly separable classes may appear to be linearly separable or better structured. The aim of kernel method is to identify a linearly separating hyperplane that separates the classes (Figure 1.2) in higher dimensional feature space. As depicted from Figure 1.2 (a), the data available was not linearly separable in two dimensional feature space. In Figure 1.2 (b), the data when mapped to a three dimensional feature space becomes linearly separable by a hyperplane. The features are the attribute that adds uniqueness to the feature vector, so that they can be uniquely identified. All kernel methods used in this research work are either dot product function e.g. global kernels or distance function e.g. local kernels.

In equation (3.6) the feature map (φ) is the mapping function that non-linearly maps the data to a higher dimensional feature space. For example, in equation (3.7) the kernel function (K) implicitly computes the dot product between two vectors $\boldsymbol{x}$ and $\boldsymbol{x_i}$ in higher dimensional feature space without explicitly transforming $\boldsymbol{x}$ and $\boldsymbol{x_i}$ to that higher dimensional feature space, this technique is known as "*Kernel trick*".

$$\varphi: R^p \longrightarrow R^q, \quad \text{where p < q} \tag{3.6}$$

$$K(\boldsymbol{x}, \boldsymbol{x_i}) = \varphi(\boldsymbol{x}) \cdot \varphi(\boldsymbol{x_i}) \tag{3.7}$$

The different kernel function used in this research work are defined as follows:

3.2.1. Local Kernels

They are based on evaluation of the quadratic distance between training samples and the mean vector of the class. Only feature vectors that are close or in proximity of each other have an influence on the kernel value. In this research, the value of the input vector was normalized between [0,1] and thus acceptable result can be produced at "σ" equals 1. The different local kernels were defined as follows:

Radial basis function (Rbf) kernel

The RBF kernel is defined by exponential function (Mittal and Tripathy, 2015) as shown in equation (3.8). Here, $\boldsymbol{x_i}$ is the feature vector in the data and $\boldsymbol{v_j}$ is the mean vector of class j. σ determines the width of the kernel, a and b are the constants. By replacing a and b by 1 the Gaussian kernel can be obtained. In this study the value of a and b were taken to be 2 and 3 respectively (Mittal and Tripathy, 2015).

$$K(\boldsymbol{x_i}, \boldsymbol{v_j}) = e^{\left(-\frac{\left\|x_i^a - v_j^b\right\|^2}{2\sigma^2}\right)} \qquad \text{where } \sigma, a, b > 0 \tag{3.8}$$

KMOD- (kernel with moderate decreasing)

KMOD is the distance based kernel function introduced by Ayat et. al., 2001, as shown in equation (3.9). It shows better result in classifying closely related datasets (highly correlated) and have shown better accuracy than Radial Basis Function (RBF) and polynomial kernel.

$$K(\boldsymbol{x_i}, \boldsymbol{v_j}) = e^{\left(\frac{\gamma}{\sigma^2 + \left\|x_i - v_j\right\|^2}\right)} - 1 \qquad \text{where } \sigma, \gamma > 0 \tag{3.9}$$

The parameter γ and σ controls the decreasing speed of the kernel function and the width of the kernel respectively. In this study the value of γ was taken to be one.

Gaussian kernel

The Gaussian kernel is a special case of radial basis function kernel (Scholkopf, 2002), shown in equation (3.10). Here**,** $\boldsymbol{x_i}$ is the feature vector in the image and $\boldsymbol{v_j}$ is the mean vector of the class.

$$K(\boldsymbol{x}_i, \boldsymbol{v_j}) = e^{\left(-\frac{\left\|x_i - v_j\right\|^2}{2\sigma^2}\right)} \qquad \text{where } \sigma>0 \qquad (3.10)$$

Inverse Multi-quadratic (IMQ) kernel

The inverse multi-quadratic kernel is defined as in equation (3.11) (Vidnerova and Neruda, 2011). Here the value of c was taken to be one.

$$K(\boldsymbol{x_i}, \boldsymbol{v_j}) = \frac{1}{\sqrt{(\left\|x_i - v_j\right\|^2 + c)}} \qquad \text{where } c>0 \qquad (3.11)$$

3.2.2. Global Kernels

In global kernels, the samples that are far away from each other have an influence on the kernel value. All the kernels which are based on the dot-product are global. The different global kernels are as follows:

Linear kernel

Linear kernel is one of the simplest kernel function. It is defined as the inner product of the input feature vectors, as shown in equation (3.12). The implementation of kernel algorithms using linear kernel is often equivalent to their non-kernel counterparts, i.e. PCM with linear kernel is equivalent to the standard PCM.

$$K(\boldsymbol{x_i}, \boldsymbol{v_j}) = \boldsymbol{x_i} \cdot \boldsymbol{v_j} \qquad (3.12)$$

Polynomial kernel

The polynomial kernel is a positive definite kernel i.e. each element of the kernel matrix (a kernel matric is a $n \times n$ matrix of feature vector) is positive, shown in equation (3.13). P defines the degree of the polynomial function and c is the constant (Hofmann et. al., 2008). The optimal value of degree of the polynomial function was identified to be at 2 and has been optimized within the range between [2,6]. The value of c was taken to be zero.

$$K(\boldsymbol{x_i}, \boldsymbol{v_j}) = (\boldsymbol{x_i} \cdot \boldsymbol{v_j} + \mathrm{c})^{P} \quad \text{where c} \geq 0 \text{ and} \tag{3.13}$$

Sigmoid kernel

Sigmoid kernel is a hyperbolic tangent function, as shown in equation (3.14). The parameter α work as scaling parameter for the kernel function and defines width of the kernel. The best possible value for α and c were when $\alpha > 0$and c$<$ 0 (Lin and Lin, 2003).

$$K(\boldsymbol{x_i}, \boldsymbol{v_j}) = \tanh\,(\alpha \boldsymbol{x_i} \cdot \boldsymbol{v_j} + \mathrm{c}) \tag{3.14}$$

3.2.3. Spectral Kernel

The spectral kernel takes into consideration the spectral signature concept, as shown in equation (3.15). These kernel are based on the use of spectral angle $a(\boldsymbol{x}, \boldsymbol{v_i})$ to measures the distance between the feature vector $\boldsymbol{x}$ and the mean vector of the class $\boldsymbol{v_i}$. It is expressed as follows:

$$a(\boldsymbol{x}, \boldsymbol{v_i}) = arccos\left(\frac{(\boldsymbol{x} \cdot \boldsymbol{v_i})}{||\boldsymbol{x}||\,||\boldsymbol{v_i}||}\right) \tag{3.15}$$

3.2.4. Hyper Tangent Kernel

The hyper tangent kernel is a hyperbolic tangent function, as shown in equation (3.16). The adjustable parameter σ defines the width or the scale of the kernel. Here x and v_i are the feature vectors in the data. It has been seen that the hyper tangent kernel outperforms other kernels when applied to a large data set (Mittal and Tripathy, 2015).

$$K(x, \boldsymbol{v_i}) = 1 - tanh\left(-\frac{||\boldsymbol{x} - \boldsymbol{v_i}||^2}{\sigma^2}\right) \tag{3.16}$$

3.2.5. Composite Kernel

The composite kernel concept is introduced to merge the efficiency of two different kernel function. The composite kernel function is formed by merging kernel function from two different kernel families, like- a global kernel and a local kernel or a local kernel or a spectral kernel. The composite kernel function may demonstrate a) improved classification accuracy as compared to primitive single kernel approach b) it provides the flexibility to adjust between the influence of the kernels by including weight factor (Camps-Valls et al., 2006). There are different methods for combining kernels such as stacked approach, direct summation kernel, weighted summation kernel and cross-information kernel. In this research work weighted summation kernel method has been adopted for composite kernel.

The composite kernel adjusts the influence of two different kernel and is formed by using weighted kernel summation approach as defined in equation (3.17) for input feature vector $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$. The weight factor λ varies between (0,1) and is optimized to get the best mixing between two kernels. Here K_a and K_b are two different kernel function that are used to form the composite kernel K.

$$K\left(\boldsymbol{x_i}, \boldsymbol{x_j}\right) = \lambda\, K_a\left(\boldsymbol{x_i}, \boldsymbol{x_j}\right) + (1 - \lambda) K_b(\boldsymbol{x_i}, \boldsymbol{x_j}) \tag{3.17}$$

3.3. Kernel Possibilistic c-Means (KPCM) classifier

The KPCM classifier is formed by using kernel methods with PCM algorithm. It is expected to handle non-linearity in the data by implementation of kernel methods. In KPCM the kernel metric is used to compute distance between the cluster prototype (the mean value of the cluster) and the feature vector (pixel), as mentioned in equation (3.18) and equation (3.21). This distance can be calculated in kernel higher dimension feature space without actual transformation of the feature vector to that higher dimensional feature space.

The distance between two vectors in higher dimensional feature space can be expressed as:

$$d_{i,j}^2 = \|\varphi(\boldsymbol{x}) - \varphi(\boldsymbol{x_i})\|^2$$

$$= \varphi(\boldsymbol{x}) \cdot \varphi(\boldsymbol{x}) - 2\varphi(\boldsymbol{x_i}) \cdot \varphi(\boldsymbol{x}) + \varphi(\boldsymbol{x_i}) \cdot \varphi(\boldsymbol{x_i}) \tag{3.18}$$

In the higher dimensional feature space, the KPCM objective function and the membership function (μ_{ij}) can be expressed as equation in (3.19) and equation (3.20) respectively. The meaning of the different terms used in these equations is same as defined for PCM and kernel functions.

$$\mathrm{J_{KPCM}}\,(\mathbf{U}, \mathbf{V}) = \sum_{i=1}^{c} \sum_{k=1}^{n} (\mu_{ik}{}^{m})\ ||\varphi(\boldsymbol{x_k}) - \varphi(\boldsymbol{v_i})||^2 + \sum_{i=1}^{c} \eta_i \sum_{k=1}^{n} (1 - \mu_{ik})^m \tag{3.19}$$

$$\mu_{ij} = \sum_{k=1}^{c} \frac{(||\varphi(\boldsymbol{x_k}) - \varphi(\boldsymbol{v_i})||)^{2/(m-1)}}{\eta_i} \tag{3.20}$$

The mapping function in the distance equation (3.18) can be replaced by kernel function (3.7) as:

$$d_{i,j}^2 = ||\varphi(\boldsymbol{x_k}) - \varphi(\boldsymbol{v_i})||^2 = K(\boldsymbol{x_k}, \boldsymbol{x_k}) - 2K(\boldsymbol{x_k}, \boldsymbol{v_i}) + K(\boldsymbol{v_i}, \boldsymbol{v_i}) \tag{3.21}$$

Thus, KPCM objective function can be obtained by replacing the Euclidean distance metric by kernel distance metric in the PCM objective function.

The Kernel Possibilistic c-Means (KPCM) clustering algorithm:

1- Identify the number of classes, and specify the class centroid based on the sample data.
2- Fix the value of m, $1< m <\infty$.
3- Compute the kernel distance vector for each feature vector from the centroid of every class (using equation (3.21)).
4- Calculate the value of η_i for each class (using equation (3.4)).
5- Using the kernel distance metric, compute membership value for each feature vector (using equation (3.20)).

3.4. SVM classification

SVM is a non-parametric classifier i.e. it doesn't depend on factors such as mean and co-variance for estimation of classification parameters for a class (Richards, 1993). SVM is based on finding a geometric decision planes to define decision boundaries for classification. The decision plane separates the pixel having different class memberships. This decision surface can be a multi-dimensional linear surface or a hyperplane. This hyperplane is constructed based on the training pixel. The training pixel considered for finding the hyperplane are those nearest to the hyperplane. And, the best hyperplane would be equidistant, between the bordering pixels for each of the two class. This hyperplane separating the classes is known as *optimal hyperplane* (Scholkopf, 2002; Scholkopf et. al., 2008). The pixels considered for constructing optimal hyperplane are known as *support vectors* (Figure 3.2). The equation for hyperplane is defined as follows:

$$W^T\boldsymbol{x} + W_{N+1} = 0 \tag{3.22}$$

In equation (3.22) $\boldsymbol{x}$ is the feature vector i.e. a column vector containing brightness value in all features, W is the set of coefficient known as weight vector. The number of weights will be equal to total number of features N plus one.

The optimal hyperplane is found by training based on the labelled feature vectors (training set). Two more hyperplanes can be drawn parallel to the optimal hyperplane that passes from the nearest training pixel from the classes. These planes are known as *marginal hyperplane* as defined by equation (3.23,3.24).

Marginal hyperplane on right side (Figure 3.2): $$W^Tx + W_{N+1} = 1 \tag{3.23}$$

Marginal hyperplane on left side (Figure 3.2): $$W^Tx + W_{N+1} = -1 \tag{3.24}$$

In Figure 3.2, the pixel that are beyond the right marginal hyperplanes can be defined by the equation (3.25).

$$W^Tx + W_{N+1} \geq 1 \tag{3.25}$$

In Figure 3.2, the pixel that are beyond the left marginal hyperplanes can be defined by the equation (3.26).

$$W^Tx + W_{N+1} \leq 1 \tag{3.26}$$

The separation between the hyperplane is known as *margin*. The best position of the separating hyperplane is the one where the margin is largest or where weight vector norm $||W||$ is smallest, as explained in equation (3.27).

$$margin = \frac{2}{||W||} \tag{3.27}$$

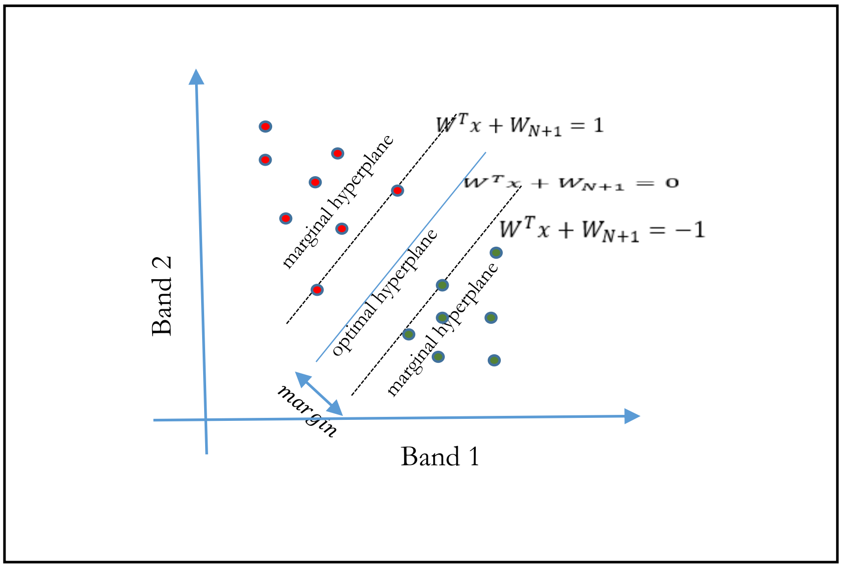


Figure 3.2. The optimal hyperplane determined by finding the maximum separation between the classes.

The linear-SVM is a classic example of linear classifier but it can also be used for classification of non-linearly distributed data by incorporating different kernels.

3.5. Accuracy assessment

Accuracy assessment is an important part of classification. To provide reliability on the classifier, it is essential to quantify the accuracy of classification. Accuracy assessment technique is used for optimizing the parameter as well as identifying the exactness of the classifier. In remote sensing the accuracy assessment of classification (specifically land-cover accuracy assessment) is usually done by comparing the result from classifier with some referenced data that is expected to reflect the true land-cover information. The referenced data includes ground truth information gathered through surveys and higher resolution images.

The error induced during supervised classification may be due to incorrect logic. For example, the classes taken during the supervised hard classification may not be mutually exclusive. The classes considered may be mislabelled during the supervised classification (Lillesand et. al., 1987). Also, using the same sampled data for all phases of classification will result in overly optimistic accuracy assessment and the results will be misleading. So in the supervised classification the sampled data need to be divided into three exclusive sets:

Training set: This set of labelled data is used to train the classifier.
Validation set: This set of labelled data is used to optimize the parameter for the classifier.
Test set: This set of labelled data is used to estimate the accuracy of the classification.

For hard classification, the confusion matrix is considered as a standard for assessing the accuracy of classification. But on the other hand there is no standard, globally accepted technique for assessing the accuracy of soft classification. This limits the usability of soft classification techniques. In this study a number of different techniques for computing the accuracy of soft classification output were used. Soft as well as hard methods of accuracy assessment were implemented for optimizing the parameters and identifying the accuracy of the classifier.

The following sub section describes the various methods for accuracy assessment.

3.5.1. Error Matrix

It is the standard for accuracy assessment in hard classification. It lists the predicted output classes present in classified data in row and the actual classes present in the referenced data in column. Each cell in the error matrix represent the pixels that are common between the referenced class and the classified class. The diagonal element contains pixels that belongs to same class in referenced data set as well as in classified data set (Richards, 1993). The total number of pixels in the column of error matrix represent the total number of labelled pixels (pixel in the test set) available per class. The row sum represents the total number of pixels classified in a particular class by the classifier. To quantify the accuracy of classification the following measures for classification accuracy were calculated:

Producer's accuracy: It is defined as the ratio of the correctly classified pixels in the (i.e. a diagonal element of a class) to the total number of pixels of that class as derived from the reference data (i.e. column total). This measures the probability of a referenced pixel being correctly classified and is measure of omission error (Lillesand and Kiefer, 1979).

User's accuracy: It is defined as the ratio of the correctly classified pixels in the class (i.e. a diagonal element of a class) to the total number of pixels that were actually classified in that class (i.e. row total), the result is a measure of commission error. This measures the probability that a pixel classified on the map actually represents the category on the ground (Richards, 1993).

Overall Accuracy: The overall accuracy is defined as the ratio of the sum of all the diagonal element of the matrix to the total number of pixel considered for accuracy assessment.

Kappa Coefficient: The Kappa coefficient is the measure of agreement between the classification map and the referenced data, shown in equation (3.28).

$$\kappa = \frac{probability\ of\ correct\ classification - probability\ of\ chance\ agreement}{1 - \ probability\ of\ chance\ agreement} \tag{3.28}$$

The probability of correct classification is given by the diagonal element and the probability of chance agreement is given by row and column total.

3.5.2. Fuzzy Error Matrix (FERM)

The fuzzy error matrix (M) is a modification of traditional error matrix for accuracy assessment of the soft classifier. Similar to the traditional error matrix the fuzzy error matrix is a square array of positive fractional value varying between [0,1]. The column R_n usually represent the sample elements assigned to the reference class n while the rows indicate the sample elements assigned to the classified class m(Binaghi et al., 1999). The element in fuzzy error matrix (M) at row m and column n for a feature vector $\boldsymbol{x}$ is computed as shown in equation (3.29).

$$M(m,n) = \sum_{\boldsymbol{x} \in \boldsymbol{X}} \min(\mu_{C_m}(\boldsymbol{x}), \mu_{R_n}(\boldsymbol{x})) \tag{3.29}$$

In equation (3.29), $\boldsymbol{X}$ is the overall sampled data set. μ_{C_m} and μ_{R_n} are the membership value for the referenced and the classified data. The "min" operator is the traditional fuzzy set operator, it returns the minimum membership value between the classified and referenced data set for a class. The various indices for accuracy assessment like overall accuracy, user's accuracy, and producer's accuracy can be calculated from FERM as in case of traditional error matrix. The value of these accuracy assessment measures range between [0,1].

Producer's Accuracy: Producer's accuracy for a class is calculated by dividing the element of major diagonal for the class by the total grade of membership found in the referenced data for the specified class (i.e. the column total).

User's Accuracy: User's accuracy for a class is calculated by dividing the element of major diagonal for the class by the total grade of membership found in the classified data for the specified class (i.e. the row total).

Overall Accuracy: Overall accuracy in FERM is calculated by dividing the sum of the major diagonal by the total grade of membership found in the reference data.

3.5.3. Sub-pixel Confusion Uncertainty Matrix (SCM)

SCM is also a modification of traditional error matrix. Unlike, traditional error matrix the entries are based on the agreement and disagreement measure for a class between the classified output and the referenced data at pixel level (Silvan-Cardenas and Wang, 2008). It presents a better insight into the per class accuracy at pixel level for soft classification. But SCM will not be used for accuracy assessment of KPCM classification because of the membership constraint to be followed in SCM (Upadhyay et. al., 2014).

The area overlap between the classes is used as a measure of agreement and disagreement between the classes to compute the entries of SCM. The operator used for computing SCM are defined as follows:

MIN operator

The MIN operator is the fuzzy set intersection operator. In case of SCM it gives the maximum sub-pixel overlap between the classes. The value with MIN operator for a pixel containing reference class m and classified class n with membership value μ_{C_m} and μ_{R_n} respectively is expressed as shown in equation (3.30).

$$MIN(m,n) = MIN(\mu_{C_m}, \mu_{R_n}) \tag{3.30}$$

$PROD$ operator

The $PROD$ operator gives the joint probability that the referenced and the classified pixel belongs to two different classes. The value with $PROD$ operator for a pixel containing reference class m and classified class n with membership value μ_{C_m} and μ_{R_n} respectively is expressed as shown in equation (3.31).

$$PROD(m, n) = \mu_{C_m} . \mu_{R_n} \tag{3.31}$$

$LEAST$ operator

The $LEAST$ operator gives the minimum possible sub-pixel overlap between two classes within a pixel. The value with $LEAST$ operator for a pixel containing class m and class n with membership value μ_{R_n} and μ_{C_m} respectively is expressed as shown in equation (3.32).

$$LEAST(m, n) = \max((\mu_{C_m} + \mu_{R_n}) - 1, 0) \tag{3.32}$$

MIN-MIN operator:

The *MIN-MIN* composite operator uses minimum operator for assigning both the diagonal and off-diagonal element, as shown in equation (3.33) and equation (3.34). It differs from the *MIN* operator because it computes the off-diagonal element based on over and under estimation errors. The over-estimation error (μ'_{R_n}) is due to the over estimation of the reference pixel membership by the classified pixel membership. The under estimation error (μ'_{C_m}) is due to under estimation of the reference pixel membership by the classified pixel membership.

Diagonal element: $\min(\mu_{C_m}, \mu_{R_n})$ (3.33)

Off-diagonal element: $\min(\mu'_{C_m}, \mu'_{R_n})$ where m≠n and $\mu'_{C_m} = \mu_{C_m} - \min(\mu_{C_m}, \mu_{R_n})$,

$\mu'_{R_n} = \mu_{R_n} - \min(\mu_{C_m}, \mu_{R_n})$

(3.34)

MIN-PROD operator

The *MIN-PROD* operator uses the minimum operator to compute the diagonal element for SCM and the normalized product operator for the off-diagonal cell, as shown in equation (3.35) and equation (3.36).

Diagonal element: $$\min(\mu_{C_m}, \mu_{R_n}) \quad (3.35)$$

Off-diagonal element: $$(\mu'_{C_m} \cdot \mu'_{R_n}) / \sum_i \mu'_{R_n} \quad \text{where m} \neq \text{n} \quad (3.36)$$

***MIN-LEAST* operator**

The *MIN-LEAST* operator uses the minimum operator to compute the diagonal element for SCM and the normalized *LEAST* operator for the off-diagonal cell, as shown in equation (3.37) and equation (3.38).

Diagonal element: $$\min(\mu_{C_m}, \mu_{R_n}) \quad (3.37)$$

Off-diagonal element: $$(\mu'_{C_m} \cdot \mu'_{R_n}) / \sum_i \mu'_{R_n} \quad \text{where m} \neq \text{n} \quad (3.38)$$

3.5.4. Entropy

Entropy has been introduced to estimate the uncertainty in the classification. It defines the degree to which the membership value is partitioned between the classes. Entropy maximises when the membership value is partitioned between the classes and minimizes when the membership value is associated entirely with a single class. Entropy measure is favoured when the classification is fuzzy and ground data is hard (Foody, 1995). It has been observed that entropy is not a satisfactory technique for estimating the accuracy of the classification but can be used for comparing the quality of the classification based on uncertainty in the results. The value of entropy for a pixel is given by the Shannon's entropy as expressed by equation (3.39) (Ricotta and Avena, 2002;Foody, 1995).

$$Entropy = -\frac{\sum_{k=1}^{c} \mu_{jk} \, log_2 \, \mu_{jk}}{\sum_{k=1}^{c} \mu_{jk}} \quad (3.39)$$

Here $\boldsymbol{\mu_{jk}}$ is the fuzzy membership value of the classified output and $\boldsymbol{c}$ is the total number classes present in the classified output. Higher value of entropy (i.e. close to one) resembles lower quality of classification.

3.5.5. Root Mean Square Error (RMSE)

RMSE measures the difference between the membership value of the classified output and the membership value of the referenced data. It determines the error in the prediction of the membership value by the classifier. The RMSE values are always greater than zero. The RMSE value close to zero shows the output from the classifier deviates less from the referenced data. In case of soft classification output the RMSE can be calculated into two ways: a) Global RMSE; as shown in equation(3.40) and b) Per class RMSE; as shown in equation (3.41) (Byju, 2015; Dehghan and Ghassemian, 2006).

Global RMSE:
$$RMSE_{Global} = \sqrt{\frac{\sum_{i=1}^{c}\sum_{j=1}^{N}\sum\left(\mu_{cl_{ij}} - \mu_{r_{ij}}\right)^2}{M \times N}} \tag{3.40}$$

Per class RMSE:
$$RMSE_{perClass} = \sqrt{\frac{\sum_{j=1}^{N}\left(\mu_{c_{ij}} - \mu_{r_{ij}}\right)^2}{M \times N}} \tag{3.41}$$

In equation (3.40) and equation (3.41), $\mu_{cl_{ij}}$ is the membership value in the classified image and $\mu_{r_{ij}}$ is the membership value in the referenced image for the feature vector j in class i. c is the total number of class, M is the total number of features present in the data and N is the total number of feature vectors per feature and $M \times N$ is the size of the image.

