EDITED-BOOTSTRAPPED SUPPORT VECTOR MACHINES FOR ONE-CLASS DATA CLASSIFICATION

By

Anne Krishna Sravanthi

Bachelor of Technology in Electronics and Communication Engineering

Nagarjuna University

Guntur, India

2003

Submitted to the Faculty of the Graduate College of the
Oklahoma State University in partial fulfillment
of the requirements for the degree of
MASTER OF SCIENCE

July 2006

EDITED-BOOTSTRAPPED SUPPORT VECTOR MACHINES FOR ONE-CLASS DATA CLASSIFICATION

Thesis Approved

Dr. Guoliang Fan (Thesis Advisor)

Dr. Rafael Fierro

Dr. Mahesh Rao

Dr. Gordon Emslie (Dean of the Graduate College)

ACKNOWLEDGEMENTS

I would like to express my most sincere thanks to my advisor, Dr. Guoliang Fan, for his guidance, support, encouragement and beyond all, his magnanimity in excusing my mistakes throughout my MS study. He has imparted not only technical knowledge, but also a rigorous attitude towards research. I would also like to acknowledge Dr. Xiaomu Song and Ginto Cherian for their guidance. I would like to express my thanks to Dr. Mahesh Rao for providing the data and help on Remote Sensing . Many thanks to Dr. Rafael Fierro for showing interest on this research and accepting me as is student. I would also like to thank my colleagues in Visual Communication Research Lab for supporting me during various stages of my thesis.

I would like to thank my best friend Chetan Yedati for helping me in writing the report and off all being very patient on my deeds. I would also like to thank my friends Nalini, Puja, Neepa and my brother Sai for their encouragement and most of all making my life happy during my stay at OSU. Last but not the least I want to thank my friend Mohan for supporting me and for taking care of me. I dedicate this work to my Parents, my Sister and Brother-in-law, and all my other close relatives for constantly motivating me and providing moral support.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

The main objective of this research is concentrated mainly on one-class data classification. In general, the remote sensing data consists of samples belonging to different classes that are highly overlapped. Classifying such kind of remote sensing data is an essential part of many remote sensing applications. If the features of all the classes are known then the remote sensing data can be classified without much difficulty by applying any multi class classifiers. But the data has to be classified by one-class classifiers if the features of only one class (target class) are known. The aim of one-class classification is to characterize one class of data from the rest in the feature space. The difficulty lies in deciding the boundary that fits around the data in each of the feature directions based on only one class data. Other advantage of one-class classification is that a multi-class problem can be converted into a simple multiple one-class problem. The first and the foremost challenge in one-class remote sensing analysis is in choosing a classifier. This is because, as mentioned earlier the remote sensing data consists of different classes that are highly overlapped resulting in linearly non-separable case. Second challenge is on choosing a perfect training data set. Training data set affects the

performance of the classifiers as a classifier learns from training data.

Many classifiers such as maximum likelihood, artificial neural networks are used in remote sensing. Neural classifiers classify the data accurately but there are some factors, which limit their use. Recently support vector machines (SVMs) are used to overcome the limits of the neural classifiers. SVMs are now widely used in remote sensing applications [9], [10] and are proved to achieve a higher level of classification accuracy than the other classifiers [11], [12]. SVMs were derived from statistical learning theory [13]. SVMs separate different classes by finding optimal classification hyper planes from training, leading to better generalization capability compared with other methods. SVMs are advantageous especially when there are only small training samples available and also in cases where training samples of only one class are available. SVM is also referred to as two-class SVM (TCSVM) [13], [14]. TSVM generates a hyperplane that maximizes the distance between two different patterns in the feature space and thus resulting in a good generalization performance. TCSVM is a supervised learning algorithm that requires training samples and cannot be applied to one-class remote sensing applications. Therefore, one-class SVM (OCSVM), which is an unsupervised classifier, is used in this research. OCSVM was developed for detecting outliers [15], [16]. OCSVM estimates a boundary that separates the majority data from the outliers. It was shown in [15] that OCSVM produces comparable or superior results over traditional classifiers. The parameter ν in OCSVM, defined as an upper bound on the fraction of outliers, is usually unknown and effects the classification results significantly.

The idea in any classifiers is to learn from a given set of data points and to predict the labels of the data points that are to be tested. If outliers are present in a learning data set then classifiers tend to pick them as potential support vectors, which in turn degrades the classification. Bootstrapping is defined as a process to create pseudo replicate datasets by re-sampling. Bootstrapping reduces the effect of outliers in the training data set during classification. Different bootstrapping

techniques have been proposed in [17] for Nearest Neighbor (NN) classifier and are proved to produce higher classification accuracies than the conventional classifiers. Beyond the bootstrapping techniques there are other different methods like condensing and editing [18], [19] that were proposed to eliminate the effect of noisy data points during classification. Editing is a process that eliminates outliers and overlapped data points from the training data set where as condensing aims in finding a minimal consistent subset of training data set that will be sufficient to classify the remaining data samples correctly.

Based on the idea of editing we proposed a new algorithm, which can be combined with one-class classifier i.e., OCSVM. The main idea of this algorithm is to purify the training data set by removing other data points that are of no interest. This algorithm along with OCSVM helps us in achieving our goals with a higher classification results. Our algorithm was applied to compliance monitoring and mapping of United States Department of Agriculture (USDA)'s Conservation Reserve Program (CRP) tracts based on Landsat imagery and other multisource GIS data. CRP is a voluntary program for agricultural landowners in which farmers are encouraged to plant long-term native plants for a period of 10-15 years. In year 2005, USDA paid around 1.69 billion dollars as annual rental payments to farmers. USDA has to monitor the lands to check whether the farmers are maintaining the CRP tracts according to the contract stipulations. The reference data provided by the UDSA is not very accurate or up-to-date for management purposes. Two different methods have been proposed for CRP monitoring [20], [21] in which TCSVM and OCSVM are used for CRP classification. A localized block-based method [8] was proposed to achieve CRP mapping. These two remote sensing issues have been combined through proposed edited-bootstrapped SVM. This reduces the cost and time for CRP management in maintaining the CRP lands.

This chapter gives an overview of the classification algorithm, Support Vector Machines (SVMs), and also some Bootstrapping techniques that were used

to enhance the performance of the Nearest Neighbor Rule. Finally, it also discusses briefly the proposed edited-bootstrapped SVM and its advantages.

1.2 Support Vector Machines (SVMs)

SVMs were derived from statistical learning theory [13]. It was first introduced in 1992. SVMs are now regarded as an important example of "kernel methods", arguably the hottest area in machine learning. Machine learning algorithm is the ability of a computer algorithm to recognize patterns that have occurred repeatedly and to improve its performance based on past experiences. SVMs separate different classes by finding optimal classification hyper planes from training, leading to better generalization capability compared with other methods. In SVMs learning, kernel methods are often used to map the data vectors in the input space into a higher dimension feature space thereby converting linearly non-separable case into a separable case. The construction of a linear classification hyper plane in this high dimension feature space is equivalent to a nonlinear decision hyperplane in the input lower dimension space.

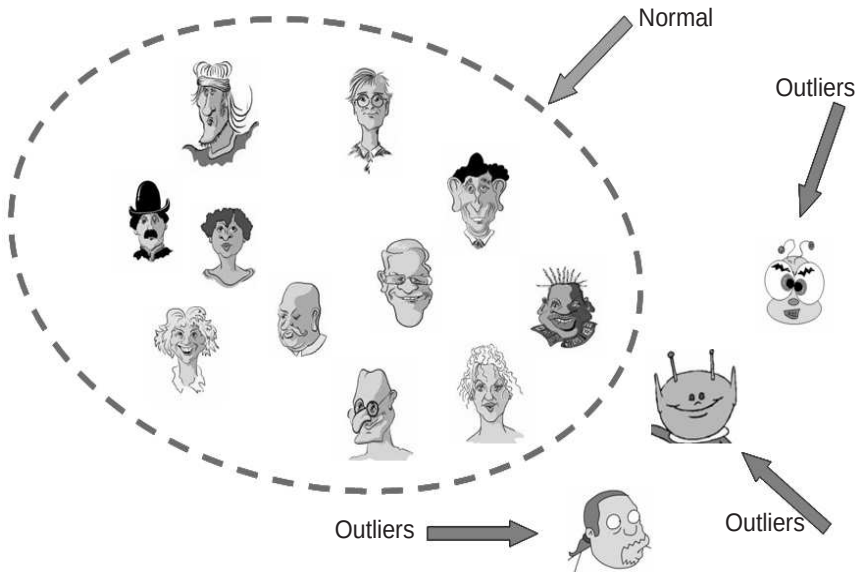


Figure 1.1: Application of Support Vector Machines (SVMs).

Figure 1.1 represents one of the application of SVMs. Common faces of the people are taken to be the training data set and the faces outside the circle represents the outliers. One-Class SVM (OCSVM) and Two-Class SVM (TCSVM) are the two different types of SVMs.

1.2.1 Two-Class SVM (TCSVM)

TCSVM is the regular SVM, which is a supervised classification. The training data in the two-class classification consists of the information about both the classes. TCSVM is further classified into two different types depending on the distribution of the data. The simplest case is the linear separable case. Here the feature vectors are assumed to be separable and can be separated by a linear decision boundary. But in most classification cases data are not separable. Non-linear separation boundaries are needed for solving general-purpose problems. SVM solves this problem by non-linearly transforming the input feature space (by kernel mapping) into a high dimensional feature space, where a linear separation is done.

1.2.2 One-Class SVM (OCSVM)

In one-class classification only the information about the target class is available. The OCSVM algorithm maps input data into a high dimensional feature space (via kernel) and iteratively finds the maximal margin hyperplane which best separates the training data from the origin. The OCSVM may be viewed as a regular two-class SVM where all the training data lies in the first class, and the origin is taken as the only member of the second class. Thus the hyperplane (or linear decision boundary) corresponds to the classification rule: $f(x) = \langle w, x \rangle + b$.

Figure 1.2 described in [1] illustrates the difference between Two-class SVM and One-class SVM. Each object has two feature values viz. weight and width. The two classes can be separated without errors by the solid line in Figure 1.2,

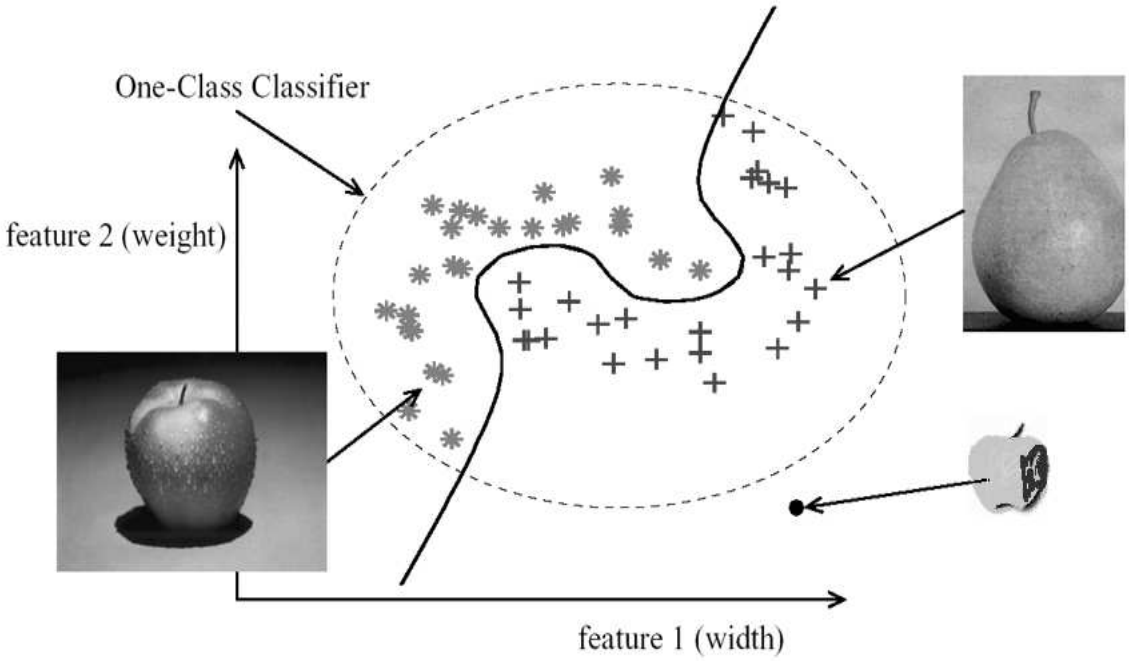


Figure 1.2: A figure taken from [1] illustrates the idea of OCSVM.

which is the normal two-class classifier as the training patterns for apples and pears are available. Consider now that a new pattern of a rotten apple in the lower right corner is introduced. It cannot be distinguished from the pears. Thus for a two-class classifier all patterns have to be either regular apples or pears and anything else won't be classified correctly. However the one-class classifier (denoted by dotted line) can differentiate the outlier after training. SVMs and different classes of SVMs will be discussed in detail in chapter 2.

1.3 Bootstrapping techniques

Bootstrapping is a process of creating pseudo replicate data sets by re-sampling in order to reduce the effect of outliers. It generates the new bootstrap samples either by re-sampling or by locally combining original training data samples. This method not only reduces the outliers but also brings the data points closer. In the conventional bootstrapping techniques the size of the new training data set in

unaltered. Therefore those techniques give smoothing of the distribution of the training samples. Different techniques have been proposed in [17] to calculate the bootstrapping samples. Similar to these bootstrapping techniques we proposed edited-bootstrapped SVM that removes outliers from the training samples in each iterations thereby purifying the training data.

Some Bootstrapping techniques which are tested using Nearest Neighbor (NN) algorithm are discussed in detail in chapter 3 and the purpose of new edited-bootstrapped SVM, its algorithm and some simulation results are discussed in chapter 4.

1.4 Conservation Reserve Program (CRP)

The proposed edited-bootstrapped SVM is tested on real data pertaining to a land that has been enrolled in Conservation Reserve Program (CRP). The US Department of Agriculture's (USDA's) Conservation Reserve Program (CRP) is a voluntary program for agricultural landowners. Through CRP, you can receive annual rental payments and cost-share assistance to establish long-term, resource conserving covers on eligible farmland. The program is administered by the Commodity Credit Corporation (CCC) through the Farm Service Agency (FSA), and program support is provided by Natural Resources Conservation Service, Cooperative State Research and Education Extension Service, state forestry agencies, and local Soil and Water Conservation Districts. The information about the CRP program is available at the CRP website [2].

The CCC makes annual rental payments based on the agriculture rental value of the land, and it provides cost-share assistance for up to 50 percent of the participant's costs in establishing approved conservation practices. Participants enroll in CRP contracts for 10 to 15 years. These CRP tracts have to be maintained according to CRP contract stipulations, which specify that the land cannot be used

for commercial purposes except for haying or gazing during the weather-related emergencies. In return annual rental payments are made to the farmers by USDA.

Figure 1.3: A figure taken from [2] represents a CRP land.

A land is chosen for the enrollment in the CRP program basing on the following factors,

- A cropland that has been planted or considered planted for 4 of the last 6 years.
- A cropland that is cultivable.
- A cropland must have a weighted average Erosion Index of 8 or greater.

Lands enrolled in the CRP program have to adhere to contract stipulations. Mainly the land has to be planted with native vegetation usually grasses. In some cases even trees are allowed. Native vegetation is preferred to improve wildlife habitats. Also farming is not allowed during the contract period.

1.5 Remote Sensing Data Analysis for CRP

1.5.1 CRP Compliance Monitoring

Though the farmers enroll their lands for CRP program some of them still continue to cultivate their lands, thereby breaking the CRP contract rules. Thus USDA has to monitor the lands frequently and has to take care that farmers are maintaining the CRP lands according to the contract stipulations. So there is a need to make sure that enrolled CRP lands are maintained properly known as compliance monitoring. Initially this problem involved manual inspection of aerial photographs which is time-consuming and costly.

Two methods were proposed to handle the problem of CRP compliance monitoring. [20] [21] discuss about the proposed algorithms on CRP compliance monitoring. Compliance monitoring methods are made by combining the OCSVMs as the first stage and the general SVM's as the second stage. Difference between the two methods lies in the way they select reliable training samples from the first stage for training the second stage. Two kernel space based methods were used for reliable sample estimation. Some machine learning approaches like decision tree also have been applied to remote sensing data as discussed in [8] [22].

1.5.2 CRP Mapping

The CRP reference data provided by the USDA's Natural Resource Conservation Service (NRCS) is not very accurate or up-to-date for management purposes. Usually, major errors in the present CRP reference data are due to the miss-location and/or misalignment of CRP tracts. This is due to the fact that the reference data is considerably old and it is possible that there have been new CRP enrollments or that old enrollments have expired and returned to agriculture. Previously CRP maps were developed based on the information provided by farmers upon

enrollment into the program and by manual delineation of aerial photographs.

An approach for accurate CRP mapping based on multi-seasonal and multi-year Land sat TM imagery is discussed in [23] [24]. An unsupervised classification was performed first to create crop and grass maps. Then after labeling these clusters manually, the CRP tracts were extracted by a post-classification comparison technique, where the areas with changed cover types can be detected. Although high classification accuracy had been achieved by this approach, the dependency on intensive human skill and labor might limit its efficiency and effectiveness in practical applications to large scale.

In [8], CRP mapping was implemented through two machine learning approaches i.e., decision tree classifier (DTC) and support vector machine (SVM). CRP mapping was implemented using pre-clustering to combine different CRP cover types and combinations of multiple OCSVMs trained on different CRP cover types.

1.6 Organization and Contributions

Chapter 2 describes the learning algorithm Support Vector Machine that plays the main role in this research. It also describes the different types of SVM's and also applications of SVM. Chapter 3 gives the idea of various Bootstrapping techniques proposed for the Nearest Neighbor (NN) algorithm. Chapter 4 describes the idea of proposed edited-bootstrapped SVM. Chapter 5 discusses the application of proposed algorithm to the CRP data. Chapter 6 is the conclusion.

The contributions of this thesis are mainly to implement the Bootstrapping techniques (discussed in chapter 3 and 4) on SVM to improve the performance of the algorithm. Based on the idea of editing algorithm new bootstrapping algorithm is proposed. The new bootstrapping technique is proposed in order to eliminate the problems faced in one-class remote sensing analysis. The algorithm performs

well even in the cases where the land area is subjected to change during different times of the year. For example, the problems of the CRP as mentioned in section 1.5 can be handled by the proposed algorithm. The other main advantage is that with the help of this new technique both the CRP problems (CRP monitoring and CRP Mapping) can be combined. The proposed algorithm is tested on texture images, random numerical data and also on the real data (discussed in chapter 5).

