Unsupervised Classification of Hyperspectral Images based on Spectral Features

Subhrajyoti Senapati
(111EC0191)

Department of Electronics and Communication Engineering
National Institute of Technology Rourkela
Rourkela – 769 008, Odisha, India

Unsupervised Classification of Hyperspectral Image based on Spectral Features

Dissertion submitted in
May 2015
to the department of
Electronics and Communication
of
National Institute of Technology Rourkela
in partial fulfillment of the requirements
for the degree of
Bachelor of Technology
by
Subhrajyoti Senapati
(Roll 111EC0191)
under the supervision of
Prof. Samit Ari

Department of Electronics and Communication Engineering
National Institute of Technology Rourkela
Rourkela – 769 008, Odisha, India

Electronics and Communication Engineering
National Institute of Technology Rourkela
Rourkela-769 008, India. www.nitrkl.ac.in

Date:

Samit Ari
Asst. Professor

Certificate

This is to certify that the work in the thesis entitled *Unsupervised Classification of Hyperspectral Image based on Spectral Features* by *Subhrajyoti Senapati*, bearing roll number 111EC0191, is a record of research work carried out by him under my supervision and guidance in partial fulfillment of the requirements for the award of the degree of *Bachelor of Technology* in *Electronics and Communication.* Neither this thesis or any part of it has been submitted for any degree or academic award elsewhere.

Dr. Samit Ari

Acknowledgement

Although this project is an individual work, it was successful with the help of many people. Simply naming them would not give justice to their help.

At first I want to thank my guide Prof. Samit Ari for spending his valuable time in guiding and inspiring me to do this project. His support has lead me to improve the the method used in this project.

I am also thankful to Prof. Kamalakanta Mahapatra for his encouragement that success comes after failure and hard work is key to get it. Its indeed a privilege to be associated with Prof. S.K. Patra, Prof. S.K. Behera, Prof. Poonam Singh, Prof. S.K. Das, Prof. S.M. Hiremath, Prof. S. Deshmukh, Prof. A.K. Swain and Prof. U.K. Sahoo. They have made valuable support in many ways.

A lot of thanks to my fellow colleagues and friends. They have removed the loneliness variable from my life and supported me during this research. Finally, my hearty gratitude to my beloved parents who have supported me in every decision i had made.

Subhrajyoti Senapati

Abstract

In this world of Big Data, large quantities of data are been created everyday from all the type of visual sensors available in the hands of mankind. One important data is that we obtain from satellite of the land image. The application of these data are numerous. They have been used in classification of land regions, change detection of an area over a period of time, detecting different anomalies in the area and so on. As data is increasing at a high rate, so manually doing these jobs is not a good idea. So, we have to apply automated algorithms to solve these jobs.

The images we see generally consists of visible light in Red, Green and Blue bands, but light of different wavelength differ in their properties of passing obstacle. So, there has been considerable research going on continuous spectra images. These images are called Hyper-spectral Image.

In this thesis, I have gone through many classic machine learning algorithms like K-means, Expectation Maximization, Hierarchical Clustering, some out of box methods like Unsupervised Artificial DNA Classifier, Spatial Spectral Information which integrates both features to get better classification and a variant of Maximal Margin Clustering which uses K-Nearest Neighbor algorithm to cross validate and get the best set to separate. Sometimes PCA is used get best features from the dataset. Finally all the results are compared.

Keywords: Hyperspectral Image, Unsupervised Classification, Maximal Margin Clustering, Artificial DNA

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In recent days, due to advancement of sensors, Hyperspectral images are pretty normal in todays scientific expedition. Maybe for Body scans, or for land images taken from satellite, they have got role to play. Hyperspectral sensors gather from a wide range of wavelength with narrow steps giving an illusion of continuity. [1]. In this project work, I have generally worked on Satellite based Hyperspectral images of Land cover. Due to huge amount of information available from these data, they are handy in tasks like segmentation of Land area, detecting changes or anomalies, also for prediction weather and many more. Manually doing these task are not possible, so I have to use computers to solve these problems. But for machines these classification tasks are difficult and hence machine learning algorithms have to be used to accomplish these problems.

Due to very few availability of ground truth data of a satellite image, unsupervised classification is much better than its counterpart and would work on almost all situations. So here, an Hyperspectral image(cube) is taken and from that segments are found belonging to different content(i.e grass, water, varied vegetation, soil type, buildings etc).

1.1 Hyperspectral Images and Multispectral Images

Its a valid question that both Hyperspectral and Multispectral images contains a number of bands, then what is the difference between the two of them.

While bands present in Multispectral images are narrow but they are taken discretely from the light spectrum where as in Hyperspectral image, bands were taken continuously over a region of spectrum. For example : bands of (10nm-20nm, 30nm-40nm, 50nm-60nm, ...) belongs to category of Multispectral images, whereas bands of (10nm-20nm, 20nm-30nm, 30nm-40nm, ...) belongs to Hyperspectral images.

Both Multispectral and Hyperspectral images have their own pros and cons. In latter the whole spectrum data is acquired and hence no prior knowledge is required and hence all information available from the dataset can be mined, which can lead to better clustering. As data present here is very large so complexity of the methods increases as well as cost. Data storing and transmission becomes another problem.

1.2 Literature Survey

Hyperspectral data, with the larger volume, high dimensionality, and temporal and spatial spectral diversity, present a challenge for the traditional unsupervised clustering processing techniques such as k-means, iterative self-organizing data (ISODATA), and fuzzy c-means clustering algorithms, which originate from multispectral imagery. In recent research, some artificial intelligence methods, such as the unsupervised artificial immune classifier [2], automatic fuzzy clustering using modified differential evolution (MoDEFC) [3], and multiobjective fuzzy clustering based on differential evolution [4] and scheme with support vector machine classifier [6], have been developed to solve the multiimage clustering problem. These methods are used to estimate the class number and the distribution of clusters by multicycle iterative optimization, but when these methods are introduced to process hyperspectral data, it is difficult to reach convergence and time-consuming to search the clusters in the sparse and skewed space.

To avoid the problems resulting from the high dimensionality of hyperspectral imagery, band selection and dimension reduction are conducted before hyperspectral image processing. Many feature extraction and dimension reduction techniques have been developed in hyperspectral imaging, such as principal component analysis (PCA) [5], independent component analysis [6], discrete wavelet transform [7], band reduction based on rough sets [8], projection pursuit algorithm [9], and clonal selection feature-selection algorithm [10]. After processing, the clustering is carried out on the dimension-reduced data by the traditional unsupervised classifiers. Recently, some new clustering methods have been developed to resolve the problems resulting from the high dimensionality. A divideand-conquer approach for contiguous subsets with similar characteristics has been proposed, which employs information fusion to classify the hyperspectral data [11]. A hierarchical clustering with local region-growing segmentation and global clustering has also been used to classify hyperspectral data [12]. Furthermore, a combination of clustering with feature selection based on particle swarm optimization has been proposed to cluster hyperspectral data by estimating the cluster statistical parameters and by detecting the most discriminative features [13]. Although these proposed methods decrease the computational complexity for the hyperspectral clustering by different strategies, the dimension reduction destroys the continuity of the spectral signatures and the spectral absorption and reflection features, such as the spectral shape, amplitude, and slope features, which cannot then be easily used to discriminate the categories of the spectral signatures.

A spectral-coding-based matching technique retains the potential discriminative information of the spectra and transforms the spectral values into a discrete set of symbols in a specific manner so that signatures can be represented and compared with each other by the new symbols more effectively [16]. Various spectral encoding and matching methods, e.g., binary coding (BC) [17], spectral-feature-based BC [18], and spectral derivative feature coding [19], have been successfully used in supervised hyperspectral data classification

1.3 Challenges faced by Unsupervised Classification of Satellite Images

Previously I have discussed about the spectral component of Image, which differentiates Hyperspectral and Multispectral Images, but here the spatial component creates the problem.