3.5.6. Pearson correlation coefficient (ρ)

The Pearson correlation coefficient is the measure of linear dependence between two variables. The value of the correlation coefficient varies between [-1, +1]. The value of Pearson correlation of -1 resembles completely negative correlation, 0 is no correlation and +1 is completely positive correlation (Weisstein, 2006).

In this research study, the Pearson correlation coefficient was calculated between the fractional image for each class of Landsat-8 and fractional image for each class of Formosat-2. The correlation between two feature vectors, $\boldsymbol{X}$ and $\boldsymbol{Y}$, can be given by the equation (3.42). The Pearson correlation coefficient ($\rho_{X,Y}$) between two feature vectors, $\boldsymbol{X}$ and $\boldsymbol{Y}$, is defined as the ratio of the covariance between the membership value of the corresponding feature vector $cov(\mu_{X_i}, \mu_{Y_i})$ to the product of their standard deviation ($\sigma_{\mu_{X_i}} \times \sigma_{\mu_{Y_i}}$).

$$\rho_{X,Y} = \frac{\sum_{i=1}^{n} cov(\mu_{X_i}, \mu_{Y_i})}{\sigma_{\mu_{X_i}} \times \sigma_{\mu_{Y_i}}} \tag{3.42}$$

3.5.7. Simulated Image Technique

In this research a new method for identifying the behaviour of fuzzy based algorithm (based on the distance measure) have been introduced. This method was developed by taking into consideration the basic idea of assigning the fuzzy membership values to feature vectors based on the distance measure from the mean vector of the classes (mean vector). The simulated image is generated based on the sample data for each class with desired number of bands. With the simulated image, it is easy to compare the outcome of the classifier with the expected known input at a particular location. Also, it makes easy to identify the behaviour of classifier with the mixed pixels. The mixed pixels can be simulated with varying proportions of different classes. As shown in Figure. 3.3, that simulated image is classified into fractional images by using soft classifier. The proportion of these classes in each individual fractional image can be identified and compared with the input. The membership value for a class in fractional image is affected by the distance criteria used for classification.

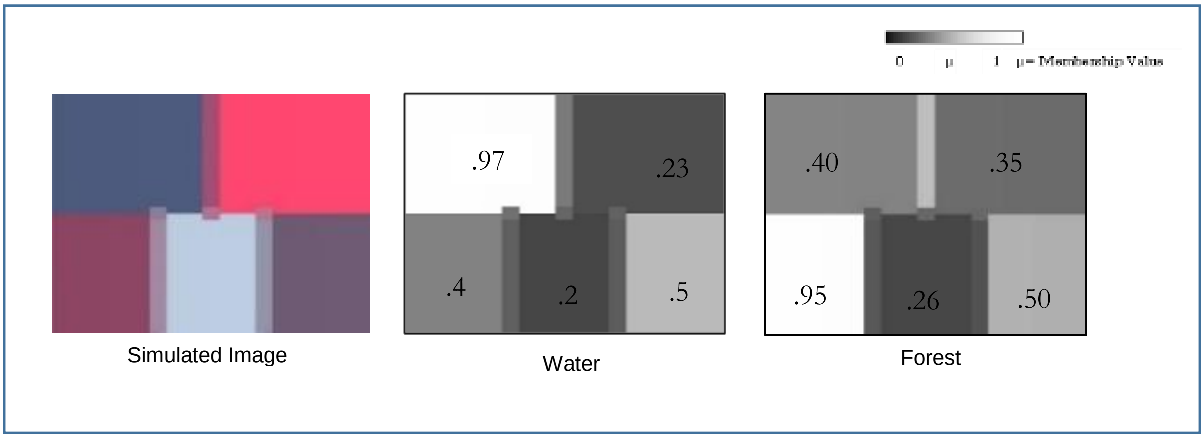


Figure 3.3. Simulated image with the fractional output along with possibilistic membership value generated by the fuzzy classifier (PCM).

CHAPTER 4

4. STUDY AREA AND METHODOLGY

4.1. Study Area

This section defines the data used for the study. It also provides explanation for selecting particular data and study area.

The site for the study work was situated in Haridwar district in the state of Uttarakhand, India. Area extends from 29°52'49" N to 29°54'2" N and 78°9'43" E to 78°11'25" E. The site was identified with six land cover classes (Figure 4.1) i.e. wheat, grassland, forest, eucalyptus, water and riverine sand (mentioned simply as sand in later text). The reasons for selecting this study area include:

- Landsat-8 and Formosat-2 images were available for selected site.
- Ground truth information was available and has been identified for six classes for Landsat-8 and Formosat-2 images.

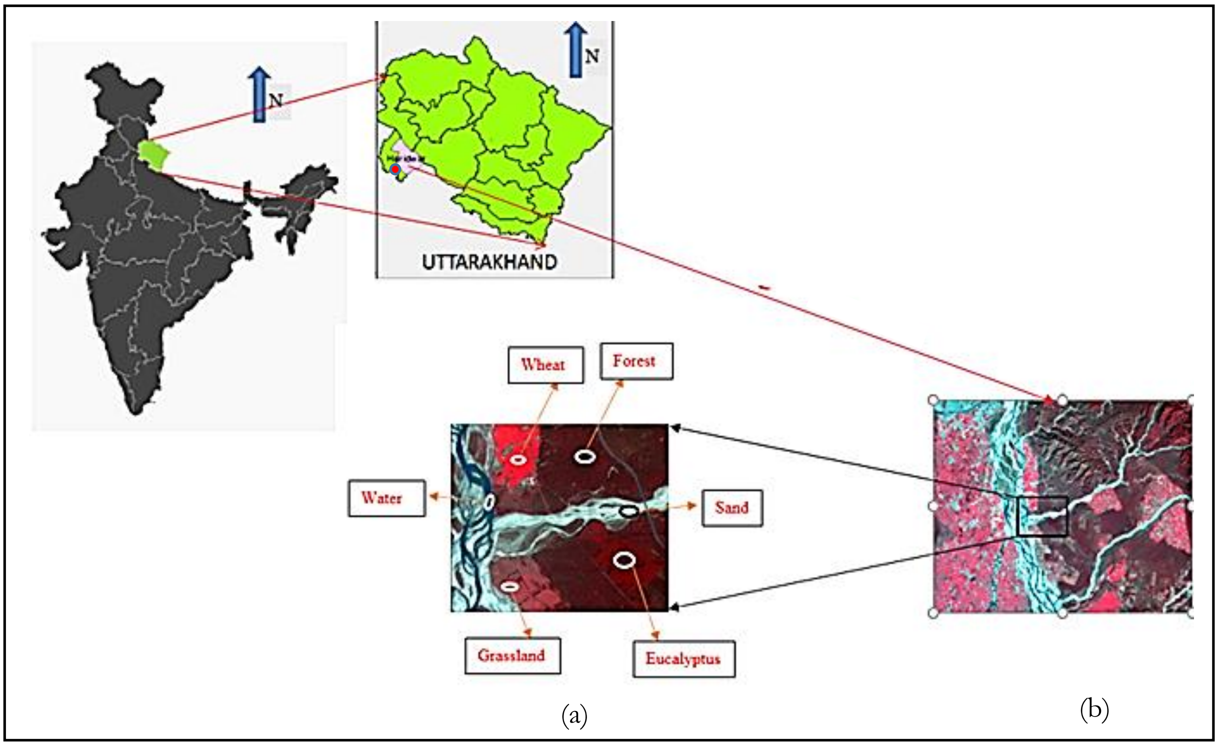


Figure 4.1. Location of area under study (a) Formosat-2 image (8 m) (a) Landsat-8 image (30 m)

4.2. Data details

The data set used in the project work was acquired from Landsat-8 and Formosat-2 satellite.

Landsat-8 is an American Earth observation satellite launched on February 11, 2013. It provides moderate-resolution imagery, from 15 meters to 100 meters, of Earth's land surface and polar regions. The used image was acquired on 12th February, 2015. Landsat 8 operates in the visible, near-infrared, short wave infrared and thermal infrared spectrums. The sensor spectral wavebands specifications are enlisted in Table 4.1.

Table 4.1. Data details for Landsat-8

Spectral Band	Wavelength	Resolution (m)
Band 1 - Coastal / Aerosol	0.433 - 0.453 μm	30
Band 2 – Blue	0.450 - 0.515 μm	30
Band 3 – Green	0.525 - 0.600 μm	30
Band 4 – Red	0.630 - 0.680 μm	30
Band 5 - Near Infrared	0.845 - 0.885 μm	30
Band 6 - Short Wavelength Infrared	1.560 - 1.660 μm	30
Band 7 - Short Wavelength Infrared	2.100 - 2.300 μm	30
Band 8 – Panchromatic	0.500 - 0.680 μm	15
Band 9 – Cirrus	1.360 - 1.390 μm	30

FORMOSAT-2 was the first remote sensing satellite developed by National Space Organization (NSPO). FORMOSAT-2 satellite carries both "remote sensing" and "scientific observation" tasks in its mission. It supports monitoring and detecting land change for any specific regions for various industries and mapping applications. The used image was acquired on 21st February, 2015. Formosat-2's ability to acquire repeat imagery of an area of interest every day and with the same viewing parameters guarantees a timely flow of compatible data, allowing to analyse and compare imagery acquired at different dates with no need of additional processing. The satellite captures panchromatic and multispectral data simultaneously with 2m and 8m spatial resolution respectively. The sensor footprint is 24×24 km and is designed in such a way to revisit the same point on the globe every day in the same viewing conditions. The sensor spectral wavebands specifications have been enlisted in Table 4.2.

Table 4.2. Data details for Formosat-2

Band	Wavelength (micrometres)	Spatial Resolution (meters)
Band 1 –Blue	0.45 - 0.52	8
Band 2 –Green	0.52 - 0.60	8
Band 3 –Red	0.63 - 0.69	8
Band 4 - Near Infrared (NIR)	0.76 - 0.90	8
P - Panchromatic	0.45 – 0.90	2

Simulated image was generated to study the behaviour of developed KPCM algorithm accurately. The different classes and their mixing present in the simulated image is explained in Figure (4.2). The advantage of using simulated image are as follows:

- The composition of each class is known.
- The mixed pixels are simulated with varying mixture of classes.
- The capability to handle the mixed pixel by the developed KPCM algorithm can also be verified.
- The pixel with known composition can easily be located within the simulated image.

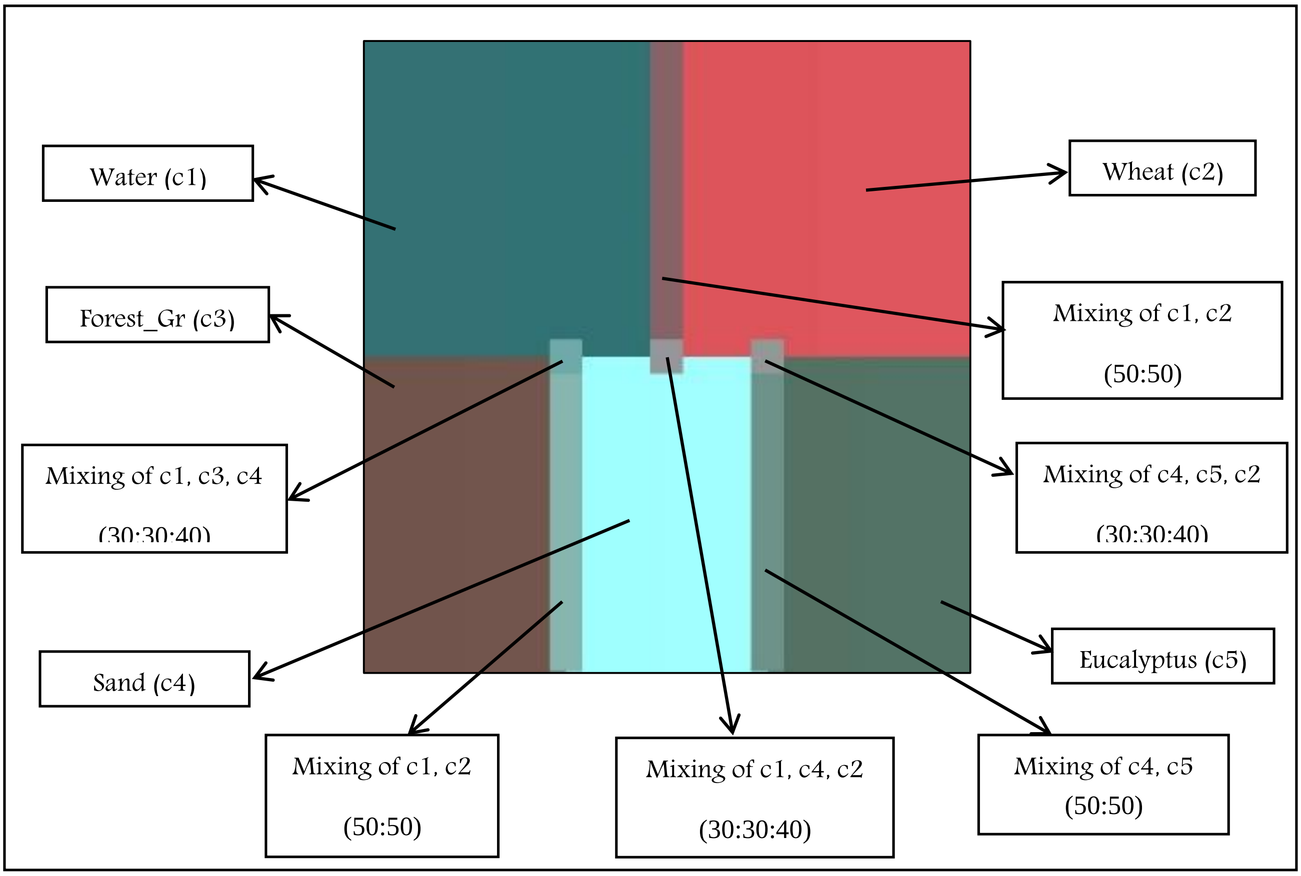


Figure 4.2. Image simulated for Formosat-2 real image. It contains five different classes with variation of 1 unit between the DN values

The available finer resolution Formosat-2 data (8m) was used as a reference data for accuracy assessment of classified results of coarser resolution Landsat-8 data (30m). In order to evaluate the accuracy of results by image to image accuracy assessment method an integral ratio between the pixel size of referenced data and classified data was needed. So, the Formosat-2 data was resampled to 10m, a ratio of three was set between the referenced and the classified image pixels. The **Nearest Neighbour Resampling** technique was used to resample the Formosat-2 data as it does not affect the pixel value of the input layer.

4.3. Methodology

This section defines the methodology that has been adopted in order to achieve the desired objective.

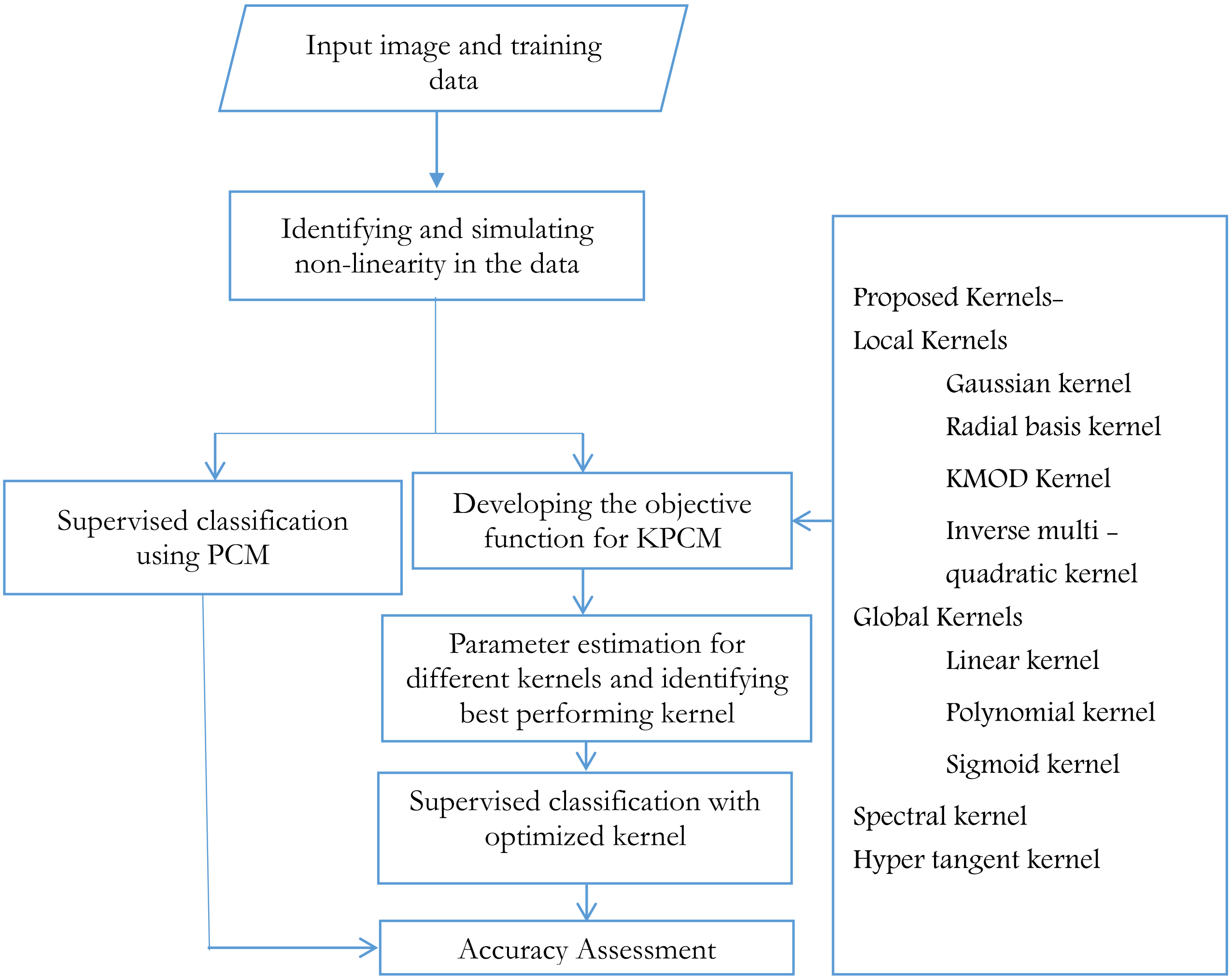


Figure 4.3. Overview of the methodology

All the steps presented in Figure (4.3) for the methodology adopted in the research study are explained below.

a) Identifying and simulating non-linearity in the input data

Initially, the study was carried out with Formosat-2 data and six different classes were identified according to the ground survey. The linear-SVM classification was performed on the Formosat-2 data. The classification results were showing 99% accuracy, thus linear-SVM classifier was successful in classifying the present dataset. Hence, it can be concluded that in the available data for Formosat-2, these classes were linearly separable (Figure A-2, Appendix A). Therefore, in order to demonstrate a useful implementation of kernel functions, non-linearity was simulated in the available dataset image using following steps:

- A new eucalyptus class was identified in the forest class. There was high mixing between these classes.
- Merging "forest" and "grassland" class into a single "forest" class, naming it as "Forest_Gr".
- Taking subset of the features. Band1 (blue) and Band2 (green) were chosen for classification.

The detail of simulating the non-linearity is explained in section A.2. in Appendix A. The Formosat-2 and Landsat-8 data now contains non-linearity in them with five different classes viz. Forest_Gr, wheat, sand, water, and eucalyptus.

b) Developing the objective function for KPCM

The Kernel based PCM classifier was formed by replacing the Euclidean distance norm present in the PCM classifier with kernel metric as described in section (3.1.3) in chapter 3.

c) Parameter estimation for different kernels and identifying best performing kernel

The parameter estimation is one of the most important step in the classification process. Choosing the optimal parameter guarantees the best results from the classifier. Here, the developed KPCM classifier was executed on the available data set for different values of fuzzy parameter ranging between [1.3,4.5] and the optimal value of fuzzy parameter was selected based on the accuracy assessment of classification. Comparing the accuracy of kernel functions on different value of fuzzy parameter the best performing kernel function was identified and was chosen for classification. The value of weight constant λ in composite kernel was also optimized within the range of (0,1). The parameter selection was based on the implementation of accuracy assessment method like Pearson correlation coefficient, FERM and RMSE.

d) Supervised classification with optimized kernel

The supervised KPCM classifier was developed with an aim to handle non-linearity between the classes. In this step, the best kernel function selected from nine different kernel functions was incorporated into PCM. The optimized value for fuzzy parameter was used for classification. The steps followed in supervised classification using KPCM classifier were as follows (Richards, 1993):

1. Identifying the required land cover classes into which the image has to be classified. This ground cover data was used for training the classifier and evaluating the accuracy.
2. Identifying the ground data in the image for each class. This data is known as *training data*. Training data was collected through ground surveys and photointerpretation method.
3. Using the training data to estimate the parameter for KPCM. These parameters are known as *signature* of the class. This step is known as *training* of classifier.
4. Using the trained KPCM classifier to calculate the membership value of feature vectors for each class. These per class classified maps are known as *thematic maps*.
5. Using the higher resolution classified results as referenced data (Formosat-2) for computing accuracy of the classification.

e) Accuracy Assessment

Several researchers have given many different techniques for accuracy assessment of the output from soft classifiers but none of them is considered as standard and universally accepted. Different accuracy assessment method used in this study are defined in section (3.5).

In this research work, a new method for evaluating the accuracy of classification was proposed. This method is known as simulated image technique. It has been used to estimate the results from PCM and KPCM classifier. As, the pixel composition is known in the simulated image so the results from the classifier can easily be verified. Other techniques like, FERM and RMSE and correlation were used for verification of fractional output from soft classifier. Apart of this, to implement the standard method of accuracy assessment i.e. error matrix, the fractional outputs from soft classifier were hardened and specific class based accuracy (i.e. user's accuracy and producer's accuracy), overall accuracy and Kappa coefficient were computed.

CHAPTER 5

5. RESULTS

This chapter shows the result of different analysis that were performed on the dataset to achieve the objective of the research study. This chapter is majorly divided into five sections. The first section (5.1) deals with the method for identifying the best kernel and optimizing the fuzzy parameter. In the second section (5.2), the classification done with KPCM classifier using the best kernel and optimised parameter obtained in the first section are dealt with. The third section (5.3) compares the PCM and KPCM classifier and elaborates the advantage of using the non-linear KPCM classifier over the linear PCM classifier. The fourth section (5.4) demonstrates the usability of composite kernel and their advantage. The fifth section (5.5) explains the effect on the KPCM classification by introducing the untrained classes in the dataset.

5.1. Identifying the best kernel and estimating the parameter

The optimal fuzzy parameter value (m) and the best performing kernel for developed KPCM algorithm with Formosat-2 data were estimated based on the simulated image. Here, the fuzzy parameter was estimated for linear Formosat-2 data as well as for non-linear Formosat-2 data, thus the effect of non-linearity on the value fuzzy parameter was observed.

Using the optimized value of fuzzy parameter and the best performing kernel function, the KPCM based classification was performed on Formosat-2 data. The classified fractional output for Formosat-2 data was used as referenced data for calculating FERM, RMSE and Pearson correlation by image to image accuracy assessment method for optimization of parameters for Landsat-8 (30m) dataset. From here, the effect of resolution on the value of fuzzy parameter can be observed. The fuzzy parameter was optimized within the range of [1.5,4]. The best kernel function was chosen from nine different kernels. Finally, with the optimized parameter value and the best kernel function, supervised KPCM classification on Landsat-8 was performed and the accuracy assessment of the classification was conducted.

As none of the accuracy assessment methods were considered as a standard for fuzzy accuracy assessment, multiple accuracy assessment techniques were implemented and compared. The techniques such as FERM, Person correlation coefficient, entropy and RMSE were used to assess the accuracy of the classification. In the following section, various techniques implemented for estimating the fuzzy parameter have been explained with their results. Each parameter estimation method measures a different attribute of the classified results.

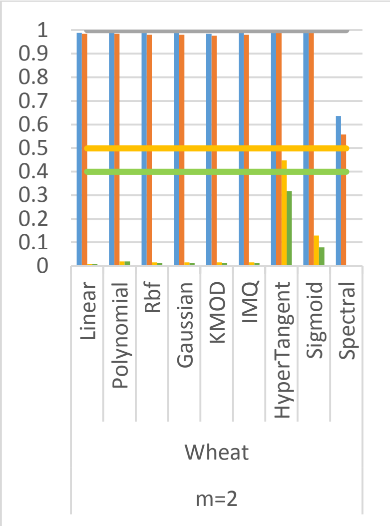
5.1.1. Optimisation of Fuzzifier (m)

Simulated image technique

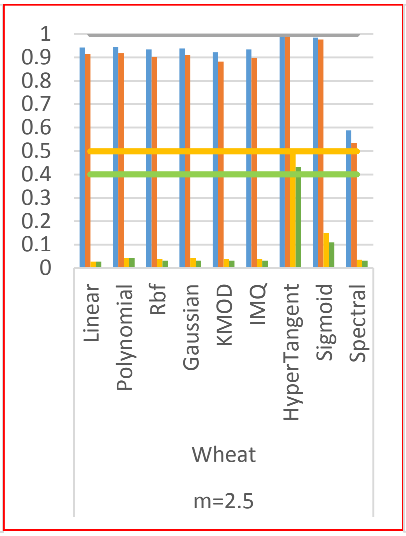
The simulated image was generated for both linear and non-linear Formsat-2 data. Firstly, the KPCM classification was performed on the linear data set (simulated Formosat-2 image) with five different classes named as Wheat, Water, Forest, Grassland and Sand. Later, the KPCM classification was performed on the non-linear data set (simulated Formosat-2 image) with five different classes namely Water, Wheat, Forest_Gr (forest and grassland merged), Sand and Eucalyptus.