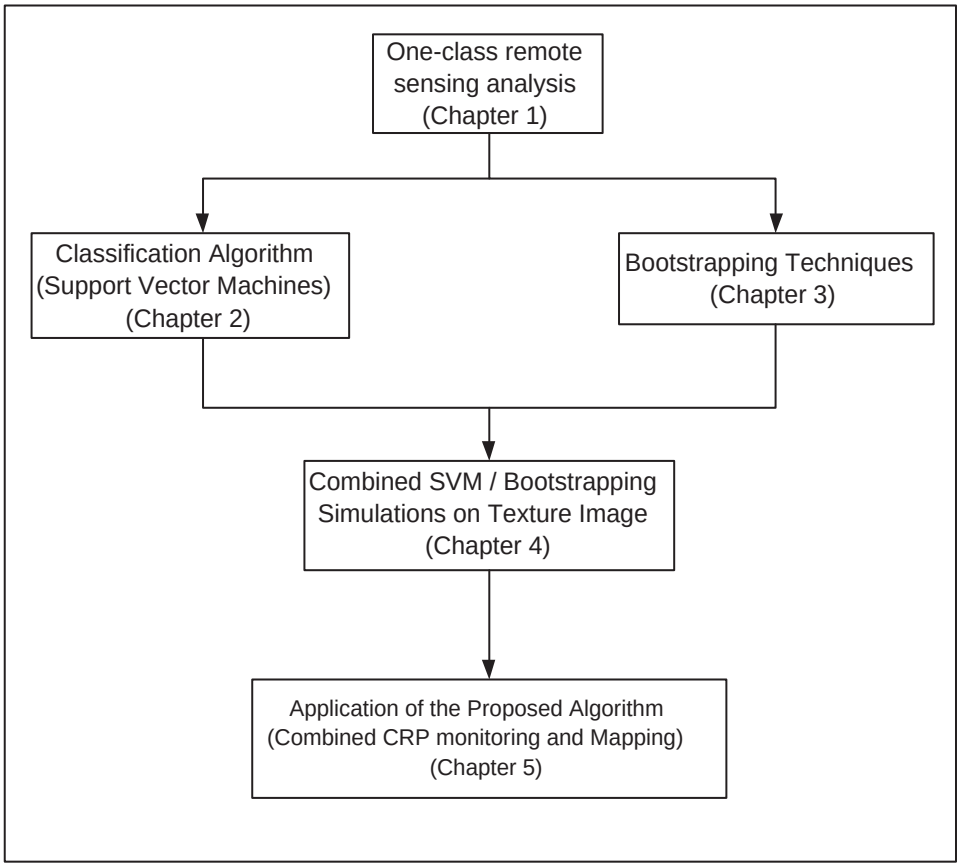


Figure 1.4: Outline of the thesis.

Chapter 2

Support Vector Machine Algorithms

Support Vector Machines (SVMs) were originated from statistical learning theory [13]. The concept of SVMs was introduced by V. N. Vapnik in the late 1970's. SVMs are the general purpose binary classification algorithm based on the margin maximizing principle. SVMs have wide applications in the fields like pattern recognition, regression analysis and density estimation discussed in [9] [25] and [26].

This chapter discusses the idea of general SVMs and different types of SVMs. It also provides a brief review on Statistical Learning Theory as this forms the main foundation for the development of SVMs.

2.1 Statistical Learning Theory

The statistical Learning theory forms the mathematical foundation to all the learning algorithms. This section gives a brief idea of statistical learning theory discussed in [13] [27] [3].

"The main goal of statistical learning theory is to provide a framework for studying the problem of inference, that is of gaining knowledge, making predictions, making decisions or constructing models from a set of data". The definition of learning theory can be defined as the following steps:

- Observe a phenomenon.
- Construct a model of that phenomenon.
- Make predictions using this model.

Two main assumptions are made by the statistical learning theory. The first assumption is that the future observations (i.e. test) and the past (i.e. training) ones are related. If there is no relation then prediction is impossible. Secondly the past and future observations are sampled independently from the same distribution (i.i.d) within the model. For a given training data a perfect function can be defined but the presence of noise leads to a bad prediction on future ones.

2.1.1 Formulation and Algorithms

Consider an input space $\mathcal{X}$ and output space $\mathcal{Y}$. As we consider binary classification $\mathcal{Y} = \{-1, 1\}$. According to the assumptions mentioned earlier the pairs $(x_i,y_i) \in \mathcal{X} \times \mathcal{Y}$ are the n i.i.d training samples distributed according to an unknown distribution P.

Now a function $f(x, \alpha)$ has to be built to predict the testing samples. Where $\alpha \in \Lambda$ is the set of function parameters also referred as the complexity of the function. This function is constructed such that the probability of error is low. Thus the *risk* is defined as

$$R(\alpha) = \int \frac{1}{2}|y - f(x, \alpha)|dP(x, y). \tag{2.1}$$

But the conflict in finding the function f is that P is unknown and thus risk cannot be calculated. In such a case *empiricalrisk* is calculated and is defined as

$$R_{emp}(\alpha) = \frac{1}{2n}\sum_{i=1}^{n}|y - f(x,\alpha)|. \tag{2.2}$$

The *empiricalrisk* can be minimized by considering infinite input space but *risk* will be maximized. However for small sample sizes large deviations are possible and overfitting may occur. Restricting the complexity of the function $f(x,\alpha)$ can avoid the problem of overfitting.

For such a learning problem the error bounds defined in [3] will apply. For any $\delta > 0$, with probability at least 1-δ the error bound is defined as:

$$R(\alpha) \leq R_{emp}(\alpha) + 2\sqrt{2\frac{\log \mathcal{S}_{\mathcal{G}}(2n) + \log\frac{2}{\delta}}{n}}. \tag{2.3}$$

where $\mathcal{S}_{\mathcal{G}}(n)$ is the VC dimension.

2.1.2 VC Dimension

In our case of binary classification the function $f(x,\alpha) \in \{-1,1\}$ VC dimension $\mathcal{S}_{\mathcal{G}}(n)$ is the maximum number of training points for which the labels can be correctly assigned in 2^n ways. That implies that VC dimension of a class $\mathcal{G}$ is the size of the largest set that it can shatter. Therefore the VC dimension is given as $\mathcal{S}_{\mathcal{G}}(n) = 2^n$.

For example, from the figure 2.1 it can be observed that a set of 3 points in dimension 2 can be shattered whereas set of 4 points cannot be shattered. In general one can shatter a set of d+1 points but no set of d+2 points. Therefore the VC dimension is d+1.

VC dimension is distribution independent, which implies same error bound

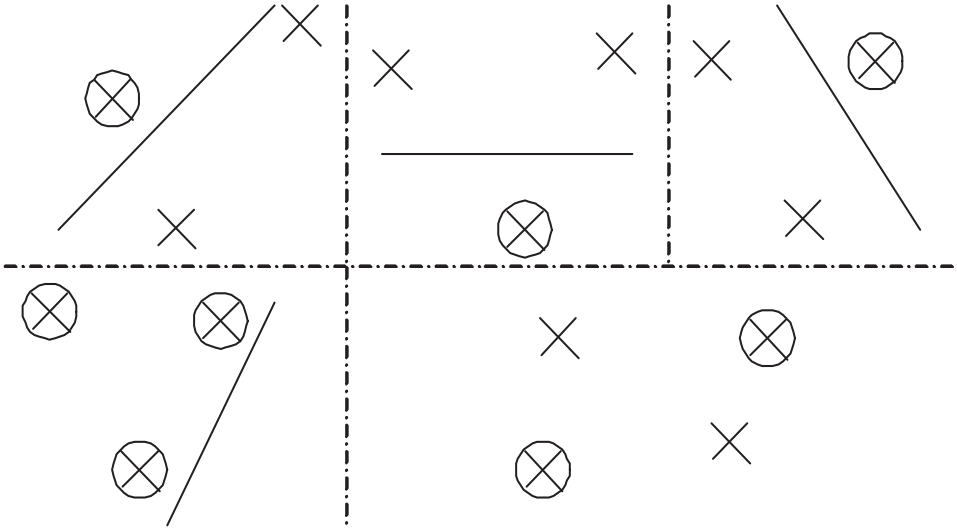


Figure 2.1: A figure taken from [3] gives a detail description of VC Dimension. In 2-dimensional space only 3 points can be shattered which implies 3 is the VC-dimension in this example.

holds for any distribution. As a matter of fact though this seem to be advantageous there is a drawback too, as the bound may be loose to some distributions.

2.1.3 Structural Risk Minimization

There are two ways in which a learning algorithm handles the problem of overfitting. First one is *Empirical Risk minimization*. The idea of this algorithm is to choose a model $\mathcal{G}$ of possible functions and then to minimize the *empirical risk* in that model. But this works well when the target function belongs to the model $\mathcal{G}$ which is a very rare case. So the model has to be enlarged as much as possible and also overfitting has to be handled.

The second method is the *Structural Risk minimization*. The idea is to choose an infinite sequence $\{\mathcal{G}_d : d = 1, 2...\}$ of models of increasing size and to minimize the *empirical risk* in each model with an added penalty for the size of the model. The penalty prefers the models with low estimation error and also measures the capacity of the model.

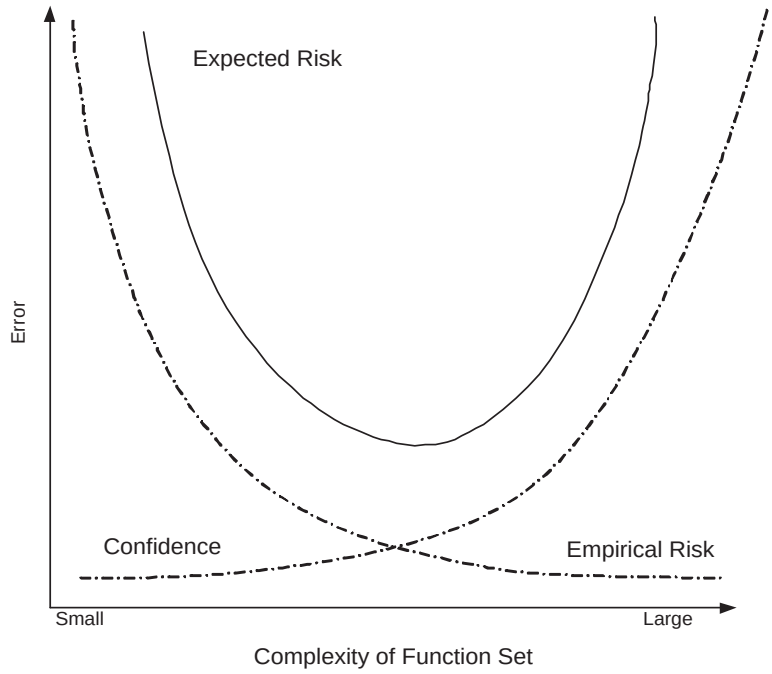


Figure 2.2: A figure taken from [4] illustrates Structural Risk Minimization.

From the figure 2.2 it can be observed that as the function complexity increases the empirical risk decreases and the VC dimension increases. For only a particular classification function the expected risk gets minimized and it guarantees a good classification if used.

2.2 Support Vector Machines

Support Vector Machine is a learning algorithm based on the principle of margin maximization. Comparative studies have found SVMs to perform as well as or better than most prevalent learning methods. There are two different types of SVMs as discussed earlier. The following sections provide brief review on Two-Class SVM and One-Class SVM.

2.3 Two-Class SVM

Two-class Support Vector Machine (TCSVM) has been widely used for remote sensing applications in recent years. As can be seen in the figure 2.3 TCSVM generates a hyper plane that maximizes the distance between two different patterns in the feature space. But TCSVM is a supervised learning algorithm i.e., it requires training samples.

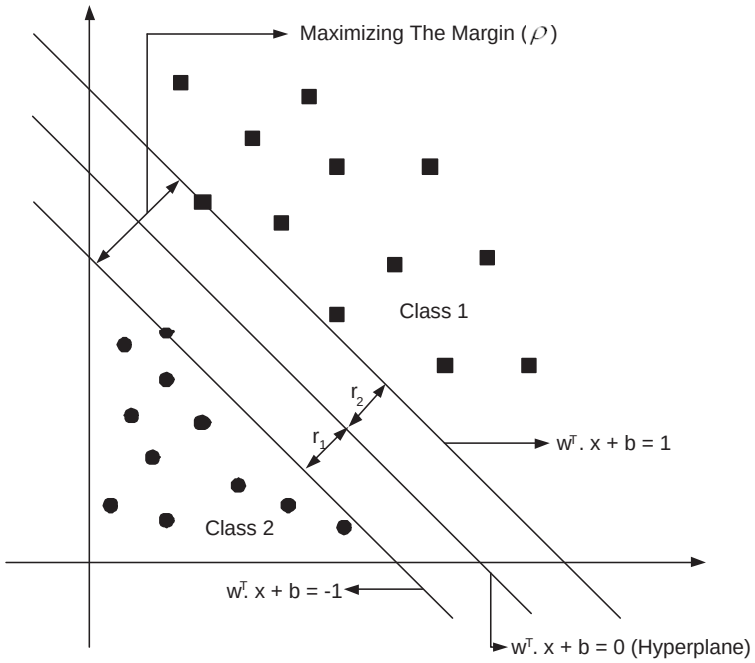


Figure 2.3: Two-Class Support Vector Machine (TCSVM).

Given the training data $((x_1,y_1), (x_m,y_m))$, x$\in$ R^n , y$\in$ (1,-1), TCSVM learning aims to find a classification hyperplane that maximally separates the two classes i.e. to maximize the margin magnitude. Consider two different patterns lying on the upper and lower margins (Support Vectors) i.e. (w.x_1) + b = -1 and (w.x_2) + b = 1. The margin is therefore the distance between these two points measured perpendicular to the hyperplane i.e $(\frac{w}{||w||}.(x_1 - x_2)) = \frac{2}{||w||}$. Thus the

best hyperplane can be found by maximizing this margin or by minimizing $\frac{2}{||w||^2}$.

$$\min_{w \in F} \frac{1}{2}||w||_2^2, \tag{2.4}$$

$$\text{subject to: } y_n((\text{w}.x_i)+\text{b}) \geq 1, \text{i}=1,...,\text{m}.$$

Where w and b are hyperplane parameters defined by (w.x) + b = 0.

Slack variables are used to convert non linear non-separable case into linear separable case. Slack variables are introduced as $\xi_i \geq 0$,i=1,...,m. Therefore the equation is transformed into

$$\min_{w \in F} C \sum_{i=1}^{N} \xi_i + \frac{1}{2}||w||^2, \tag{2.5}$$

$$\text{subject to : } y_n((\text{w}.x_i)+\text{b}) \geq 1 - \xi_i, \text{i}=1,...,\text{m}.$$

Where C is called the regularization constant which determines the penalty on the errors. The kernel methods are often used to project the original feature space into a higher dimensional feature space, and thus a linear classification in the high dimensional feature space is equivalent to a nonlinear classification in the original feature space. Radial basis function (RBF) kernel is defined as:

$$\kappa(x, x_i) = e^{-\gamma||x-x_i||^2}, \tag{2.6}$$

where γ is the kernel width.

2.3.1 Linear Separable Case

This is the simplest case of pattern recognition. Here the feature vectors are assumed to be separable and can be separated by a linear decision boundary. The decision boundary is a hyperplane as the input space can be of any dimension.

We have a set of example data samples, which are referred as feature vectors x_i and corresponding labels y_i which form an independent and identical distribution related according to an unknown probability distribution P(x,y).

Now for the data set defined, it is possible that there are set of hyperplanes called canonical hyperplanes capable of separating the two classes of data. Our objective is to select the hyperplane separating the two classes of data with the maximum margin. The hyperplanes are of the form,

$$x \in \mathbb{R}^N : (w.x) + b = 0, \tag{2.7}$$

where b is the offset from the origin and w is the normal to the hyperplane.

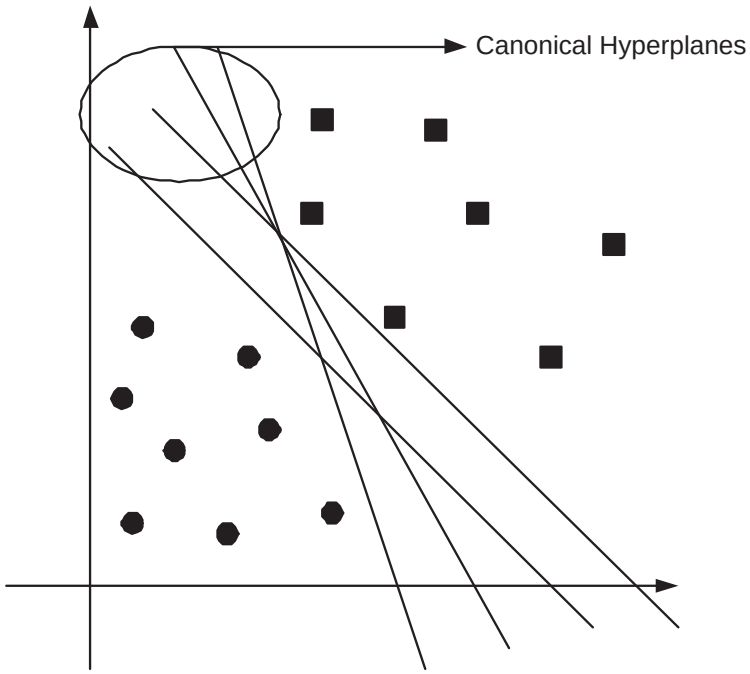


Figure 2.4: Canonical Hyperplanes.

Therefore the condition for classifying the data samples without error is

$$y_i((w.x_i) + b \geq 1, i = 1, ..., m. \tag{2.8}$$

where m is the number of data samples.

Now consider two different patterns lying on the upper and lower margins (Support Vectors) i.e., (w.x_1) + b = 1 and (w.x_2) + b = -1.

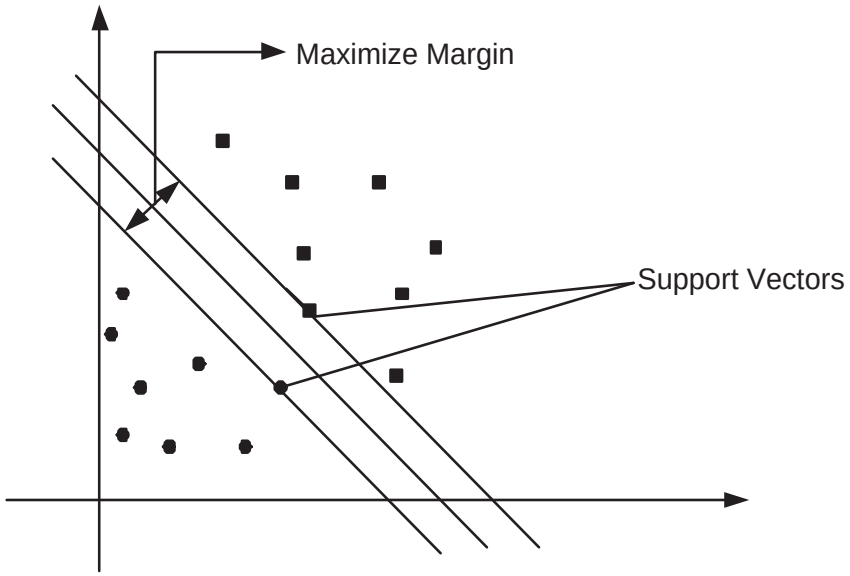


Figure 2.5: Margin Maximization.

The margin is therefore the distance between these two data points measured perpendicular to the hyperplane i.e.,

$$\frac{w}{||w||}.(x_1 - x_2) = \frac{2}{||w||}, \tag{2.9}$$

thus the best can be found by maximizing this margin or by minimizing,

$$\tau(w) = \frac{1}{2}||w||^2, \tag{2.10}$$

subject to: y_i ((w.x_i) + b) = 1, i = 1,,m.

2.3.2 Non-Linear Separable Case

Non-linear separation boundaries are needed for solving general purpose problems. SVMs tries to solve this problem by non-linearly transforming the input feature

space by a mapping function

$$\Phi : x_i \rightarrow z_i, \tag{2.11}$$

into a high dimensional feature space where a linear separation is done.