- Sensors present in the satellite are at a huge distance from earth. Although their zooming and resolution capacities are tremendous, theres a limit to that. So, when image taken from that far away, one pixel(the smallest unit in the image) represents an area of 30 X 30 meters or so. In that area there may be more than on component(for example both soil and water) present. This leads the sensor to pick up mixed signals. Hence these signals will not look similar to their parent data.

- In Unsupervised Classification, no prior information is available and with the large amount of data the ”curse of dimensionality”, Hyperspectral Images give, this creates a problem for unsupervised Algorithms.

1.4 Objective

This research project was carried out to satisfy the following objectives :

- To study the approaches taken to find different land cover given a Hyperspectral Satellite Image.

- To contribute to better classification of the data.

- Finally compare all the methods implemented.

For this project my main goal was to implement the latest classification algorithms. Extraction of satellite images from their respective formats, noise removal, number of class determination are not considered. Some of the algorithms are specifically for images while other can be applied to any data type.

1.5 Theory Required

In this chapter I am going to discuss about basics of classification and some simple methods that help it.

1.5.1 What is Classification/Clustering

Basically we are given data about something of particular interest. each data point will contain a array of values, which basically relates what the data is from a given training set that contains the true class of the data.But When the data is not available, we go for Unsupervised Learning. In Unsupervised Classification/Clustering, the task is to group the data that are most similar. Also termed as segmentation analysis, taxonomy analysis, these methods are most helpful when ground truth cases are not given.

Many algorithms have been developed to attempt clustering problem. Some of the classic and well known are : *K-Means Clustering*, *Gaussian Model*

Clustering using Expectation Maximization, *Hierarchical Clustering*. Here I have used K-means algorithm for several purposes. Also many supervised algorithms are used in this project like K-Nearest Neighbor(KNN), SVR(Support Vector Regression). These supervised algorithms though can't be directly applied to the image, they were used with those clustering algorithms to improve the result obtained from basics.

1.5.2 Feature Extraction

Our data may contain a lot of input features which may or may not be redundant. Least informant features need to be removed to both decreases the complexity and dimensionality curse. The minimum discriminative features can be found by different greedy approaches. Numerous of the features are really a combination of many information removing which we may lose important data needed. Uprooting such a highlight would uproot more data than required. In the following passages, PCA is presented as a feature extraction answer for this issue, and its internal workings are discussed.

As a rule, features are related. As a sample, consider the situation where we need to utilize the red, green and blue parts of every pixel in a picture to arrange the picture (e.g. identify cats versus dogs). Picture sensors that are most touchy to red light additionally catch some blue and green light. Correspondingly, sensors that are most touchy to blue and green light likewise show a certain level of affinity to red light. Thus, the R, G, B parts of a pixel are measurably mixed up. Subsequently, essentially disposing of the R segment from the feature vector, likewise certainly removes data from the G and B channels. As it were, before taking out features, we might want to change the complete feature space such that the basic uncorrelated segments are acquired.

By and large, PCA permits us to get a M-dimensional subspace of the first N-dimensional information, where M ¡= N. Besides, if the obscure, uncorrelated parts are Gaussian distributed, then PCA really goes about as an ICA(Independent Component Analysis) since uncorrelated Gaussian variables are measurably free. Be that as it may, if the hidden segments are not regularly dispersed, PCA simply creates decorrelated variables. For this situation, algorithms with dimensionality reduction may be a better option.

The PCA formula : Taking into account the past segments, we can now list the straightforward formula used to apply PCA for highlight extraction:

1. ***Center the information* :**

   It is demonstrated that the covariance network can be composed as a grouping of straight operations (scaling and revolutions). The eigendecomposition removes these change matrices. The eigenvectors speak to the pivot framework while the eigenvalues speak to the scaling variables. Nonetheless, the covariance matrix does not contain any data identified with the interpretation of the information. For sure, to speak to interpretation, a relative change would be required rather than a direct change. Any current move needs to be countered by subtracting the mean of the information from every information point before applying PCA to uncorrelated the data. This essentially relates to focusing the information such that its normal turns into zero.

2. ***Normalize the information* :**

   The eigenvectors found from the covariance matrix points to the direction of largest eigenvalue or variance of the data. On the other hand, fluctuation is a flat out number, not a relative one. This implies that the change of information, measured in centimeters (or inches) will be much bigger than the difference of the same information when measured in meters (or feet). Consider the sample where one feature speaks to the length of an item in meters, while the second highlight speaks to the width of the article in centimeters. The biggest fluctuation, and consequently the biggest eigenvector, will verifiably be characterized by the first highlight if the information is not standardized.

To dodge this scale-subordinate nature of PCA, it is helpful to standardize the information by isolating every highlight by its standard deviation. This is particularly essential if diverse highlights relate to distinctive measurements.

3. ***Calculate the eigendecomposition* :**

   Since the information will be anticipated onto the biggest eigenvectors to diminish the dimensionality, the eigendecomposition needs to be gotten. Singular Value Decomposition(SVD) standout amongst the most broadly utilized systems to proficiently ascertain the eigendecomposition

4. ***Project the information* :**

   To lessen the dimensionality, the information is just anticipated onto the biggest eigenvectors. Let V be the grid whose segments contain the biggest eigenvectors and let D be the first information whose sections contain the diverse perceptions. At that point the anticipated information D' is gotten as $D' = V^T D$. We can either pick the quantity of remaining measurements, i.e. the sections of V, straightforwardly or we can characterize the measure of change of the first information that needs to keep while taking out eigenvectors. If N eigenvectors are kept, and 'e' speak to the relating eigenvalues, then the measure of fluctuation that remaining parts in the wake of anticipating the first d-dimensional information can be computed as:

$$sq = \frac{\sum_{i=0}^{N} e_i}{\sum_{j=0}^{d} e_j}$$

1.5.3 K-Means

k-means clustering [14] is a partitioning strategy. Not at all like hierarchical clustering, k-means grouping works on genuine perceptions (as opposed to the bigger arrangement of divergence measures), and makes a solitary level of groups. The qualifications imply that k-means grouping is regularly better than hierarchical clustering for a large dataset.

K-means treats every perception in your information as an object having an area in input space. It discovers a partition in which objects inside every group are as near to one another as would be prudent, and as a long way from objects in different groups as possible.First, you indicate ahead of time what number of clusters are being looked for: this is the parameter k. At that point, k points are picked indiscriminately as bunch focuses. All examples are appointed to their nearest group focus as per the customary Euclidean separation metric. Next the centroid, or mean, of the cases in every cluster is computed this is the ”signifies” part. These centroids are taken to be new central values for their individual groups. At long last, the entire procedure is rehashed with the new group focuses. The cycle proceeds until the same centers are appointed to every group in back to back rounds, at which organize the cluster focuses have settled and will continue as before for eternity.

Every cluster in the partition is characterized by its part questions and by its centroid or center. The centroid for every group is the point to which the whole of separations from all items in that cluster is minimized. means figures group centroids distinctively for every separation measure, to minimize the aggregate regarding the measure that you indicate.

Indeed, this is valid for all functional grouping methods: it is quite often infeasible to discover all around ideal bunches. To expand the possibility of discovering a worldwide least individuals regularly run the calculation a few times

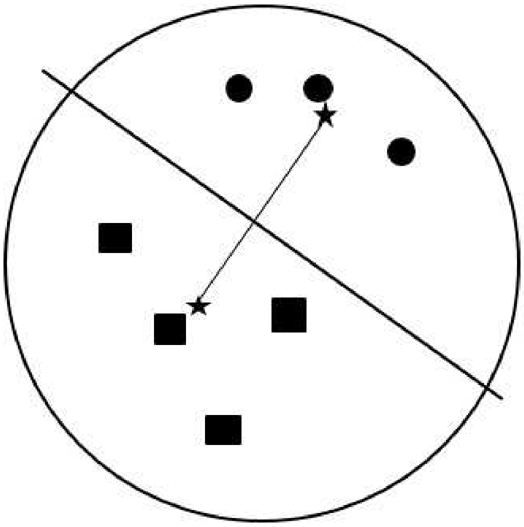

Figure 1.1: K-Means Algorithm in 2D space

with distinctive beginning decisions and pick the best last resultthe one with the littlest aggregate squared separation.