At first, the KPCM classifier implementing kernel functions were applied to Formosat-2 linear simulated dataset. The value for the fuzzy parameter was selected based on the classification results (Figure 5.1). For the optimal classification, the membership value of pure pixel in the classified output of a class must be maximum (shown as Pure Pixel (I) in Figure 5.1) and it must also show the variation present within the class (shown as Pure Pixel (II) in Figure 5.1). The mixed pixels were simulated with two variations, one with composition of 50:50 (shown as Mixed Pixel (50:50) in Figure 5.1) between two different classes and other with composition of 30:30:40 (shown as Mixed Pixel (30:30:40) in Figure 5.1) among three different classes. The membership value of a pixel is shown with three different horizontal lines i.e. grey, yellow and green colour representing the target values for a pure pixel, mixed pixel (50:50) and mixed pixel (30:30:40) respectively. The target membership value expected from the pixel with full belongingness to a class must be close to 1, the target membership value of 0.50, 0.40 and 0.30 is expected from the pixel with 50%, 40% and 30% belongingness for a class respectively.

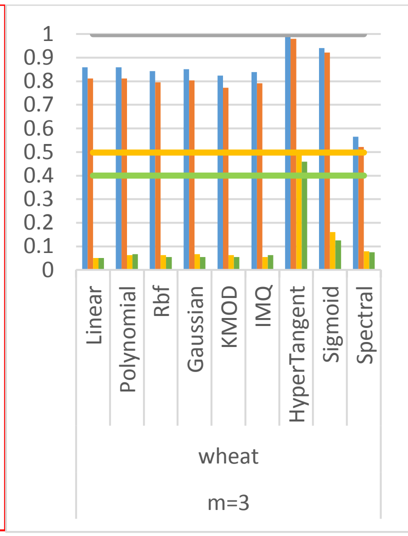
As evident from the histograms in Figure 5.1, the optimal value for fuzzy parameter was obtained at 2.5 for hyper tangent kernel for all five classes. Later, the developed KPCM classifier was applied on Formosat-2 non-linear simulated image and as illustrated in figure 5.2, the optimal value for fuzzy parameter was obtained at 3 for hyper tangent kernel for all classes. In both cases, the hyper tangent kernel function outperforms all other kernel functions followed by the sigmoid kernel function.


Linear
Polynomial
Rbf
Gaussian
KMOD
IMQ
HyperTangent
Sigmoid
Spectral
Wheat
m=2

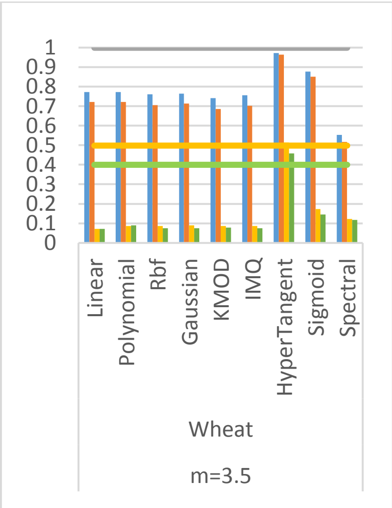
(a)


Linear
Polynomial
Rbf
Gaussian
KMOD
IMQ
HyperTangent
Sigmoid
Spectral
Wheat
m=2.5

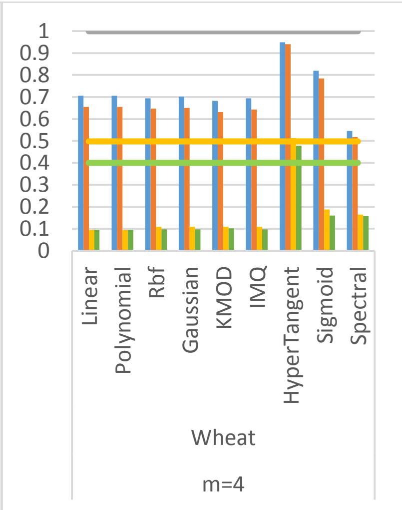
(b)


Linear
Polynomial
Rbf
Gaussian
KMOD
IMQ
HyperTangent
Sigmoid
Spectral
wheat
m=3

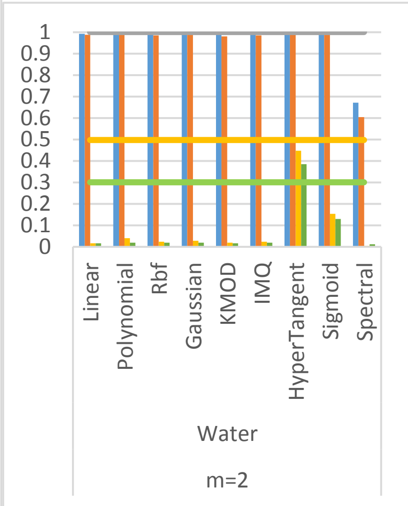
(c)


Linear
Polynomial
Rbf
Gaussian
KMOD
IMQ
HyperTangent
Sigmoid
Spectral
Wheat
m=3.5

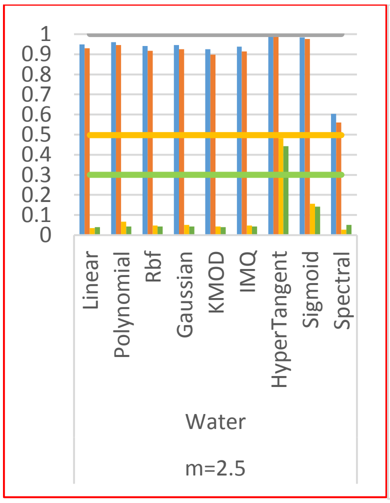
(d)


Linear
Polynomial
Rbf
Gaussian
KMOD
IMQ
HyperTangent
Sigmoid
Spectral
Wheat
m=4

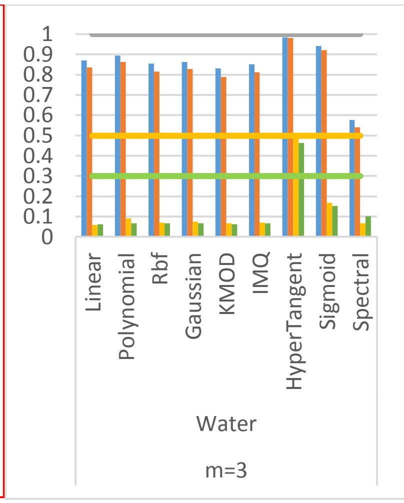
(e)


Linear
Polynomial
Rbf
Gaussian
KMOD
IMQ
HyperTangent
Sigmoid
Spectral
Water
m=2

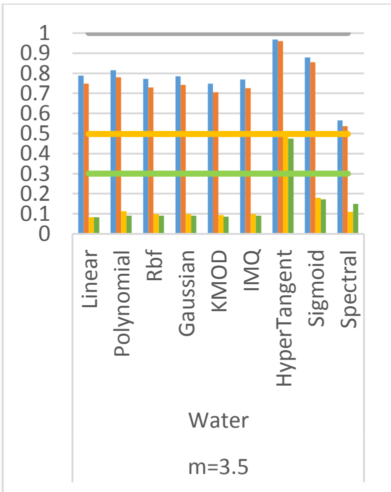
(f)


Linear
Polynomial
Rbf
Gaussian
KMOD
IMQ
HyperTangent
Sigmoid
Spectral
Water
m=2.5

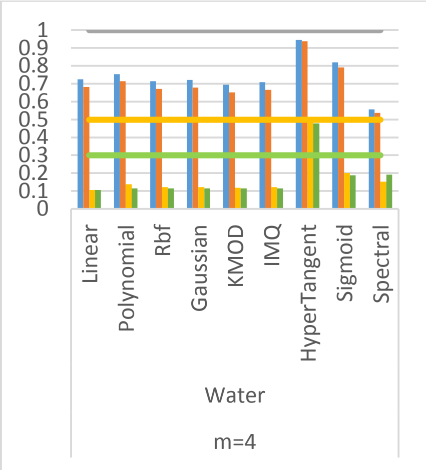
(g)


Linear
Polynomial
Rbf
Gaussian
KMOD
IMQ
HyperTangent
Sigmoid
Spectral
Water
m=3

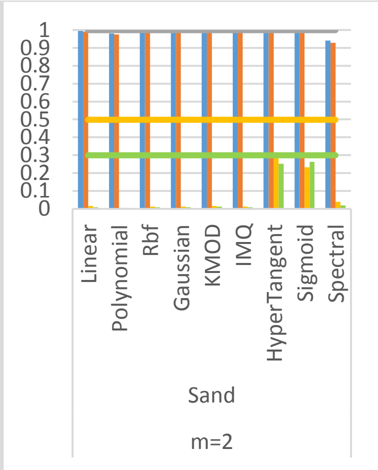
(h)


Linear
Polynomial
Rbf
Gaussian
KMOD
IMQ
HyperTangent
Sigmoid
Spectral
Water
m=3.5

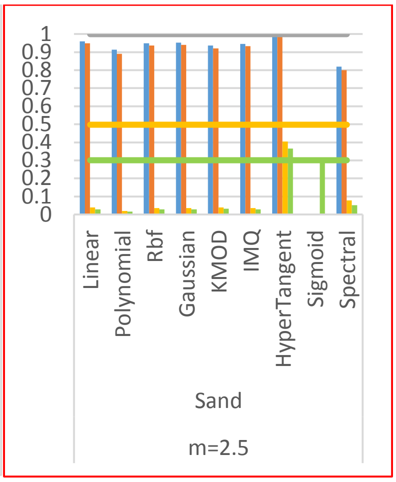
(i)


1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Linear
Polynomial
Rbf
Gaussian
KMOD
IMQ
HyperTangent
Sigmoid
Spectral
Water
m=4

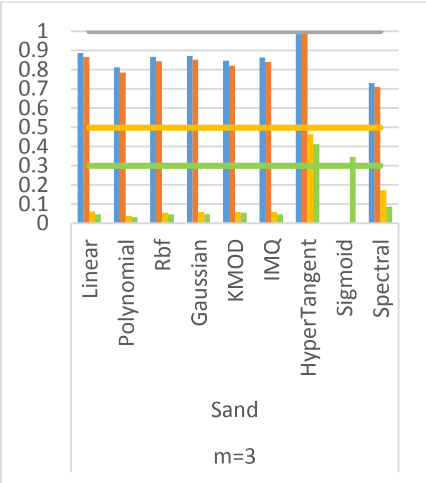
(j)


1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Linear
Polynomial
Rbf
Gaussian
KMOD
IMQ
HyperTangent
Sigmoid
Spectral
Sand
m=2

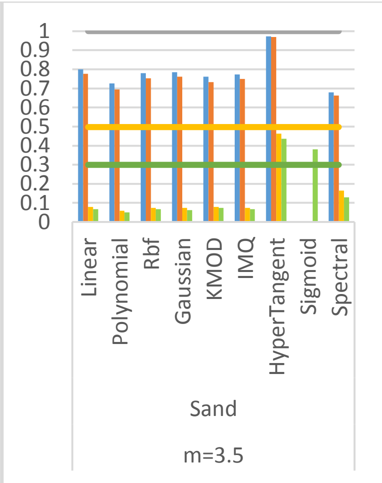
(k)


1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Linear
Polynomial
Rbf
Gaussian
KMOD
IMQ
HyperTangent
Sigmoid
Spectral
Sand
m=2.5

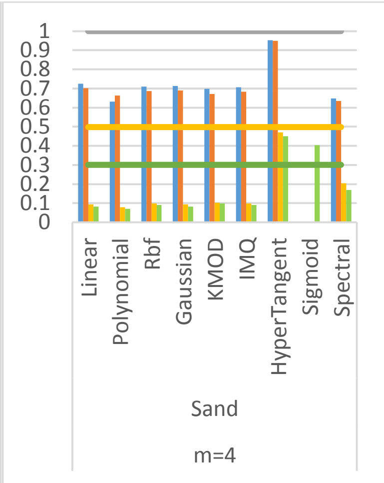
(l)


1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Linear
Polynomial
Rbf
Gaussian
KMOD
IMQ
HyperTangent
Sigmoid
Spectral
Sand
m=3

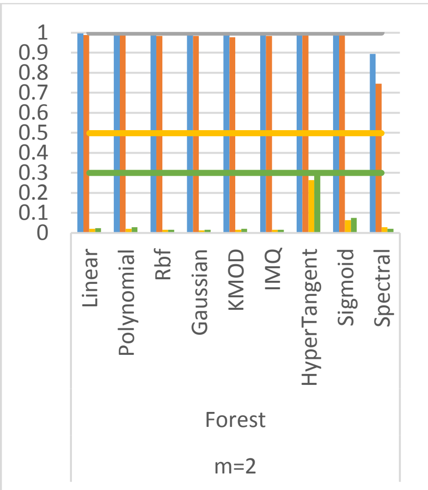
(m)


1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Linear
Polynomial
Rbf
Gaussian
KMOD
IMQ
HyperTangent
Sigmoid
Spectral
Sand
m=3.5

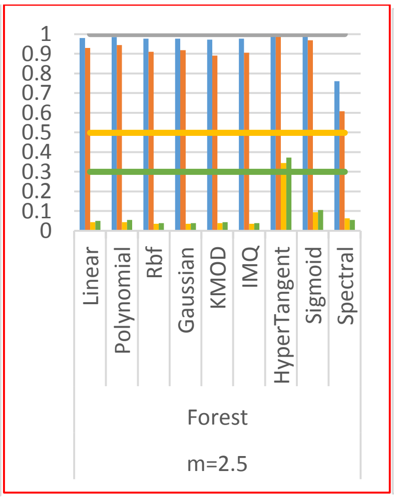
(n)


1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Linear
Polynomial
Rbf
Gaussian
KMOD
IMQ
HyperTangent
Sigmoid
Spectral
Sand
m=4

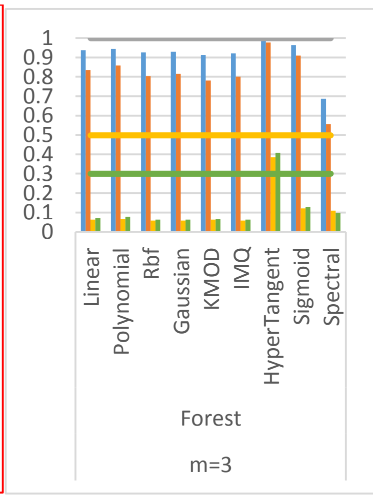
(o)


1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Linear
Polynomial
Rbf
Gaussian
KMOD
IMQ
HyperTangent
Sigmoid
Spectral
Forest
m=2

(p)


1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Linear
Polynomial
Rbf
Gaussian
KMOD
IMQ
HyperTangent
Sigmoid
Spectral
Forest
m=2.5

(q)


1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Linear
Polynomial
Rbf
Gaussian
KMOD
IMQ
HyperTangent
Sigmoid
Spectral
Forest
m=3

(r)

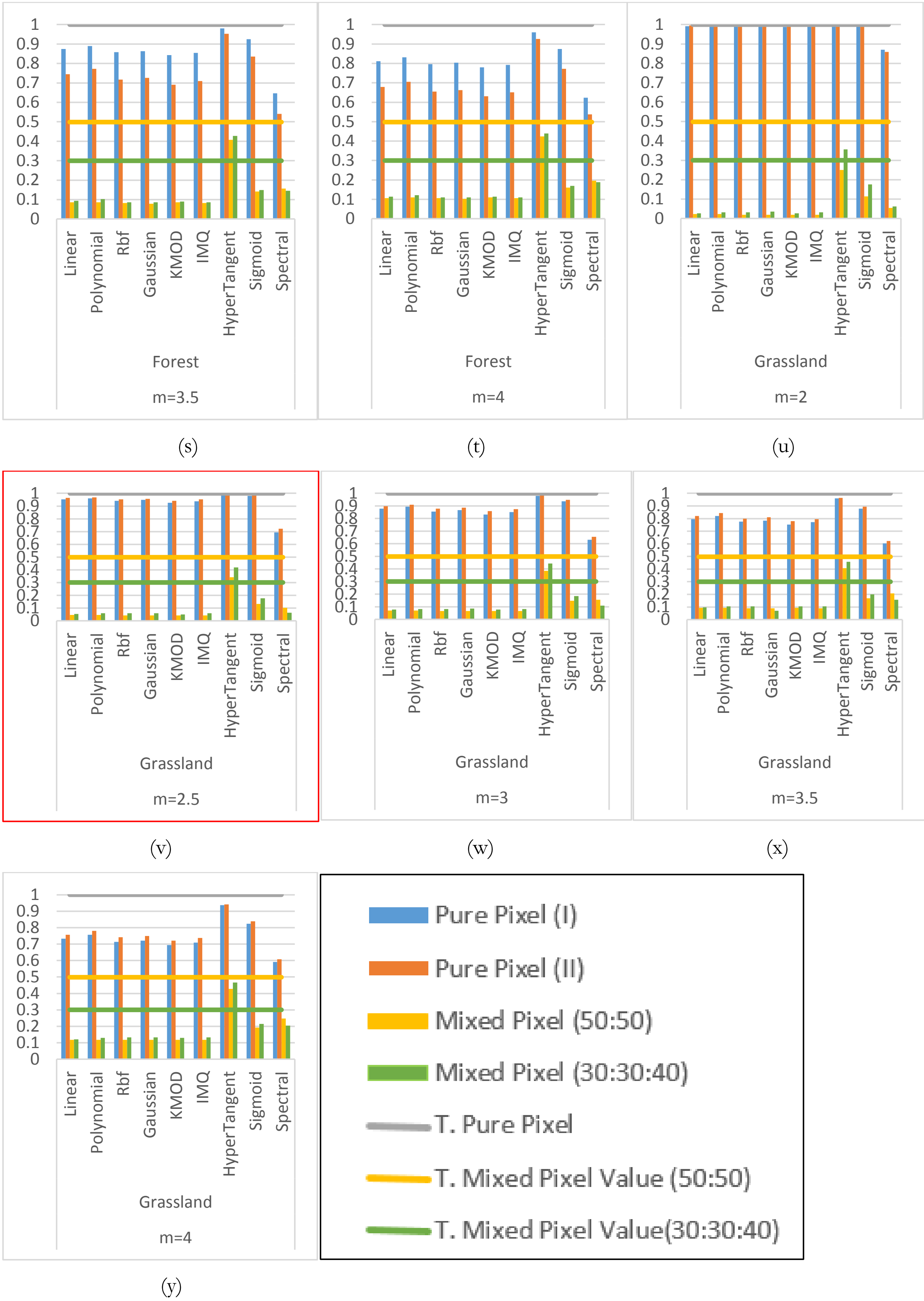
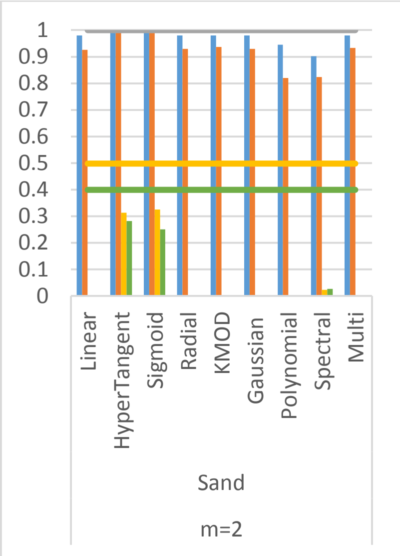
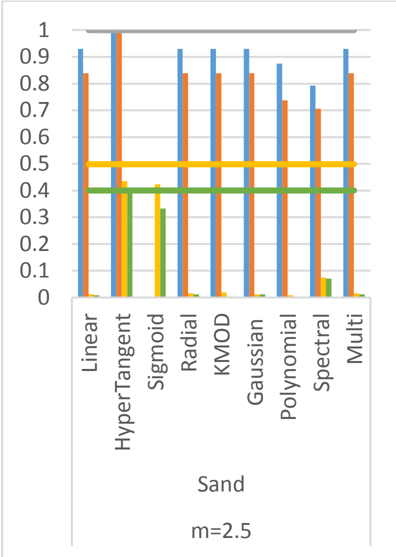


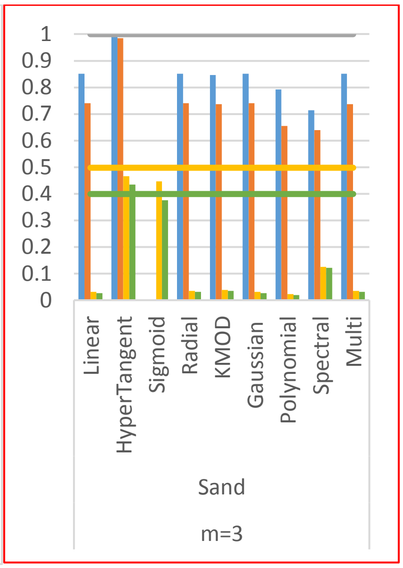
Figure 5.1. (a)-(y) Comparison of the membership for KPCM classified linear simulated image. The plot with red boundaries are the optimal value of fuzzy parameter m.


Linear
HyperTangent
Sigmoid
Radial
KMOD
Gaussian
Polynomial
Spectral
Multi
Sand
m=2

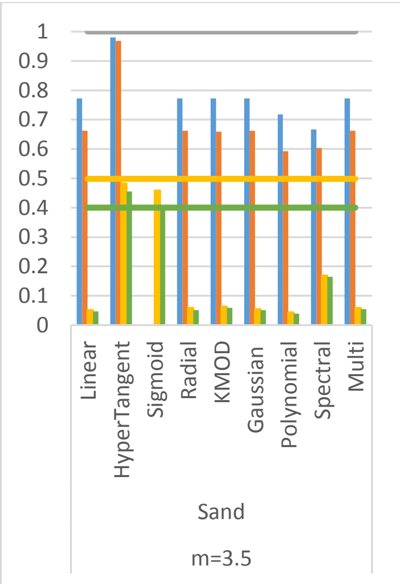
(a)


Linear
HyperTangent
Sigmoid
Radial
KMOD
Gaussian
Polynomial
Spectral
Multi
Sand
m=2.5

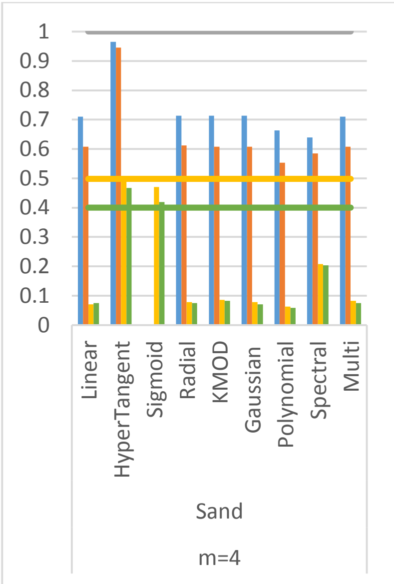
(b)


Linear
HyperTangent
Sigmoid
Radial
KMOD
Gaussian
Polynomial
Spectral
Multi
Sand
m=3

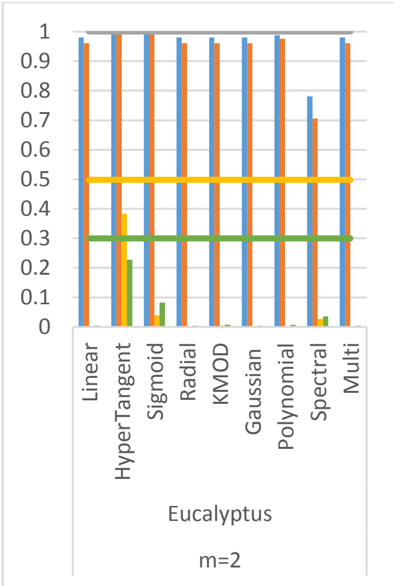
(c)


Linear
HyperTangent
Sigmoid
Radial
KMOD
Gaussian
Polynomial
Spectral
Multi
Sand
m=3.5

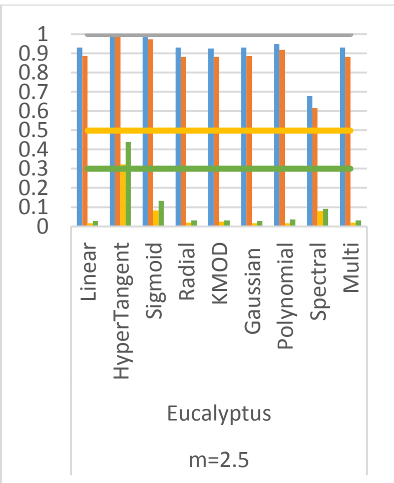
(d)


Linear
HyperTangent
Sigmoid
Radial
KMOD
Gaussian
Polynomial
Spectral
Multi
Sand
m=4