A decision function of the following form is obtained;

$$f(x) = sgn(\sum_{i=0}^{m} \alpha_i y_i (\Phi(x).\Phi(x_i)) + b), \tag{2.12}$$

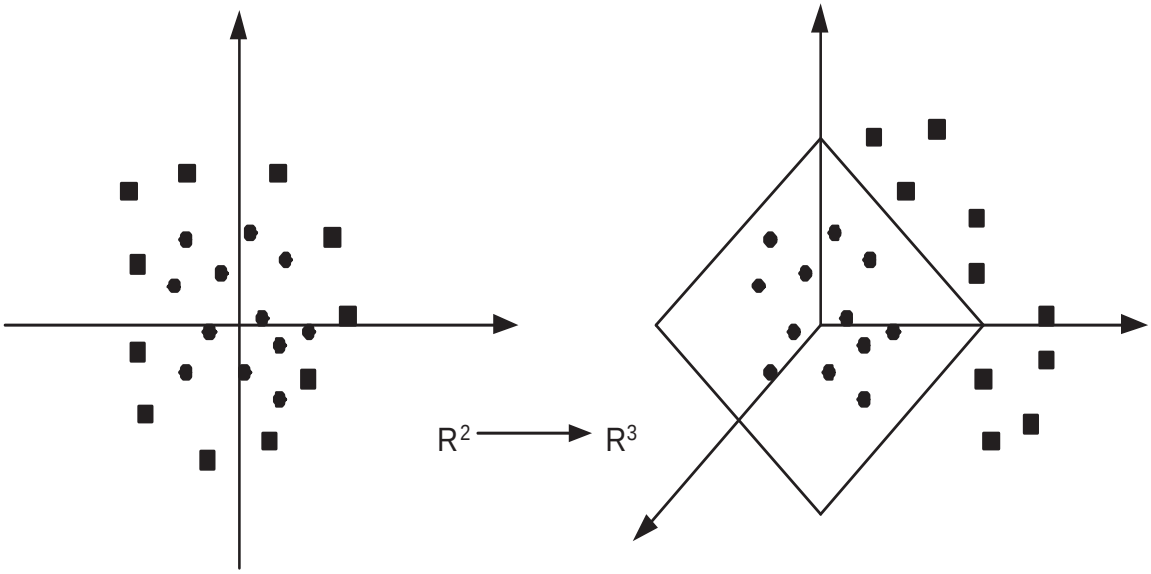


Figure 2.6: Projection of data into higher dimensional feature space.

where $(\Phi(\mathrm{x}).\Phi(x_i))$ are dot products computed in the projected space. Generally used kernels are;

- Polynomial Kernel: $\mathrm{k(x},x_i) = (\mathrm{x}.x_i)^d$,
- Guassian Kernel: $\mathrm{k(x},x_i) = \exp(-\gamma||\mathrm{x}\text{-}x_i||^2)$,
- Sigmoid Kernel: $\mathrm{k(x},x_i) = \tanh(\mathrm{m}.(\mathrm{x}.x_i) + \Theta)$.

The kind of kernels used in SVMs algorithms are Mercer Kernels. The

kernel that is normally used is the Gaussian kernel. This kernel allows tighter decision boundaries and thus provides better classification as discussed in [28]. This kernel is independent of the position of the patterns with respect to the origin, it utilizes the distances between the patterns.

2.4 One-Class SVM

The OCSVM is an unsupervised classification algorithm i.e., the information about the target class is unavailable. The boundary between the data of the target class, which is provided, and all other data, considered as outliers has to be estimated based on the data of the target class alone. This method is also referred to as novelty detection.

There are two methods for the implementation of OCSVM. Support Vector Data Description is one of the proposed methods. In this method a small hypersphere is found containing majority of the data. Vectors lying outside the sphere are classified as outliers. The second method is ν-SVM.

2.4.1 Hyperplane Based Model

The objective in this method is to separate the origin from the projected data using a hyperplane with the maximum margin ρ . ν is an important parameter that decides the percentage of outliers and its value lie between 0 and 1.

f_w(x)=(w.Φ(x)) computes the distance from the origin and decides that a pattern belongs to one class when ever f_w(x)$\geq \rho$. The hyperplane is constructed by solving the equation:

$$\min_{w \in F, \xi \in \mathbb{R}^n, \rho \in \mathbb{R}} \frac{1}{2}||w||^2 - \nu\rho + \frac{1}{n}\sum_{i=1}^{n}\xi_i, \tag{2.13}$$

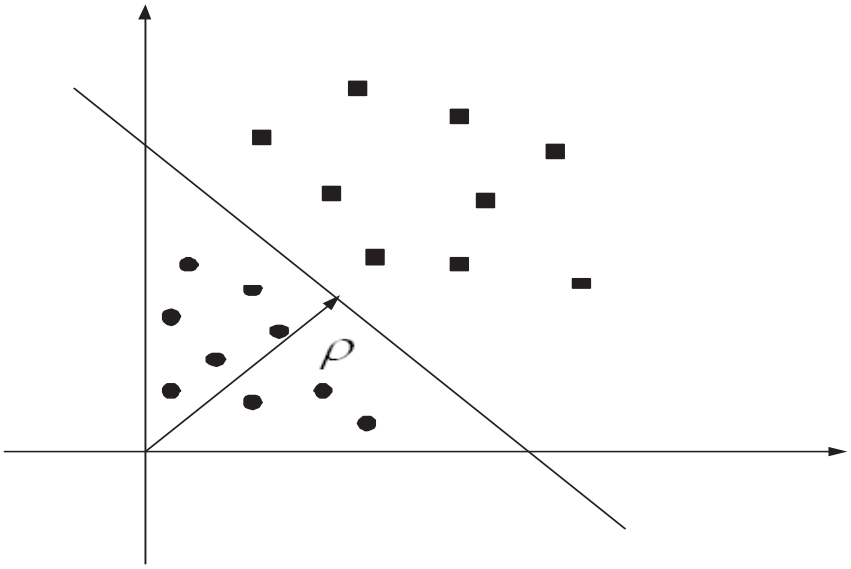


Figure 2.7: Hyperplane based OCSVM.

subject to: $y_i((\mathrm{w}.x_i)+\mathrm{b}) \geq \rho$ - ξ_i,i=1,...,n.

ν is defined as an upper bound on the fraction of outliers and a lower bound on the fraction of support vector's. For practical implementations it can be roughly taken as the percentage of outliers. The value of ν majorally effects the classification results but correct ν value can be chosen only when we know the exact percentage of outliers in the data. The proposed edited-bootstrapped algorithm overcomes this problem by assigning a small value to ν. Thus ν no more effects the classification results.

2.4.2 Hypersphere Based Model

This model is also referred to as Support Vector Data Description. Here the objective is to create a hypersphere with the minimum volume in kernel space containing all the data. The sphere is defined by a center (a) and a radius R>0. The volume is minimized by minimizing R^2 and the sphere has to contain all the training objects x_i.

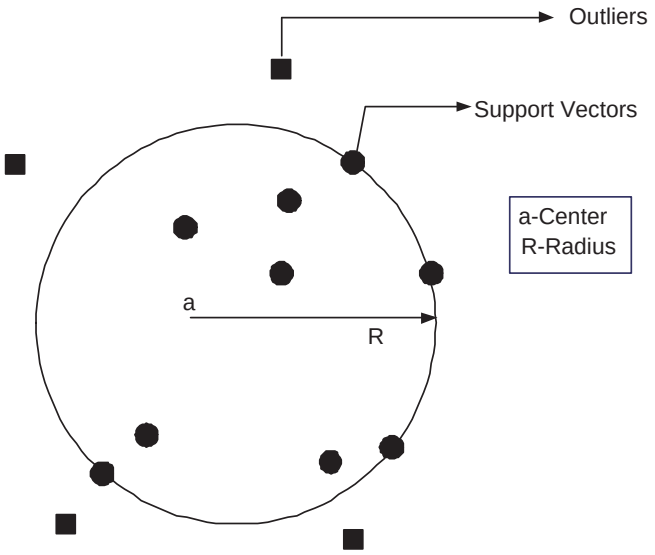


Figure 2.8: Hypersphere based OCSVM.

The error function to minimize is:

$$F(R, a) = R^2, \tag{2.14}$$

$$\text{subject to:}||x_i\text{-a}||^2 \leq R^2\text{,i=1,...,n.}$$

To allow for the possibility of outliers in the training data, slack variables $\xi_i \geq 0$ are introduced. The distance from x_i to the center a should be strictly smaller than R^2, but larger distances will be penalized. So maximization problem changes to,

$$F(R, a) = R^2 + C \sum \xi_i, \tag{2.15}$$

$$\text{subject to:}||x_i\text{-a}||^2 \leq R^2 + \xi_i\text{,}\xi_i \geq 0\text{,}\forall \text{ i.}$$

Where the parameter C controls the tradeoff between the volume and the errors. C is similar to the ν of the hyperplane model.

2.5 Applications of SVMs

Now-a-days Support Vector Machines are widely used in many classification problems like hyperspectral remote sensing images, texture classification, face detection,texture detection in digital video, Landsat TM Imagery and Image Retrieval. SVMs is used in all the above domains in [25] [29] [26] [30] and [31]. The advantages of SVMs over other conventional learning algorithms are discussed in the papers [30] [29] [9] and [31].

In [30] experimental analysis was aimed at understanding and assessing the potentialities of SVMs classifiers in hyper dimensional feature spaces. Also the performance of SVMs was compared with two other nonparametric classifiers, radial basis function neural networks and the k-nearest neighbor classifier. The analysis was done on hyper spectral remote sensing images. The paper concludes that SVMs performed well rather than the other conventional nonparametric classifiers in terms of classification accuracy, computational time and stability to parameter setting. [29] discusses the application of SVMs in text detection in digital video and it also compares SVMs with other classifiers. Explicit texture feature extraction schemes are not used in this paper. Instead they fed the gray level values of the raw pixels directly to the classifier. It concludes that SVM based text detection method was more efficient than Neural Network ones.

The performance of the SVMs can further be increased by applying the idea of bootstrapping techniques (discussed in chapter 3) and the results obtained by applying the idea of proposed edited-bootstrapped SVM will be shown in chapters 4 and 5.

Chapter 3

Bootstrapping Techniques

Bootstrapping is defined as a process to create pseudo replicate data sets by re-sampling. It is an iterative technique to eliminate the effect of outliers during the process of obtaining a classification boundary in any classification algorithm. This method not only reduces the outliers but also brings the data points closer. The idea in any classifiers is to learn from a given set of data points and to predict the labels of the data points that are to be tested. If outliers are present in a learning data set then classifiers tend to pick them as potential support vectors, which in turn degrade the classification. Thus the outliers have to be eliminated from the training data set in order to purify the training data set. This technique gives smoothing to the distribution of training samples.

In [17] different techniques to calculate the bootstrapping samples are discussed. There are four different bootstrapping techniques. Their performance was tested on three artificial data sets and one real data set. All the experiments were performed on Nearest Neighbor (NN) classifier and were compared with that of conventional k-NN classifiers. The paper concludes that the NN classifier designed on the bootstrap samples performs better when compared with that of the conventional NN classifiers as mentioned above.

3.1 Bootstrapping Techniques

In general there are many ways to generate the bootstrap samples out of which four are discussed in [17]. There are similar resampling techniques like cross-validation and jackknifing [32].

Let $X_N^i = x_1^i,x_2^i,,x_N i^i$ be an original training data set. Now four different bootstrap techniques are discussed to generate bootstrap set $X_N i^B = x_i 1^b,x_i 2^b,,x_N i^b$ of size N_i.

3.1.1 Bootstrapping I

1. Select a sample x_i randomly from the original training set.
2. Choose r nearest samples of x_i.
3. Compute the bootstrap sample $x_i^b = \sum_{j=0}^{r} w_j x_j$ where w_j is a weight. The weight w_j is given by

   $$w_j = \frac{\Delta_j}{\sum_{c=0}^{r} \Delta_c},$$

   where Δ_j is chosen from a uniform distribution [0,1].
   Note that $\sum_{j=0}^{r} w_j = 1$.
4. Repeat steps 1,2 and 3 until the size of the bootstrap training data samples is same as the original training set.

In this method the samples are randomly chosen. The set generated finally by choosing the samples randomly from the original set is known as Efron's bootstrap set. If an original sample is surrounded by many outliers then the new bootstrap sample obtained by r nearest neighbors results in wrong classification. In which case, the classification accuracy decreases. Thus Efron's set avoids such situations as there is every chance for the re-sampled samples is chosen again.

3.1.2 Bootstrapping II

1. Select a sample x_i from the original training set.

2. Choose r nearest samples of x_i.

3. Compute the bootstrap sample $x_i^b = \sum_{j=0}^{r} w_j x_j$ where w_j is a weight. The weight w_j is given by

$$w_j = \frac{\Delta_j}{\sum_{c=0}^{r} \Delta_c},$$

   where Δ_j is chosen from a uniform distribution [0,1].
   Note that $\sum_{j=0}^{r} w_j = 1$.

4. Repeat steps 1,2 and 3 for all the training data samples in the original training set.

In this technique samples are chosen so that no sample is selected more than once. In both bootstrapping techniques I and II, the original training samples are linearly combined by using random weights.

3.1.3 Bootstrapping III

1. Select a sample x_i randomly from the original training set.

2. Choose r nearest samples of x_i.

3. Compute the bootstrap sample $x_j^b = \frac{1}{r+1} \sum_{j=0}^{r} x_j$.

4. Repeat steps 1,2 and 3 until the size of the bootstrap training data samples is same as the original training set.

In this technique, either Efron's set or random weights were considered. As can be seen in step 3 the new bootstrap sample is the local sample mean.

3.1.4 Bootstrapping IV

1. Select a sample x_i from the original training set.
2. Choose r nearest samples of x_i.
3. Compute the bootstrap sample $x_j^b = \frac{1}{r+1} \sum_{j=0}^{r} x_j$.
4. Repeat steps 1,2 and 3 for all the training data samples in the original training set.

In step1 of bootstrapping IV, the samples are chosen so that no sample is selected more than once.

The only parameter to be taken care in all the four bootstrapping techniques is r, which decides the number of nearest neighbors. [17] concludes that the bootstrap technique for the Nearest Neighbor classifier outperforms the conventional k-NN classifiers and also the Edited Nearest classifiers (discussed in next section) [19] [18] [33]. Particularly in high dimensional spaces bootstrap techniques has to be used to design 1-NN classifier.

3.2 Condensed Nearest Neighbor Rule (CNN)

According to the NN rule an unclassified sample is assigned to the same class as its nearest neighbor. The n reference points, classified by an external source, are used to classify the unclassified samples. As it is required to store the reference points, NN rule is not used in many practical applications. NN rule is expensive because it has to compute the distance to all the training samples. As the dimensionality increases the amount of training data required increases exponentially and computational cost increases.

CNN rule was proposed in order to preserve the idea of NN rule without imposing such inflexible idea of storing the reference points. In this rule much

of the redundant data will be removed from the sample set. This reduces the number of training data points thereby eliminating the problem of large storage. Computational cost also reduces as the sample set has been condensed.

The main idea of the CNN is centered in finding a minimal consistent subset of the reference set used in conventional NN rule. A minimum number of reference points which classifies the remaining samples correctly form the minimal consistent subset of a sample set. The assumption made in CNN rule is that the original sample set is arranged in some order and to bins named as *STORE* and *GARBAGE*. The algorithm of CNN in [18] is as follows:

1. First sample is placed in *STORE*.
2. The second sample is classified by the NN rule, using as a reference set the current contents of *STORE*. (Since *STORE* has only one point, the classification is trivial at this stage). If the second sample is classified correctly it is placed in *GARBAGE*; otherwise it is placed in *STORE*.
3. Proceeding inductively, the i^th sample is classified by the current contents of the *STORE*. If classified correctly it is placed in *GARBAGE*; otherwise it is placed in *STORE*.
4. After one pass through the original sample set, the procedure continues to loop through *GARBAGE* until termination, which can occur in one of the two ways:
   (a) The *GARBAGE* is exhausted, with all its members now transferred to *STORE* (in which case, the consistent subset found is the entire original set), or
   (b) One complete pass is made through *GARBAGE* with no transfers to *STORE*. (If this happens, all subsequent passes through *GARBAGE* will result in no transfers, since the underlying decision surface has not been changed.)

5. The final contents of *STORE* are used as reference points for the NN rule; the contents of *GARBAGE* are discarded.

If the feature vectors in the classes have small overlap then the algorithm removes the data points that are close to the boundary between the classes. The data points have small overlap implies that Bayes risk is small. In the second case, if the data points of the two classes overlap to a greater extent i.e., the Bayes risk is high then *STORE* contains almost all the data points in the original sample set. In such a case there will be no reduction in the sample set.

Figure 3.1 demonstrates the idea of Condensed Nearest Neighbor algorithm. As discussed earlier it is an iterative process which ends after the correct classification of all the data points in the sample set. Figure 3.1(a) shows the original training data set. From figure 3.1(b) to (g) data points are picked in random and NN rule is applied. If the selected data points generates a boundary that classifies all other remaining data points correctly then the iteration stops. Then those data points form the condensed consistent sample set.

If the training data contains noise and overlapping classes i.e., Bayes risk is high then there will be no reduction in the sample set.

3.3 Edited Nearest Neighbor Rule (ENN)

Presence of outliers or strong overlapping among the classes produces training data sets that contain arbitrarily large number of bad samples if CNN algorithm is used for classification. In simple words editing is a process which removes the noisy data points and the overlapped points to smoothen the decision boundary. The editing algorithm was first introduced by Wilson [34]. The purpose of this algorithm was to remove the possible outliers that degrade the performance of the NN rules. Secondly these editing algorithms when combined with condensing algorithm results in a good classification accuracy. Thus editing before applying

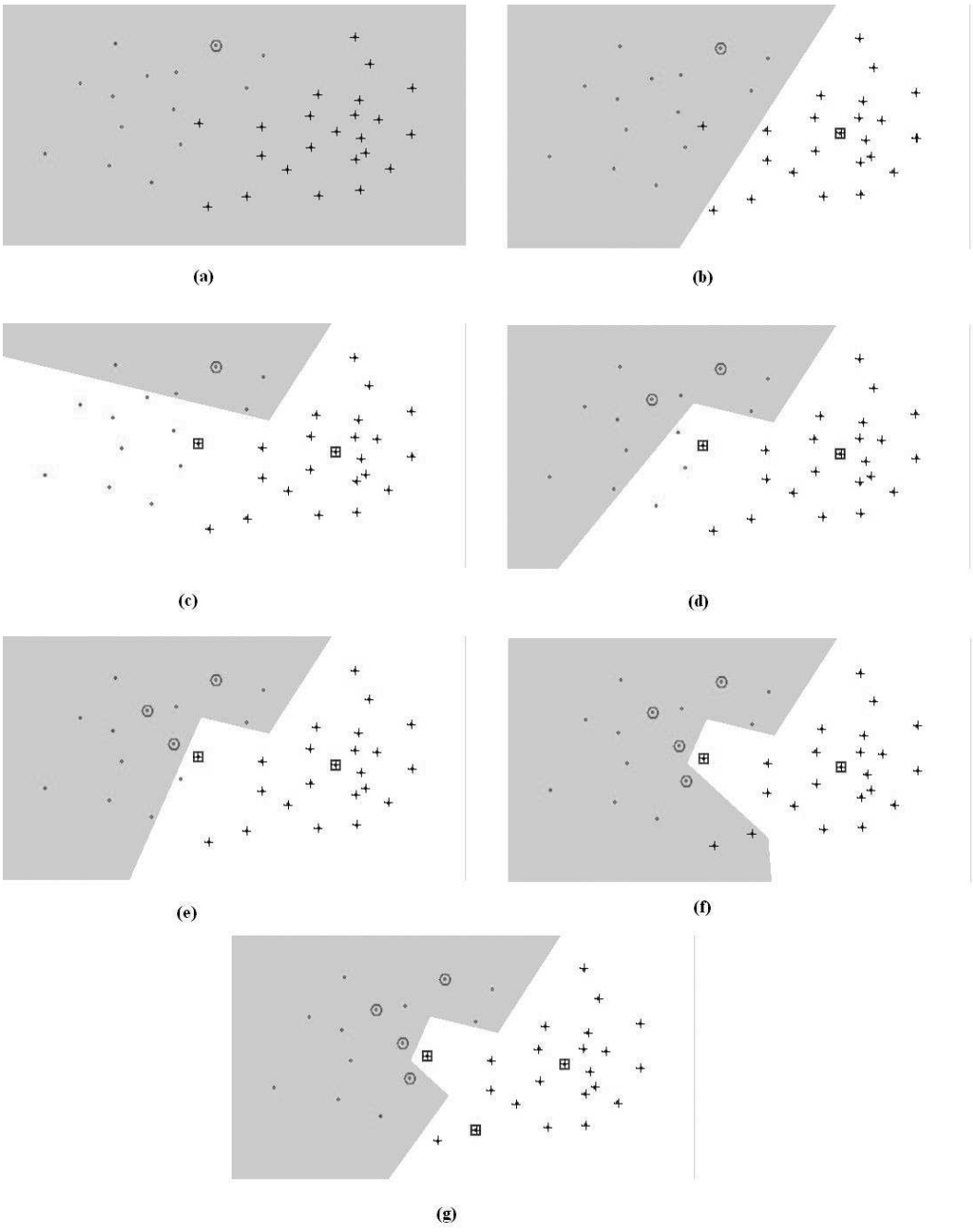


Figure 3.1: Figure is taken from [5].(a)-(f)Represents the iterative CNN algorithm. (g)Represents the result after the final iteration, where the marked samples represent the training data set.

the idea of condensing became compulsory.