It is anything but difficult to envision circumstances in which k-means neglects to locate a decent clustering. Consider four examples at the vertices of a rectangle in the two-dimensional space. There are two regular groups, framed by gathering together the two vertices at either end of a short side. Anyhow, assume that the two introductory group focuses happen to fall at the midpoints of the long sides. This structures a steady setup. The two groups every contain the two examples at either end of a long side regardless of how awesome the distinction between the long and the short side is.

Algorithm: K-Means Clustering

```
Repeat till convergence
1. for i=1 to m
        c(i) := index (from 1 to K) of cluster centroid closest to x(i)

2. for k = 1 to K
        myu(k) := average(mean) of points assigned to cluster k.
```

1.5.4 K-Nearest Neighbor(KNN)

k-Nearest Neighbors is an algorithm that doesn't use any parameter helps in regression and classification. Here the input data comprises of the nearest located k instance in the highlighted domain. The yield relies on upon utilization on Regression and Classification.:

k-NN method returns class membership values. Maximum vote given by neighbors characterize the data, with the data being allocated to the class most regular among its k closest points (k is positive integer and its not very large). In the event that k is 1, then the item is basically allocated to the class of that closest neighbor.

k-NN regression returns the estimation of the object. This k closest neighbors constitute this value by averaging the distance. k-NN is a apathetic realizing, where the capacity is just approximated by regional standards and all processing is conceded until order. The k-NN algorithm is one of the easiest in machine learning.

Both for regression and classification, it can be valuable to weight. Weight gives a measure of what neighbors are what distance. Naturally, nearest neighbors give more weight than the far ones. So one way is to give weight 1/d to the neighbors where d represents the distance of the neighbors from the test data.A deficiency of this calculation is that it is sensitive to the nearby structure of the information. The calculation is not similar to that of k-Means, another well-known machine learning method.

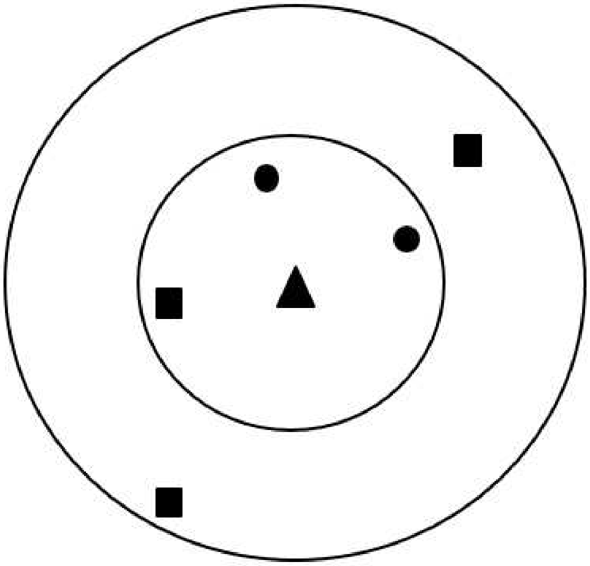

Figure 1.2: How k-NN classifies test data.

A usually utilized separation metric used while operating on continuous vars is Euclidean distance. [15] For discrete variables, for example, for content order, another metric can be utilized, for example, the cover metric. Frequently, the characterization exactness of k-NN can be enhanced essentially if the separation metric is found out with specific calculations, for example, Neighborhood component analysis or Large Margin Nearest Neighbor.

A drawback of the essential ”larger part voting” grouping happens when the class dispersion is skewed. One approach to defeat this issue is to weight the arrangement, taking into record the separation from the test point to each of its k closest neighbors. The class (or quality, in classification issues) of each of the k closest center is reproduced by a weight corresponding to the opposite of the separation starting there to the test point. Another approach to overcome skew is by deliberation in information representation. That is, examples of a more frequent class tend to dominate the prediction of the new example.

1.5.5 Support Vector Regression

Support Vector Machine is basically a Large Margin Classifier. It tries to separate the data by the maximum margin by which they can be separated. So, it doesn't faces the problem of Local Minima. Also using the Kernel trick, this classifier can extend to a non-linear one.

Although Support Vector Machines are generally used for classification problems, they can also be applied to regression problems while retaining properties of Maximum Margin algorithms. Similar to classification approach, here the aim is to optimize the bounds or distance of separation between data points. Similarly for regression case, a loss function is defined thats going to ignore errors that are within a certain distance. this is called epsilon loss function. According to epsilon-SVR, error for boundary on training is measured for each data and all points inside the boundary are given a loss value of zero. As the calculations are done using only the support vectors, so complexity of program is reduced than the direct approach. So, this regression method finds the global minimum as well as optimizes the problem to a small subset containing some specific data(support vector).

For a linear regression problem,

$$f(x, w) = \sum_{q=1}^{m} w_q t_q(x) + b$$

where $t_q(x), q = 1, 2, 3, ..., m$ denotes non-linear transformation of the data and b is the bias term. The estimation accuracy is measured by a loss function, as proposed by Vapnik:

$$L_s(y, f(x,w)) = max(0, |y - f(x,w)| - \epsilon)$$

SVM finds the linear regression from the high dimensional data's feature space using epsilon-sensitive loss as well as reduces $||w||^2$. This process can be explained by taking into consideration non-negative slack variables $insertetaiandetai*$. This measures the error of data points that are outsize the epsilon boundary.

$$min_{w,b,\xi_i,\xi_i^*}\frac{1}{2}||w||^2 + C\sum_i(\eta_i\xi_i + \eta^*\xi_i^*)$$
$$\text{s.t.}\left\{\begin{array}{c} y_i - < w, x_i > -b <= \eta + \xi_i \\ < w, x_i > +b - y_i <= \eta + \xi_i^* \\ \eta_i, \xi_i, \eta_i^*, \xi_i^* >= 0 \end{array}\right\}$$

The ooptimization problem is henced changed to its dual problem and hence he solution is given by

$$f(x) = \sum_{q=1}^{sv}(\alpha - \alpha^*)K(x_i, x);$$

where sv is the cardinality of set of Support Vectors. and the kernel function is

$$K(x_i, x) = \sum_{q=1}^{m} t_q(x)t_q(x_q)$$

SVR's performance same way as its counterpart depends on the value of C(regularization parameter) and epsilon. So to reduce the complexity these parameters are user defined. Knowledge about the matter we are solving the problem for is required to give corresponding Kernel function and hence value of epsilon. Parameter C punishes the deviation from boundary and determines trade-off between the degree of error and the complexity of the model i.e the flatness of the model.

1.6 Dataset

All the algorithms mentioned here were tested using basically two datasets.

1. ***SalinasA Scene***

   This scene was collected by the 224-bandAVIRIS sensorover Salinas Valley, California, and is characterized by high spatial resolution (3.7-meter pixels). The area covered comprises $86X83$ pixels and 224 spectral reflectance bands .

2. ***Real and fake fruit dataset***

   Although this is not a satellite image, it was taken from CAVE laboratory using Apogee Alta U260 CCD camera with wavelength range of 400-700 nm with 10 nm steps. This image was taken as there were unavailability of many small dataset that can be run using all our algorithm.

1.7 Thesis Outline

This thesis contains four main chapters each consisting of sections.

Chapter 2: Theory of Basic Methods

In this chapter, approach to classification, basic theory of PCA Data extraction Algorithm, basic classification Algorithms like K-Means, K-Nearest Neighbor are explained. With the basic algorithms that are going to be used here are cleared in the this chapter, I am going to introduce some new algorithms some of which are specifically for the task of classification of land cover in Hyperspectral images and one is a general classifier technique in the next chapter onwards.