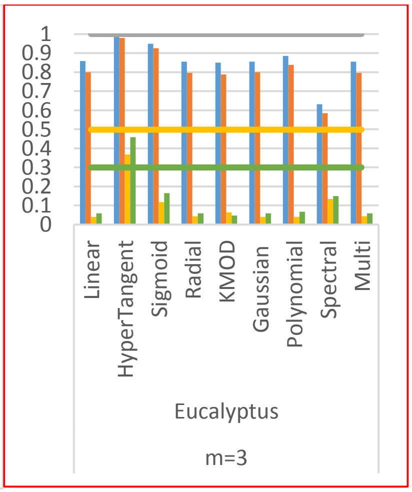
(e)


Linear
HyperTangent
Sigmoid
Radial
KMOD
Gaussian
Polynomial
Spectral
Multi
Eucalyptus
m=2

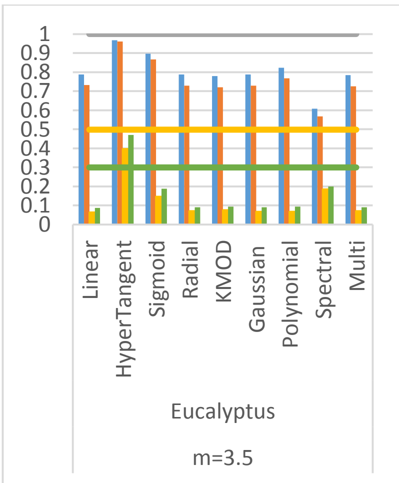
(f)


Linear
HyperTangent
Sigmoid
Radial
KMOD
Gaussian
Polynomial
Spectral
Multi
Eucalyptus
m=2.5

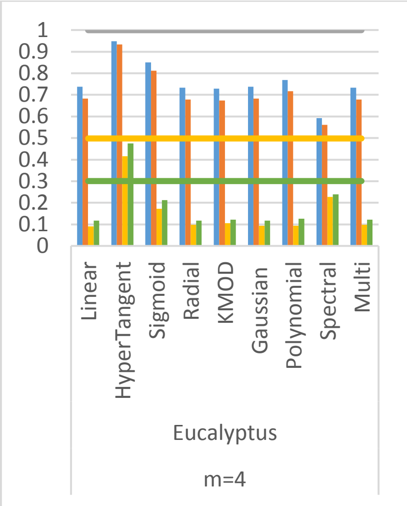
(g)


Linear
HyperTangent
Sigmoid
Radial
KMOD
Gaussian
Polynomial
Spectral
Multi
Eucalyptus
m=3

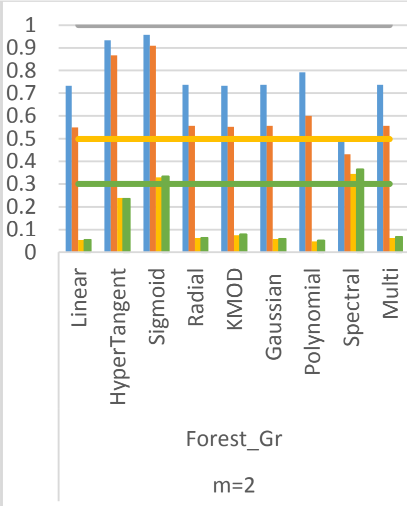
(h)


Linear
HyperTangent
Sigmoid
Radial
KMOD
Gaussian
Polynomial
Spectral
Multi
Eucalyptus
m=3.5

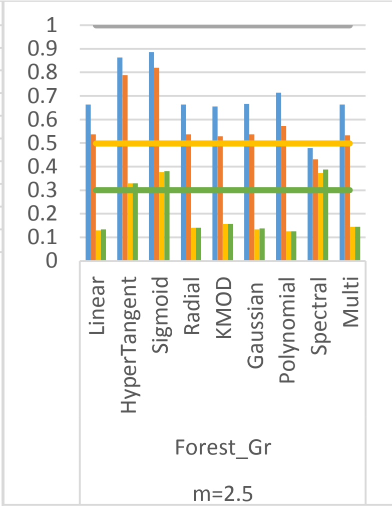
(i)


1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Linear
HyperTangent
Sigmoid
Radial
KMOD
Gaussian
Polynomial
Spectral
Multi
Eucalyptus
m=4

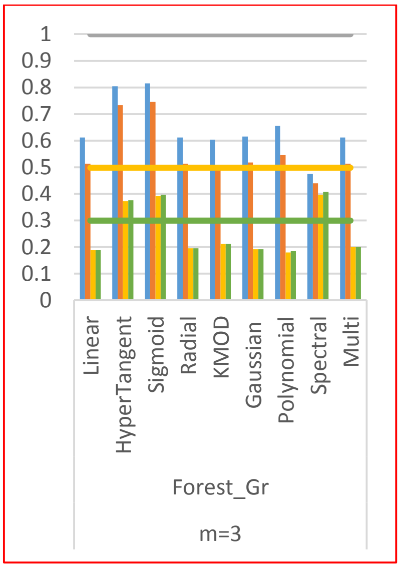
(j)


1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Linear
HyperTangent
Sigmoid
Radial
KMOD
Gaussian
Polynomial
Spectral
Multi
Forest_Gr
m=2

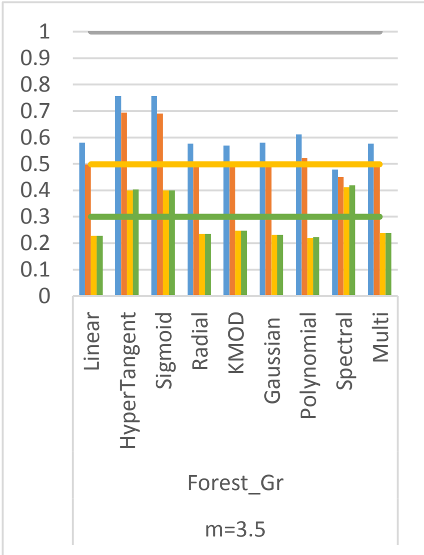
(k)


1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Linear
HyperTangent
Sigmoid
Radial
KMOD
Gaussian
Polynomial
Spectral
Multi
Forest_Gr
m=2.5

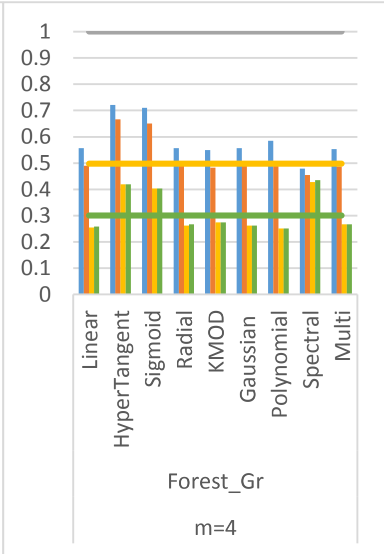
(l)


1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Linear
HyperTangent
Sigmoid
Radial
KMOD
Gaussian
Polynomial
Spectral
Multi
Forest_Gr
m=3

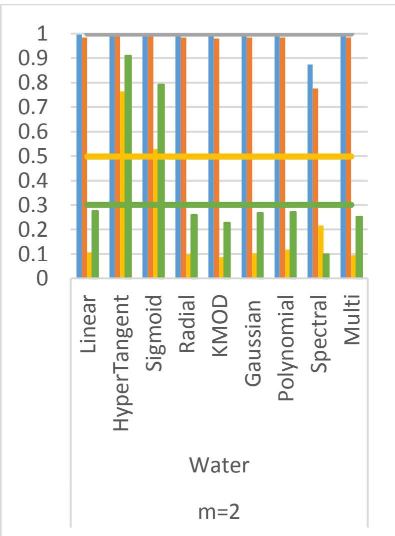
(l)


1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Linear
HyperTangent
Sigmoid
Radial
KMOD
Gaussian
Polynomial
Spectral
Multi
Forest_Gr
m=3.5

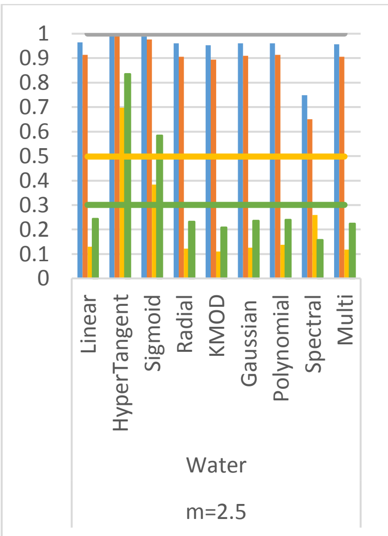
(m)


1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Linear
HyperTangent
Sigmoid
Radial
KMOD
Gaussian
Polynomial
Spectral
Multi
Forest_Gr
m=4

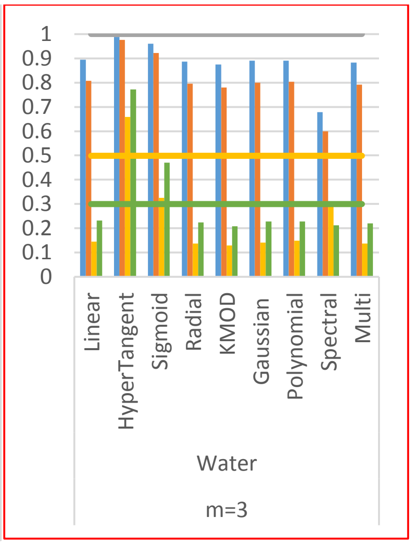
(n)


1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Linear
HyperTangent
Sigmoid
Radial
KMOD
Gaussian
Polynomial
Spectral
Multi
Water
m=2

(o)


1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Linear
HyperTangent
Sigmoid
Radial
KMOD
Gaussian
Polynomial
Spectral
Multi
Water
m=2.5

(p)


1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Linear
HyperTangent
Sigmoid
Radial
KMOD
Gaussian
Polynomial
Spectral
Multi
Water
m=3

(q)

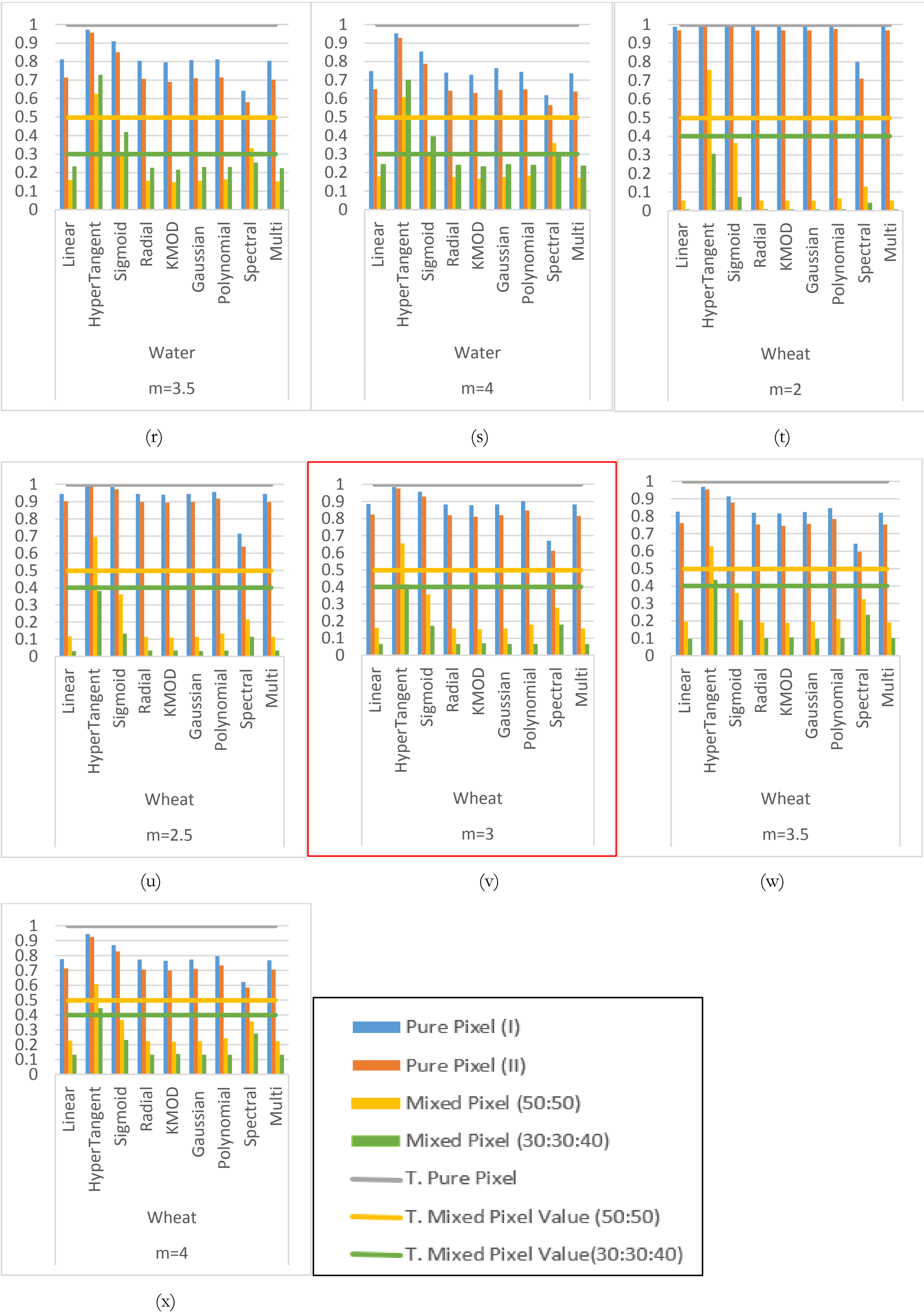


Figure 5.2. (a)-(y) Comparison of the membership for KPCM classified non-linear simulated image. The plot with red boundaries are the optimal value of fuzzy parameter m.

Fuzzy Error Matrix (FERM)

The image to image accuracy assesment techniques were used to estimate the fuzzy parameter for Landsat-8 data. The optimal value of fuzzy paramter for Formosat-2, used as reference image for parameter optimisation was estimated at 3 using simulated Formosat-2 image. Fuzzy error matrix (FERM) was used to estimate the optimal value of fuzzy parameter for Landsat-8 data with reference to Formosat-2 data. As shown in Figure 5.3, the highest overall accuracy of 98.87% was achieved with hypertangent kernel function for fuzzy parameter equals to 2.7. The second best result was observed with the sigmoid kernel with maximum overall accuracy of 92.60% at fuzzy parameter value equals to 1.5.

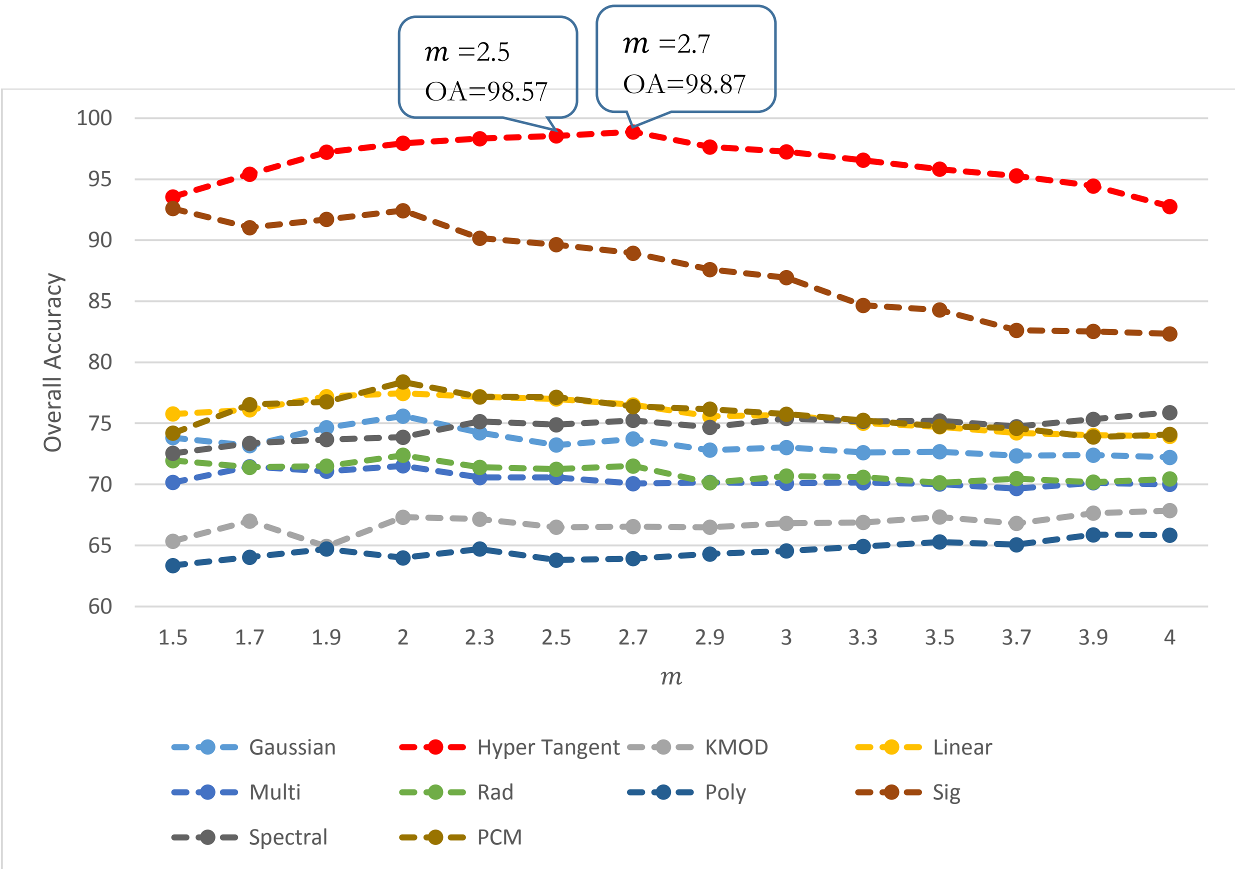


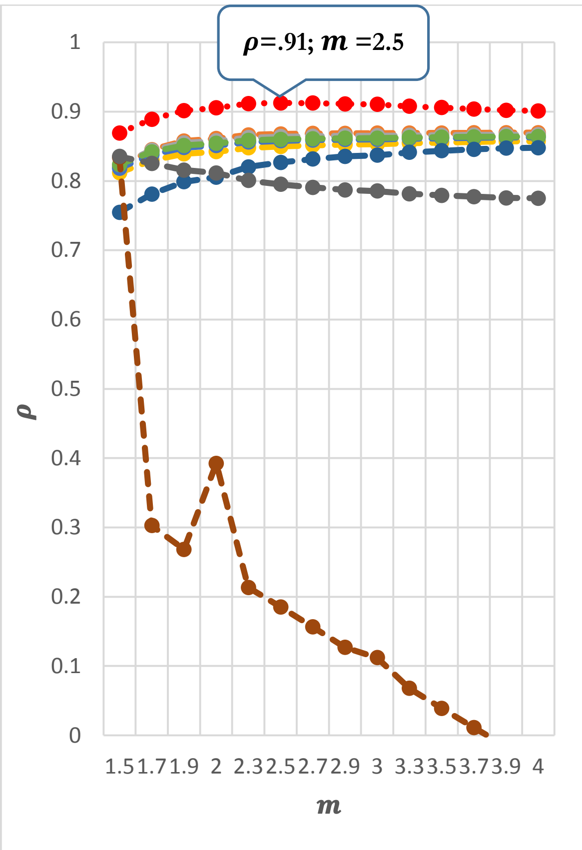
Figure 5.3. Overall accuracy of different kernels using FERM with respect to fuzzy parameter ($\boldsymbol{m}$).

Pearson Correlation Coefficient (ρ)

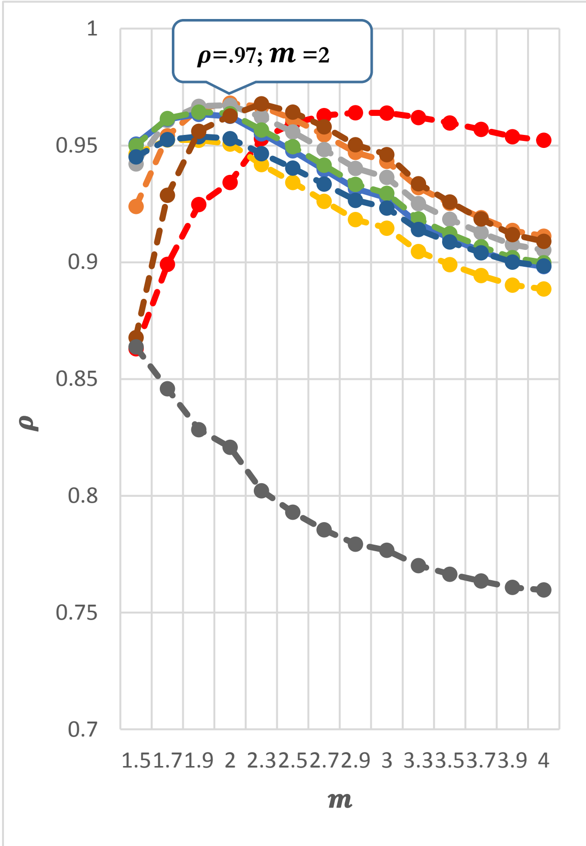
Pearson correlation was computed with reference to Formosat-2 dataset. It can be observed from Figure 5.4, that for different classes high correlation was achieved with different kernels and at different values for fuzzy parameter (m). Overall, better results were achieved with hyper tangent kernel in all classes within the range of [2.5,3] as shown in Table 5.1 and Figure 5.4. Similarly, as in case of optimizing parameter with simulated image and FERM, here also, sigmoid kernel provides better results after hyper tangent kernel function. The highest ρ of magnitude 0.97 was attained for eucalyptus class with linear KPCM classifier at fuzzy parameter value equals to 2. The lowest ρ of magnitude 0.79 was calculated for water class with sigmoid-KPCM classifier at fuzzy parameter value equals to 2.3.

Table 5.1. The value of ρ for all class with the best performing kernel at an optimal value of m. It also shows the highest ρ value for hyper tangent kernel and corresponding value of fuzzy parameter.