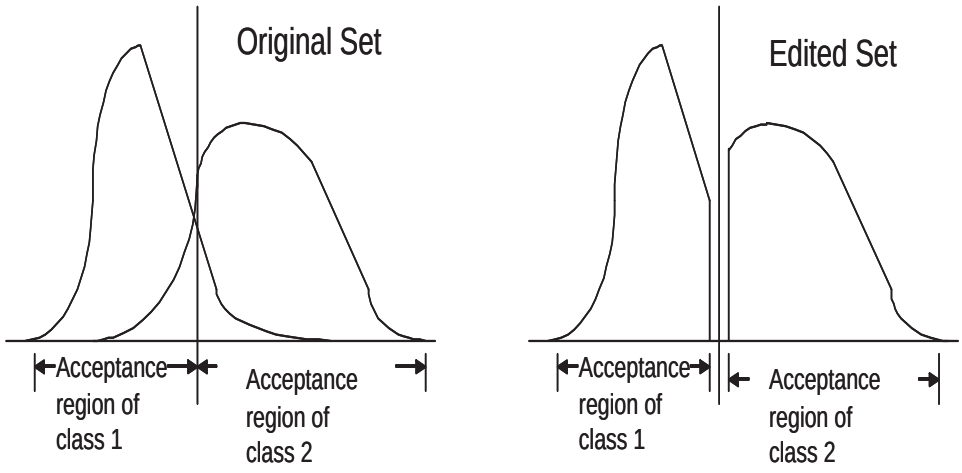


Figure 3.2: Figure taken from [6] illustrates the editing procedure using the probability density functions of the two different classes.

Figure 3.2 illustrates the editing procedure. As can be seen before editing the probability density functions of two classes overlap and the data samples falling in the overlapping region has to be removed in order to obtain an optimal decision boundary. From practical point of view it is not possible to remove only outliers from an overlapping region without removing the correct samples. Hence these lead to suboptimal results and most of the editing algorithms have different tradeoffs between removal of too many samples and leaving a small overlapping data points.

The Wilson's editing algorithm as in [6] is as follows:

Let $\mathcal{R}$ be the initial set of prototypes

1. Let $\mathcal{S}$ be the subset of data points, d, that are misclassified using the k-NN rule with $\mathcal{R}$ - $\{d\}$.
2. Return $\mathcal{R}$ - $\mathcal{S}$

Based on Wilson's editing algorithm many other algorithms were proposed to eliminate the drawbacks in the Wilson's editing technique. Wilson's editing

technique sometimes may lead to overestimation of the classification accuracy on the training data points. Secondly it can also be seen from the Wilson's editing algorithm that there is an statistical dependence between the training and the testing data points. Multiedit algorithm was proposed by Devijver and Kittler in 1982 to overcome the drawbacks in the Wilson's editing technique. In this algorithm independent reference set is used to classify the data samples. The algorithm given in [6] and [35] is as follows:

1. $Diffusion$: Let $\mathcal{R}$ be the initial set of data samples with known class labels. A random partition of $\mathcal{R}$ is made into B subsets, $\mathcal{R}^1, ..., \mathcal{R}^B$ $(B \geq 3)$.

2. $Classification$: Classify the samples in $\mathcal{R}^i$ using the 1-NN rule with $\mathcal{R}^{(i+1)modB}$ as the training data set.

3. $Editing$: Discard all the samples that were misclassified at step 2.

4. $Confusion$: Pool all the remaining data to constitute a new set $\mathcal{R}$.

5. $Termination$: If the last I iterations produced no editing, exit with the final set $\mathcal{R}$, else go to Step 1.

The original data sets in the figure 3.3(a),(c),(e) are partitioned into regions within which all the points are closer to some particular point than to any other point. Such a diagram is known as $Voronoi\ diagram$. When classifying a new data point if it falls into a particular Voronoi region then it is classified to the class in that region. It can be seen from figure 3.3(f) that Multiedit algorithm produces a good training data set even though the original data set has overlapped data points.

The second main purpose of editing technique is to combine it with the condensing algorithm to increase the performance of the NN rule. The figure 3.4(a) shows the original overlapped data set. First edit the data to remove the noise and smooth boundary as shown in figure 3.4 (b). Then condense to obtain a smaller training data set as in figure 3.4 (c)-(d).

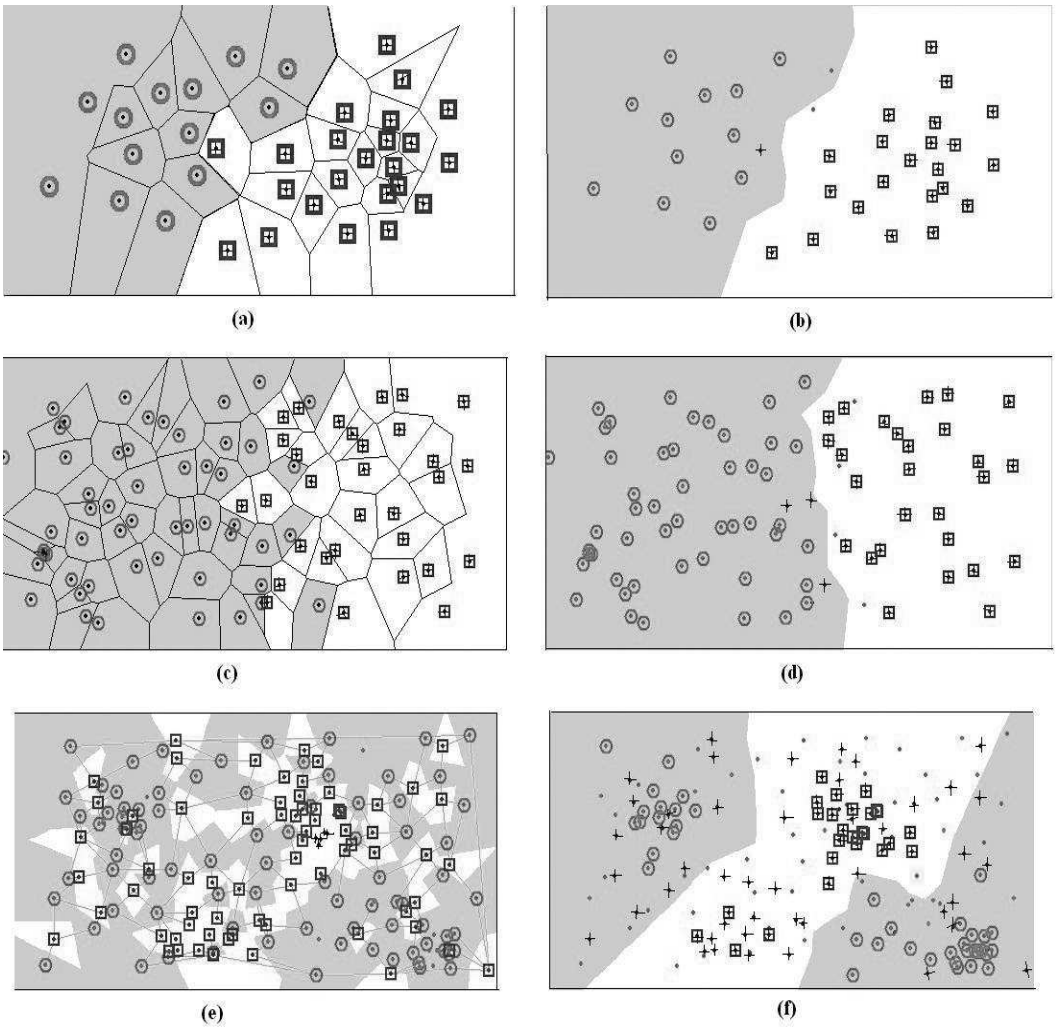


Figure 3.3: Figure is taken from [5]. (a) Original data set. (b) The result of Wilson's editing technique implemented on the data in (a). (c) Overlapped original data set. (d) The result of applying Wilson's editing algorithm on the data in (c). (e) Overlapped original data set. (f) The result of applying Multiedit algorithm on the data in (e).

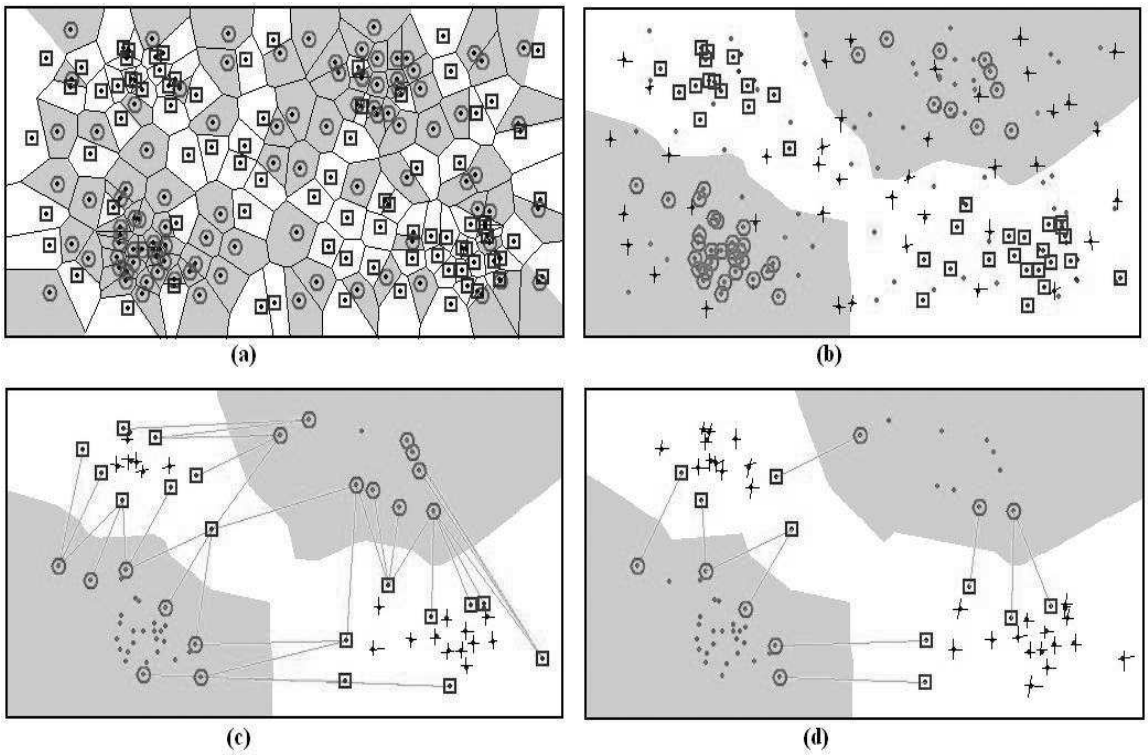


Figure 3.4: Figure is taken from [5]. (a) Original data set. (b) The result of applying Multiedit algorithm on the data in (a). (c) - (d) The result of condensing the data set.

3.4 Conclusions

The idea of condensing though it ended up with a negative experimental conclusion it has given rise to editing technique. If the training data set consists of outliers and the data samples are overlapped then the condensing algorithm does not produce better results. But if the training data set is edited before condensing then the performance of the classifier increases and it also speeds up the classification process. The performance of the bootstrapping techniques discussed in section 3.1 is compared with that of the editing techniques in the paper [17]. The performance of the NN-classifier based on bootstrap samples was demonstrated on several data sets. The results proved that the bootstrapping techniques outperform the editing technique.

Table 3.1: Comparison of Bootstrapping techniques with Editing technique.

	Bootstrapping Techniques	Combined Editing and Condensing Techniques
Methodology	Locally Combining the original training samples	Removal of misleading data samples from the training data set
Distribution of data samples	Smoother	overlap between the data samples is eliminated
Performance in higher-dimensional spaces	Comparatively higher than ENN and CNN algorithms	Comparatively less than bootstrapping techniques

In [36], a sequential bootstrapped SVM was proposed which aims in decreasing training time. The idea of the algorithm is similar to multiedit algorithm. The training data set is divided into two groups of different sizes. The large set is chosen to be the testing data set and the small set is considered to be the training data set. Now after classification the farther data points that are misclassified are transferred to the training data set. This iterates until there are no misclassifications. This algorithm reduces the size of the training data set, thereby

speeding up the training process of SVM. The bootstrapping technique II in [17] was implemented along with SVM in [37] for efficient handwritten recognition. These editing and bootstrapping algorithms result in higher classification results and also reduce the computational time by reducing the size of the training data set.

Chapter 4

Proposed Edited-Bootstrapped SVM

The proposed edited-bootstrapped SVM stand out at finding solutions for problems related to one-class remote sensing analysis such as having no one (no reference data that can verify the classification accuracy) and wide range of parameters. For example, consider the area that contains forested wetland and dryer forested land. In this case, during different times of the year there is every possibility for the wetland to become dryer forest land. If both the classes are of interest then it could be complicated to classify the pixels in the area. In such cases a slight shift in an image location places a given area in within different pixels and results in a slight change within a single pixel. Strong overlapping between the classes is also another problem in remote sensing. Overlapping implies that pixels in some areas the image could belong to different classes simultaneously. The classification algorithm results in different final solutions for the changes in the definition of classes.

The proposed edited-bootstrapped SVM algorithm and the result of applying the proposed algorithm on texture images and on the random data are discussed in this chapter.

4.1 Proposed Algorithm

Edited-bootstrapped Support Vector Machine is proposed based on the bootstrapping techniques reviewed in chapter 3. The algorithm is derived based on the principle that, classification algorithms produce better results if the training data is pure. The two main steps in any classification algorithm are training and testing. Out of which training plays a major role. The algorithm learns during the training period and then it classifies the testing data depending on what it learnt. Therefore if the training data is pure then the performance of the classification algorithms improves.

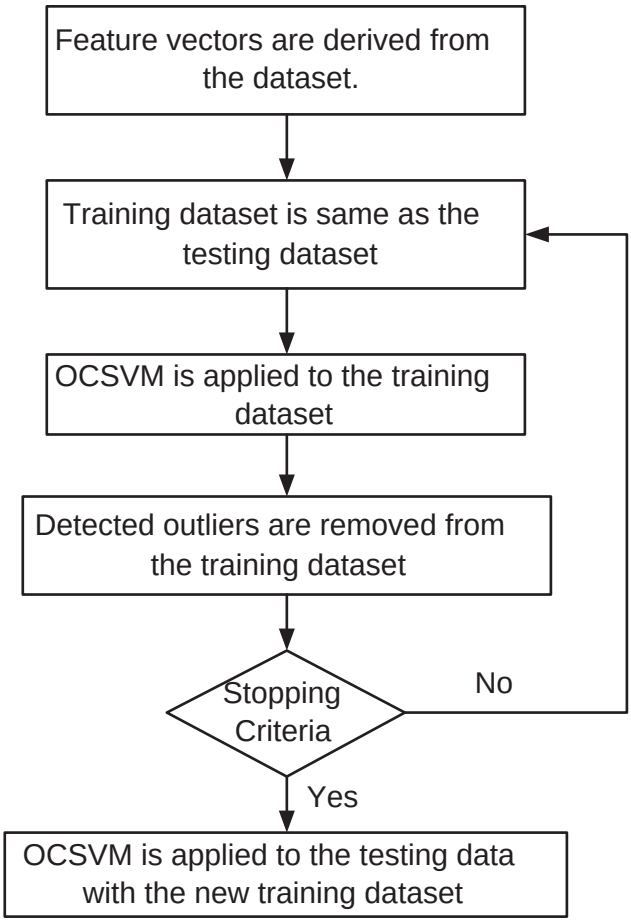


Figure 4.1: Flow chart representing the proposed edited-bootstrapped SVM.

The proposed algorithm can be visualized as shown in fig.4.2. Initial train-

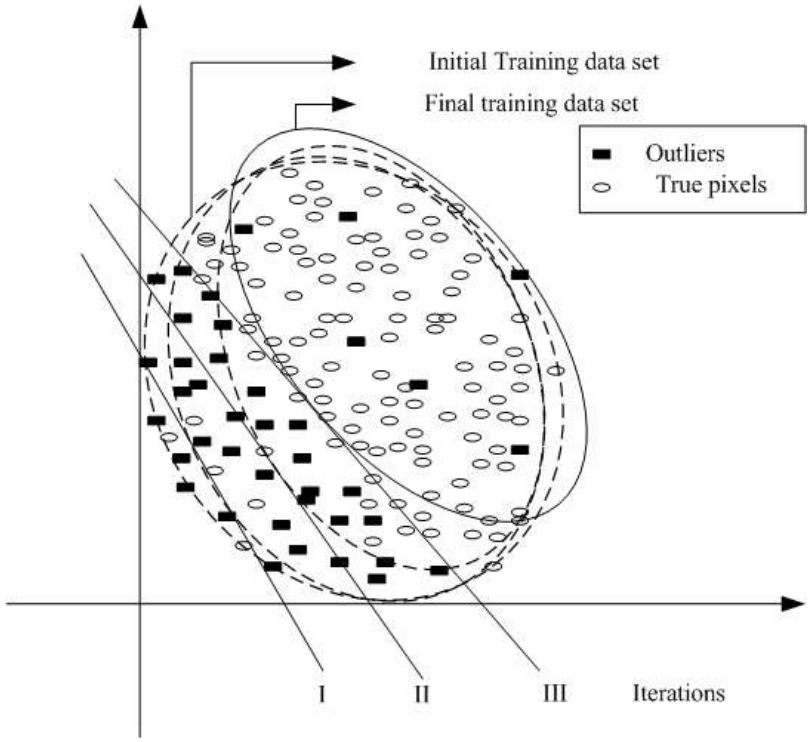


Figure 4.2: Training data set in each iteration.

ing data set consists of outliers and it can be seen that at the end of final iteration outliers are eliminated from the set. A few true pixels are also eliminated as they are highly overlapped with the outliers. The major advantages of this proposed algorithm can be listed as follows:

- The parameter ν in OCSVM is no longer an unknown parameter, as it is initialized to a small known value irrespective of total number of outliers present in the data set.
- It improves the performance of the classifier by removing the outliers and the overlapped data points from the training data set.
- This algorithm produces better results than that of the existing bootstrapping techniques in the presence of high percentage of outliers in the data set.
- This algorithm eliminates the problems faced in one-class remote sensing

analysis such as having no perfect reference data and high overlapping between the classes.