Chapter 3: Integrating Spatial Feature in Classification

This chapter uses the location or nearness information along with spectral data to cluster the land regions.

Chapter 4: An Unsupervised DNA Classifier

In this chapter implementation of a bio-inspired algorithm is used. Here features are thought to be as DNA strands and they are used to both compress data as well as classify it.

Chapter 5: Maximum-Margin Clustering with Hierarchical Tree

In this method, i have implemented a technique to use Support Vector Regression in an unsupervised approach with validation using k-NN so as to get a better result.

Chapter 6: Conclusions and future work

Here the conclusion of this research project is discussed along with the trails of it in the future.

Chapter 2

Integrating Spatial and Spectral Data using Spectral Information Divergence

This algorithm is basically similarity measures calculated between spatially adjacent pixels using the spectral information of these pixels. So it uses both spectral as well as spatial information to enhance the performance/accuracy. For similarity measure of information it uses Spectral Information Divergence(SID) thats well known in information theory domain.

So, After eliminating noisy bands this algorithm can be divided into 3 parts :

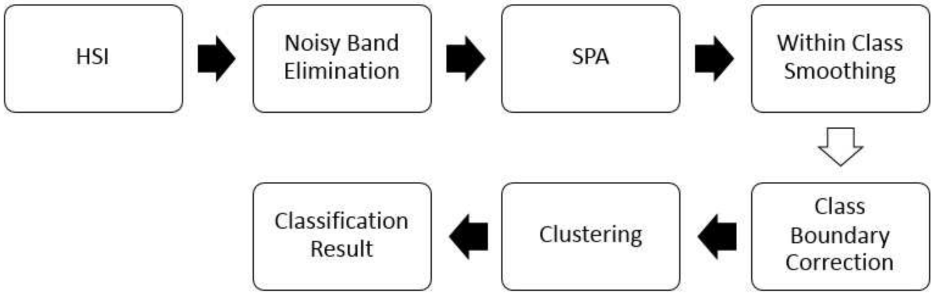

Figure 2.1: This shows the basic steps involved in implementing SPA algorithm in Hyperspectral Images.

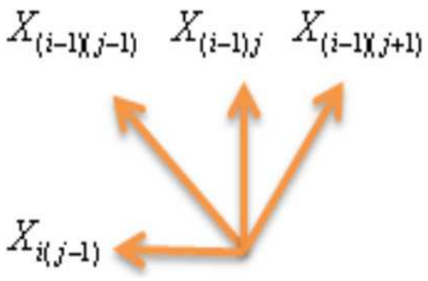

Figure 2.2: Getting the Maximum Spectral Similarity

- Location-wise Separation
- Class Boundary Correction (CBC)
- Final Clustering

2.1 Location-wise Clustering

The term explains its meaning. Its giving each pixel a score based on how related they are to their neighboring pixels. This relation is based on the information obtained from the spectral data.

Let the pixel located at (i,j)th location be X(i,j). So, there are eight pixels nearest to it. Of those eight, if we start from one corner it is only possible to compare that pixel with four of those neighbors because taking all eight ones there is going to be a unreliability in giving acquaintance. For example if we take the top left corner as our starting point, our nearest points taken into account here are $(i-1, j-1), (i-1, j), (i, j-1), (i-1, j+1)$ and from these four similarities only neighboring similarity is considered.

$$SS_{max} = max_i\{SS_i \ for \ all \ i = 1 \ to \ 4\}$$

After getting the maximum similarity of each pixel for its neighboring pixel, we are going to assign to which class current pixel belongs to. It should be the same as that of the neighboring pixel which had the most similarity. But there may happen a condition the maximum similarity is very less(lower than a threshold). For that case the current pixel is going to be assigned to a new class, and the process continues for the next pixel.

$$if\ SS_{max} >= threshold\ then\ class(i,j) = class(most\ similar\ pixel)$$
$$else\ class(i,j) = new\ class$$

2.2 Spectral Information Divergence (SID)

Now for similarity measures there are many there is Euclidean distance, SAM distance, etc but they fail in Hyperspectral dataset. So Spectral Information Divergence(SID) [16] that comes from Information theory is used here.

Let there be 2 pixels X_i and X_j. with $X_i = [d_{i1}, d_{i2}, ..., d_{iB}]$ and $X_j = [d_{j1}, d_{j2}, ..., d_{jB}]$ and $p^T = [p_1, p_2, , p_B]$ and $q^T = [q_1, q_2, , q_B]$, where p and q are probability mass function of X_i and X_j respectively. Then

- Self information from each band is :

  $I_b(X_i) = -log(p_b)$ and $I_b(X_j) = -log(q_b)$

- Diversity in Band 'b' is :

  $D_b(X_i||X_j) = I_b(X_j) - I_b(X_i)$

- over all bands:

  $D(X_i||X_j) = \sum_{q=1} D_q(X_i||X_j)$

- So to make a symmetrical measure

  $SID(X_i||X_j) = D(X_i||X_j) + D(X_j||X_i)$

2.3 Class Boundary Correction (CBC)

So using both SID and SPA measure we assigned each pixel to the grouping they belongs to. Now we have to smooth those labels, to remove those labels that do not have a good support from its neighbors. This is done to remove the problem of merged adjacent classes [16] due to the particular nature of Land Images. Lets assume that for each pixel there are less than equal 3 neighboring classes.

- *Case 1*: c1 pixels containing all only class c1 neighbors.
- *Case 2*: c1 pixels contains c2 or c3 pixel as one or more neighbors.
- *Case 3*: c1 pixels contains more neighbors of class c3 or c2.

Generally the pixels that are going to be most affected are the boundary pixels. So the class bound correction method tries to check it again to observe which class it belongs to.

- $if(central\ pixel = c1)\ \&\ (\ (case1\ |case2\ |case3))\ class(i,j) = c2$
- $if(centralpixel = c2)\ \&\ (c2\ neighbor < 2\ \&\ c3\ neighbor > 5)\ class(i,j) = c3$
- $if(centralpixel = c3)\ \&\ (c3\ neighbor < 2\ \&\ c2\ neighbor > 5)\ class(i,j) = c2$

2.4 Clustering

After finding class numbers from Boundary Correction algorithm, we have classes much more than real number of clusters, but also very few than the data points. So each class represents a group of data. Now, we take the average of each group and put those few values in a unsupervised classifier. Here K-Means is implemented for these problem due to its simple and less complex calculation.

After K-Means clustering all data points are assigned to the label given by K-Means.

2.5 Result

I used the threshold parameter to be 0.004 and the number of classes to be 6.

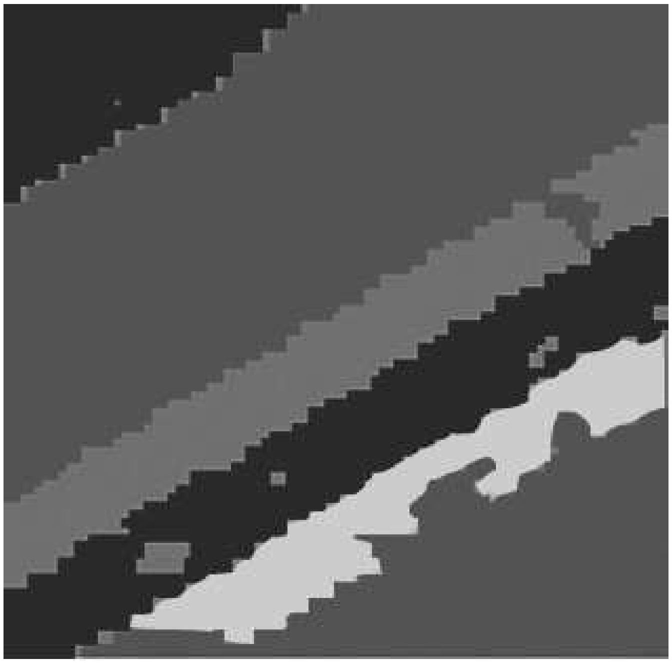

Figure 2.3: Result obtained for salinasA dataset using SPA algorithm.