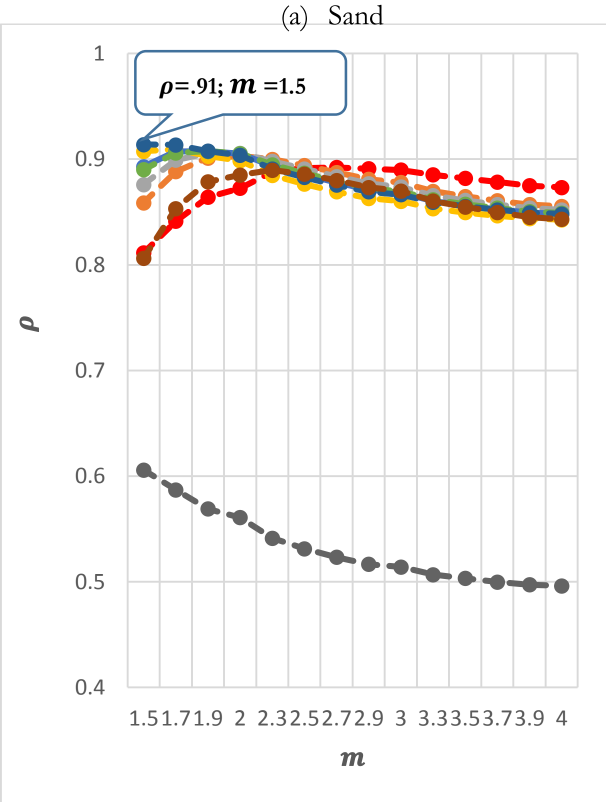
	Class – with the best performing Kernel	**Max. corelation value**	m	**ρ with Hyper Tangent Kernel**
1.	Sand- **Hyper Tangent**	**0.91**	2.5	**0.91 (m=2.5)**
2.	Eucalyptus- Linear	**0.97**	2	**0.96 (m=2.9)**
3.	Forest_Gr- Polynomial	**0.91**	1.5	**0.89 (m=2.7)**
4.	Water- Sigmoid	**0.79**	2.3	**0.73 (m=2.7)**
5.	Wheat- **HyperTangent**	**0.94**	2.9	**0.94 (m=2.9)**



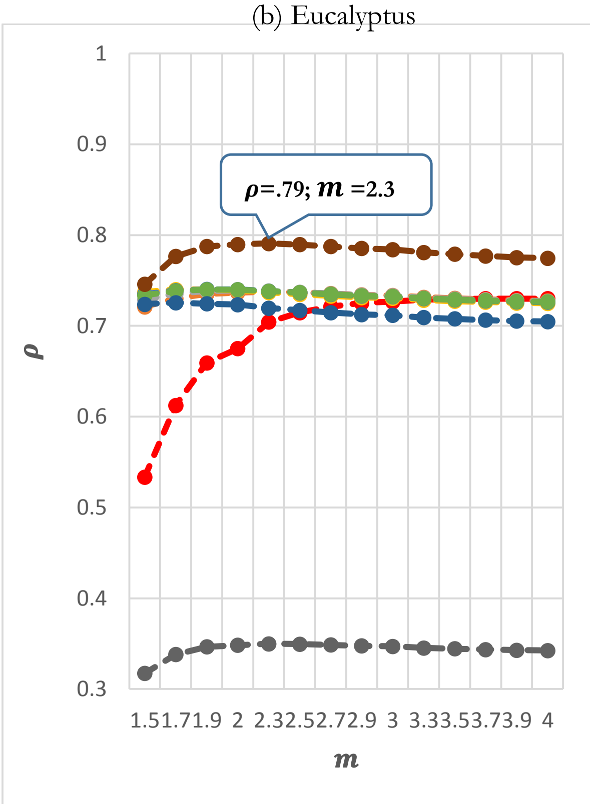
(a) Sand



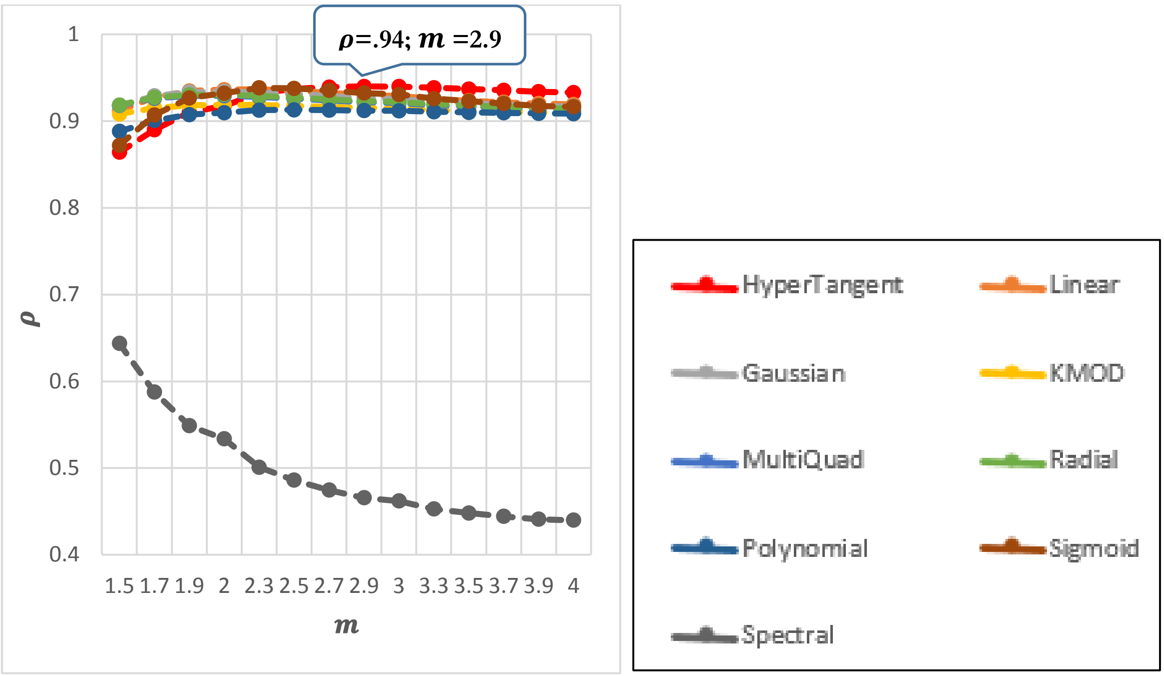
(b) Eucalyptus



(c) Forest_Gr



(d) Water



(e) Wheat

Figure 5.4. (a-e) Pearson correlation coefficient ($\boldsymbol{\rho}$), for different kernel functions, with respect to fuzzy parameter ($\boldsymbol{m}$)

5.1.2. Optimization of weight factor (λ) for composite kernel

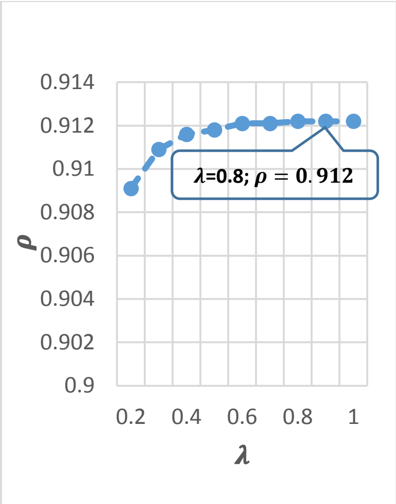
The weight factor (λ) for composite kernel was optimized using RMSE, Pearson correlation coefficient ($\boldsymbol{\rho}$) and FERM. The weight factor in composite kernel was optimized within the range of [0.1,0.9]. The best performing hyper tangent kernel and sigmoid kernel were combined together as given in equation (5.1). In equation 5.1, the composite kernel function is implemented on the input feature vector $\boldsymbol{x}, \boldsymbol{x_i}$, combining the best performing kernels implemented with their optimal parameter values.

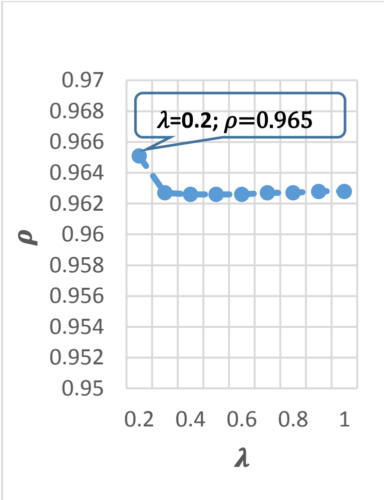
$$K(x, x_i) = \lambda \times HyperTangent(x, x_i) + (1 - \lambda) \times Sigmoid(x, x_i) \qquad (5.1)$$

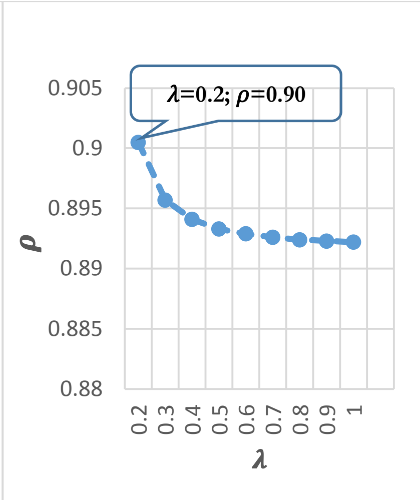
Pearson correlation coefficient

The value of weight factor in composite kernel was optimized with different values of fuzzy parameter. The optimal value of weight factor was identified at 2.7 fuzzy parameter value. From Figure 5.5, it can be observed that for different class, the best value for $\boldsymbol{\rho}$ was obtained at different values of weight factor. Table 5.2 shows the best value of $\boldsymbol{\rho}$ for different classes and the corresponding optimal values of λ. The maximum

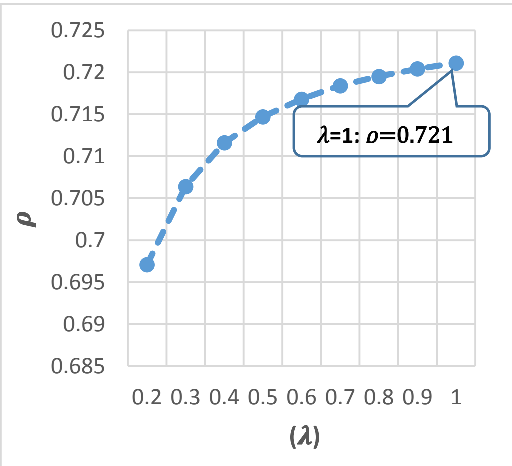
$\boldsymbol{\rho}$ of 0.965 was achieved for eucalyptus class at $\boldsymbol{\lambda}$ equals to 0.2 and lowest $\boldsymbol{\rho}$ value was attained for water of magnitude 0.72 at $\boldsymbol{\lambda}$ which equals to 1.

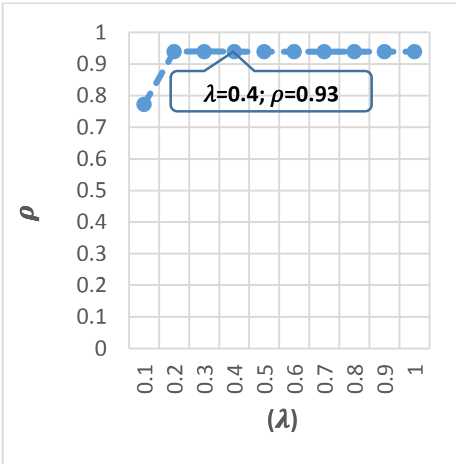






(a) Sand (b) Eucalyptus (c) Forest_Gr





(d) Water (e)Wheat

Figure 5.5. (a-e) Pearson correlation coefficient ($\boldsymbol{\rho}$) for different kernel functions with respect to weight constant ($\boldsymbol{\lambda}$) for each class.

Table 5.2. The maximum value of correlation coefficient ($\boldsymbol{\rho}$) and corresponding fuzzy parameter ($\boldsymbol{m}$) for different classes using composite kernel.

	Class	$\boldsymbol{\rho}$	$\boldsymbol{\lambda}$
1.	Sand	0.91	**0.8 - 1**
2.	Eucalyptus	0.96	**0.2**
3.	Forest_Gr	0.90	**0.2**
4.	Water	0.72	**1**
5.	Wheat	0.94	**0.2- 1**

Root Mean Square Error (RMSE)

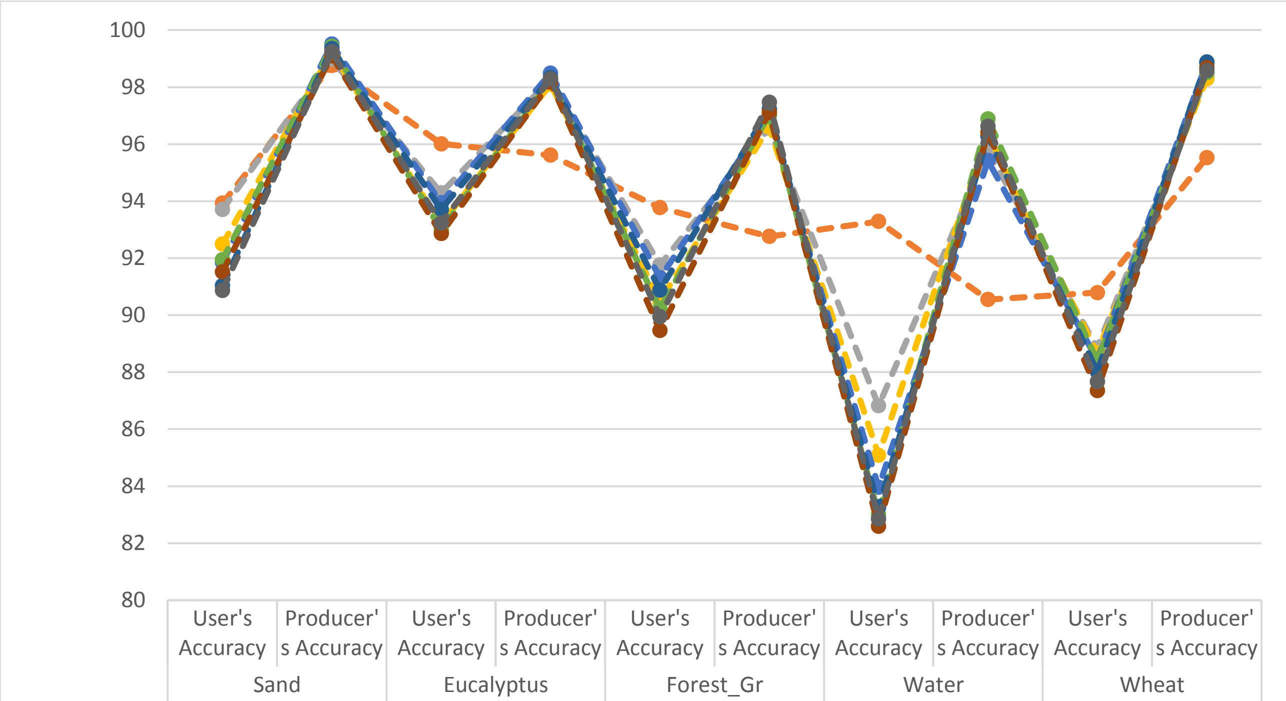
Further the value of λ was optimized using RMSE for composite kernel. The per class RMSE value was lowest at λ value (0.2) for class Sand, Eucalyptus, Forest_Gr and Wheat. The global RMSE was lowest with value being 0.223 at λ as 0.2 and then increases as the share of hyper tangent kernel increases in the composite kernel (Table 5.3). The value of RMSE was highest when the composite kernel had 90% share for sigmoid kernel. The lowest class based RMSE value of 0.0648 was achieved for eucalyptus at λ value of 0.2 and highest for Water with value 0.5556 at λ value of 0.1.

Table 5.3. RMSE value for different classes using composite kernel

Lambda	Sand	Eucalyptus	Forest_Gr	Water	Wheat	Global RMSE
0.1	0.5099	0.2286	0.3415	0.5556	0.3632	0.9325
0.2	0.0725	0.0648	0.1077	0.126	0.1028	0.2180
0.3	0.0756	0.0674	0.111	0.1257	0.1064	0.2230
0.4	0.0788	0.0693	0.1142	0.1323	0.1088	0.2311
0.5	0.0804	0.0703	0.1161	0.1361	0.1102	0.2357
0.6	0.0812	0.071	0.1172	0.1382	0.1111	0.2384
0.7	0.0817	0.0713	0.1179	0.1396	0.1116	0.2401
0.8	0.0819	0.0715	0.1183	0.1405	0.1119	0.2411
0.9	0.0819	0.0716	0.1187	0.1412	0.1121	0.5257

FERM

The λ value was again optimized using FERM for composite kernel. The class based accuracy was analysed for each class and it was observed that maximum value of user's accuracy in all classes were obtained for composite kernel (hyper tangent-sigmoid) at λ=0.2 (Figure 5.6). Further, it was observed that maximum producer's accuracy in Eucalyptus (98.49%), Sand (99.51%) and Wheat (98.88%) class were obtained for composite kernel at λ=0.5, and for Forest_Gr (97.47%) and Water (96.88%) at λ=0.9 and λ=0.6 respectively. The value of overall accuracy increases as the value of λ tends to 1 (98% at λ=0.9), that is when the composite kernel is composed of only hyper tangent kernel.



	Sand User's Accuracy	Sand Producer's Accuracy	Eucalyptus User's Accuracy	Eucalyptus Producer's Accuracy	Forest_Gr User's Accuracy	Forest_Gr Producer's Accuracy	Water User's Accuracy	Water Producer's Accuracy	Wheat User's Accuracy	Wheat Producer's Accuracy
0.2	93.93	98.76	96.01	95.61	93.77	92.76	93.29	90.55	90.79	95.52
0.3	93.71	98.98	94.29	98.28	91.76	96.56	86.82	95.76	88.85	98.38
0.4	92.49	99.23	93.2	98.1	90.5	96.59	85.09	96.34	88.68	98.29
0.5	91.85	99.51	93.94	98.49	91.31	97.01	83.96	95.42	88.44	98.84
0.6	91.94	99.43	92.99	98.25	90.23	97.03	83.01	96.88	88.39	98.51
0.7	91.04	99.35	93.71	98.33	90.88	97.23	83.25	96.28	88.02	98.88
0.8	91.52	99.1	92.86	98.17	89.46	97.11	82.59	96.42	87.35	98.68
0.9	90.87	99.24	93.23	98.29	89.95	97.47	82.85	96.63	87.67	98.58

Figure 5.6. User's and Producer's accuracy obtained through FERM (Fuzzy error matrix) for different values of weight constant (λ) for composite kernel.

So, the optimal value for fuzzy parameter was identified at 2.7. The value of fuzzy parameter was selected based on the simulated image technique, high Pearson correlation value (Table 5.1) and high overall accuracy shown in FERM (Figure 5.3). The optimal value for λ was identified at 0.5 because of its overall high correlation (Figure 5.5), low RMSE (Table 5.3) and high producer accuracy (Figure 5.6) in all classes. The hyper tangent kernel was identified as the best performing kernel for all classes.

5.2. Accuracy assessment

5.2.1. Accuracy assessment based on soft classification

The optimized values of parameters were used with kernel based PCM classifier to classify Landsat-8 data. The soft outputs from the classifier were then analysed with various accuracy assessment methods to quantify the accuracy of classification. There are various techniques mentioned in section 3.5 (chapter 3) for accuracy assessment of fuzzy output. The accuracy of classified Landsat-8 image was computed with respect to Formosat-2 image.

Soft classified output using KPCM classifier

Figure 5.7 shows the soft classified output for Landsat-8 dataset using KPCM classifier at the optimal value for fuzzy parameter. Each fractional image represents the membership value of a pixel in a particular class. All five classes were visible in their respective fractional images. The dark pixels in the fractional image of particular class shows lower degree of belongingness for the class and the brighter pixels shows higher degree of belongingness. Eucalyptus, water and sand, classes were clearly visible in their corresponding fractional images whereas in Forest_Gr class, the grassland class which was merged with forest was not well classified. Also, due to very closely located mean values of wheat and the Forest_Gr class, the wheat class was given high membership value in the fractional image for Forest_Gr class and vice-versa.

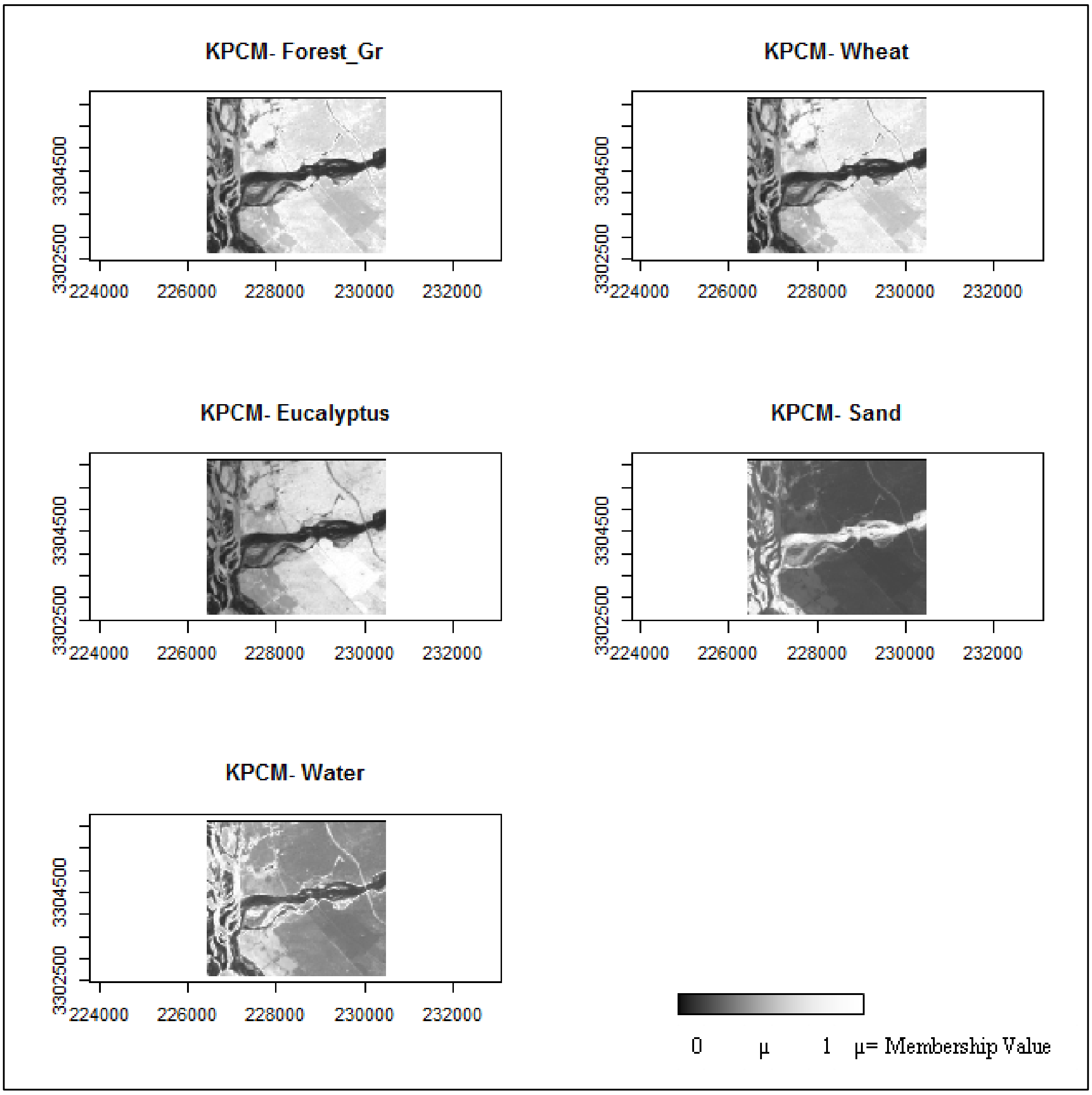


Figure 5.7. Fractional output for KPCM classification using hyper-tangent kernel.

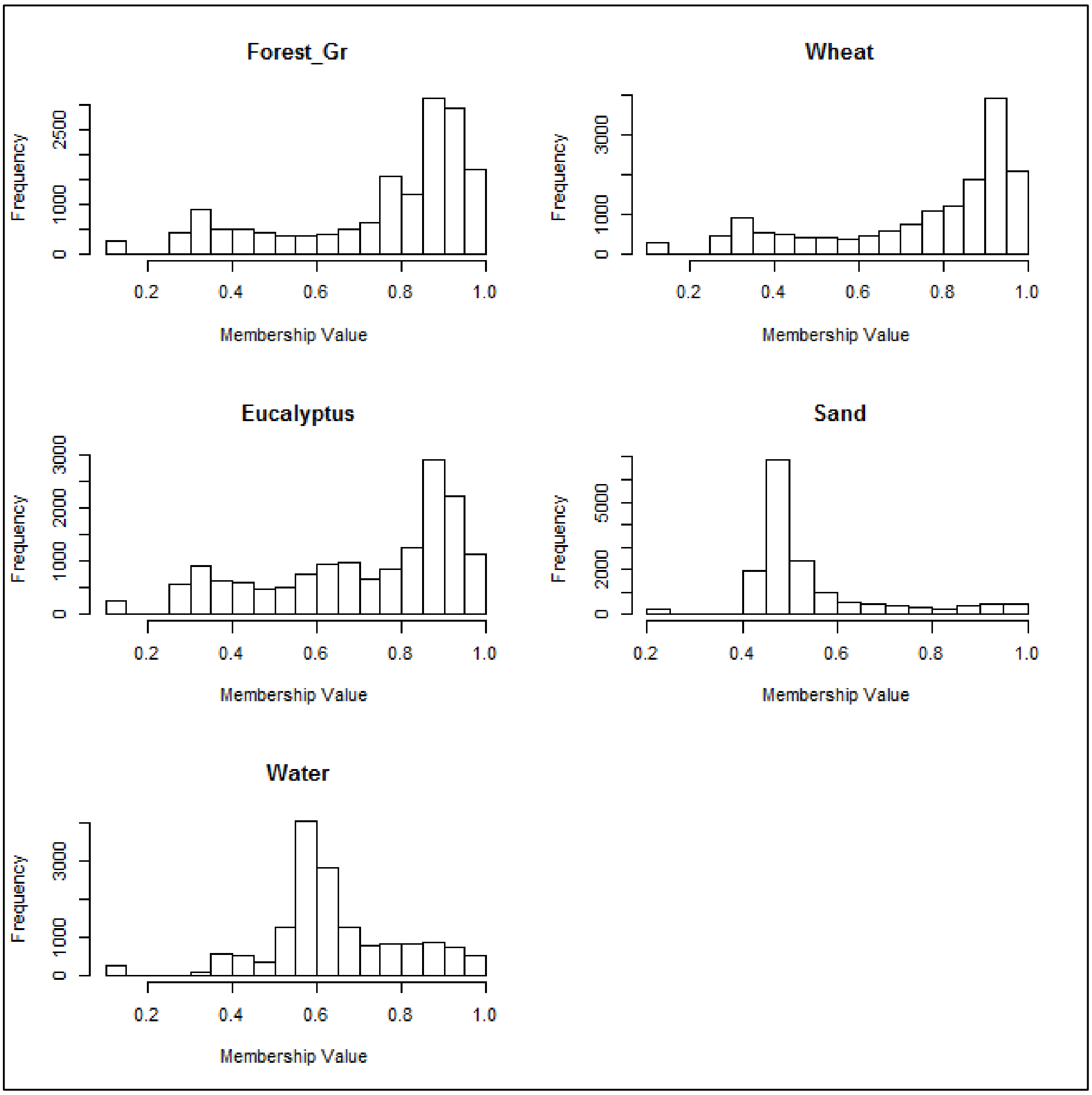


Figure 5.8. Membership value for KPCM classification using hyper tangent kernel.

Figure 5.8 shows the distribution of membership values in the fractional output for Landsat-8 dataset using KPCM classifier. Each histogram plot shows the distribution of membership values of a pixel within the range of [0,1]. In all classes, the membership value of feature vectors tends to be maximum. The maximum number of pixels are having membership value in range of [0.8,1] for Forest_Gr, Wheat and Eucalyptus class. Most of the pixels in sand class are having membership value between [0.4,0.6]. The membership value of most pixels in water class is in the range of [0.5,0.7].

Correlation (ρ), RMSE and FERM results using hyper-tangent KPCM classifier

To further assess the accuracy of the classified results for Landsat-8 data with hyper-tangent kernel based PCM at fuzzy parameter equals to 2.7, the Pearson correlation coefficient, RMSE, FERM and entropy were calculated. Table 5.4 shows the ρ and RMSE value achieved for all classes. The highest value for ρ was obtained for eucalyptus class with magnitude being to 0.95 and the water class shows the lowest value for ρ, the value being 0.72. Correspondingly, the RMSE value is lowest for eucalyptus class (0.07) and highest for water class (0.14). The results of KPCM classifier were further analysed using FERM (Table 5.5). The overall accuracy of 98.37% was achieved with FERM. The highest user's accuracy was attained for eucalyptus class (93.26%) and lowest user's accuracy was obtained for water class (82.68%). The results obtained from supervised KPCM classification of Landsat-8 data at $m = 2.7$ were further evaluated based on the entropy values. The entropy value for hyper-tangent KPCM classification was 0.4758.

