MODULE – 5 LECTURE NOTES – 2

IMAGE CLASSIFICATION- UNSUPERVISED

1. Introduction

As compared to supervised classification technique, unsupervised classification requires minimum initial input from the analyst. The natural groupings of spectral information present in pixels are examined in the multispectral feature space. In unsupervised classification, instead of letting the user collect the training data, the computer is allowed to select the class mean and covariance matrices which will be further used for classification. The automatic process of classification is left entirely to the system and hence the name unsupervised. The number of clusters to be formed is decided by the user. After classification, the analyst will assign these spectral classes to the information classes of interest. The analyst should be well versed with the spectral characteristics of the terrain that is being classified in order that the clusters may be labeled as containing useful information or as meaningless. The analyst relies on whatever reference information (ground truth) he has about the classified ground to identify the spectral classes created by an unsupervised classifier. For this reason, the term exploratory is sometimes used in place of unsupervised classification. Numerous clustering algorithms have been developed over the years which differ in terms of the clustering efficiency and decision rules used to perform classification. All these algorithms involve some form of iterative calculations in order to arrive at an optimum set of decision rules for the data set.

2. Chain Method of Unsupervised Classification

This algorithms operates in two steps. The first step reads through the dataset to identify possible clusters together with estimating mean pixel value to the formed clusters. The second step estimates a distance measure on a pixel by pixel basis in order that each pixel be classified to one of the clusters created in the previous step. The steps are elaborated below:

a) Generation of clusters

This classification technique requires the user to provide the following information :

(i) Radius to determine the formation of a new cluster

In a spectral space where pixel values stand for digital numbers providing brightness/spectral information, radius is usually specified in terms of brightness units (e.g., 30 brightness value units)

(ii) Parameter to merge clusters

A distance parameter in spectral space is required so that two or more clusters close enough with one another can be merged and treated as a single cluster. Again, in spectral space this is expressed in terms of brightness value units.

(iii) Number of pixels evaluated

The total number of pixels that should be analysed/evaluated before major merging of clusters should be specified (e.g., 3000 pixels)

(iv) Maximum number of clusters which need to be identified

This can be based on expert advice or user's familiarity with the area to be classified. Certain indices can be used which enable estimation of optimum number of clusters existing within an imagery. However, these will not be discussed in this module.

2.1 Merging of clusters

Consider a remotely sensed imagery with two bands and n number of land cover types. As mentioned earlier, the values of an image can be referenced using row and column numbers. Beginning from the origin (row1, column1), the pixels are usually evaluated from left to right in a sequential manner like a chain and hence the name. Assume that the pixel value at first location be considered as mean value of cluster 1 and let it be *M1* (30,30). A multispectral image can have say m number of bands, wherein each pixel value will be represented by m values. However, for simplicity this discussion considers just 2 bands and hence two values associated with each pixel. Now assume that pixel 2 is considered as the mean of cluster 2 with a mean value of *M2* (10, 10). The spectral distance between cluster 2 and cluster 1 is estimated using some distance measure (like Euclidean or Mahalanobis etc). If this distance be greater than the user specified radius (to determine formation of a new cluster), then

cluster 2 will be cluster 2. Else, the mean data of cluster 1 will be considered to be an average of the first and second pixels of brightness values. In such cases, pixel 2 will fail to pass the distance measure to be classified as belonging to cluster 2 and hence ti will be considered as being in cluster 1. Averaging the values of pixel 1 and pixel 2 will yield a new location for cluster 1. This process is continued with pixel 3 and so on until the number of pixels evaluated becomes larger than the total number specified by the user (e.g., 3000). At this point, the program calculates the distances between each pair of clusters. The user defined parameter to merge clusters is made use of until there are no clusters with a separation distance less than the parameter values. The entire imagery is analyzed using this process.

3. Iterative Self Organizing Data Analysis Technique (ISODATA)

ISODATA is self organizing because it requires relatively little human input. When compared with the chain method explained in section 2, ISODATA does not select the initial mean vectors based on the pixels present in the first line of data. Instead it is iterative in nature i.e., it passes through the image sufficient number of times before coming to a meaningful conclusion. Classification using the ISODATA algorithm normally requires the analyst to specify the following criteria.

(i) The maximum number of clusters to be identified by the algorithm ($C_{\max}$) .

(ii) The maximum percentage of pixels whose class values are allowed to be unchanged between iterations (*T*). This is regarded as a termination criterion for the ISODATA algorithm.

(iii) The maximum number of times ISODATA is to classify pixels so that cluster mean vectors can be recalculated (*M*) . This is also regarded as a decision rule to terminate the ISODATA algorithm.