The proposed algorithm performs well in the case of one-class remote sensing analysis problems discussed above. The training data is purified by iteratively removing the detected outliers. This algorithm not only eliminates the outliers but also removes the true data points that are highly overlapped with the outliers. OCSVM is used in this algorithm. The parameter ν in OCSVM is the upper bound of the number of outliers and lower bound on the number of support vectors. Thus the parameter ν plays a major role in classification. The proposed algorithm is independent of this parameter as the outliers are removed in a small quantity in each iteration. The size of the training data reduces in each iteration as the outliers are eliminated $R_0 > R_1 > R_2... > R_i$. The algorithm is as follows:

Let $\mathcal{R}$ be the initial set of training data and the parameter ν is assigned to a lower value. Let γ be the stopping criteria.

1. $\mathcal{S} = \mathcal{R}$ is the training data set.
2. $\mathcal{R}$ is classified by One-Class SVM algorithm, considering $\mathcal{S}$ to be the training data set. The misclassified data points ($\mathcal{U}$) are considered to be the outliers and are removed from the original data set $\mathcal{R}$.
   $\mathcal{R} = \mathcal{R}$ - $\mathcal{U}$.
3. If γ then stop else go to 1.
4. Finally the testing data is classified by considering the modified R as the training data set.

The stopping criteria can be changed according to the information available. If the reference data provided is perfect then the stopping criteria can be based on the classification accuracy. The iterations stop once the classification accuracy reaches a desired value. But if the reference data is not perfect, discussed

as one of the problems in one-class remote sensing problems, then the iterations stop when there are no more outliers to detect.

4.2 Experimental Setup

The algorithm is tested on two different texture images one with 25% outliers and the other with only 4% outliers. The features are extracted from texture pixels within a 7X7 window, resulting in a 25 dimensional feature space as proposed in [38]. OCSVM is implemented on the texture data using the software LIBSVM [39]. Gaussian kernel γ is taken to be 0.000001. The ν is assigned to a small value (0.05).

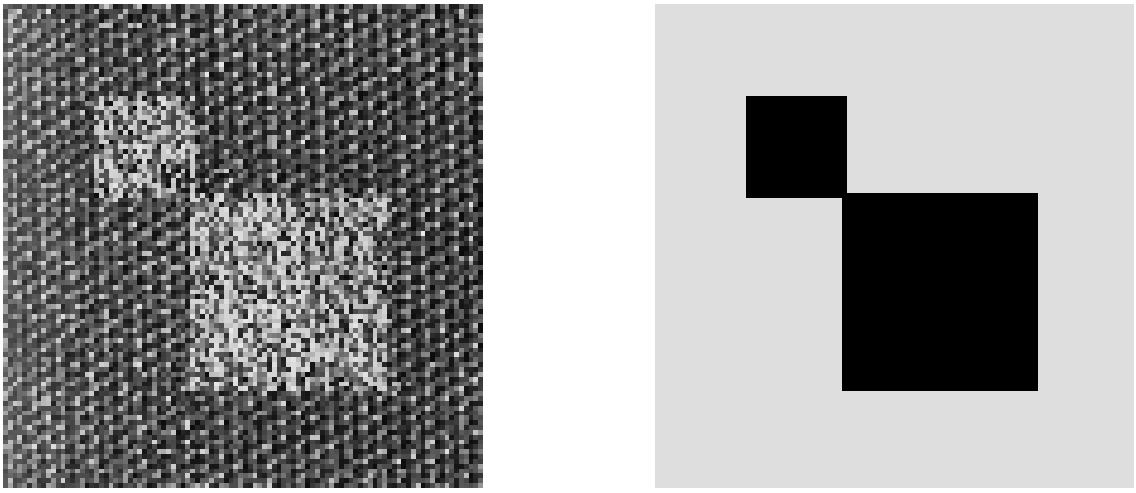


Figure 4.3: Experimental data to test the proposed algorithm. (a)Texture image. (b)Ground truth containing 25% outliers.

Figure 4.3 represents the texture image with 25% outliers and the ground truth data. The ground truth data can be used to calculate the classification accuracy as it is not subjected to change. Thus in the cases of texture images the stopping criteria can be the classification accuracy or number of outliers removed in each iteration. Similarly figure 4.4 represents the texture image containing

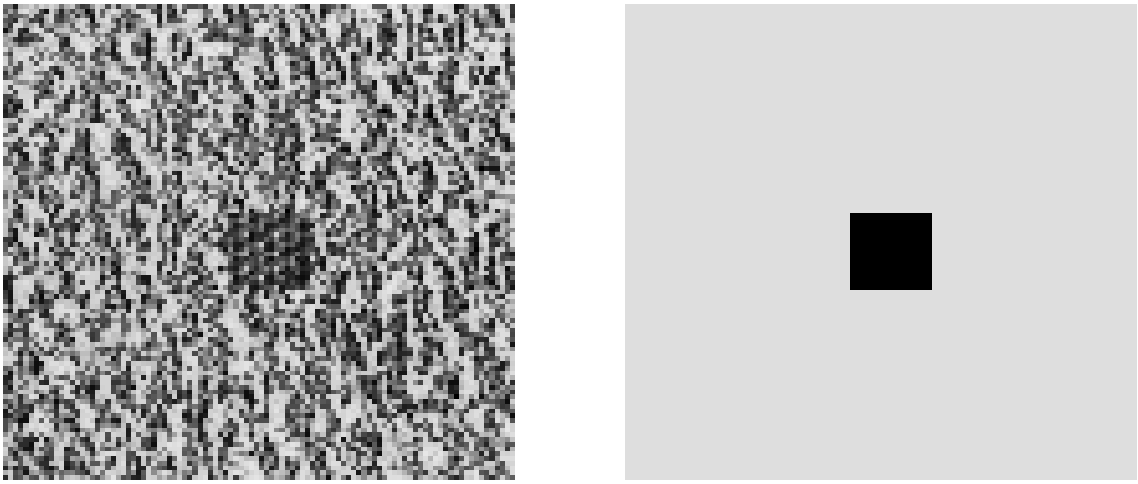


Figure 4.4: Experimental data to test the proposed algorithm. (a)Texture image. (b)Ground truth containing 4% outliers.

4% outliers and its ground truth. To test the algorithm in many conditions two texture images with different percentage of outliers are chosen.

4.2.1 Simulation Results and Discussions

Simulations are performed on the texture images mentioned in section 4.2. Deliberately those two images are chosen in order to test the performance of the proposed algorithm. The bootstrapping techniques discussed in chapter 3 are also applied to the texture images to compare the performance of the proposed algorithm. The parameter r, which decides the number of nearest neighbors, in the bootstrapping techniques is taken to be 5.

First the result of applying the edited-bootstrapped SVM algorithm is discussed and later the results are compared with that of the bootstrapping techniques. In order to compare the performances three measurements are used: *Classification accuracy* (P_a) is defined as the percentage of pixels that are correctly classified in terms of correct data points and outliers. *Precision* (P_b) in-

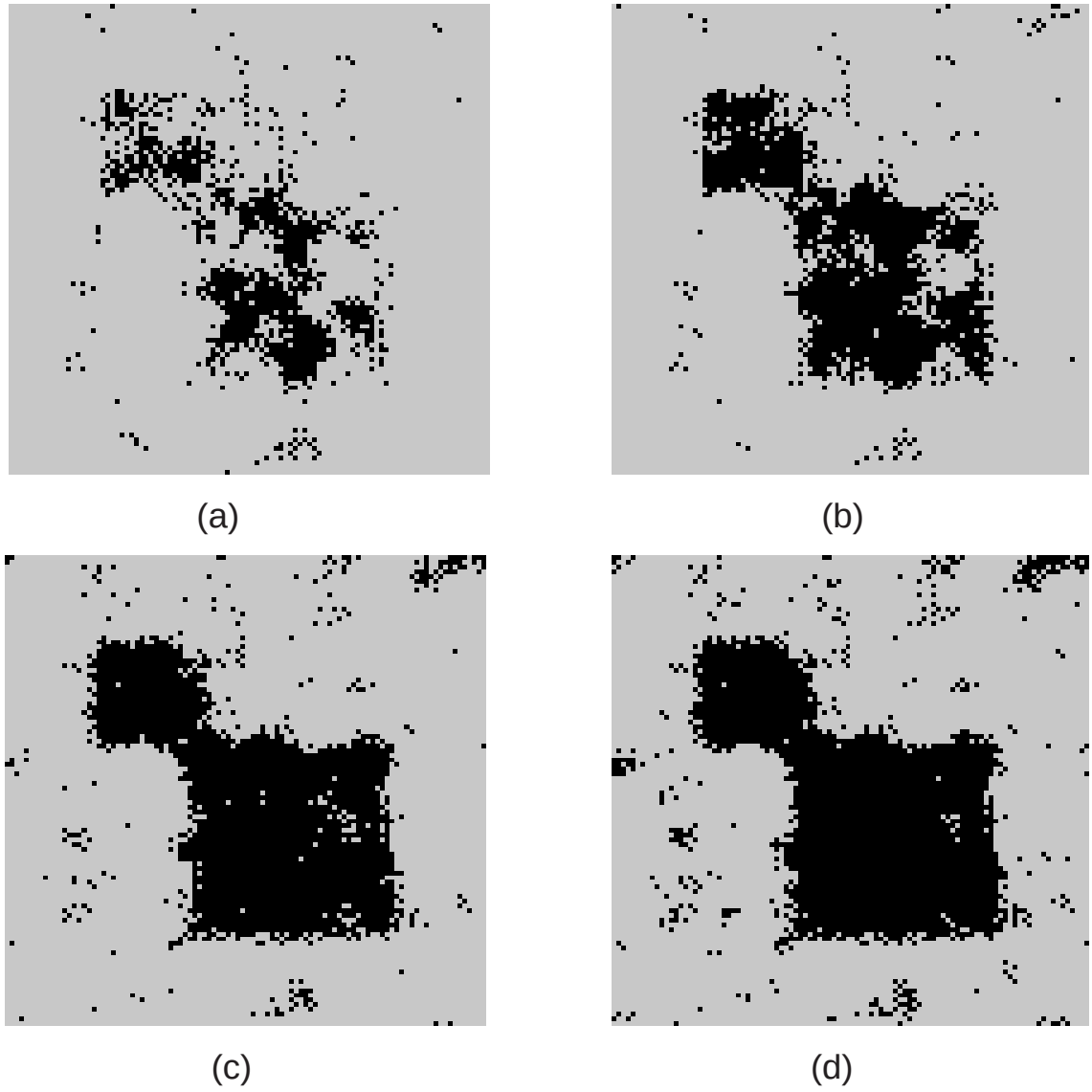


Figure 4.5: Result of applying edited-bootstrapped SVM to the texture image containing 25% outliers.(a)-(d)The results after each iteration.

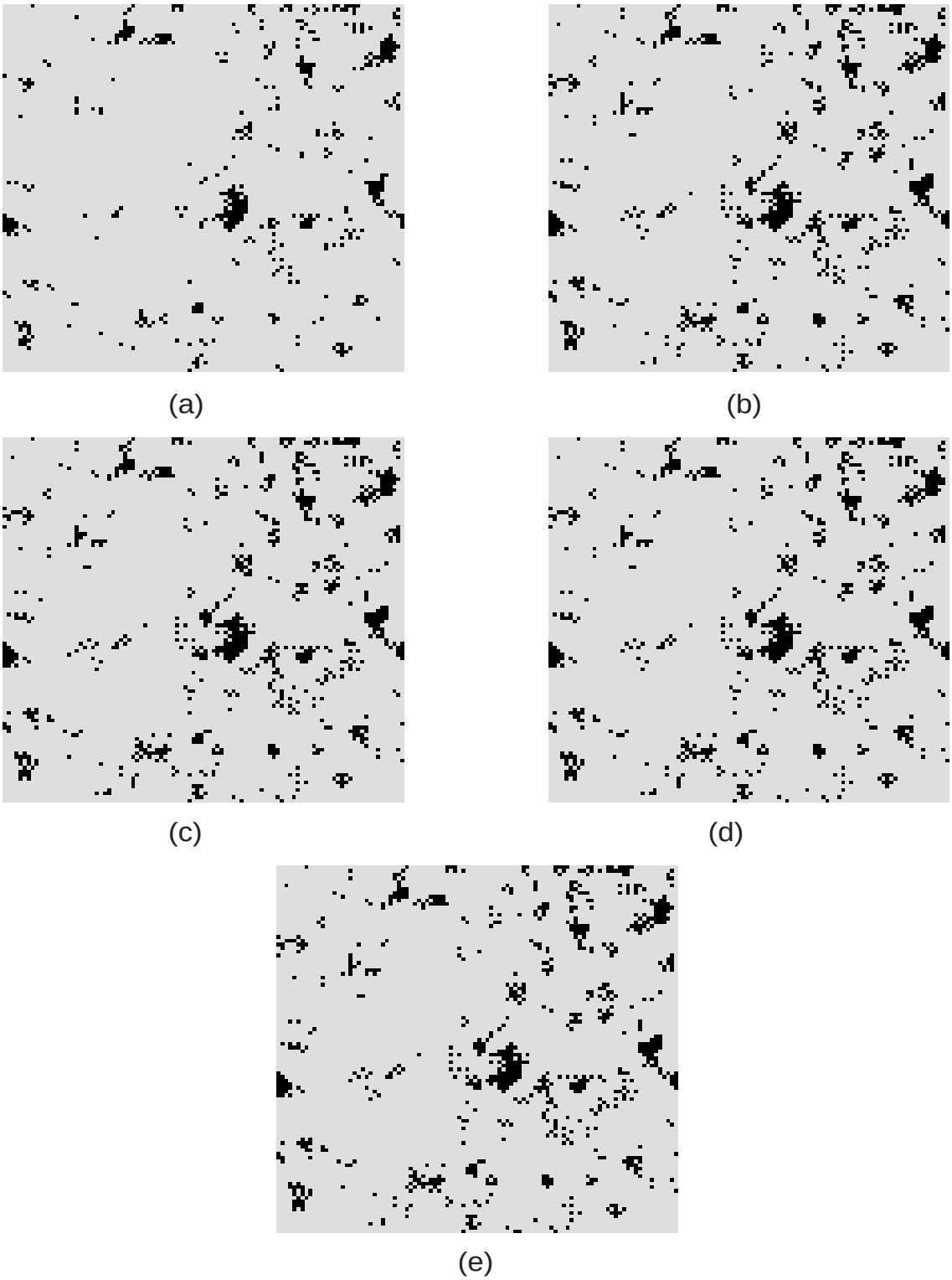
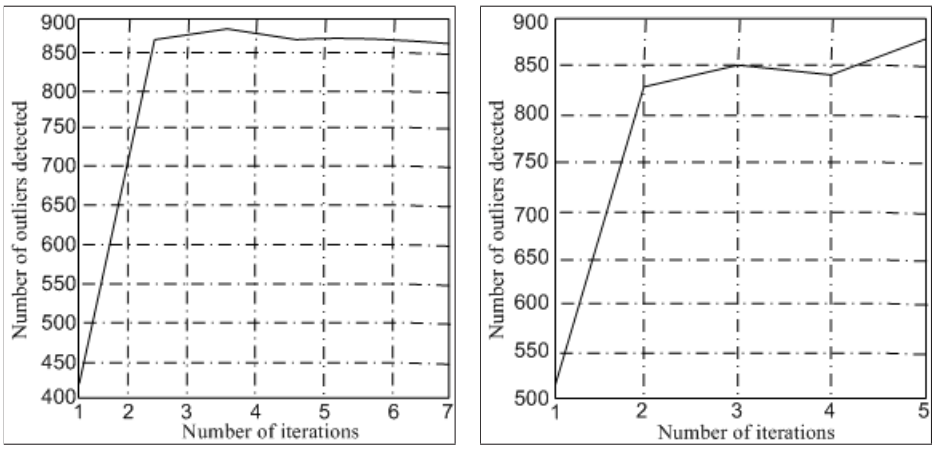


Figure 4.6: Result of applying edited-bootstrapped SVM to the texture image containing 4% outliers.(a)-(d)The results after each iteration.

dicates the percentage of detected correct pixels that are true ones. *Recall* (P_c) is the percentage of true correct pixels that can be detected. In any classification system if the classification accuracy is considered to be a constant then there will be tradeoff between *Precision* and *Recall*. Figure 4.5 represents the result of applying the proposed algorithm to the texture data with 25% outliers. Initially as the training data contains some outliers the classification accuracy is low as can be seen in figure 4.5(a). The final result is obtained after the fourth iteration. The algorithm is stopped once the number of outliers detected got reduced as can be seen in the figure 4.7.

Similarly figure 4.6 represents the result of applying proposed edited-bootstrapped SVM to the texture image containing only 4% of outliers. Fig.4.7(a)-(b) repre-



(a) (b)

Figure 4.7: Graphs representing the stopping criteria. (a) Mosiac with 25% outliers. (b) Mosiac with 4% outliers.

sents the number of outliers removed in each iteration when OCSVM is applied to the texture image with 25% and 4% outliers respectively. It can be inferred that number of outliers reduce in the case of 25% outliers whereas if the image has only 4% outliers then true pixels are also detected as outliers due to overestimation.

Figure 4.8(a)-(b) is the result of applying the idea of bootstrapping II and bootstrapping IV to the texture image containing 25% outliers and (c)-(d) is the result of applying the same techniques to the texture image with 4% outliers.