Table 5.4. RMSE and correlation value using hyper-tangent kernel based PCM classifier.

Class	**Correlation (ρ)**	**RMSE**
Sand	**0.91**	**0.08**
Eucalyptus	**0.95**	**0.07**
Forest_Gr	**0.89**	**0.11**
Water	**0.72**	**0.14**
Wheat	**0.94**	**0.11**

Table 5.5. FERM based accuracy assessment for classified result of Landsat-8 dataset using hyper-tangent kernel based PCM.

Accuracy Assessment methods	**FERM (%)**
Sand	
User's Accuracy	**92.08**
Producer's Accuracy	**99.46**
Eucalyptus	

User's Accuracy	**93.26**
Producer's Accuracy	**98.61**
Forest_Gr	
User's Accuracy	**90.24**
Producer's Accuracy	**97.42**
Water	
User's Accuracy	**82.68**
Producer's Accuracy	**97.53**
Wheat	
User's Accuracy	**87.59**
Producer's Accuracy	**99.01**
Average User's Accuracy	**89.17**
Average Producer's Accuracy	**98.41**
Fuzzy overall Accuracy	**98.37**
Entropy	**0.4758**

5.2.2. Accuracy assessment based on hard classification

By hardening the output from the KPCM classifier the membership of a pixel was forced to belong to a single cluster. This provides the crisp interpretation of the soft output from the KPCM (hyper tangent) classifier. The hard classification of a fuzzy soft output provides a general idea about the magnitude of membership value in different classes. The Forest_Gr was having lowest producer accuracy of 9.045%. The overall accuracy with confusion matrix was 58.18% which is quite low as demonstrated with soft classifier (Table 5.6)

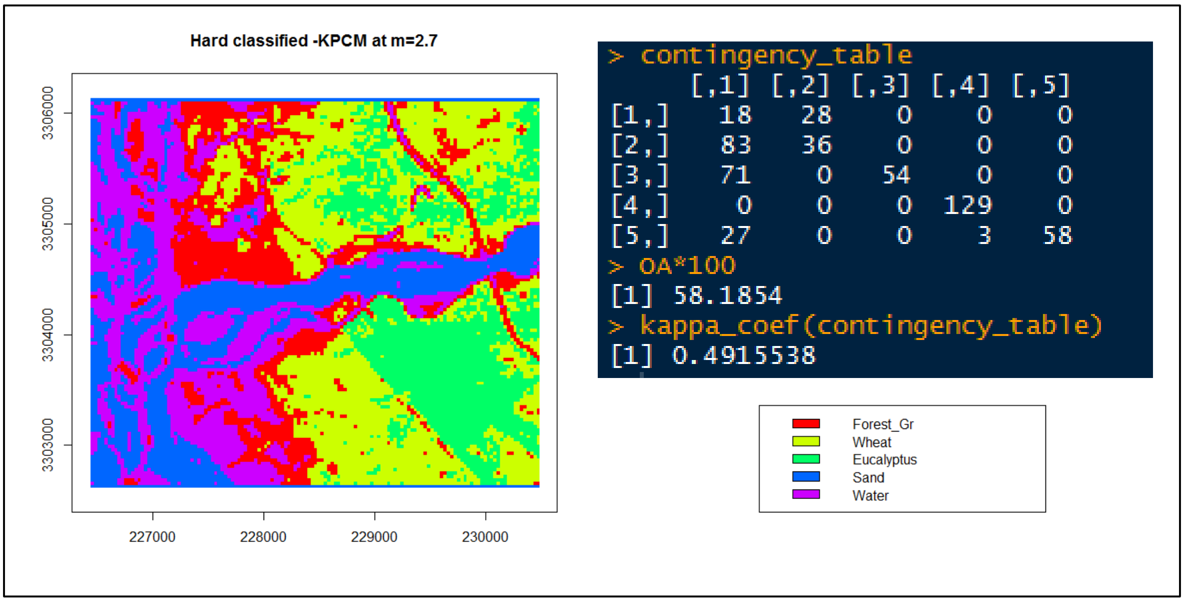


Figure 5.9. Hard classification of the fractional image from hyper tangent kernel based possibilistic c-means classifier.

Table 5.6. Error matrix for the hard classified KPCM classification.

Producer's Accuracy (omission error)		User's Accuracy (commission error)	
Forest_Gr=$\frac{18}{199}$=**9.045%**	**90.95% (omission error)**	Forest_Gr=$\frac{18}{46}$= **39.13%**	**60.87%(commission error)**
Wheat=$\frac{36}{64}$= **56.25%**	**43.75%(omission error)**	Wheat=$\frac{36}{119}$= **30.25%**	**69.75%(commission error)**
Eucalyptus=$\frac{54}{54}$=**100%**	**0%(omission error)**	Eucalyptus =$\frac{54}{125}$= **43.2%**	**56.80%(commission error)**
Sand=$\frac{129}{132}$= **97.72%**	**2.28% (omission error)**	Sand=$\frac{129}{129}$= **100%**	**0%(commission error)**
Water=$\frac{58}{58}$= **100%**	**0% (omission error)**	Water =$\frac{58}{88}$= **65.90%**	**34.10%(commission error)**

5.3. Comparison of PCM and KPCM

To identify the advantage of KPCM classifier over PCM classifier, the classified outputs of hyper-tangent KPCM classifier were compared with outputs from PCM classifier. For this, accuracy assessment method like entropy, simulated image technique, RMSE, correlation cofficient and FERM were used for comparison. The value of the fuzzy parameter in PCM was optimised at 2 for Landsat-8 data, it was optimized based on high correlation value, low RMSE and high overall accuracy with FERM for all classes. It should be noted that the result of PCM classifier were similar to linear-KPCM classifier (Figure 5.3).

Simulated image technique

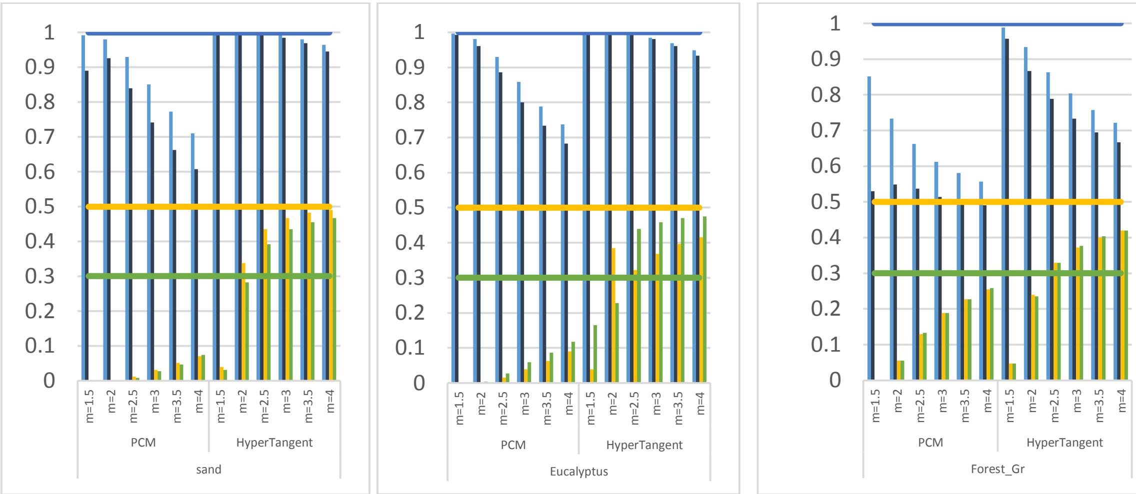
In figure 5.10, the classified results from PCM and KPCM classifier of simulated Landsat-8 data for different value of fuzzy parameter were evaluated. In figure 5.10 the membership value of pure pixel is shown as Pure Pixel (I) and the variation present within the class is shown as Pure Pixel (II). The mixed pixels were simulated with two variations, one with composition of 50:50 shown as Mixed Pixel (50:50) formed by mixture of two different classes and other with composition of 30:30:40 shown as Mixed Pixel (30:30:40) formed by mixture of three different classes. In Figure 5.10 the three different horizontal lines grey, yellow and green colour representing the target values for a pure pixel, mixed pixel (50:50) and mixed pixel (30:30:40) respectively. The target membership value expected from the pixel with full belongingness to a class must be close to 1, the target membership value of .50, .40 and .30 is expected from the pixel with 50%, 40% and 30% belongingness for a class respectively.

Here the PCM and hyper tangent-KPCM classifiers at different value of fuzzy parameter (m) were compared using simulated Landsat-8 data. From figure (5.10) it can be observed that the hyper-tangent KPCM classifier shows better classification results in comparison to PCM classifier. Also, the influence of the fuzzy parameter on the classification results of hyper-tangent KPCM classifier was low as compared to PCM classifier. In Table 5.7, value of the classified pixels for optimal value of fuzzy parameter in PCM and KPCM classifier are compared and it can be observed that the value of the pure pixel and mixed pixel are more near to the target values (shown as T.Pure Pixel, T. Mixed Pixel (50:50), T. Mixed Pixel (30:30:40)) for KPCM classification as compared to PCM classification.

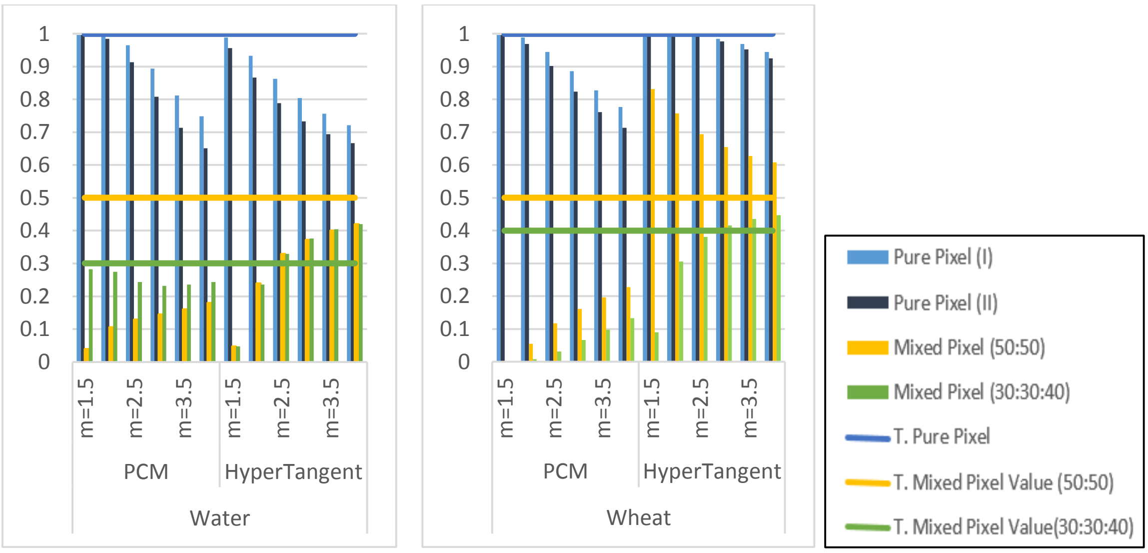
Table 5.7. Comparison between the pixel values for classified output for simulated Landsat-8 data for optimal m.

Classifier	m	Class	Pure Pixel (I)	Pure Pixel (II)	T. Pure Pixel	Mixed Pixel (50:50)	T. Mixed Pixel (50:50)	Mixed Pixel (30:30:40)	T. Mixed Pixel (30:30:40)
PCM		Sand	0.9803	0.9254	**1**	0	**0.5**	0	**0.3**
		Eucalyptus	0.9803	0.9607	**1**	0	**0.5**	0.0039	**0.3**
	2	Forest_Gr	0.7333	0.5490	**1**	0.0549	**0.5**	0.0549	**0.3**

		Water	0.9960	0.9843	**1**	0.1058	**0.5**	0.2745	**0.3**
		Wheat	0.9882	0.9686	**1**	0.0549	**0.5**	0.0078	**0.4**
KPCM		Sand	0.9960	0.9960	**1**	0.4352	**0.5**	0.3921	**0.3**
		Eucalyptus	0.9960	0.9921	**1**	0.3215	**0.5**	0.4392	**0.3**
		Forest_Gr	0.8627	0.7882	**1**	0.3294	**0.5**	0.3294	**0.3**
		Water	0.8627	0.7882	**1**	0.3294	**0.5**	0.3294	**0.3**
	2.7	Wheat	0.9960	0.9921	**1**	0.6941	**0.5**	0.3803	**0.4**



(a) (b) (c)



(d) (e)

Figure 5.10. (a-e) Comparison of PCM and KPCM classification for simulated data set with different fuzzy parameter (m) values.

Soft classified output with PCM classifier

Figure 5.11 shows the classified output for PCM classifier when applied to Landsat-8 dataset at the optimal value of fuzzifier parameter as 2. Each fractional image represented the membership value of a pixel in a particular class. The eucalyptus, water and sand, these classes are clearly visible in their corresponding fractional images. But in the Forest_Gr class the grassland class which was merged with forest was not well classified. Also due to closely located mean values of wheat and Forest_Gr class, the wheat class was given high membership value in the fractional image for Forest_Gr and vice-versa.

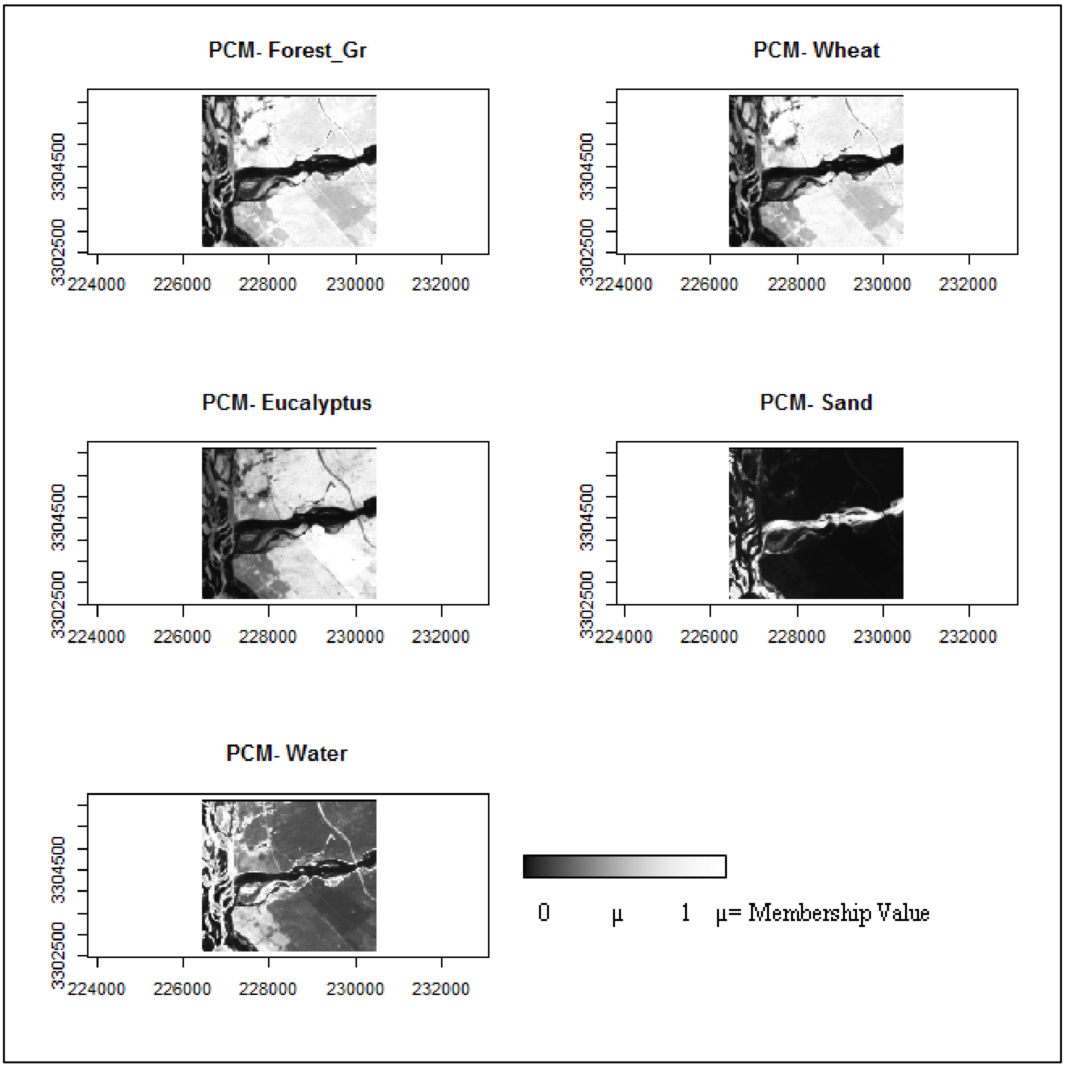


Figure 5.11. Fractional output of PCM classification on Formosat-2 (non-linear) data set.

On visual comparison between the output of PCM and KPCM classifier, it can be seen that the classes like sand, eucalyptus, water are visually more evident in their corresponding fractional images from KPCM classifier though on first look, the PCM seems to be more promising because of the contrast present in the fractional image due to which the feature seems to be revealing. This higher contrast is due to the lower value of fuzzy parameter in PCM classifier.

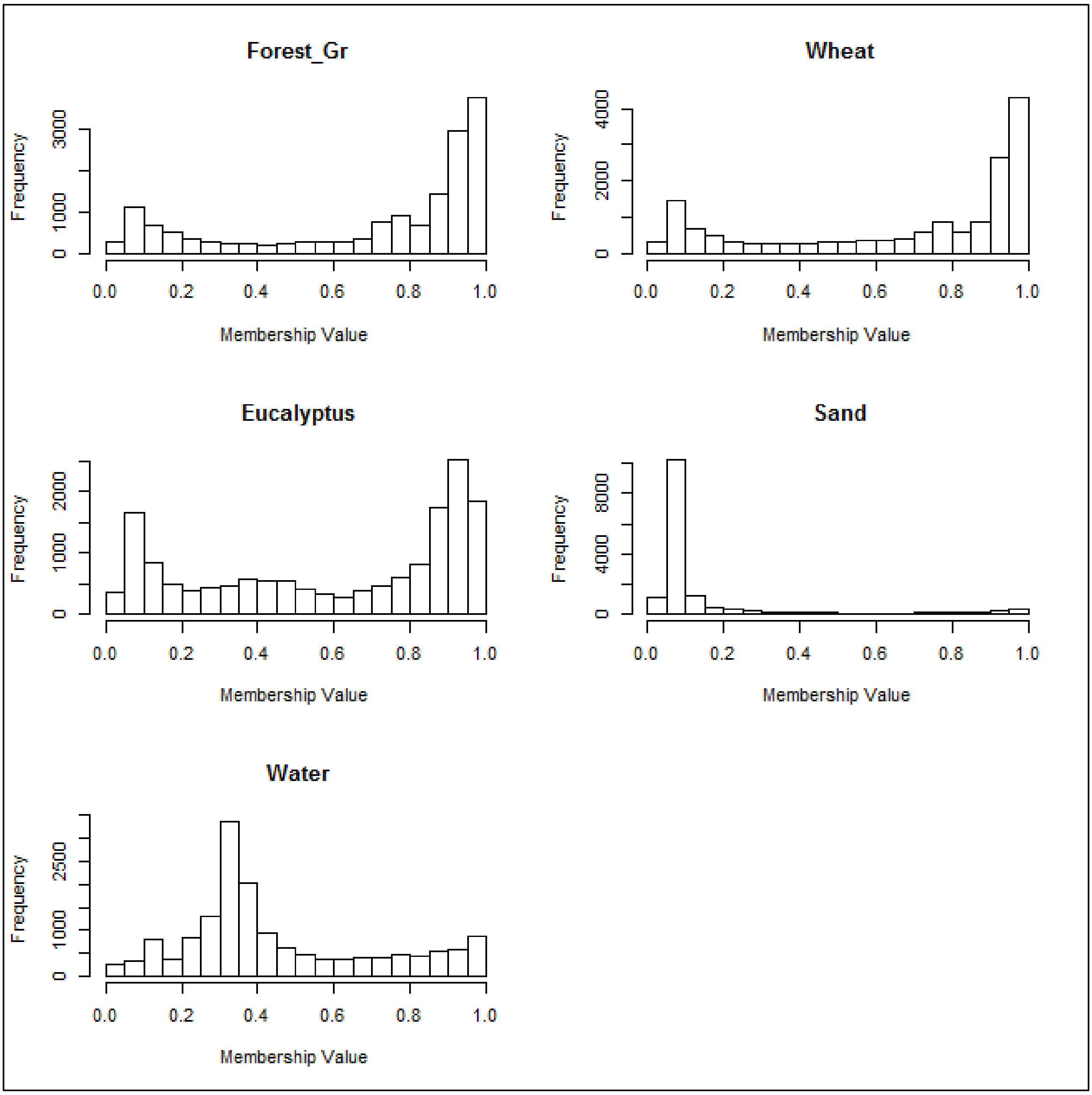


Figure 5.12. Membership value for different classes in the output from PCM classifier.

Figure 5.12 depicts the distribution of membership value in the fractional output for each class using PCM classifier. Each histogram shows the distribution of membership value within the range of [0,1]. In Table 5.8, the range of the membership value for maximum number of pixel in each class is compared for PCM and KPCM classifier. Also, it can be observed that the membership value increases for pixels within each class, and randomness decreases in KPCM classification outputs as compared to PCM classification.

Table 5.8. Comparison of the range of membership value where maximum pixels lies within each class.

Classifier	m	Class	Range of membership value
PCM	2	Sand	0-0.1
		Eucalyptus	0-1
		Forest_Gr	0.8-1
		Water	0.2-0.4
		Wheat	0.8-1
KPCM	2.7	Sand	0.4-0.6
		Eucalyptus	0.8-1
		Forest_Gr	0.8-1
		Water	0.5-0.7
		Wheat	0.8-1

Pearson correlation coefficient (ρ), RMSE and FERM results using PCM classifier

To evaluate the accuracy of PCM classification results the ρ, RMSE, FERM and entropy were used. Table 5.9 shows the values for ρ and RMSE for PCM classification. The highest value for ρ was obtained for eucalyptus class with magnitude being equal to 0.97 and the water class shows the lowest value for ρ as 0.73. The correlation values for each class in KPCM outputs (except for sand class) is similar as calculated for PCM output. The correlation value for the sand class in PCM outputs was 0.86 whereas it was 0.91 for the sand class in KPCM classification. The RMSE value was lowest for eucalyptus class (0.14) and highest for sand class (0.37). The RMSE values are higher for the outputs of PCM classifier as compared to the KPCM classifier. For PCM classification output, the overall accuracy of 78.38% was obtained with FERM which is less than the overall accuracy of 98.37% achieved with KPCM classification (Table 5.10). The entropy value of 0.5430 for PCM classifier is also significantly large than the hyper-tangent KPCM classification (0.4758). PCM classification results showed better value for average users accuracy of 91.42% compare to KPCM classifier with value of 89.17%. This shows higher quality of classification results of KPCM classifier in comparison to PCM classifier.

Table 5.9. RMSE and correlation value for classified results of PCM classifier.

Class	Correlation (ρ)	RMSE
Sand	**0.86**	**0.37**
Eucalyptus	**0.97**	**0.14**
Forest_Gr	**0.90**	**0.14**
Water	**0.73**	**0.20**
Wheat	**0.93**	**0.16**

Table 5.10. Result of FERM for classified results of PCM classifier

Accuracy Assessment methods	FERM (%)
Sand	
User's Accuracy	**91.19**
Producer's Accuracy	**29.70**
Eucalyptus	
User's Accuracy	**96.24**
Producer's Accuracy	**87.84**
Forest_Gr	
User's Accuracy	**89.49**
Producer's Accuracy	**93.04**
Water	
User's Accuracy	**91.95**
Producer's Accuracy	**76.23**

Wheat	
User's Accuracy	**88.21**
Producer's Accuracy	**91.63**
Average User's Accuracy	**91.42**
Average Producer's Accuracy	**75.69**
Fuzzy overall Accuracy	**78.38**
Entropy	**0.5430**

5.4. Results of kernel based PCM classifier using composite kernel

The composite kernels were developed to combine the capabilities of different kernel and to enhance the efficiency of individual kernel. For composite kernel, the hyper tangent kernel and the sigmoid kernel are preferred choice because of the high overall accuracy demonstrated by them in FERM (Figure 5.3 and Figure 5.4). To combine the spectral information in a classifier, the spectral kernel was combined with the best performing hyper tangent kernel but the result obtained were not satisfactory (not even visually) so this combination was rejected and not considered for analysis.

The equation for composite kernel ($K(x, x_i)$) formed with the optimized λ value is shown in equation 5.2. Though in general, the maximum value of correlation for all classes was attained at λ being equal to 1 (Table 5.2) i.e. fully hyper Tangent Kernel. For analysis, the lambda value was selected on the basis of producer's accuracy, which was highest at 0.5 for most of the classes. Also, at λ equals to 0.5, the composite kernel contains the property of both the kernels by equal proportion.

$$K(x, x_i) = 0.5 * HyperTangent(x, x_i) + 0.5 * Sigmoid(x, x_i) \qquad (5.2)$$

Soft classified output using composite-KPCM classifier

Figure 5.13 shows the classified output for Landsat-8 dataset using KPCM classifier (hyper Tangent-Sigmoid kernel) at the optimal value for "m" being 2.7 and "λ" as .5. Each fractional image represents the membership value of a pixel in a particular class. All five classes were clearly visible in their respective fractional images.