(iv) Minimum members in a cluster : In case a cluster seems to contain members that are less than the minimum percentage of members, that cluster is deleted and the members are assigned to an alternative cluster. In most of the image processing softwares, the default minimum percentage of members is often set to 0.01.

(v) Maximum standard deviation: This specified value of standard deviation of cluster is used to decide on whether a cluster needs to be split into two or not. When the standard deviation for a cluster exceeds the specified maximum standard deviation and when the number of members in a class is found to be greater than twice the specified minimum members in a class, that cluster is split into two clusters.

(vi) Minimum distance between cluster means: This distance measure calculated in the spectral feature space is used to merge two or more clusters with one another.

Once all the user defined inputs are supplied to the classifier, the algorithm performs in the following manner:

The mean vectors of all the clusters are arbitrarily assigned along a m dimensional feature space. This process of choosing mean vectors ensures that the first few lines of pixel data do not influence or bias the creation of clusters. With the mean vectors, the algorithm searches through the image data wherein each pixel gets compared to each cluster mean using some distance measure and is assigned to that cluster mean to which it lies closest in the spectral feature space. ISODATA can progress either line by line or in block by block manner. Either way it will have some influence on the resulting mean vectors. At the end of first iteration, a new mean vector is calculated for each cluster based on the new pixel locations that have been assigned to each cluster based on distance measure. To perform this calculation, the minimum members within a cluster, their maximum standard deviation and the minimum distance between clusters need to be taken into consideration. Once a new set of cluster means are calculated, this process is repeated wherein each pixel is again compared to the new set of cluster means. This iteration continues until one of the thresholds is reached i.e., either there is very little change in the class assignment or the maximum number of iteration are attained.

The ISODATA classification algorithm is a slow technique in which the analysts allow the algorithm to commence by a large number of iterations in order to generate meaningful results. Details regarding the algorithm steps are schematically shown in Figure 1. The output of an unsupervised classification will be a set of labelled pixels which range from 1 to *k* where k stands for the total number of classes picked out by the classification algorithm. The output image of classification can be displayed by assigning color to each of these class labels. The geographical location of each pixel of each class can be assessed to evaluate the

land cover represented by these pixels. Unsupervised classification can be used as an initial method to refine the classes present within an image before carrying out the supervised classification.