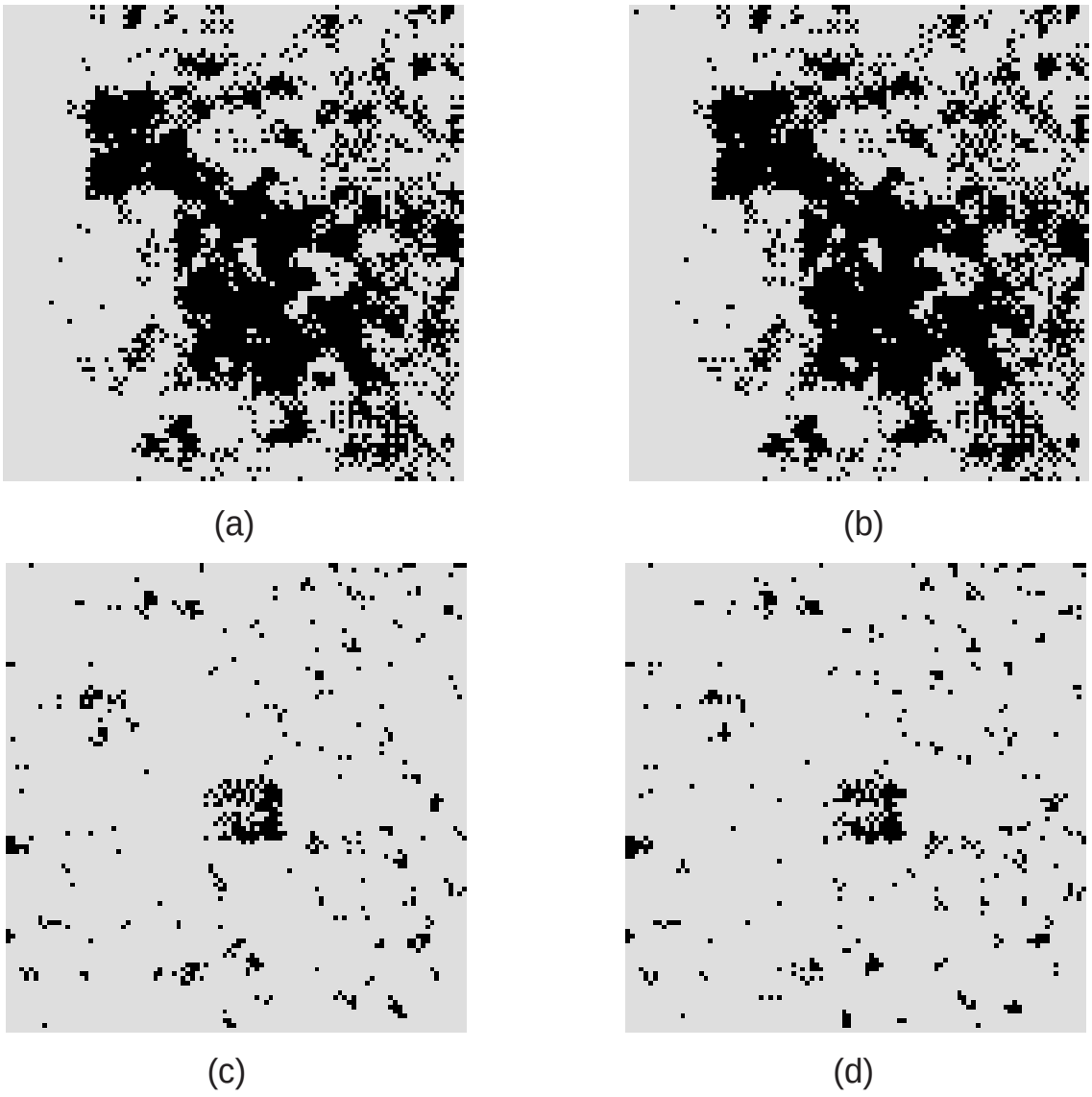


Figure 4.8: (a)-(b)Result of applying the idea of bootstrapping techniques II and IV respectively to the texture image containing 25% of outliers. (c)-(d)Result of applying the idea of bootstrapping techniques II and IV respectively to the texture image containing 4% of outliers.

By comparing the final results in figures 4.5 4.6 and 4.8, obtained after applying the proposed edited-bootstrapped SVM and the bootstrapping techniques II and IV respectively, it can be concluded that the proposed edited-bootstrapped SVM algorithm achieves higher accuracy than the existing bootstrapping techniques if the percentage of outliers are more in the original data set. The result is more obvious from the tables 4.1 and 4.2.

Table 4.1: Comparison of proposed edited-bootstrapped SVM and bootstrapping Techniques II and IV through accuracies obtained by applying the techniques to the texture image containing 25% outliers.

$\nu = 0.2$	Classification Accuracy (P_a)	Precision Accuracy (P_b)	Recall Accuracy (P_c)
Bootstrapping II	77.6	85.2	86.2
Bootstrapping IV	76.9	85.5	85.1
Edited-bootstrapped SVM ($\nu = 0.05$)	82.4	98.2	79.0

In the table 4.1, which lists the accuracies of applying edited-bootstrapped SVM and the two bootstrapping techniques to the texture image containing 25% outliers, the precision accuracy of edited-bootstrapped SVM is relatively higher than the bootstrapping techniques II and IV. Similarly from the table 4.2, which lists the accuracies of applying edited-bootstrapped SVM and the two bootstrapping techniques to the texture image containing 4% outliers, it can be derived that the bootstrapping techniques produced high accuracy when compared to edited-bootstrapped SVM.

4.2.2 Random Data and Simulation Results

The purpose of the random data is to compare the performance of the proposed edited-bootstrapped SVM with the bootstrapping techniques II and IV by considering a random data. The data samples were independently generated from 30 dimensional normal distributions with zero mean. Outliers were generated from

Table 4.2: Comparison of proposed edited-bootstrapped SVM and bootstrapping Techniques II and IV through accuracies obtained by applying the techniques to the texture image containing 4% outliers.

$\nu = 0.04$	Classification Accuracy (P_a)	Precision Accuracy (P_b)	Recall Accuracy (P_c)
Bootstrapping II	94.9	98.5	96.0
Bootstrapping IV	94.7	98.5	96.0
Edited-bootstrapped SVM ($\nu = 0.05$)	89.4	98.2	90.7

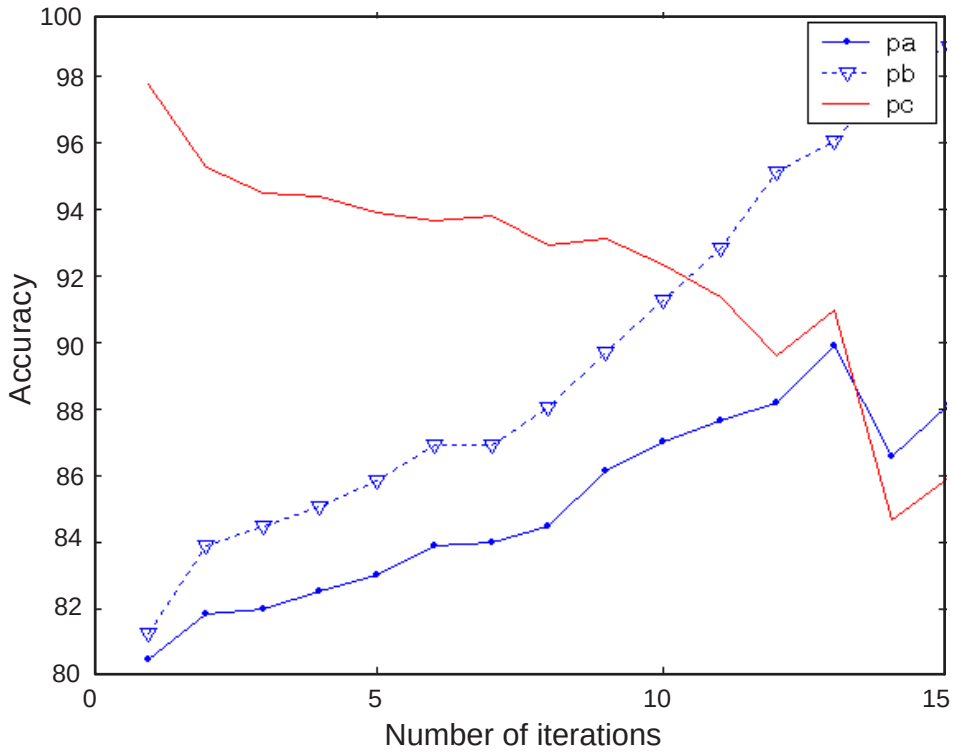


Figure 4.9: Accuracies obtained by applying the edited-bootstrapped SVM on the random data are plotted at different iterations.

same 30 dimensional normal distributions but with a mean equal to 2.56. The generated outliers are added to the data samples with zero mean. The final data contains true data points and also the outliers. The percentage of outliers present in the final data is 25%.

The proposed edited-bootstrapped SVM is applied to the random data generated. The accuracies are plotted against the iterations. In this case the stopping criteria can be the point where the *Precision* and *Recall Accuracies* meet. As can be seen from the figure 4.9 both the accuracies meet at 11th iteration.

Data: : Normally distributed
Dimensionality: : 30
No. of test samples: : 10000
No. of outliers: : 2500
Classifier: : OCSVM
Algorithms tested: : Bootstrapping II & IV, Edited-bootstrapped SVM

The performance of the bootstrapping Techniques II and IV are compared with the edited-bootstrapped SVM. It can be inferred from the table 5.1 that the proposed edited-bootstrapped SVM gives a high accuracy when compared with the other two bootstrapping techniques.

Table 4.3: Comparison of proposed edited-bootstrapped SVM and bootstrapping techniques II and IV through accuracies obtained by applying the techniques to the random data.

$\nu = 0.2$	Classification Accuracy (P_a)	Precision Accuracy (P_b)	Recall Accuracy (P_c)
Bootstrapping II	81.1	87.2	89.2
Bootstrapping IV	81.5	86.8	90.3
Edited-bootstrapped SVM ($\nu = 0.05$)	88.2	92.8	91.4

4.3 Conclusions

The idea of the proposed edited-bootstrapped SVM was applied to different types of data and is also compared with the bootstrapping techniques discussed in chapter 3. Simulation results show that the proposed edited-bootstrapped SVM algorithm results in a higher accuracy when compared with the other bootstrapping techniques if the original data set consists of large number of outliers. Bootstrapping techniques and the edited-bootstrapped SVM have a similar performance if the percentage outliers are low in the original data set.

The two major advantages of the proposed edited-bootstrapped SVM is that even though the original data set consists of huge percentage of outliers the algorithms performs well. Secondly, the unknown parameter in OCSVM (ν) is taken care off by assigning a small value to it. Thus no more the parameter ν is an unknown parameter. Therefore the proposed algorithm performs well in the cases where the data changes in different durations of the year. This algorithm is also tested on such real data i.e., the CRP data which will be discussed in next chapter.

Chapter 5

Application to CRP Analysis

5.1 Remote Sensing Data Analysis for CRP

As mentioned in section 1.4 USDA has to monitor the lands that are enrolled in CRP to check whether the farmers are maintaining the CRP tracts according to contract stipulations. USDAs Common Land Unit (CLU) data used for general compliance issues is generated from aeriel photographs, which are updated every 1-2 years and may not be very efficient for CRP compliance monitoring on a large scale. The existing CRP reference data obtained from USDAs Natural Resource Conservation Service (NRCS) is not very accurate or up-to-date for the management purpose. Automatic compliance monitoring method was proposed discussed in [20], which examines CRP tracts more efficiently and promptly with minimum human involvement.

The reference data provided by NRCS is outdated and has errors due to misalignment of the CRP tracts. The CRP maps were developed based on information provided by the farmers upon enrollment into the program and by manual delineation of aerial photographs. A method was developed in order to periodically update CRP maps based on the reliable training samples. These maps are required for reference purposes and for various assessment activities.

5.1.1 CRP Compliance Monitoring

The previous method involves both SVM and OCSVM. OCSVM is used for the classification of CRP and non-CRP areas. The OCSVM results are used to train a SVM to further refine the previous results. The OCSVM produces a complex decision boundary marked by a large number of support vectors, whereas SVM provides a more natural decision boundary. CRP tracts contain many CRP grass species. To get a more natural decision hyperplane, TCSVM is applied to the results obtained after OCSVM application. The initial results from OCSVM selects some training samples for CRP and non-CRP classes, and then TCSVM is trained and used to reclassify the whole CRP region.

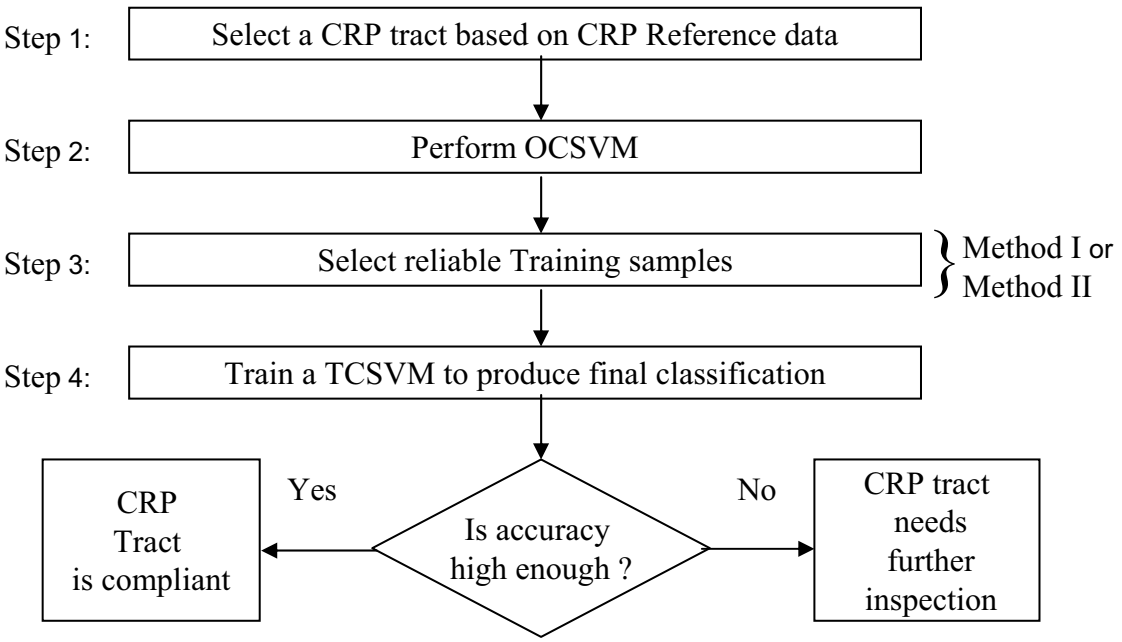


Figure 5.1: Figure taken from [7] represents the flowchart of CRP Compliance Monitoring.

CRP clips are chosen based on the reference data. The major part of the clip is considered to be compliant and also some non-CRP areas were included. The OCSVM is trained several times for different values of ν. The trained OCSVM

is applied to the entire clip. The resulted training samples were used for TCSVM training. The whole CRP clip is reclassified using the trained TCSVM. Final result was compared with the reference data to decide compliance of the clip.

Two different methods were proposed in [7]. The two methods differ in choosing the training data samples as in step3 of figure 5.1. In method I the training samples were chosen in two steps. Model selection was done for the OCSVM as discussed in [7]. The spatial properties of the data were used to decide the reliable samples among the ones classified belonging to both the majority (CRP) and outlier or minority (non-CRP) class. Method II is a ν - insensitive approach as discussed in [20]. This method makes use of pattern distribution within the kernel space to decide on the reliable training patterns.

Thus for CRP compliance monitoring both OCSVM and TCSVM were incorporated together to accomplish self-supervised CRP classification. The consistency of the clip was calculated based on the reference data.

5.1.2 CRP Mapping

CRP mapping is a complex classification problem where both CRP and non-CRP areas are composed of various cover types having highly overlapped clusters in the spectral space of the satellite imagery. The reference data provided by Natural Resources Conservation Service is not accurate and is old. Therefore based on the mapping results the present reference data can be updated. CRP mapping is an uneven classification task where CRP tracts usually consist of less than 10% over all study areas. As CRP mapping is unsupervised classification accuracy, precision and recall accuracies are used to evaluate the overall CRP mapping performance.

In this work two types of machine learning algorithms such as Decision Tree Classifier (DTC) and Support Vector Machine (SVM) can be used. The principle of DTC is to break up a complex classification problem into a union of several

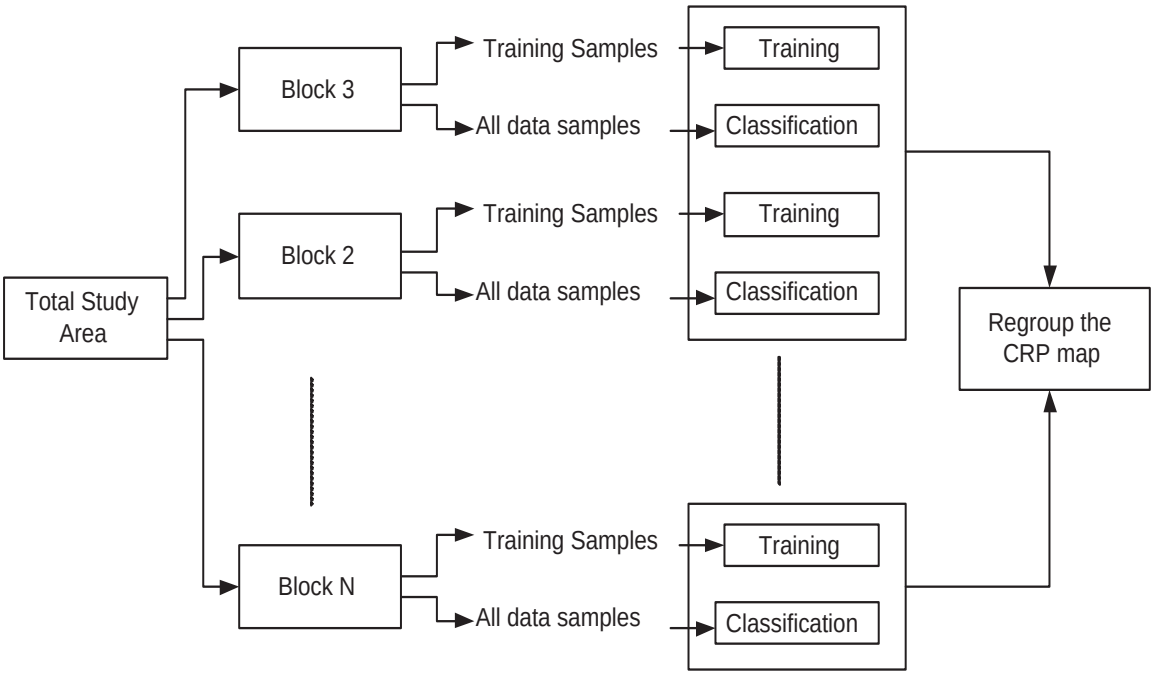


Figure 5.2: Figure taken from [8] represents the Block-based Classification framework.

simpler classification issues. DTC has shown advantages in real remote sensing applications for more than ten years. But because of over fitting problem the performance was poor. Thus SVM is suggested as an alternative to the DTC.

The CRP clip, study area, contains a total of 14 CRP species and also there are other cover types in non-CRP regions. Therefore a localized block-based technique was used to achieve automatic CRP mapping efficiently. That is SVM training and classification is performed on each block initially and their results are combined to rebuild the whole CRP map.

The proposed block-based technique assumes independence across all local blocks. But in reality each block is not completely independent to other blocks. When the block size is smaller, the CRP and non-CRP cover types in one block tend to be purer and more distinct compared with other blocks. If block size is larger, due to more complex cover types in each block, the applicability of a locally

trained SVM to other areas is not good.

5.1.3 Limitations on existing research

It was assumed in compliance monitoring that the majority of a CRP tract is compliant. But there may be the case where the majority of a CRP tract is not compliant. There was a problem in assigning a value to the parameter ν in OCSVM because there is no information about the outliers present in the CRP tracts. OCSVM was used even in CRP Mapping by assigning an arbitrary value to the parameter ν. Localized block-based technique was followed to achieve automatic CRP mapping.

Our proposed edited-bootstrapped SVM combines the CRP compliance monitoring and CRP mapping into a joint framework. It also eliminates the problem of assigning an appropriate value to the parameter ν in OCSVM. The parameter ν is assigned to a smaller value. This helps in removing the outliers in each iteration, thereby purifying the training data samples. It is necessary to develop a method to purify the training data samples as the CRP maps are updated based on the training data samples. Some of the locality errors and spatial misalignment of the CRP tracts can be rectified by choosing reliable training data samples. Instead of dividing the CRP tract into blocks they are divided based on the species and OCSVM training and classification are performed within each species independently. Then the outputs of all species are combined to rebuild the whole CRP map.