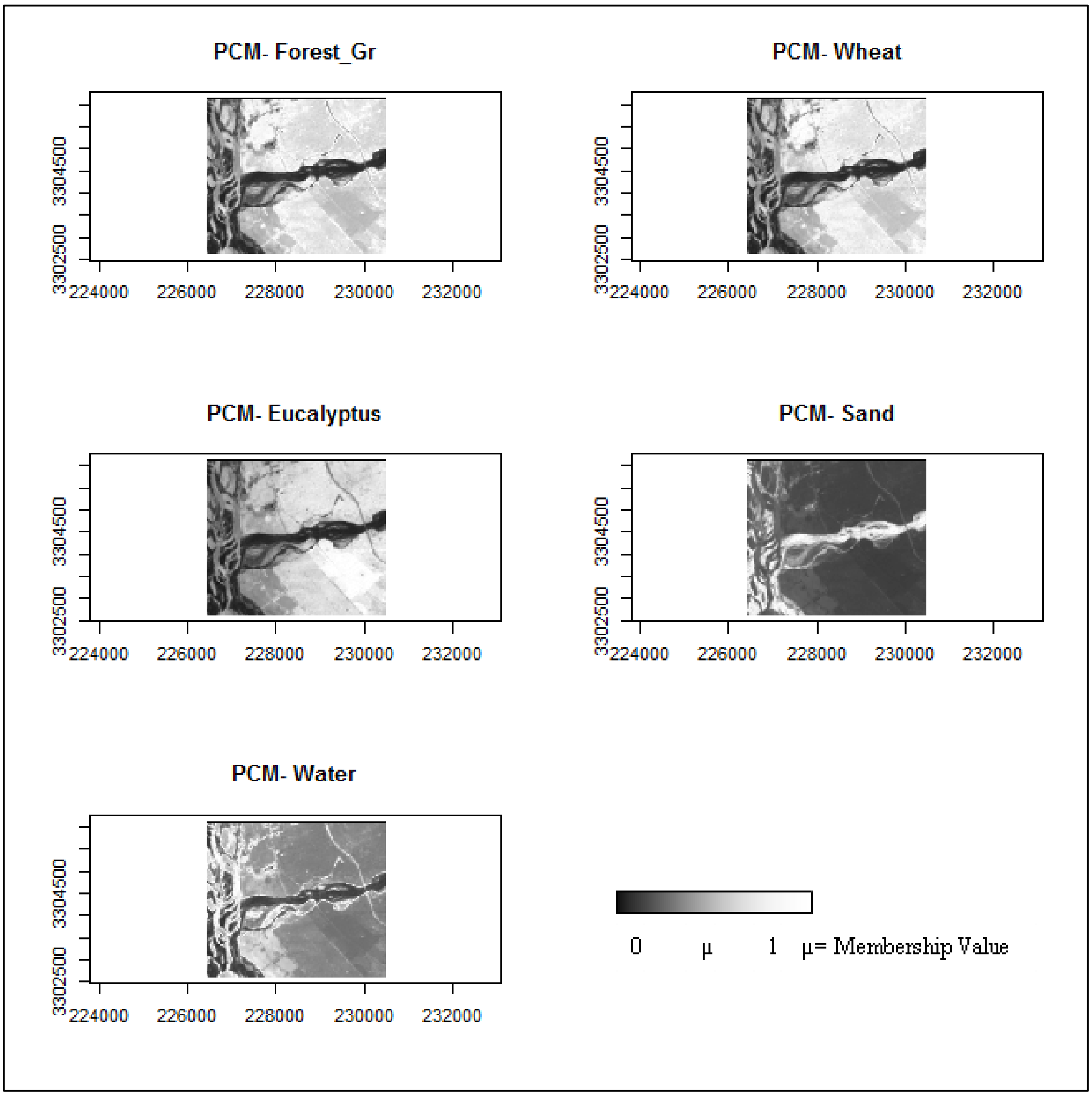


Figure 5.13. Fractional output for composite kernel based classification (hyper tangent - Sigmoid) at λ=0.5.

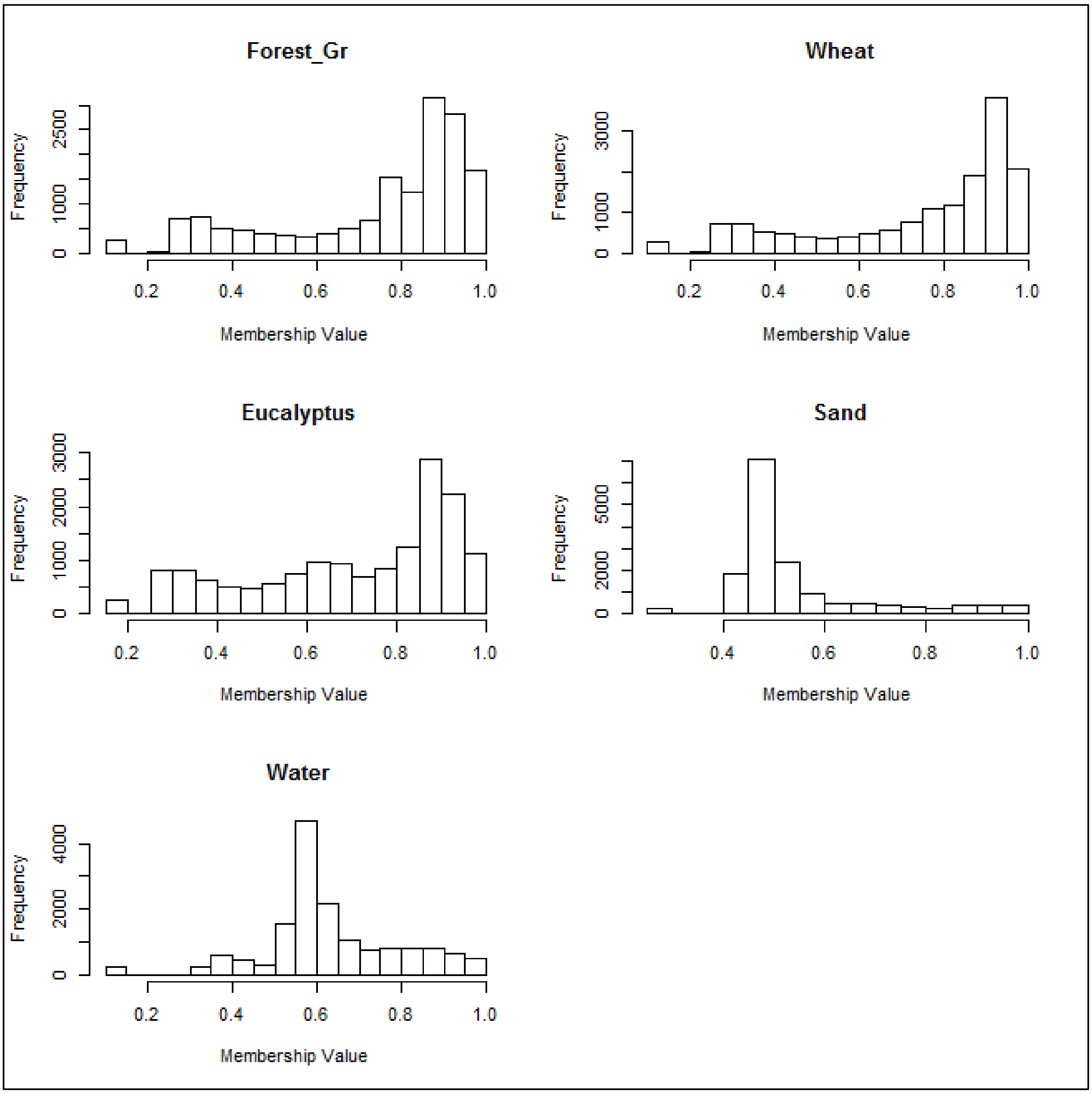


Figure 5.14. Membership value for composite kernel based classification (hyper tangent - Sigmoid) at λ=0.5.

In Figure 5.14, the histogram plots were similar as obtained from the classification with hyper tangent-KPCM classifier. The range of membership value for maximum pixels in different classes is same as with the output for KPCM classifiers.

Correlation (ρ), RMSE and FERM results for hyper tangent-sigmoid KPCM classifier

The results of hyper tangent-sigmoid KPCM classifier were further analysed on the basis of Pearson correlation coefficient, RMSE, Entropy and FERM measures. The correlation and RMSE value calculated for the outputs of composite kernel were identical to the corresponding value attained with KPCM classifier

(Table 5.11). Table 5.12 shows the FERM accuracy assessment results for the output of hyper tangent-sigmoid KPCM classifier. The overall accuracy of 97.84% and entropy as 0.48 were attained.

Table 5.11. RMSE and correlation value with composite-kernel.

Class	**Correlation (ρ)**	**RMSE**
Sand	0.91	0.08
Eucalyptus	0.96	0.07
Forest_Gr	0.89	0.11
Water	0.71	0.13
Wheat	0.93	0.11

Table 5.12. Result of FERM with composite kernel

Accuracy Assessment methods	**FERM %)**
Sand	
User's Accuracy	**91.85**
Producer's Accuracy	**99.51**
Eucalyptus	
User's Accuracy	**93.94**
Producer's Accuracy	**98.49**
Forest_Gr	
User's Accuracy	**91.31**
Producer's Accuracy	**97**
Water	
User's Accuracy	**83.96**
Producer's Accuracy	**95.42**
Wheat	

User's Accuracy	**88.44**
Producer's Accuracy	**98.84**
Average User's Accuracy	**89.90**
Average Producer's Accuracy	**97.85**
Fuzzy overall Accuracy	**97.84**
Entropy	**0.4838**

5.5. Untrained classes

The KPCM algorithm using hyper-tangent kernel function was further analysed for the effect of the presence of untrained classes during classification, because in case of FCM, it has been observed that the presence of untrained classes affects the membership value and decreases the correspondence between the estimated and the actual class composition (Byju, 2015). This happens with FCM because of its membership constraint (Foody, 2000).In supervised classification, the untrained class is generated by escaping the training of classifier for a particular class.

From Table 5.9, Table 5.10 and Table 5.11, it can be interpreted that the possibilistic membership value in KPCM are not affected by the presence of untrained classes. Pearson correlation coefficient, RMSE values, class based accuracy as well as overall accuracy remains unaffected in the presence of untrained class. These values are obtained by removing one class each time from training stage.

Table 5.13. Correlation for PCM and KPCM classification results. One untrained class was considered at a time.

Algorithm	**Untrained class**	**Correlation Coefficient**				
		Sand	**Eucalyptus**	**Forest_Gr**	**Water**	**Wheat**
PCM	**None**	**0.8684**	**0.9546**	**0.8874**	**0.7356**	**0.9325**
	Sand	**-**	**0.9646**	**0.8932**	**0.7332**	**0.9389**
	Eucalyptus	**0.8805**	**-**	**0.89019**	**0.7347**	**0.9335**
	Forest_Gr	**0.8794**	**0.9607**	**-**	**0.7353**	**0.9231**

	Water	**0.8792**	**0.9637**	**0.8915**	**-**	**0.9374**
	Wheat	**0.8795**	**0.9589**	**0.8747**	**0.7353**	**-**
KPCM (Hyper Tangent)	**None**	**0.9122**	**0.9628**	**0.8922**	**0.7211**	**0.9392**
	Sand	**-**	**0.9624**	**0.8919**	**0.7210**	**0.9391**
	Eucalyptus	**0.9122**	**-**	**0.8922**	**0.7211**	**0.9393**
	Forest_Gr	**0.9122**	**0.9629**	**-**	**0.7211**	**0.9393**
	Water	**0.9122**	**0.9629**	**0.8922**	**-**	**0.9393**
	Wheat	**0.9122**	**0.9629**	**0.8923**	**0.7211**	**-**

Table 5.14. RMSE for PCM and KPCM classification results. One untrained class was considered at a time.

Algorithm	**Untrained class**	**RMSE Error**				
		Sand	**Eucalyptus**	**Forest_Gr**	**Water**	**Wheat**
PCM	**None**	**0.3006**	**0.1795**	**0.1325**	**0.1650**	**0.1144**
	Sand	**-**	**0.0001**	**0.0010**	**0.0014**	**0.0006**
	Eucalyptus	**0.0017**	**-**	**0.0004**	**0.0012**	**0.0002**
	Forest_Gr	**0.0018**	**0.0005**	**-**	**0.0010**	**0.0006**
	Water	**0.0018**	**0.0002**	**0.0006**	**-**	**0.0004**
	Wheat	**0.0018**	**0.0006**	**0.0003**	**0.0011**	**-**
KPCM (Hyper Tangent)	**None**	**0.0819**	**0.0716**	**0.1188**	**0.1417**	**0.1122**
	Sand	**-**	**0.0013**	**0.0018**	**0.002143**	**0.0016**
	Eucalyptus	**0.0009**	**-**	**0.0018**	**0.002143**	**0.0016**
	Forest_Gr	**0.0009**	**0.0012**	**-**	**0.002143**	**0.0016**
	Water	**0.0009**	**0.0012**	**0.0018**	**-**	**0.0016**

	Wheat	**0.0009**	**0.0012**	**0.0018**	**0.002143**	**-**

Table 5.15. FERM for PCM and KPCM classification results. One untrained class was considered at a time.

Algorithm	**Untrained class**	**Sand**		**Eucalyptus**		**Forest_Gr**		**Water**		**Wheat**		**Overall Accuracy**
		UA	**PA**	**UA**	**PA**	**UA**	**PA**	**UA**	**PA**	**UA**	**PA**	
KPCM	**None**	**92.08**	**99.46**	**93.26**	**98.61**	**90.24**	**97.42**	**82.68**	**97.53**	**87.59**	**99.01**	**98.37**
	Sand	**-**	**-**	**93.47**	**98.43**	**90.52**	**97.53**	**81.47**	**97.10**	**87.83**	**98.98**	**98.05**
	Eucalyptus	**92.48**	**99.46**	**-**	**-**	**90.23**	**97.41**	**82.23**	**97.90**	**87.59**	**99.06**	**98.41**
	Forest_Gr	**91.57**	**99.38**	**94.42**	**98.33**	**-**	**-**	**81.66**	**97.19**	**88.37**	**98.99**	**98.48**
	Water	**91.57**	**99.45**	**93.41**	**98.44**	**90.16**	**97.38**	**-**	**-**	**88.39**	**98.88**	**98.46**
	Wheat	**91.39**	**99.42**	**93.81**	**98.31**	**90.73**	**97.21**	**82.31**	**96.86**	**-**	**-**	**97.89**

CHAPTER 6

6. DISCUSSION

The present chapter explains the results obtained in chapter 5. The main objective of this research work was to develop a kernel based possibilistic c-means algorithm for handling non-linear classes present in the dataset by identifying the best kernel methods and optimizing the parameters.

The most important parameter in PCM classifiers was the fuzzy parameter (m). It handles the randomness in the classification results. The fuzzy classifiers like PCM allow to adjust the randomness present in the ground through the fuzzy parameter. The value for fuzzy parameter increased from 2.5 to 3 when non-linearity was incorporated in the linear Formosat-2 data. The change in spatial resolution didn't have significant effect in the value of fuzzy parameter. The reason for this may be associated with less difference in spatial resolution between the Formosat-2 and Landsat-8 data. It has been observed from the classification of simulated image that the mixed pixels were better handled with the non-linear hyper tangent-KPCM classifier as it assigns the expected membership values to the mixed pixels (section 5.1 in chapter 5). The optimal value of fuzzy parameter was kernel specific and thus, needed to be optimized very carefully for implementation of kernel with fuzzy classifier.

The accuracy for classification of a class depends on the physical properties of that class and on the accuracy assessment technique. For instance, the water class in KPCM classification have shown lower user's accuracy, low value for Pearson correlation coefficient and high RMSE value when calculated by using image to image accuracy assessment method. This may be because the water body present in the site was river which is not having a defined boundary and neither was a uniform continuous feature. This may result into large number of mixed pixels. So, when it was accessed through image to image accuracy assessment technique with higher resolution formosat-2 image as reference image, the correspondence between the water pixels in the classified and the referenced image may not be high and thus, resulting into lower accuracy.

The handling of non-linearity can't be observed visually from the classification results but can be depicted from the improved accuracy results. The non-linear KPCM classifier handled the non-linearity present in the data as evident from its high overall accuracy of 98.37% in comparison to PCM classifier with comparatively low overall accuracy of 78.38%. A total increase of 19.99% in the overall accuracy of classification was observed. The quality of the results of non-linear KPCM classifier were better than the linear PCM classifier as evident from the low entropy value for results of KPCM classifier (0.4758) as compared to high entropy for results of PCM classifier (0.5430).

The composite kernel were formed by fusing the two best kernels through weighted summation technique (Camps-Valls et al., 2006). The hyper tangent and the sigmoid kernel were fused in the ratio of 50:50. The resultant composite kernel provided satisfactory classification with fuzzy overall accuracy of 97.84% equivalent to 98.37% achieved with hyper tangent KPCM classifier. With composite kernel, there was an improvement in average user's accuracy from 89.17% in case of KPCM to 89.90% for composite kernel. Thus, composite kernel can be used to add merits of different individual kernel function into a single kernel function

It has been observed from the previous studies that the accuracy of FCM classifiers decreased with the introduction of the untrained classes (Byju, 2015). But in case of PCM, shown in Table 5.9 and Table 5.10, the value of correlation, RMSE does not change with the introduction of untrained classes. Table 5.11 depicts the result for FERM where the class based accuracy was same and didn't change with introduction of untrained classes. This resistance by PCM classifier for the presence of untrained classes was due to the possibilistic (probabilistic) membership value which was calculated independently without considering the presence of other classes, thus, ignoring the untrained classes.

It has been seen that the implementation of kernels was data specific (Mittal and Tripathy, 2015), so in this research nine different kernel functions were implemented to identify the best kernel function. Among, nine different kernel functions implemented, hyper tangent kernel function performed best followed by sigmoid kernel function. The hyper tangent kernel generally showed good results for large numeric data set (Mittal and Tripathy, 2015). The high accuracy of classification of hyper tangent kernel in presence of non-linearity and mixed pixels showed its high sensitivity for the spectral variations. As the raster satellite data is the best example of large numeric data and they also have high spectral variations, the hyper tangent can have broader applicability in remote sensing image classification. The sigmoid kernel was also a hyperbolic tangent function which may be the reason for its better accuracy.

To conclude, the classifier developed by incorporating the hyper tangent kernel function with Possibilistic c-Means classifier not only handles the mixed pixels but also provide solution for the non-linearity present between the classes and handles untrained classes. The behavior of kernel functions was not affected by change in the dataset which may be due to the similar spectral variations in the dataset. As in this study the effect of data on kernel function is not considered, it provides scope for kernel function being studied with different dataset.

CHAPTER 7

7. CONCLUSION AND RECOMMENDATION

This chapter discusses the conclusion (section 7.1) and recommendation (section 7.2), it describes the final take away from this study. The section 7.1 concludes by answering the research questions framed at the beginning of this research and provides the concluding remark. The section 7.2 provides the recommendation for further research work which can be extended from this study work.

7.1. Conclusion

The main objective of this research work was to develop a kernel based Possibilistic c-Means (KPCM) classifier for handling the non-linear data. The Possibilistic c-Means (PCM) approach was used for handling the mixed pixel and kernel functions were used for handling the non-linearity. Nine different kernel functions were chosen for implementation. The incorporation of kernel function into PCM involves replacing of the Euclidean distance measure in PCM to kernel distance measure that takes care of the complicated non-linear boundaries between the classes.

To access the accuracy of classification different accuracy assessment methods were incorporated. Their results have shown different dimensions for comparing the accuracy of results, like correlation and RMSE provides the correspondence between the reference data and the classified output, entropy gives the degree of randomness in the result and shows the quality of classification. As each accuracy assessment method provides a new measure for evaluation of classifier, therefore, no accuracy assessment method can be said to be the best.

The thematic maps generated from hyper tangent-KPCM classifier were more accurate as the output of this classifier shows overall accuracy of 98.37%which was better than all other eight kernel function implemented. The lowest entropy value 0.47 attained with hyper tangent KPCM classifier guarantees better classification quality of the fuzzy output. It performs far better than all other kernel functions in absence as well as presence of non-linearity as demonstrated for Formosat-2 data (Fig 5.1 and Fig. 5.2) and with different resolution data set as shown with Formosat-2 and Landsat-8 data. So, it is evident that the classification results with hyper tangent kernel are quite stable and accurate for different datasets.

Though it is known that PCM classifier handles the untrained class quite well but it has been observed that the kernel (hyper-tangent) implementation with PCM further improves the capability to handle the untrained classes. It can be observed from Table 5.13, Table 5.14 and Table 5.15 that for KPCM classifier, the correlation, class based accuracy values remain constant and high (or vary within a negligible range) whereas for PCM classifier these values are comparatively varying and are low.

Answers to the research questions

Research Question 1: How well non-linearity between classes in the input feature space will be handled by KPCM?

Answer 1: The non-linearity handled with KPCM classifier has been proved by comparison of non-linear KPCM (with hyper tangent kernel) classifier with linear-PCM classifier. The hyper tangent-KPCM classifier shows accuracy of 98.37% as compared to PCM with low overall accuracy of 78.38% (Table 5.5). The KPCM classification outcome shows high correlation and low RMSE value with respect to the referenced data (Table 5.3 and Table 5.4). The lower entropy value (0.47) for classified results of KPCM in comparison to PCM classifier further guarantees the improved handling of non-linearity present in the input space.

Research Question 2: How can mixed pixels be handled using KPCM?

Answer 2: From the classification of simulated image, it can be observed that the mixed pixels were better handled by hyper tangent KPCM classifier as compared to linear PCM classifier (Figure 5.1). For example- in Figure 5.1 (b), a mixed pixel with composition of 50% for wheat class shows the membership value of 0.49 for wheat class whereas for the same pixel, with linear PCM classifier have very low membership value of 0.027 for the wheat class.

Research Question 3: How well KPCM performs in case of untrained classes (considering one or more than one classes at a time)?

Answer 3: The capability to handle the untrained classes increases with hyper tangent-KPCM classifier. As it is evident from the Table 5.13, the correlation values are high and remain constant for KPCM classifier in comparison to PCM classifier where the correlation value were low and varying. The similar affect was observed with RMSE, and FERM from Table 5.14 and Table 5.15 respectively. For example- In Table 5.13, for sand class the correlation value with PCM classifier is in between [0.86-0.88] whereas the correlation value for sand class remains constant (.9122) and comparatively high.

Research Question 4: How can we evaluate the performance in terms of accuracy and robustness during classification with single/composite kernel in KPCM?

Answer 4: There are various accuracy assessment techniques that can be employed for the accuracy assessment of the soft classified output. But none of the soft accuracy assessment techniques are considered as standard as each of the technique is having its own merit and provides a unique measure for assessment. So, in in this research multiple accuracy assessment techniques like Pearson correlation coefficient, RMSE, entropy and FERM were used. The Pearson correlation coefficient, RMSE and FERM were calculated relative to the referenced data, whereas the entropy is an absolute measurement i.e. it does not require any referenced data for computing entropy. The Pearson correlation coefficient, RMSE and FERM quantifies the correspondence between the classified and the reference data set. The entropy on the other hand quantifies the classification quality by giving lower value of entropy to the classification output having lower randomness.

7.2. Recommendations

The present research has scope for improvement as well as extensibility by enhancing the existing implementation and by incorporating of new and advance techniques. The KPCM classifier can be improved by taking following point into consideration:

- A detailed study is needed on parameter optimisation for different kernel functions as well as new method for parameter estimation can be implemented.
- A more rigorous study on the effect of multi sensor data on kernel implementation is needed.
- The KPCM classifier in this research study was developed by incorporating the kernel function with primitive PCM algorithm other new PCM algorithm like FPCM (Pal et al., 1997), IPCM (Zhang and Leung, 2004), PFCM (Pal et. al., 2005) and EPCM (Xie et. al., 2008) can be experimented with aim to get higher accuracy.
- More variability can be incorporated within the classes in simulated image so that it can represent the real image more accurately.
- The research can take a new dimension by performing KPCM in unsupervised mode.

List of References

Aggarwal, C. C. (2015). Outlier Analysis. In *Data Mining* (pp. 237–263). New York, USA: Springer International Publishing. http://doi.org/10.1007/978-3-319-1412-8

Ayat, N. E., Cheriet, M., Remaki, L., and Suen, C. Y. (2001). KMOD-A New Support Vector Machine Kernel With Moderate Decreasing. In *Document Analysis and Recognition, 2001. Proceedings. Sixth International Conference* (pp. 1215–1219). http://doi.org/10.1109/ICDAR.2001.953976

Bezdek, J. C., Ehrlich, R., and Full, W. (1984). FCM: The fuzzy *c*-means clustering algorithm. *Computers and Geosciences*, *10*(2–3), 191–203. http://doi.org/10.1016/0098-3004(84)90020-7

Binaghi, E., Brivio, P. a., Ghezzi, P., and Rampini, A. (1999). A fuzzy set-based accuracy assessment of soft classification. *Pattern Recognition Letters*, *20*(9), 935–948. http://doi.org/10.1016/S0167-8655(99)00061-6

Byju, A. P. (2015). *Non-Linear Separation of classes using a Kernel based Fuzzy c -Means (KFCM) Approach*. ITC, University of twente,The Netherlands.