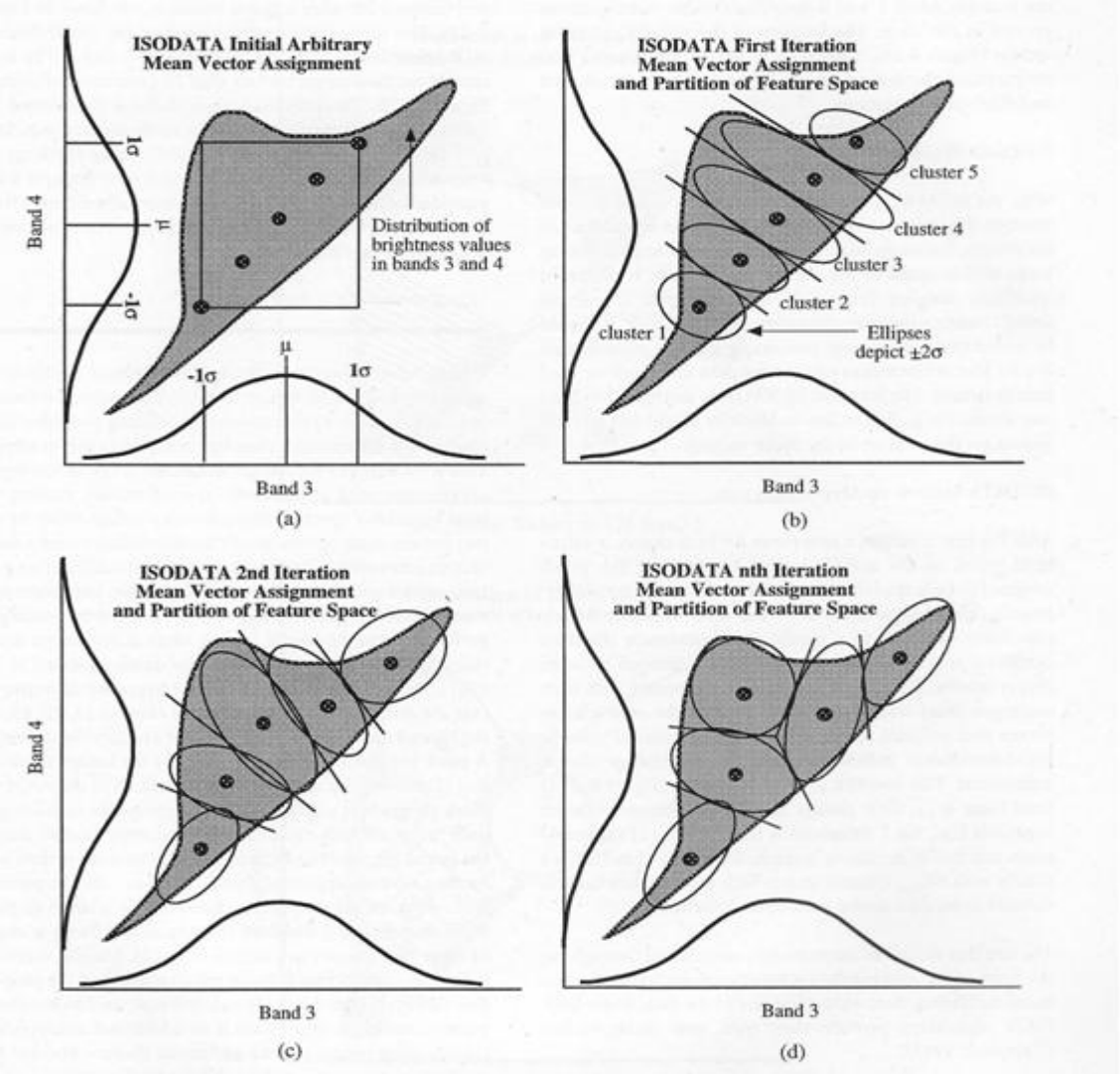

Figure 1: Schematic showing (a) ISODATA initial distribution of 5 hypothetical mean vectors in both bands as beginning and ending points (b) In the first iteration, each candidate pixel is compared to each cluster mean and assigned to the cluster whose mean is closest in Euclidean distance (c) during the second iteration, a new mean is calculated for each cluster based on the actual spectral locations of the pixels assigned to each cluster, instead of the intial arbitrary calculation. This involves analysis of several parameters to merge or split clusters. After the new cluster mean vectors are selected, every pixel in the scene is once again assigned to one of the new clusters. (d) This split-merge-assign process continues until

there is little change in class assignment between iterations (the threshold is reached) or the maximum number of iterations is reached.

4. K means algorithm

One of the most commonly used non-parametric unsupervised clustering algorithms, well known for its efficiency in clustering large data sets, is that of *K*-means. In general, all the pixels are classified based on their distances from the cluster means. Once this is done, the new mean vectors for each cluster are computed. This procedure is iteratively carried out until there is no variation in the location of cluster mean vectors between successive iterations.

Similar to the ISODATA algorithm, Kmeans algorithm also assigns initial cluster vector. The difference is that *k* means algorithm assumes that the number of clusters is known a priori. The main objective of *k* means clustering approach is to estimate the within cluster variability.

Step-1: Locate initial cluster centers

The cluster centers generated at the end of first iteration are taken as the initial cluster centre. If X represents the sample space of data having elements as x, N be the total number of elements in sample space and Bnd be the total number of bands, then the mean for data points for i^{th} cluster in j^{th} dimension is given by Equation (3.1):

$$v_{ij} = \sum_{j=1}^{Bnd} \left(\frac{\sum_{k=1}^{N} x_{kj}}{N} \right), 1 \le i \le c \qquad (3.1)$$

Distance of each pixel from all existing clusters is computed and it is assigned to the cluster yielding the minimum distance. Recalculate the cluster centers using Equation (3.1). The program terminates once the maximum number of iterations have been reached or by the minimization of the objective function J i.e. the within cluster sum of squares as given by Equation (3.2).

$$J = \sum_{i=1}^{c} \sum_{j=1}^{Bnd} \sum_{k=1}^{N} \left\| x_{kj} - v_{ij} \right\|^2 \qquad (3.2)$$

Step-2: Merging of clusters

Several measures are available for cluster merging. Some of these adopted in this work are enlisted below:

1. Root mean square (RMS) Euclidean distance for each cluster

   The RMS distance of pixels in the i^{th} cluster from their cluster centre is given by Equation (3.3)

$$RMS_i = \sqrt{\frac{1}{N_i}\sum_{x\in X}\sum_{j=1}^{Bnd}(v_{ij} - x_{ij})^2} \tag{3.3}$$

2. Matrix of Euclidean distances between cluster centers.

   The average Euclidean distance of the i^{th} cluster center to the other cluster centers is given by the Equation (3.4)

$$A_i = \frac{1}{c-1}\sum_{i=1}^{c} d_{ij} \tag{3.4}$$

   Here, d_{ij} is the Euclidean distance between the i^{th} and j^{th} cluster centers.

   The advantages of using this technique are that it is a simple, computationally fast clustering approach which produces tighter clusters.