Table 2.1: Confusion Matrix using UADNA

	1	2	3	4	5	6
1	389	0	0	1	0	1
2	0	840	18	487	2	6
3	0	12	58	499	20	7
4	0	5	21	1491	18	0
5	0	0	0	82	560	32
6	0	0	10	14	57	728

Table 2.2: Overall Accuracy of Location-wise Clustering using SID

Class	1	2	3	4	5	6	Overall
Accuracy	99.49	62.08	9.73	97.13	83.09	89.99	73.59

2.6 Summary

Here in this method I have implemented the Spatial Clustering of Hyperspectral Images where I have also used the location/spatial information to cluster the data, As there are mixed pixels in the image, so we have to take the information about the nearest classes present in the image to get the required information.

From the result, we found that the overall accuracy is 73.59 percent. This is due to the fact one of the class is not separated at all. This may result from the fact that it's nearest class is much similar to it.

Chapter 3

Unsupervised Artificial DNA Classifier

Large dimension is a difficulty for traditional algorithms and all bands are not relevant to all the clusters. So, here Unsupervised Artificial DNA clustering technique is introduced. In this method, certain irrelevant bands are neglected during calculation for each class. This classifier uses the following algorithm :

- All the spectral data are transformed/converted to DNA strands, which help in bot effective storing and comparison in low dimension space.
- The irrelevant bands are replaced with '0' code and are found using voting method.
- To calculate similarity between DNA strands a normalized spectral similarity measure is chosen.

Similar to a DNA molecule, these spectral data are converted to string of 4 values G,A,C,T. So data size is decreased as well as complexity also decreases. Key ideas to solve problems through Artificial DNA Computing is :

- Spectral Features are first encoded to DNA strands.

- These DNA strands are to be matched to obtain their cluster number.
- Then these DNA strands are recombined to solve the problem.

3.1 ADNA Model

DNA Strand: The whole data is composed of discrete codes similar to DNA as code(A,G,C,T).

- Length of DNA Code : $||code|| = 1\ if\ code = G, A, C, T$
- Length of DNA Strand: $||DNA|| = \sum_i ||code_i||$
- AND Operation: $\bigcap_{i=1}^{M} code_i = max_{code}(num_{code})\ if\ max_{code}(num_{code})/M > \gamma$ where γ is set as 0.5.

  Similarity Measure to be taken care of while using DNA codes.

- DNA Similarity : $||code_i \oplus code_j||_{DNA} = \left\{\begin{matrix} 1,\ if\ code_i = code_j \\ 0,\ otherwise \end{matrix}\right\}$
- DNA similarity in Real : $||code_i \oplus code_j||_{real} = \left\{\begin{matrix} 4,\ if\ code_i = G \\ 3,\ if\ code_i = A \\ 2,\ if\ code_i = C \\ 1,\ if\ code_i = T \end{matrix}\right\}$
- Measure of Similarity in DNA strands :

  $||DNA_i \oplus DNA_j||_{real/DNA} = \sum ||code_i \oplus code_j||_{real/DNA}$

3.2 Pre-processing for Spectral Image Data

Hyperspectral images consist of b bands and T pixels($T = R * C$). Pixels are represented by vector $D_i = [d_i^1, d_i^2, d_i^3, ..., d_i^b],\ i = 1, 2, 3, ...T$. The q^{th} band is represented as $D^q = [d_1^q, d_2^q, ..., d_T^q]$.

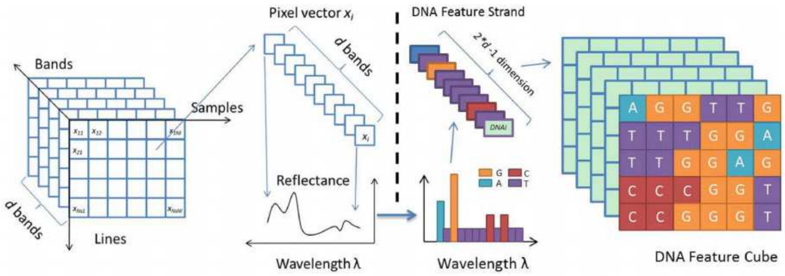

Figure 3.1: Preproceessing is done to get DNA features(from [17]).

DNA encoding for Spectra: DNA encoding is used to extract spectral reflection features. So, three feature encodings are done :

1. **Spectral Shape Feature Encoding.**

   The original spectrum data is used for ”spectral shape” encoding process. Let the pixel be $D_i = [d_i^1, d_i^2, d_i^3, ..., d_i^b]$, where $i = 1, 2, 3, ..., T$. Now all these values are sorted in ascending order. Then we have our minimum value as x_i^{min} and maximum value as x_i^{max}.
   So three quartile thresholds obtained are: $T_i^1 = x_i^{\frac{b\prime}{4}}$, $T_i^2 = x_i^{\frac{b\prime}{2}}$, $T_i^3 = x_i^{\frac{3b\prime}{4}}$.
   So for each pixel the j_{th} data is transformed as:

$$DNA_{i,j}^{shape_char} = \begin{Bmatrix} G, \ if \ x_i^j \epsilon [T_i^3, x_i^{max}] \\ A, \ if \ x_i^j \epsilon [T_i^2, T_i^3] \\ C, \ if \ x_i^j \epsilon [T_i^1, T_i^2] \\ T, \ if \ x_i^j \epsilon [x_i^{min}, T_i^1] \end{Bmatrix}$$

   ,

2. **Spectral Amplitude Feature Encoding.**

   Here DNA encoding is done for each band separately. Then the procedure of sorting is applied. Thresholds are taken and the whole band is changed to string of G,A,C,T. So, for each pixel there are b values.

3. **Spectral Slope Feature Encoding.**

   First the derivative of the features of Spectral Shape feature is found and then those slope features are used to do similar encoding as done in ”spectral shape” encoding. All these strings are stored in $DNA_i^{der_char}$ where $i = 1, 2, 3, ..., T$ represents different bands of image. Here are $b-1$ data for each pixel as derivative is taken.

3.3 Clustering Procedure

We get DNA strands from Hyperspectral Image Cube by using those preprocessing steps mentioned before. The shape is of measurement $R*C*b$ converts to $R*C$ DNA strands each of length $3d-1$. So unsupervised order can be determined via naturally examining and recognizing sensible features contained in the DNA strands.

It fundamentally Consists of :

- **Closeness Tests.**

  Similarity evaluates how near the DNA_i are to DNA_{DCs}. This Algorithm finds the DNA_{DCs} that is closest to the current DNA_i. Then index of the clustering center is assigned to the current DNA. Similarity Condition is given as :

$$max_i(\frac{||DNA^i_{DCs} \oplus DNA_i||}{||DNA^i_{DCs}||}) > \beta$$

- **Voting Process for overhauling DNAs.**

  The voting methodology is done to discover predictable DNA highlight codes in particular position. The are near to specific are utilized to vote in favor of new DCs in DNA encoding subspace.