5.2 Study Area

The study area that we have chosen is Texas County (Figure 5.3), Oklahoma. As of October 2003 Texas County had 217,802 acres out of the total 1,036,441 acres enrolled in Oklahoma. Here different grass species are grown on the CRP tracts.

This allows us the opportunity to analyze the CRP plots better as plots will have be less variation due to changes in topography and other factors.

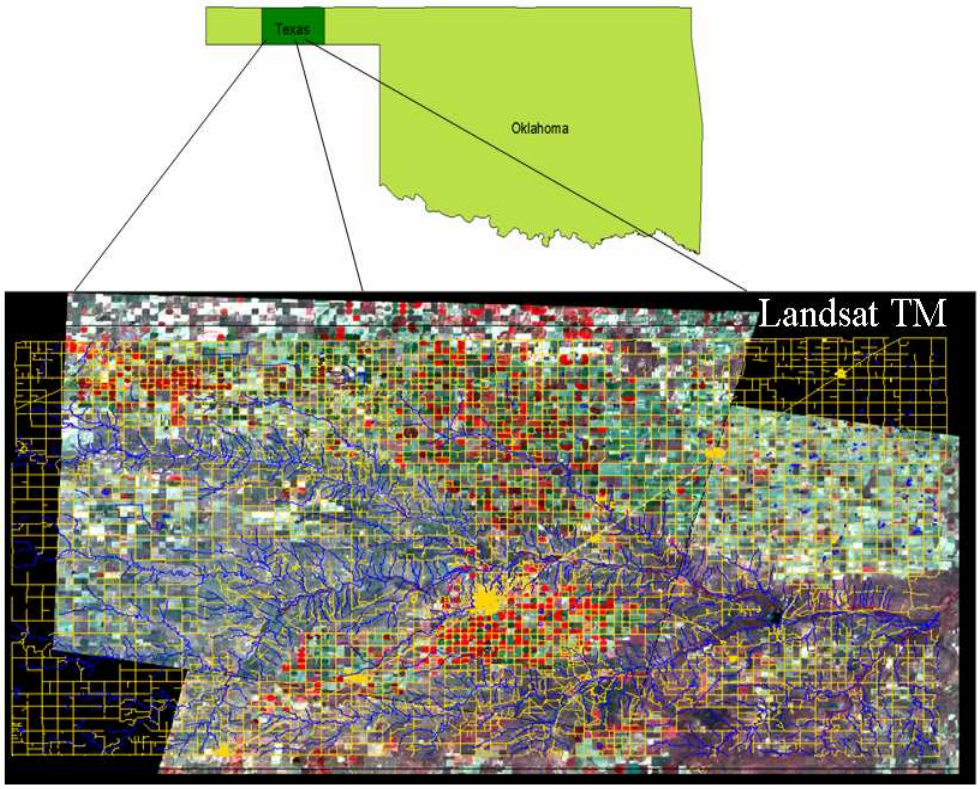


Figure 5.3: Texas County Landsat data superimposed with Road and Stream network information (Courtesy of Dr. Mahesh Rao of the Oklahoma State University's Geography Department).

In our work we are using Landsat TM (Thematic Mapper) images obtained for February, 2000 and June, 2000 (so as to get information for both winter and summer seasons) covering Texas County, Oklahoma. Hence this data set is multi-temporal. Each pixel of this image covers an area of $30m \times 30m$. Landsat TM generated imagery covers seven spectral bands viz. Band 1 (Blue), Band 2 (Green), Band 3 (Red), Band 4 (Near infrared(IR)), Band 5 (Mid IR), Band 6 (Thermal IR) and Band 7 (Mid IR).

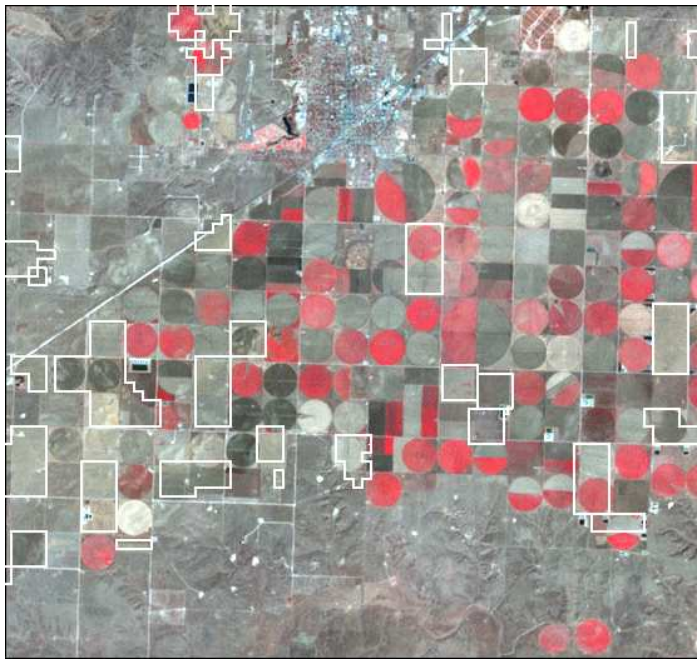


Figure 5.4: Clip of February 2000 Landsat TM image with superimposed CRP ground data (in white polygons).

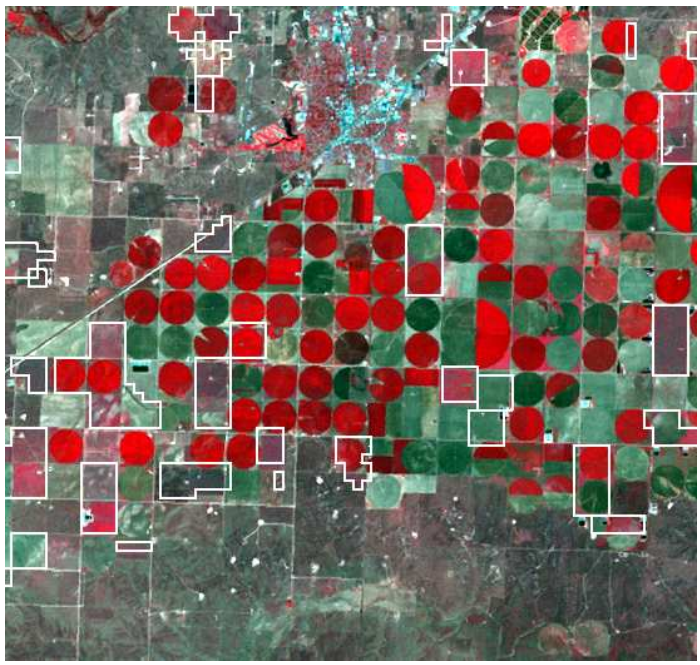


Figure 5.5: Clip of June 2000 Landsat TM image with superimposed CRP ground data (in white polygons).

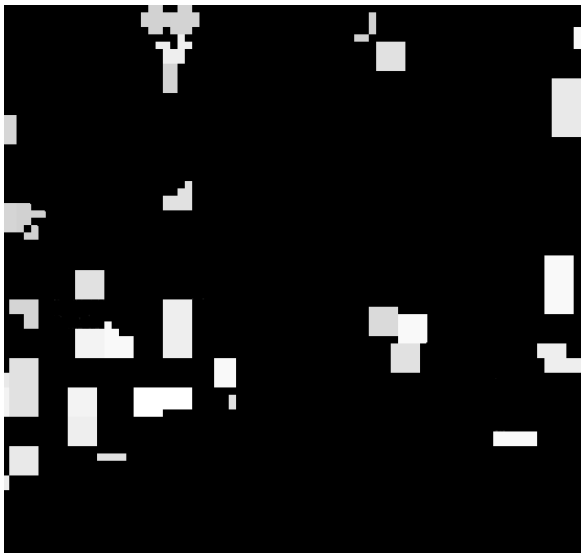


Figure 5.6: Ground data for the simulations. Black regions are non-CRP and grey regions are CRP.

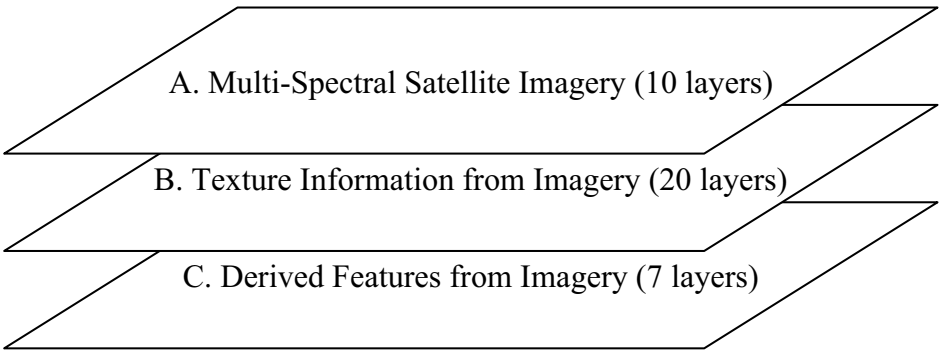


Figure 5.7: Different layers in each pattern.

CRP Reference data (Figure 5.6) is obtained from Natural Resources Conservation Service (NRCS) and is used as the ground data. This data shows the CRP tracts as well as the cover types in each CRP tract as per the information given by the farmer when his/her land is enrolled in the CRP program.

Feature Extraction is adapted from [8]. Each pattern (representing a single pixel) composed of totally 37 layers generated solely from the Landsat TM images as shown in the figure 5.7. The first 10 layers of each pattern consist of Landsat TM bands from each TM image. Bands 1 (i.e. blue band is prone to haze) and 6 (i.e. thermal band which has a different resolution and is not useful in vegetation studies) were excluded. The following 20 layers are texture information that includes the local mean and local variance within a 3×3 window of each band in each season. The last 7 layers consist of different derived features like, Normalized Difference Vegetation Index (NDVI), Band Ratios and Band Differences.

5.3 Experimental Results

The data discussed in section 5.2 is used for the simulations. The area is a clip (figure 5.9) from multi-temporal Landsat data of Texas county of size 552×523 pixels. The clip consists of 14 different CRP species. Simulations are done on each individual species using OCSVM's and then combined together for the final result. OCSVM implementation as in [39] was used for simulation. For OCSVM Guassian kernel was used and the γ value was set to 1 to achieve tighter classification boundaries. The parameter ν was assigned to a lower value i.e., 0.02. In each iteration 5% outliers are removed in each species. The flow chart for the CRP mapping is shown in the figure 5.8.

The result of applying OCSVM to all the 14 CRP species and then combining them to get the final result is shown in the figure 5.9. In this case the % of outliers are assumed to be about 10% in all the CRP species i.e.,value of ν is set

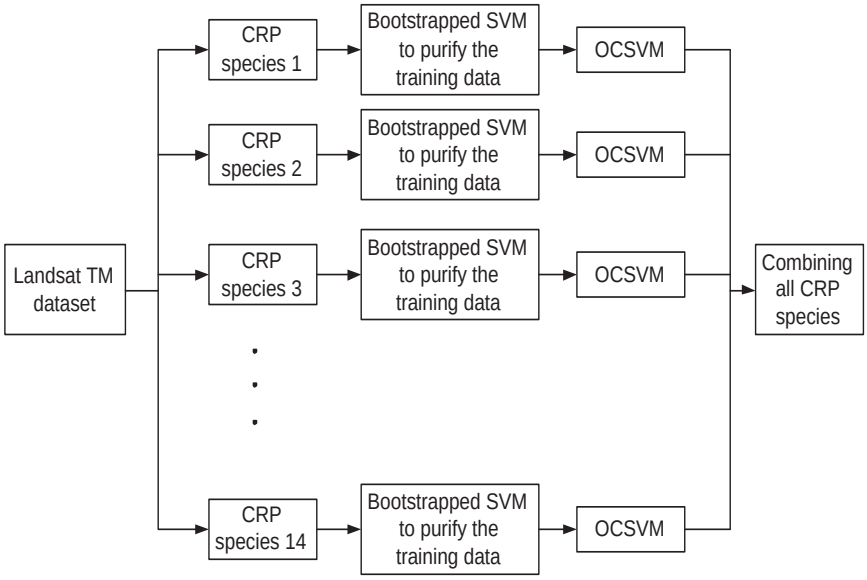


Figure 5.8: Flow chart to represent the process of CRP Mapping.

to 0.1. The classification, precision and recall accuracies obtained in this case are 91.5, 57.7 and 66.0 respectively. Figure 5.10 is the combined result of applying the idea of edited-bootstrapped SVM to all the CRP species individually. The parameter ν is initialized to 0.02. Therefore in each iteration 2% of outliers are removed from each CRP species and then the final results of each species are combined to achieve the final result of the entire study area. The classification, precision and recall accuracies obtained in this case are 95.7, 82.7 and 73.7 respectively. The accuracies increased to a large extent by applying the idea of edited-bootstrapped SVM to the study area. This is because of purifying the training data for each CRP species. The performance of OCSVM increased drastically by the idea of proposed edited-bootstrapped SVM.

5.3.1 Comparison with Bootstrapping Techniques II & IV

The bootstrapping techniques discussed in section 3.1 is applied to the CRP species individually. OCSVM is applied to all the species using the new training data sets calculated by the bootstrapping techniques II and IV. The final classification result

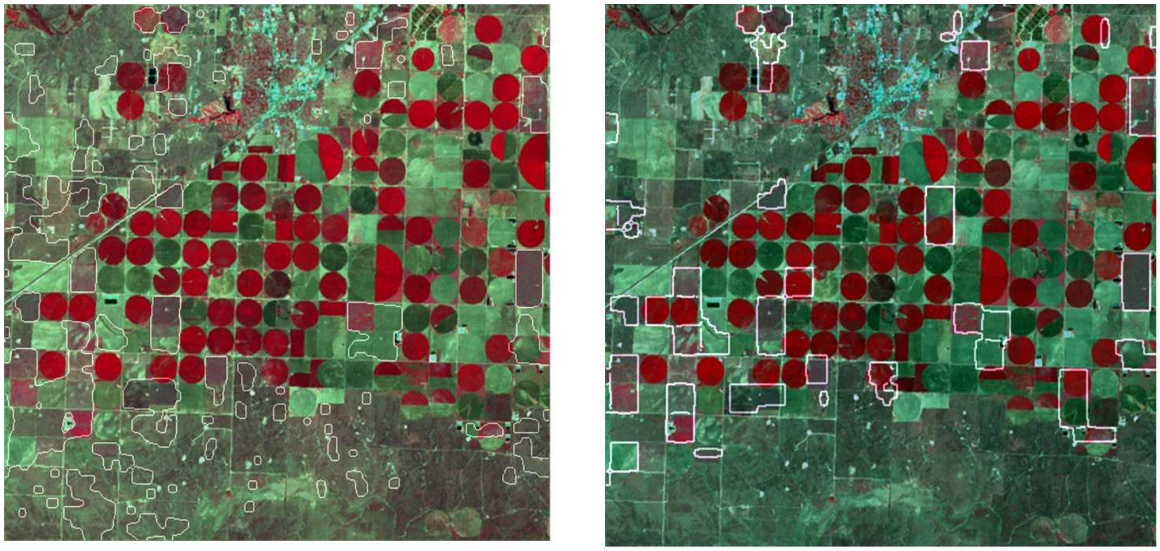


Figure 5.9: (a) Classification result obtained after applying OCSVM and is superimposed on June 2000 TM image. (b) Reference data superimposed on June 2000 TM image.

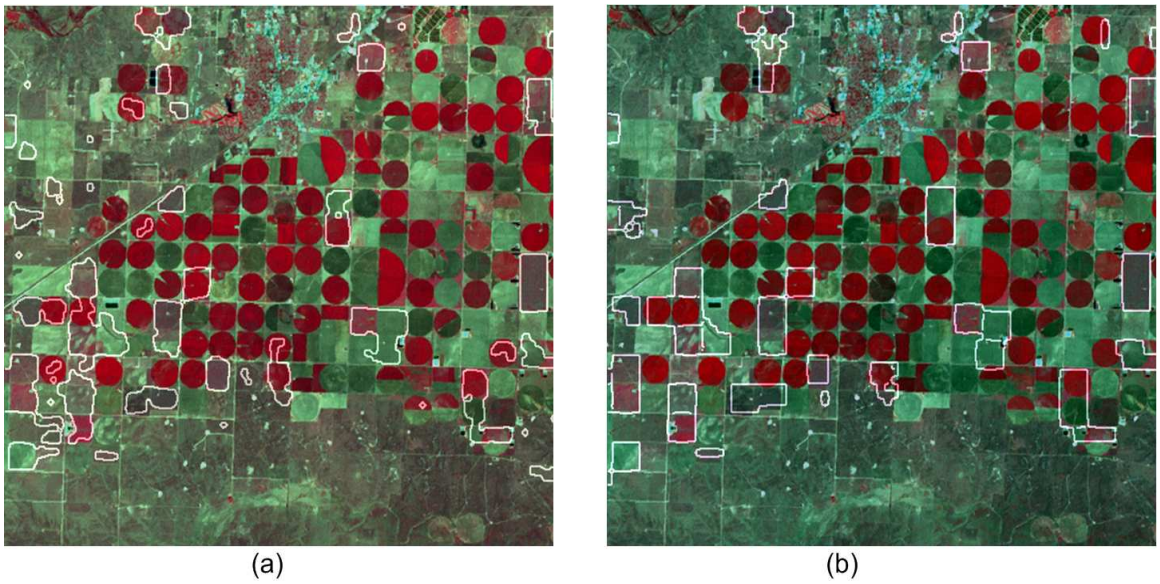


Figure 5.10: (a) Classification result obtained after applying proposed edited-bootstrapped SVM and is superimposed on June 2000 TM image. (b) Reference data superimposed on June 2000 TM image.

after applying the bootstrapping techniques II and IV are shown in the figures 5.11 and 5.12 respectively. In both the cases the ν value is set to 0.1.

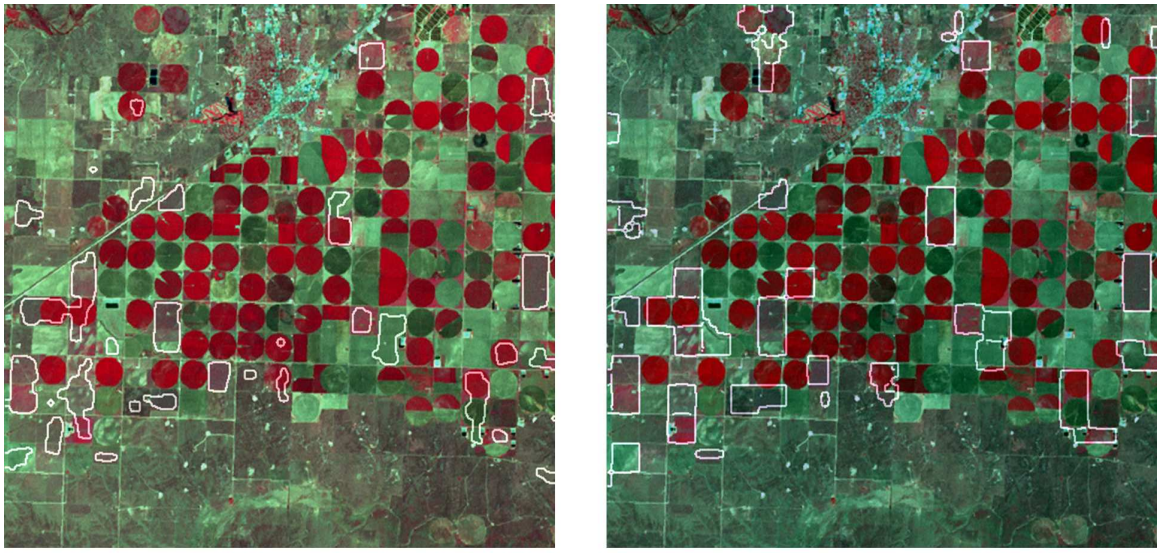


Figure 5.11: (a) Classification result obtained after applying bootstrapping technique II and is superimposed on June 2000 TM image. (b) Reference data superimposed on June 2000 TM image.