Camps-Valls, G., Gomez-Chova, L., Munoz-Mari, J., Vila-Frances, J., and Calpe-Maravilla, J. (2006). Composite kernels for hyperspectral image classification. *IEEE Geoscience and Remote Sensing Letters*, *3*(1), 93–97. http://doi.org/10.1109/LGRS.2005.857031

Dehghan, H., and Ghassemian, H. (2006). Measurement of uncertainty by the entropy: application to the classification of MSS data. *International Journal of Remote Sensing*, *27*(18), 4005–4014. http://doi.org/10.1080/01431160600647225

Filippone, M., Masulli, F., and Rovetta, S. (2010). Applying the Possibilistic c-Means Algorithm in Kernel-Induced Spaces. *IEEE Transactions on Fuzzy Systems*, *18*(3), 572–584. http://doi.org/10.1109/TFUZZ.2010.2043440

Foody, G. M. (1995). Cross-entropy for the avaluation of the accuracy of a fuzzy land cover classification with fuzzy ground data. *ISPRS Journal of Photogrammetry and Remote Sensing*, *50*, 2–12. http://doi.org/0924-2716/95

Foody, G. M. (2000). Estimation of sub-pixel land cover composition in the presence of untrained classes. *Computers and Geosciences*, *26*(4), 469–478. http://doi.org/10.1016/S0098-3004(99)00125-9

Ganesan, P., and Rajini, V. (2010). A method to segment color images based on modified Fuzzy-Possibilistic-C-Means clustering algorithm. *Recent Advances in Space Technology Services and Climate Change 2010 (RSTS and CC-2010)*, 157–163. http://doi.org/10.1109/RSTSCC.2010.5712837

Hofmann, T., Scholkopf, B., and Smola, A. J. (2008). Kernel methods in machine learning. *The Annals of Statistics*, *36*(3), 1171–1220. http://doi.org/10.1214/009053607000000677

Hu, Y., Zuo, C., Qu, F., and Shi, W. (2012). Unsupervised Possibilistic Clustering Based on Kernel Methods. *Physics Procedia*, *25*, 1084–1090. http://doi.org/10.1016/j.phpro.2012.03.203

Krishnapuram, R., and Keller, J. M. (1993). Possibilistic approach to clustering. *IEEE Transactions on Fuzzy Systems*, *1*(2), 98–110. http://doi.org/10.1109/91.227387

Krishnapuram, R., and Keller, J. M. (1996). The possibilistic C-means algorithm: insights and recommendations. *IEEE Transactions on Fuzzy Systems*, *4*(3), 385–393. http://doi.org/10.1109/91.531779

Krishnapuram, R., and Keller, J. M. (1996). The possibilistic C-means algorithm: insights and recommendations. *IEEE Transactions on Fuzzy Systems*, *4*(3), 385–393. http://doi.org/10.1109/91.531779

Kumar, A., Ghosh, S. K., and Dadhwal, V. K. (2005). STUDY OF MIXED KERNEL EFFECT ON CLASSIFICATION ACCURACY USING DENSITY ESTIMATION. *ISPRS Journal of Photogrammetry and Remote Sensing*. Retrieved from http://www.isprs.org/proceedings/XXXVI/part7/PDF/206.pdf

Kumar, A., Ghosh, S. K., and Dadhwal, V. K. (2006). A comparision of the performance of fuzzy algorithm versus statistical algorithm based sub-pixel classifier for remote sensing data. In *ISPRS Commission VII Mid-term Symposium "Remote Sensing : From Pixels to Processes"* (pp. 1–5). Enschede, Netherlands. Retrieved from http://www.isprs.org/proceedings/XXXVI/part7/PDF/153.pdf

Kumar, A., Ghosh, S. K., and Dadhwal, V. K. (2014). Full fuzzy land cover mapping using remote sensing data based on fuzzy c -means and density estimation. *Canadian Journal of Remote Sensing*, *33*(2), 81–87. http://doi.org/10.5589/m07-011

Lillesand, T., Kiefer, R. W., and Chipman, J. (1987). Thematic Map Accuracy Assessment. In *Remote Sensing and Digital Image Processing* (p. 495). http://doi.org/10.1080/01431168708954824

Lillesand, T. M., and Kiefer, R. W. (1979). Thematic Information Extraction:Pattern Recognition. In *Remote Sensing and Image Interpretation* (5th ed., p. 612). Wiley India.

Lin, H., and Lin, C. (n.d.). A Study on Sigmoid Kernels for SVM and the Training of non-PSD Kernels by SMO-type Methods, 1–32.

Lin, H., and Lin, C. (2003). A study on sigmoid kernels for SVM and the training of non-PSD kernels by SMO-type methods. *Submitted to Neural Computation*, 1–32. http://doi.org/10.1.1.14.6709

Mittal, D., and Tripathy, B. K. (2015). Efficiency Analysis of Kernel Functions In Uncertainty Based C-Means Algorithms. *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 807–813.

Pal, N. R., Pal, K., and Bezdek, J. C. (1997). A mixed c-means clustering model. *Proceedings of 6th International Fuzzy Systems Conference*, *1*, 11–21. http://doi.org/10.1109/FUZZY.1997.616338

Pal, N. R., Pal, K., Keller, J. M., and Bezdek, J. C. (2005). A Possibilistic Fuzzy c-Means Clustering Algorithm. *IEEE Transactions on Fuzzy Systems*. http://doi.org/10.1109/TFUZZ.2004.840099

Rhee, F., Choi, K., and Choi, B. (2012). Kernel Approach to Possibilistic C-Means Clustering. *International Journal of Intelligent Systems*, *24*(3), 272–292. http://doi.org/10.1002/int.20336

Richards, J. a. (1993). *Remote Sensing Digital Image Analysis*. http://doi.org/10.1007/978-3-642-88087-2

Ricotta, C., and Avena, G. C. (2002). Evaluating the degree of fuzziness of thematic maps with a generalized entropy function: a methodological outlook. *International Journal of Remote Sensing*, *23*(December), 4519–4523. http://doi.org/Doi 10.1080/01431160210153985

Scholkopf, B. (2002). Learning with kernels. *Journal of the Electrochemical Society*, *129*(November), 2865. http://doi.org/10.1198/jasa.2003.s269

Scholkopf, B., Burges, J. C., and Smola, A. J. (2008). Advances in Kernel Methods: Support Vector Learning, 373. http://doi.org/10.1016/j.neuropsychologia.2011.01.001

Silvan-Cardenas, J. L., and Wang, L. (2008). Sub-pixel confusion-uncertainty matrix for assessing soft classifications. *Remote Sensing of Environment*, *112*(3), 1081–1095. http://doi.org/10.1016/j.rse.2007.07.017

Upadhyay, P., Ghosh, S. K., and Kumar, A. (2014). A Brief Review of Fuzzy Soft Classification and Assessment of Accuracy Methods for Identification of Single Land Cover. *Studies in Surveying and Mapping Science (SSMS)*, *2*(Mlc), 1–13.

Vidnerova, P., and Neruda, R. (2011). Evolving Sum and Composite Kernel Functions for Regularization Networks. In *Adaptive and Natural Computing Algorithms* (pp. 180–189). http://doi.org/10.1007/978-3-642-20282-7_19

Weisstein, E. W. (2006). Correlation Coefficient. Retrieved February 4, 2016, from http://mathworld.wolfram.com/CorrelationCoefficient.html

Wu, X. (2006). A Possibilistic C-Means Clustering Algorithm Based on Kernel Methods. In *2006 International Conference on Communications, Circuits and Systems* (Vol. 3, pp. 2062–2066). IEEE. http://doi.org/10.1109/ICCCAS.2006.285084

Xie, Z., Wang, S., and Chung, F. L. (2008). An enhanced possibilistic C-Means clustering algorithm EPCM. *Soft Computing*, *12*(6), 593–611. http://doi.org/10.1007/s00500-007-0231-6

Zadeh, L. A. (1965). Fuzzy sets. *Information and Control*, *8*(3), 338–353. http://doi.org/10.1016/S0019-9958(65)90241-X

Zadeh, L. A. (1978). Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, *100*, 9–34. http://doi.org/10.1016/S0165-0114(99)80004-9

Zhang, J.-S., and Leung, Y.-W. (2004). Improved Possibilistic C-Means Clustering Algorithms. *IEEE Transactions on Fuzzy Systems*, *12*(2), 209–217. http://doi.org/10.1109/TFUZZ.2004.825079

APPENDIX A

A.1: For Generation of referenced image from formosat-2 data, for image to image based accuracy assessment technique involves resampling of formosat-2 data from 8m to 10m. The generated Formosat-2 data is shown in Figure A-1.

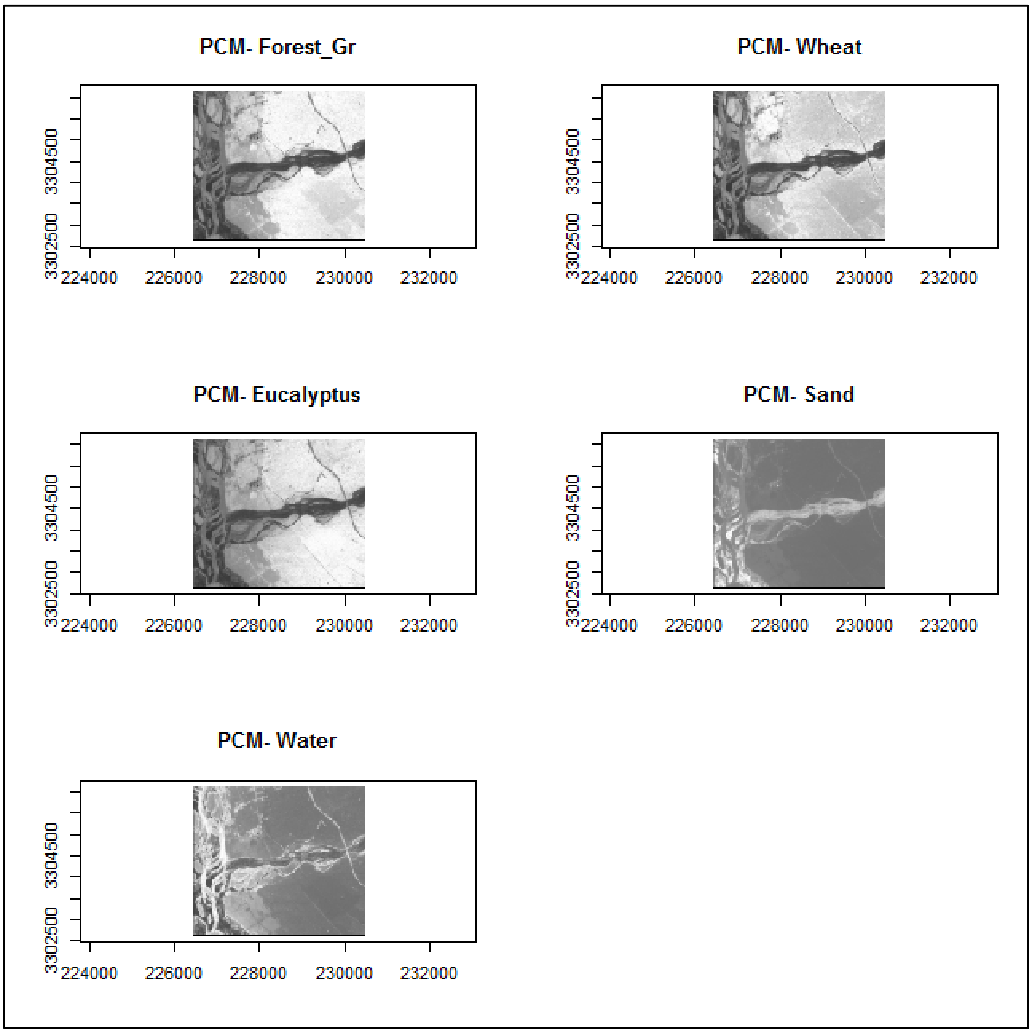


Figure A-1: Classified results for Fromsat-2 used as reference data for accuracy assessment.

A.2 Identifying and Simulating non-linearity in the data

At first, the linear-SVM classification was performed on Formosat-2 data for identifying non-linearity. The classification results showed 99.81% accuracy and linear-SVM classifier was successful in classifying the data (Figure A-2).

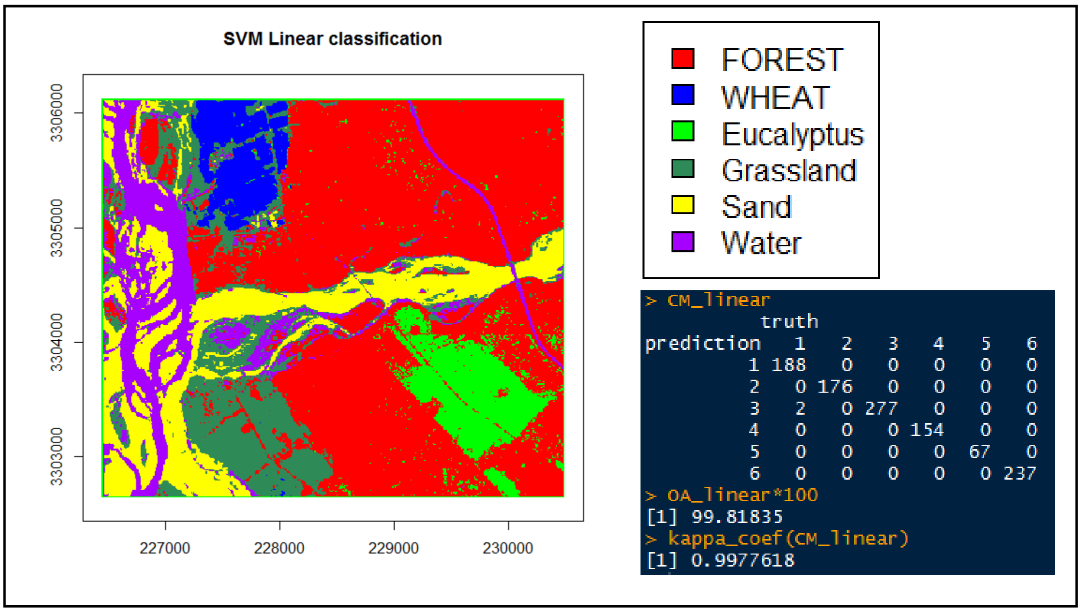


Figure A-2. Linear SVM classification for linear Formosat-2 data.

To demonstrate a useful implementation of kernel functions, non-linearity was simulated in the data with following steps:

- A new eucalyptus class was identified in the forest class. There was high mixing between these classes.
- Merging "forest" and "grassland" class into a single "forest" class, naming it as "Forest_Gr".
- Taking subset of the features. Band1 (blue) and Band2 (green) were chosen for classification.

On the simulated non-linear data linear-SVM classification was performed. High misclassification was observed between wheat and forest class. Steps involved in verifying the non-linearity in data by linear-SVM are as follows:

Training the classifier, i.e., identifying "W" of the optimal hyperplane, as shown in equation (3.26).

The SVM classifier was trained with the available training data to identify the optimal value of the weight vector (W).

Tuning the classifier, i.e., choosing the optimal value for the cost factor (Figure A-3).

The optimal value of cost factor was estimated based on the kappa value of classification result. The kappa value was calculated for the range of cost factor between [1*10^{-7} ,400]. The optimal value for cost factor was identified at 290.

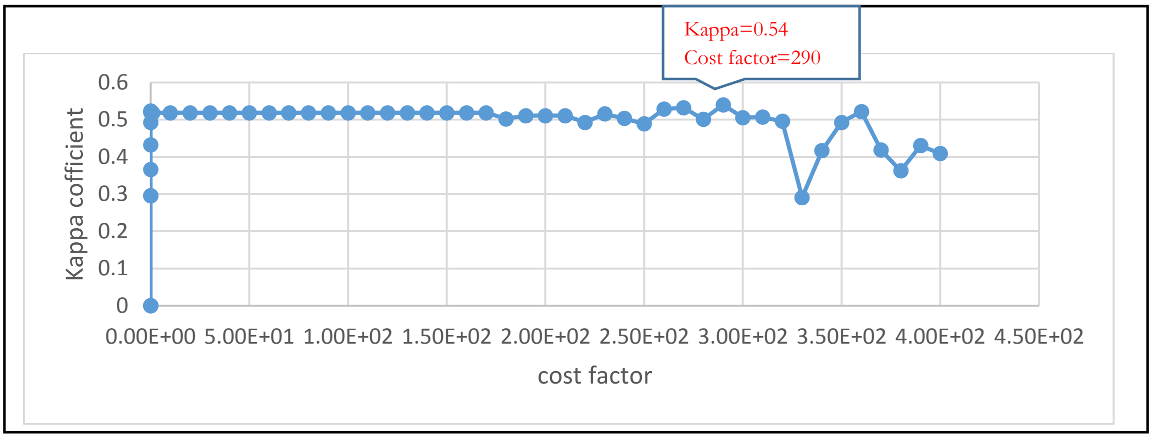


Figure A-3. Plot between kappa values for different value of cost factor

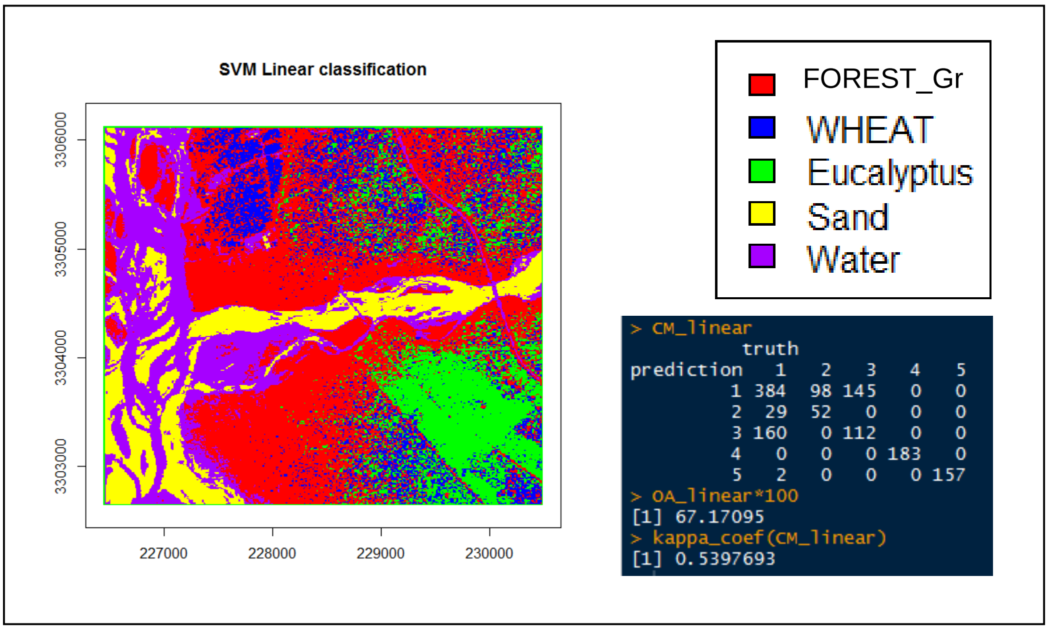


Figure A-4. Classified result for linear-SVM classification with non-linear data.

Accuracy Assessment, i.e., estimating classification accuracy

To evaluate the accuracy of linear-SVM classification the test data was used to identify the accuracy of each class. Here high misclassification was observed between wheat, eucalyptus and Forest_Gr class (Figure A-4). For the simulated dataset the overlap between these classes (forest and wheat) was depicted from the feature space as shown in Figure A-5. It can be seen that the wheat class is within the convex hull drawn (based on the co-variance of the classes) for Forest_Gr class. So, the non-linearity was present between Forest_Gr and wheat class.

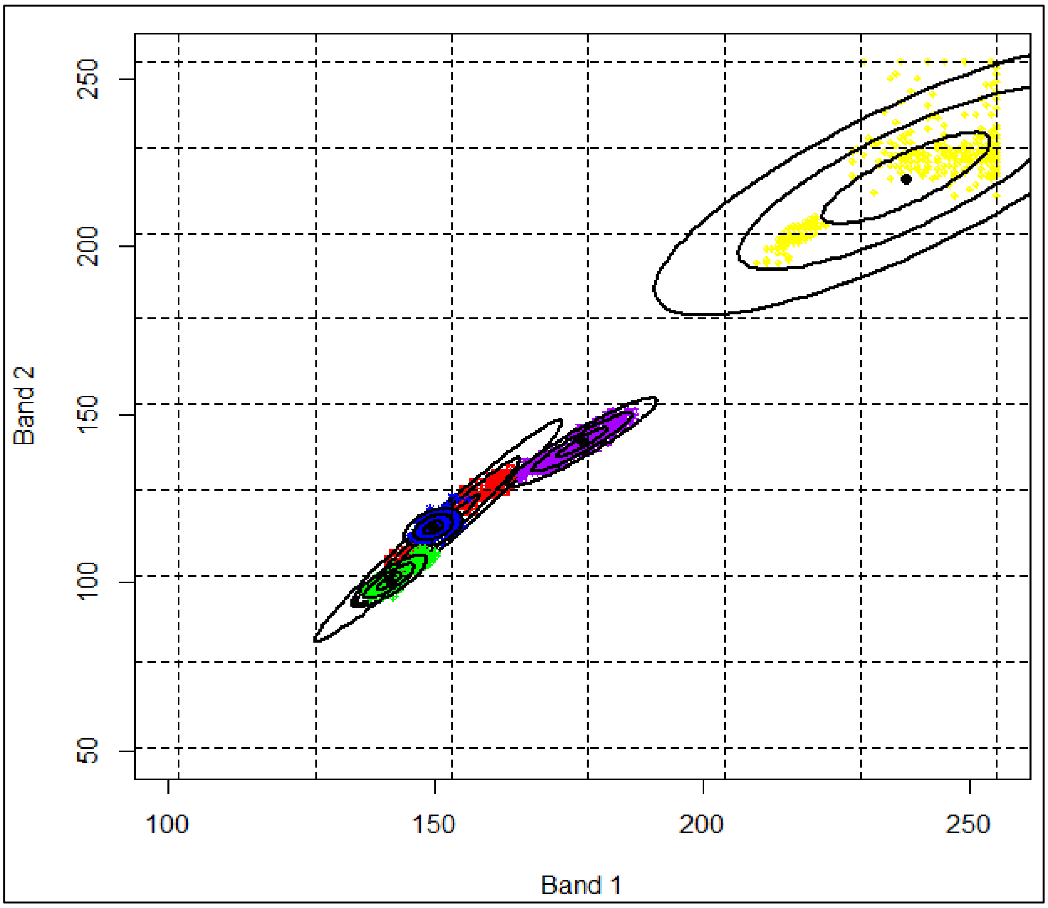


Figure A-5. Feature space for classes identified in Formosat-2

To further assure the presence of non-linearity in the data, classification was performed with non-linear SVM using RBF kernel with optimized value of cost factor and gamma at 230 and .01 respectively. The classified output for non-linear SVM (RBF kernel) shows higher accuracy than linear SVM kernel (Figure A-6).

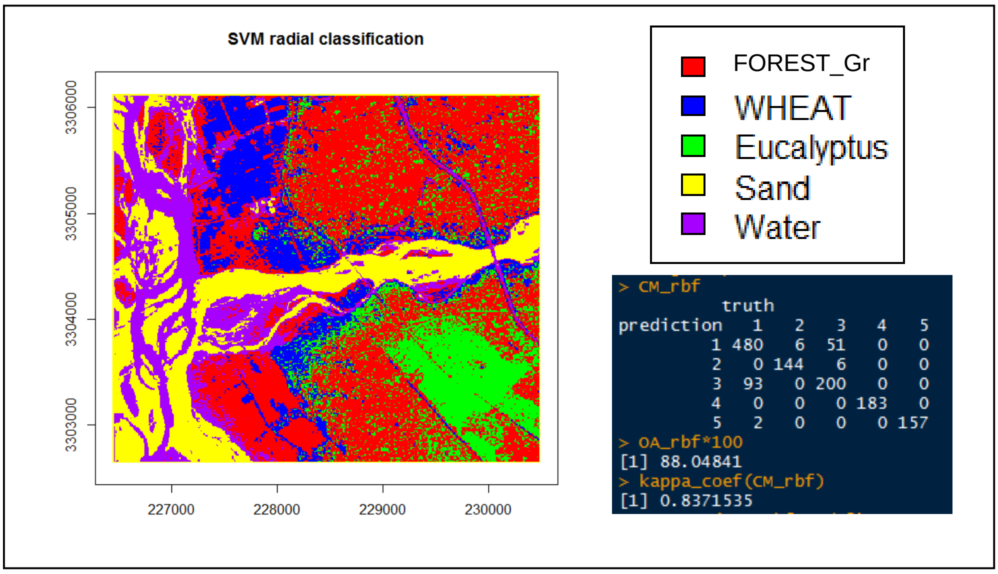


Figure A-6. Classified result for non-linear SVM (radial) classification. It shows the classified image, legend and error matrix for linear-SVM classification.

As, higher accuracy in classification was attained using non-linear SVM (with radial kernel), thus it is clear that non-linearity was successfully incorporated in the data set.

APPENDIX B

- A draft for the research paper has been submitted to guides. And is aimed to be published in a peer reviewed journal. - "Non-Linear separation of classes using kernel based possibilistic c-means". Nitin Kandpal, Anil Kumar, Valentyn Tolpekin