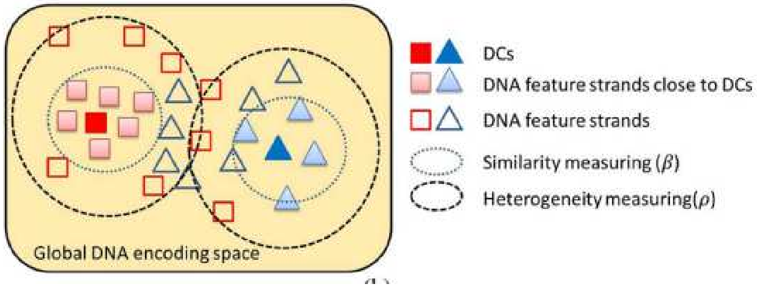
DCs
DNA feature strands close to DCs
DNA feature strands
Similarity measuring (β)
Heterogeneity measuring(ρ)
Global DNA encoding space

Figure 3.2: Concept Diagram of Similarity test and Heterogeneity test (from [17])

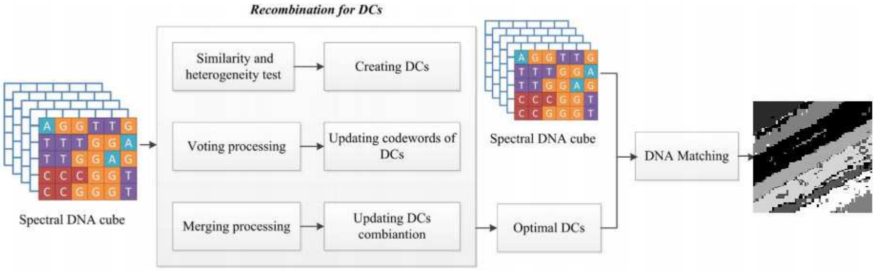

Figure 3.3: Basic Procedure for clustering using ADNA (from [17])

Here the ith code-word will be supplanted by $\cap_{q=1}^{3d-1}$: . The code-word that remaining parts same are stable code-words and the ones conflicting are supplanted with "0" code-word.

We set the parameter γ as 0.5.

The process is continued iteratively, clustering the current DNA strands with the updated DNA clustering centers and will only stop when the new ones are same as the old ones.

- **After Convergence, grouping of DNA strands to distinctive classes.** After voting more clusters are obtained than whats required. So simple clustering algorithm is used to classify the small number of data to the the required number of clusters. Here for simplicity, K-Means is used.

There are maintained few DNA "clustering centers" called DNA_{DCs}. These clustering centers are empty initially and are gradually filled up. Maximum number of clustering centers are fixed initially.

3.4 Result

Taking the value of $\alpha = 0.85$ and $\beta = 0.75$, we get the following result in salinasA dataset.

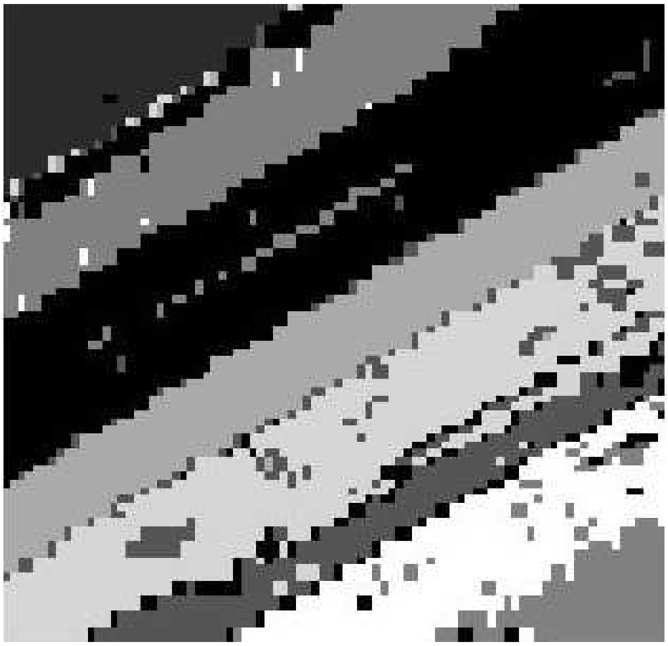

Figure 3.4: Different Class shown using UADNA Clustering technique

Table 3.1: Confusion Matrix using UADNA

GT/OUT	1	2	3	4	5	6
1	381	0	0	7	0	3
2	0	730	59	551	1	12
3	0	12	468	89	20	7
4	0	5	81	1431	18	0
5	0	0	0	2	640	32
6	0	0	10	14	6	769

Table 3.2: Overall Accuracy using UADNA

Class	1	2	3	4	5	6	Overall
Accuracy	97.44	53.95	78.52	93.22	94.96	96.25	85.72

3.5 Summary

This unsupervised Classifier uses method of Artificial DNA to both compress the data as well as easily cluster it. As it removes irrelevant bands while classification, it can be improved to get better accuracy. We observed that this method detects the class that was not earlier detected by the spatial based algorithm. The overall accuracy we get from this method is 85.72 percent.

Chapter 4

Maximal Margin Clustering using Hierarchical Tree

Maximal Margin Clustering, which is the unsupervised counter part of Support Vector Machine(SVM). As it tries to separate the data maximally, it generally gives better result than other methods as data is separated maximally. As the method is comparatively slow, so I am taking 20-25 best features from the data by first finding the Principal Component Analysis(PCA) of the image data. In general MMC's aim is to optimize the problem :

$$min_y \ max_{\alpha,\alpha^*}(\alpha-\alpha^*)^T y - \tfrac{1}{2}(\alpha-\alpha^*)K(\alpha-\alpha^*)$$
$$s.t. \ (\alpha-\alpha^*)^T = 0$$
$$0 <= \alpha, \alpha^* <= Ce.$$

4.1 Iterative SVR

Now, this problem can be solved by first taking y constant, solving the dual of the Iterative SVR problem taking parameters as α,α^*. Then we put α,α^* constant and minimize the value of the objective function.

As this algorithm tends to find the maximum distance, so the trivial or most optimal solution according to the computer is to classify all data as one cluster, so margin is infinite. For this reason to check and remove the trivial solution a class balance constraint parameter is used here. Let that variable be r. The statement can be represented as $-\frac{r}{n} <= \sum_{i=1}^{n} y_i <= \frac{r}{n}$. This prevents the trivial solution.

After fixing α,α^* fixed, we have w fixed, so our problem is reduced to

$$min_{y,b} \; | < w, \phi(x) > \; + b \; - \; y_i|$$

So, we can now simply shift b in range $-rn$ to rn so that our problem is optimized and also the maximum balance constraint are satisfied.

4.2 Hierarchical Structure for Multiple Classes

As this Iterative SVR can only solve binary clustering problem and we don't have training data with us to create one-vs-one or one-vs-all classifier, we have to go for some other method to separate all the classes.As the clusters separated using Iterative SVR must contain at least one class and the separation margin is maximum in the whole dataset. Then we have 2 clusters, each containing more than or equal to one class.

Let d be a function that defines how well the clusters are separated.
So, now we maintain a *Tree*, whose leaf nodes are going to be clustered. The leaf node with minimum value of d is going to be clustered and the process continues iteratively until the number of leaf nodes reaches the given number of class(c). (Here d is taken to be the cross validation after clustering using k-NN algorithm) Each of the final leaf nodes constitutes a unique label.

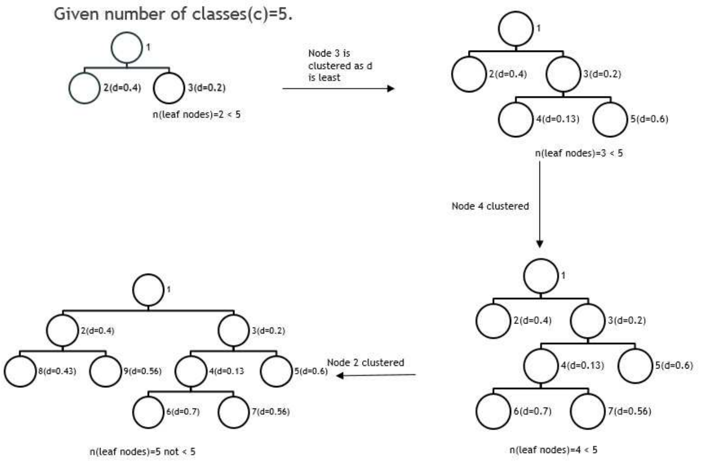

Figure 4.1: Hierarchical Structure used with MMC for Multiple Class Clustering

4.3 Result

Here we had tested this algorithm with both the datasets.The parameter r is taken here to be 0.3 and ϵ is taken to be 0.1. The results are shown below.

Table 4.1: Confusion Matrix using MMC with Hierarchical Tree

GT/OUT	1	2	3	4	5	6
1	389	0	0	1	0	1
2	0	739	0	537	5	12
3	0	0	563	53	0	0
4	0	0	0	1525	0	0
5	0	0	0	0	672	2
6	0	0	0	3	11	785

Table 4.2: Comparison of Overall Accuracy for UADNA, H2NMF, MMC methods

Method/Class	1	2	3	4	5	6	Overall
UADNA	97.44	53.95	78.52	93.22	94.96	96.25	85.72
H2NMF	99.49	54.15	3.73	100	97.92	98.77	75.69
MMC_HT	99.49	55.03	91.40	100	99.70	98.25	90.64

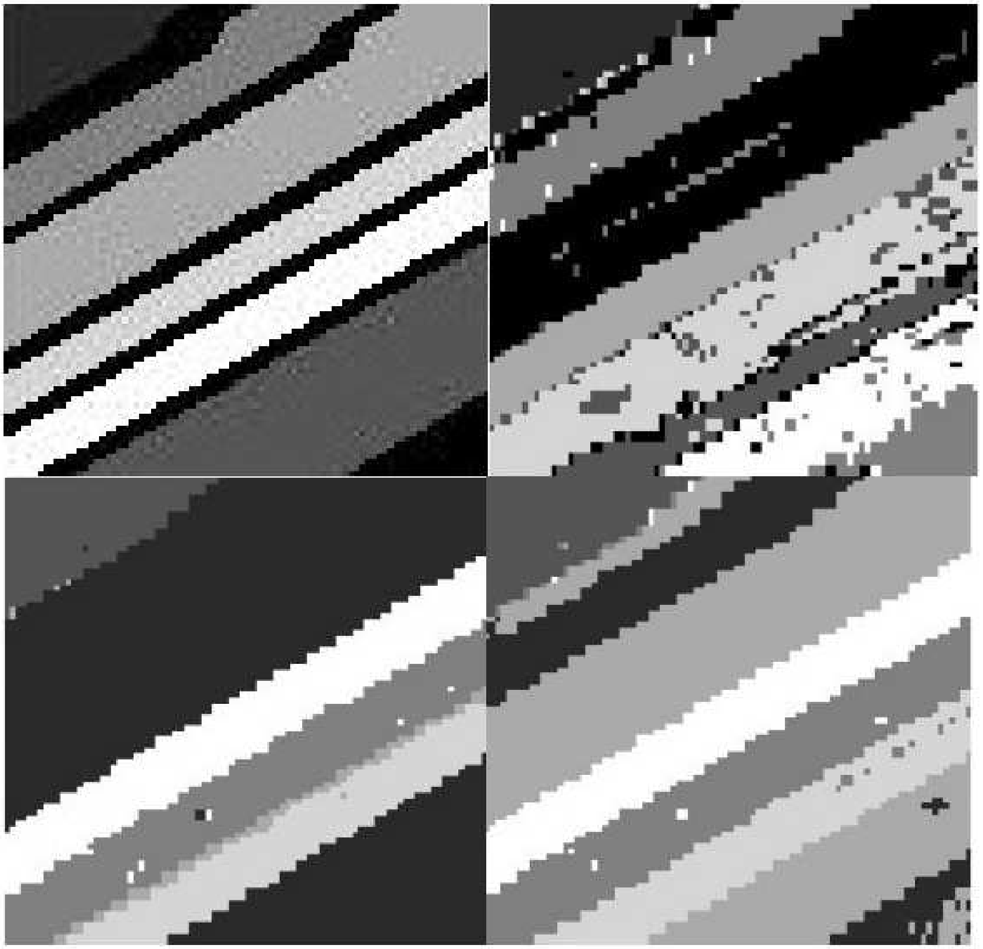

Figure 4.2: From top left (a) Ground Truth. (b) Clustering using Unsupervised DNA Classifier(using $\alpha = 0.85$ and $\beta = 0.75$) (c) Clustering using Hierarchical Rank-2 NMF. (d) Clustering using MMC using Hierarchical Tree with $r = 0.3$.

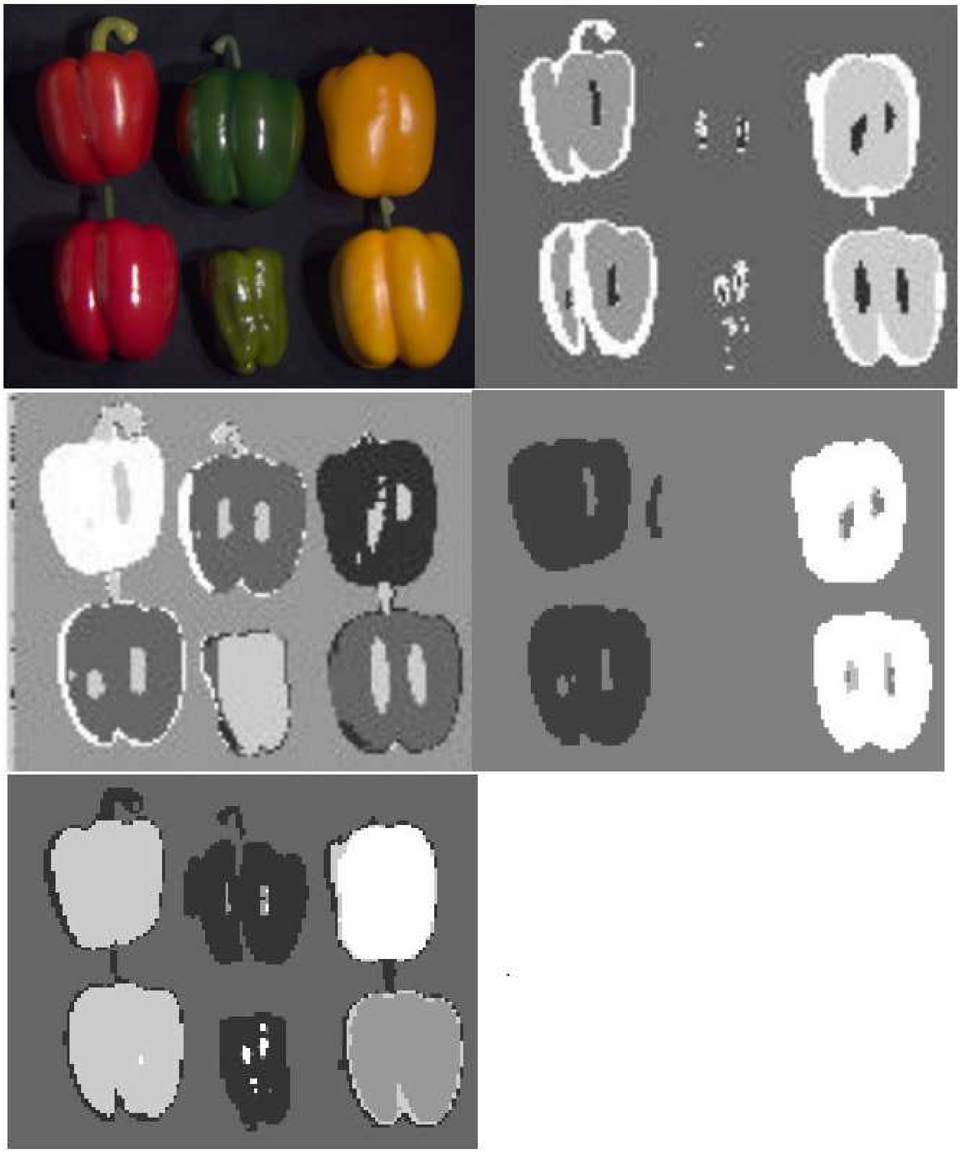

Figure 4.3: From top left (a)Image in RGB Format. (b)Clustering done through K-Means and PCA. (c) Clustering through Gaussian Expectation Maximization. (d) Clustering through Hierarchical Rank-2 NMF. (e) Clustering through MMC using Hierarchical Tree with $r = 0.3$.

4.4 Summary

Maximal Margin Clustering is a really challenging task for computers as it computationally intensive and the exact algorithm is an exponential one. So, we managed to reduce the algorithm to an iterative polynomial one at the cost of little approximations and reduced accuracy. Both fruit database and salinasA dataset are used here.

The result shown from this method surpasses the others mentioned in this thesis, but they also have a disadvantage of taking more time to execute. This can be basically termed as an unsupervised Support Vector Machine. Here, we obtain an overall accuracy of 90.64 percent, which is good for a unsupervised method. For the fruit dataset, we don't have any truth case. So only the visual result are shown.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

Unsupervised Classification/Clustering is a more difficult challenge when compared to supervised Classification, but it has its merit and is essential when no truth case is given. This is a totally automated technology. No Human interaction is needed except for setting some parameters.

In this project, I started with the basic clustering approach, used spatial location data along with spectral data to increase accuracy. As this method uses the spectral Information Divergence along with the neighbors data, we can say that it somewhat removes the mixed pixel problem.

We also tried to implement new algorithm that has theoretical base on DNA mechanism. Its inspired by our DNA molecules that recombines to get a better configuration. This mechanism is computationally intensive than the previous one and parameters like α and β has to be adjusted to get better result.

Finally, I implemented a Maximum Margin Clustering which basically clusters data to two sets. The parameter we have to change are basically ϵ which is generally taken as 0.1 and class balance constraint ratio r which everywhere i have taken as 0.3. So, we have to use a hierarchical tree to get multiple clusters.

Finally, with the limited dataset I had, we could see that the final one gave the best result of all.

5.2 Future Work

Maximal Margin Clustering is a really slow method. So, it has difficulty in high resolution images. Hence, In future my aim would be to decrease the complexity of the algorithm. Also the cross checking measure used is not that good. So, had to try for better functions to cross-check.

Bibliography

[1] C. L. Chan, A. K. Katsaggelos, and A. V. Sahakian, "Image Sequence Filtering in Quantum-Limited Noise with Applications to Low-Dose Fluoroscopy," *IEEE Transactions on Medical Imaging*, vol. 12, no. 3, pp. 610 – 621, September 1993.

[2] C. M. Bishop *et al.*, *Pattern recognition and machine learning.* springer New York, 2006, vol. 4, no. 4.

[3] Y. Zhong, L. Zhang, B. Huang, and P. Li, "An unsupervised artificial immune classifier for multi/hyperspectral remote sensing imagery," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 44, no. 2, pp. 420–431, 2006.

[4] U. Maulik and I. Saha, "Automatic fuzzy clustering using modified differential evolution for image classification," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 48, no. 9, pp. 3503–3510, 2010.

[5] A. Mukhopadhyay and U. Maulik, "Unsupervised pixel classification in satellite imagery using multiobjective fuzzy clustering combined with svm classifier," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 47, no. 4, pp. 1132–1138, 2009.

[6] M. D. Farrell and R. M. Mersereau, "On the impact of pca dimension reduction for hyperspectral detection of difficult targets," *Geoscience and Remote Sensing Letters, IEEE*, vol. 2, no. 2, pp. 192–195, 2005.

[7] J. Wang and C.-I. Chang, "Independent component analysis-based dimensionality reduction with applications in hyperspectral image analysis," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 44, no. 6, pp. 1586–1600, 2006.

[8] S. Kaewpijit, J. Le Moigne, and T. El-Ghazawi, "Automatic reduction of hyperspectral imagery using wavelet spectral analysis," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 41, no. 4, pp. 863–871, 2003.

[9] H. Shi, Y. Shen, and Z. Liu, "Hyperspectral bands reduction based on rough sets and fuzzy c-means clustering," in *Instrumentation and Measurement Technology Conference, 2003. IMTC'03. Proceedings of the 20th IEEE*, vol. 2. IEEE, 2003, pp. 1053–1056.

[10] L. O. Jimenez and D. A. Landgrebe, "Hyperspectral data analysis and supervised feature reduction via projection pursuit," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 37, no. 6, pp. 2653–2667, 1999.

[11] L. Zhang, Y. Zhong, B. Huang, J. Gong, and P. Li, "Dimensionality reduction based on clonal selection for hyperspectral imagery," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 45, no. 12, pp. 4172–4186, 2007.

[12] Y.-Q. Zhao, D. Zhang, and S. G. Kong, "Band-subset-based clustering and fusion for hyperspectral imagery classification," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 49, no. 2, pp. 747–756, 2011.

[13] S. Lee and M. M. Crawford, "Unsupervised multistage image classification using hierarchical clustering with a bayesian similarity measure," *Image Processing, IEEE Transactions on*, vol. 14, no. 3, pp. 312–320, 2005.

[14] Wikipedia, "K-means clustering — wikipedia, the free encyclopedia," 2015, [Online; accessed 9-May-2015]. [Online]. Available: http://en.wikipedia.org/w/index.php?title=K-means_clustering&oldid=660233806

[15] ——, "K-nearest neighbors algorithm — wikipedia, the free encyclopedia," 2015, [Online; accessed 9-May-2015]. [Online]. Available: http://en.wikipedia.org/w/index.php?title=K-nearest_neighbors_algorithm&oldid=661537193

[16] B. Baassou, M. He, S. Mei, and Y. Zhang, "Unsupervised hyperspectral image classification algorithm by integrating spatial-spectral information," in *Audio, Language and Image Processing (ICALIP), 2012 International Conference on*. IEEE, 2012, pp. 610–615.

[17] H. Jiao, Y. Zhong, and L. Zhang, "An unsupervised spectral matching classifier based on artificial dna computing for hyperspectral remote sensing imagery," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 52, no. 8, pp. 4524–4538, 2014.

[18] N. Gillis, D. Kuang, and H. Park, “Hierarchical clustering of hyperspectral images using rank-two nonnegative matrix factorization,” 2013.

[19] K. Zhang, I. W. Tsang, and J. T. Kwok, “Maximum margin clustering made practical,” *Neural Networks, IEEE Transactions on*, vol. 20, no. 4, pp. 583–596, 2009.

[20] H. Drucker, C. J. Burges, L. Kaufman, A. Smola, V. Vapnik *et al.*, “Support vector regression machines,” *Advances in neural information processing systems*, vol. 9, pp. 155–161, 1997.

[21] B. Schölkopf, P. Simard, V. Vapnik, and A. Smola, “Improving the accuracy and speed of support vector machines,” *Advances in neural information processing systems*, vol. 9, pp. 375–381, 1997.

[22] H. Jiao, Y. Zhong, and L. Zhang, “Artificial dna computing-based spectral encoding and matching algorithm for hyperspectral remote sensing data,” *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 50, no. 10, pp. 4085–4104, 2012.

[23] J. R. Jensen and K. Lulla, “Introductory digital image processing: a remote sensing perspective,” 1987.

[24] Y. Yan, Y. Zhao, H.-f. Xue, X.-d. Kou, and Y. Liu, “Integration of spatial-spectral information for hyperspectral image classification,” in *Geoscience and Remote Sensing (IITA-GRS), 2010 Second IITA International Conference on*, vol. 1. IEEE, 2010, pp. 242–245.

[25] Y. Du, C.-I. Chang, H. Ren, C.-C. Chang, J. O. Jensen, and F. M. DAmico, “New hyperspectral discrimination measure for spectral characterization,” *Optical Engineering*, vol. 43, no. 8, pp. 1777–1786, 2004.

[26] A. Plaza, J. A. Benediktsson, J. W. Boardman, J. Brazile, L. Bruzzone, G. Camps-Valls, J. Chanussot, M. Fauvel, P. Gamba, A. Gualtieri *et al.*, “Recent advances in techniques for hyperspectral image processing,” *Remote sensing of environment*, vol. 113, pp. S110–S122, 2009.