By comparing the accuracies obtained in all the four cases it can be concluded that bootstrapping II - IV and edited-bootstrapped SVM improve the performance of SVM. But the recall accuracy which gives the percentage of true CRP pixels that can be detected is low in bootstrapping techniques II and IV when compared with the proposed edited-bootstrapped SVM. The low recall accuracy is also due to the inaccurate reference data. Bootstrapping techniques reduced the effect of outliers by re-sampling. But in this case the percentage of outliers is very high which implies that the true pixels are surrounded by more number of outliers. Therefore there is every possibility that during the process of locally combining original training samples the true pixel gets converted into an outlier. Thus it can be concluded that proposed edited-bootstrapped SVM works better than the bootstrapping techniques II and IV if the percentage of outliers are more in the real data. Our algorithm performance reduces if outliers are very less in

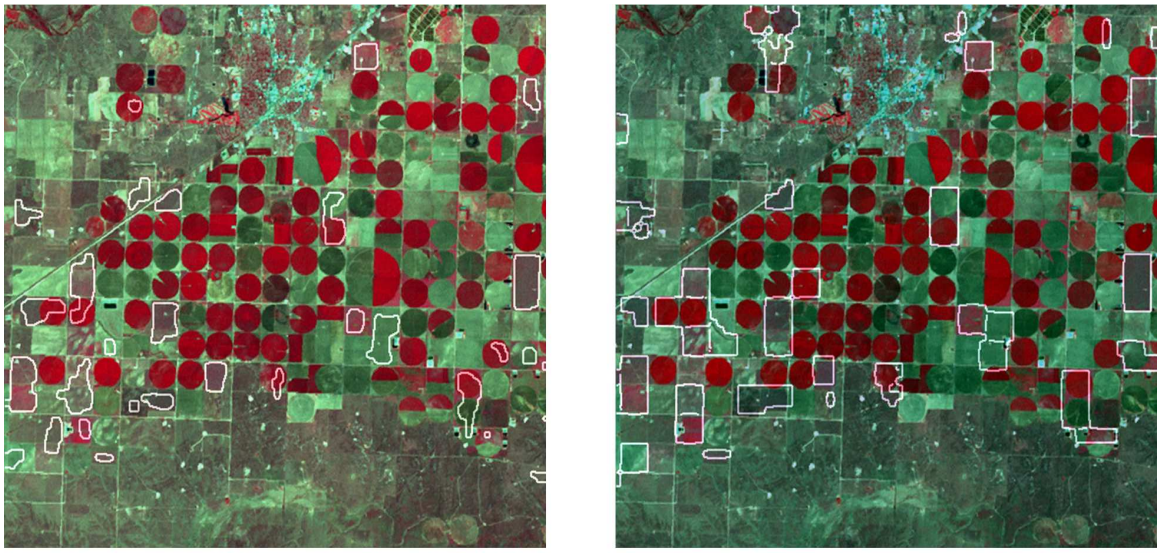


Figure 5.12: (a) Classification result obtained after applying bootstrapping technique IV and is superimposed on June 2000 TM image. (b) Reference data superimposed on June 2000 TM image.

the data set this is because of overestimation.

Table 5.1: Comparison of proposed edited-bootstrapped SVM and bootstrapping techniques II and IV through accuracies obtained by applying the techniques to the real data.

	Classification Accuracy (P_a)	Precision Accuracy (P_b)	Recall Accuracy (P_c)
General SVM	91.5	57.7	66.0
Bootstrapping II	93.3	81.1	46.1
Bootstrapping IV	93.0	82.5	40.4
Edited-bootstrapped SVM	95.7	82.7	73.7

The accuracies are calculated with respect to the reference data shown in the figure 5.6. In the previous work, CRP monitoring and CRP mapping were two different tasks to deal with. But by applying the proposed edited-bootstrapped SVM, CRP monitoring and mapping are combined together. Reasons for not detecting some of the CRP tracts are, either the lands are no more enrolled in

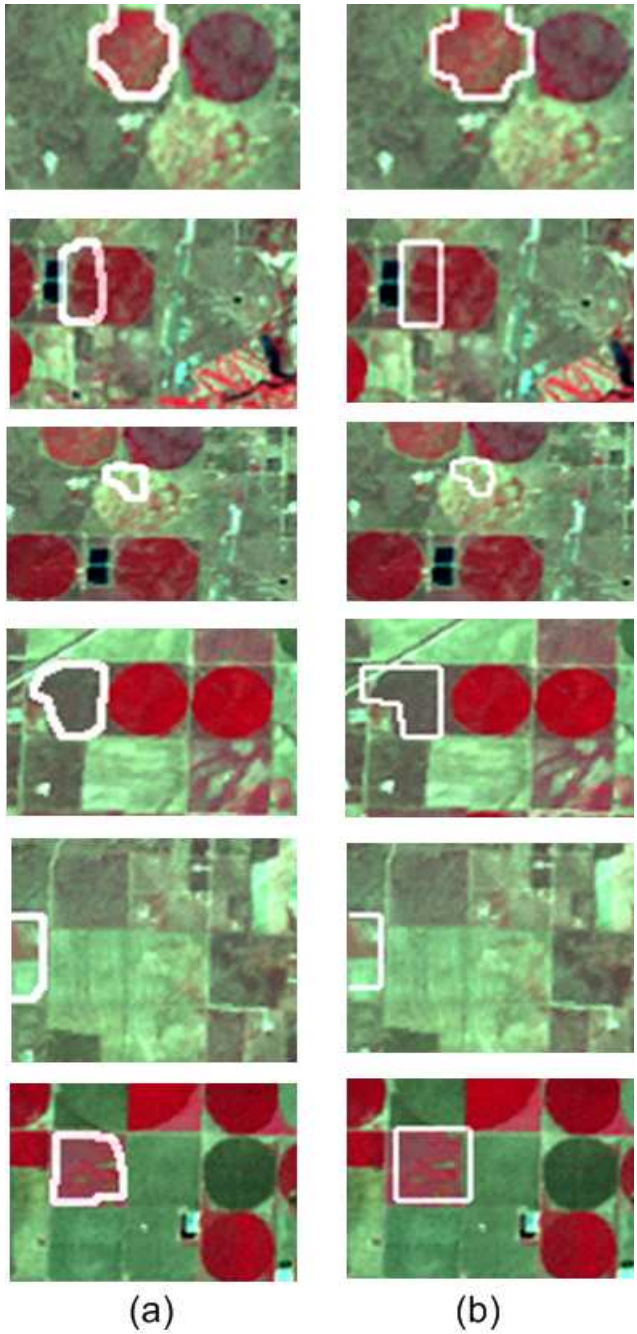


Figure 5.13: (a) Classification results of different CRP species superimposed on June 2000 Landsat TM image. (a) Different CRP species superimposed on June 2000 Landsat TM image based on the reference data.

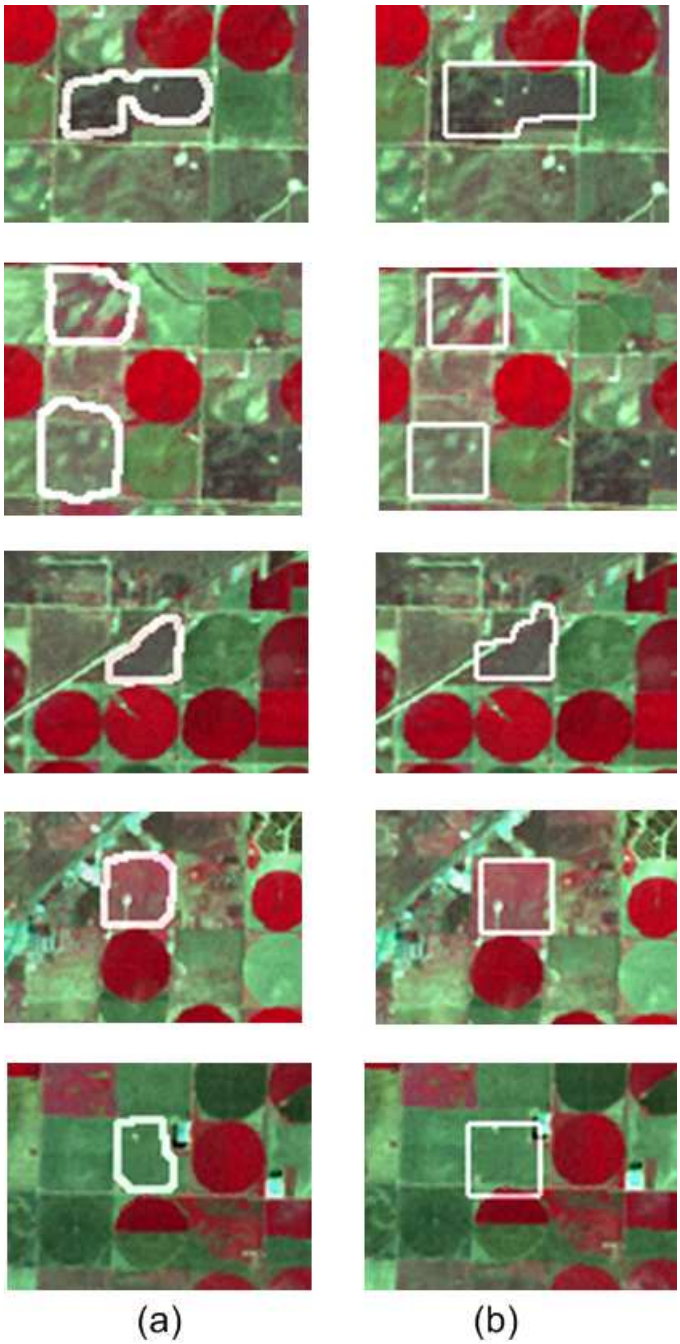


Figure 5.14: (a) Classification results of different CRP species superimposed on June 2000 Landsat TM image. (a) Different CRP species superimposed on June 2000 Landsat TM image based on the reference data.

CRP or the farmers are not following the contract stipulations. In CRP compliance monitoring we assumed that most of the area is compliant. Figure 5.13 and 5.14 represents classification results of different CRP species that are correctly classified. CRP compliance monitoring failed in the cases where most of the training data set is non-compliant. For example, figure 5.15 represents different CRP

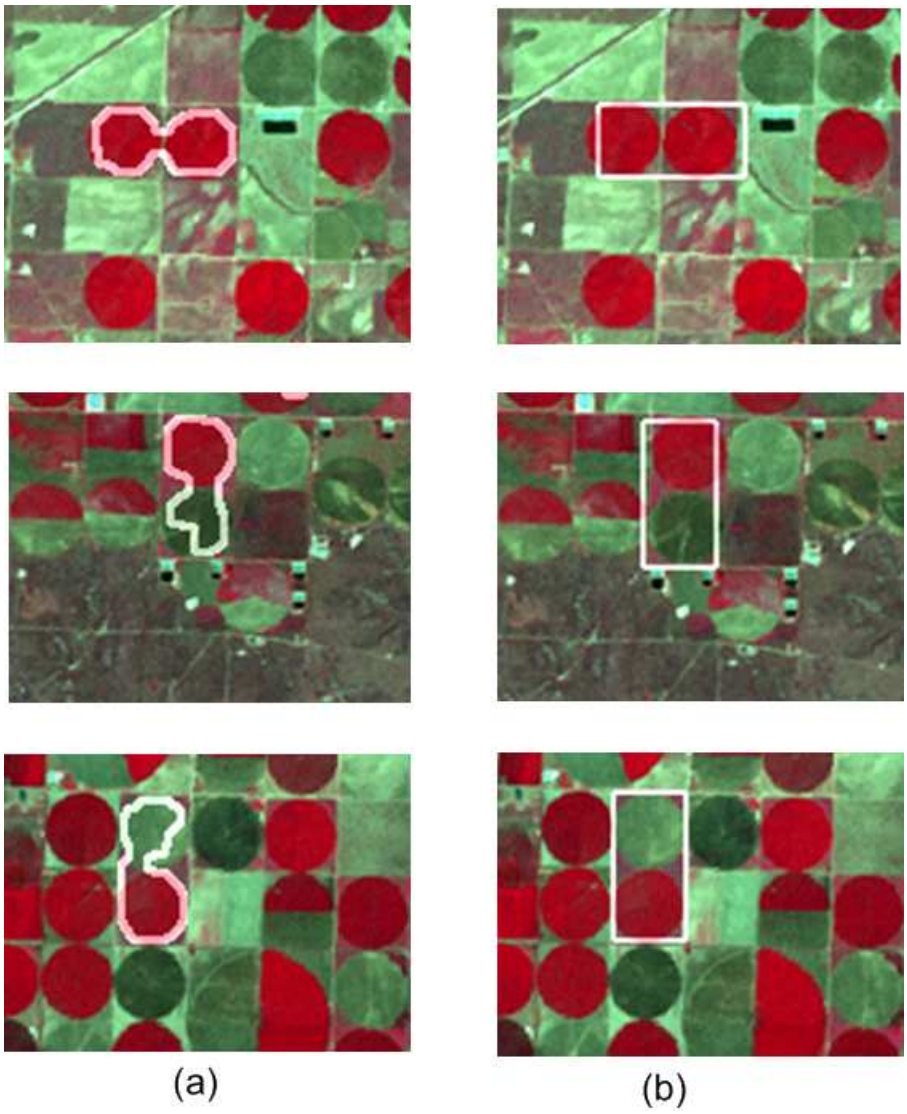


Figure 5.15: (a) Classification results of different CRP species superimposed on June 2000 Landsat TM image that are misclassified. (a) Different CRP species superimposed on June 2000 Landsat TM image based on the reference data.

species wherein our assumption fails and thus leads to misclassification as most of the tract is non-compliant (red color represents Non-CRP land). In such cases

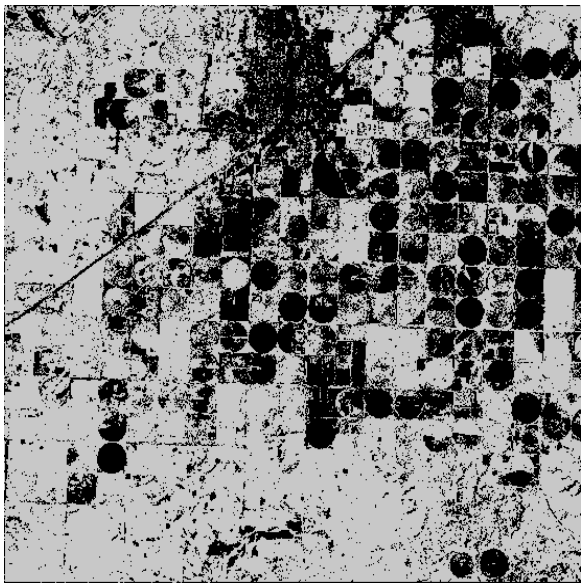


Figure 5.16: Classification result obtained by using 10 dimensional feature vectors.

Non-CRP land cannot be detected as most of the training data is non-compliant. Reducing the dimension of the feature vectors in the feature space results in even poor classification results. Figure 5.16 represent the classification result obtained by reducing the feature vector dimension from 37 to 10 (5 bands of data in two different seasons).

Chapter 6

Conclusions and Future work

In this thesis, we have studied different bootstrapping techniques to improve the performance of SVM classifier. A new edited-bootstrapped SVM was proposed which was derived from edited NN rule. This algorithm was proposed in order to overcome the problems faced in One-class remote sensing analysis. Two specific remote sensing issues, CRP compliance monitoring and CRP mapping, related to USDA's CRP program have been combined by the proposed edited-bootstrapped SVM. In these applications we were able to obtain satisfactory results. This report can be concluded as follows:

- During training, classifiers learn through a given set of data points. Then during testing, classifiers predict the labels of the data points that are to be tested based on what it learnt. If there are any outliers present in the training data set it picks them as the support vectors and thus leads to wrong classifications. Thus edited-bootstrapped SVM algorithm was proposed in which outliers were removed from the training data set, thereby purifying the training data set. The performance of the classifier improved resulting in higher classification accuracies.

- The parameter ν in OCSVM is defined as an upper bound on the fraction

of outliers and a lower bound on the fraction of support vectors. ν can be initialized to an exact value if we have any knowledge about the outliers present in the data set. Wrong initialization leads to misclassification of the testing data set. In edited-bootstrapped SVM the parameter ν is initialized to a low value thereby eliminating small percentage of outliers in each iteration. Thus ν is no more an unknown parameter in OCSVM.

- The two remote sensing issues related to USDA's CRP program are CRP compliance monitoring and CRP mapping. Earlier they were two different issues and many methods were proposed for both the issues individually. By applying the idea of proposed edited-bootstrapped SVM, CRP compliance monitoring and Mapping were combined. This reduces the cost and time for CRP management in maintaining and planning of the CRP lands.

The prospectives of the future research relating to our present research can be listed as follows:

- All the experiments were performed on a small CRP area in Texas County. The idea of edited-bootstrapped SVM can be applied to entire Texas County by dividing the Texas County into different blocks. The performance of the algorithm and SVM can be tested by increasing the size of the study area.

- The proposed edited-bootstrapped SVM can be tested with other classifiers like Nearest Neighbor and genetic algorithms. The classifiers can be trained to classify both the majority and minority data points. For example, in CRP mapping the classifier is trained only on CRP data points and OCSVM is used for classification. Now the training data consists of both CRP and non-CRP data points and Two-Class SVM can be used as a classifier.

- The idea of condensing can be included in the proposed edited-bootstrapped SVM. Thus the size of the training dataset can further be reduced resulting in less computational time.

Bibliography

[1] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York: Spinger-Verlag, 1995.

[2] "Conservation reserve program.." http://www.fsa.usda.gov/dafp/cepd/crp.htm.

[3] O. Bousquet, S. Boucheron, and G. Lugosi, *Introduction to Statistical Learning Theory*, vol. Lecture Notes in Artificial Intelligence 3176, pp. 169–207. Heidelberg, Germany: Springer, 2004.

[4] K. R. Muller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf, "An introduction to kernel-based learning algorithms," *IEEE Transactions on Neural Networks*, vol. 12, pp. 181–201, March 2001.

[5] C. F. Eick, "Data mining dm05, university of houston." Course Document, Fall 2005.

[6] F. J. Ferri, J. V. Albert, and E. Vidal, "Considerations about sample-size sensitivity of a family of edited nearest-neighbor rules," *IEEE transactions on Systems, Man, and Cybernetics-part B: Cybernetics*, vol. 29, pp. 667–672, August 1999.

[7] G. Cherian, "Support vector machines for conservation reserve program (crp) mapping and compliance monitoring," Master's thesis, Oklahoma State University, December 2004.

[8] X. Song, G. Fan, and M. Rao, "Automated crp mapping using non-parametric

machine learning approaches.," *IEEE Trans. Circuits and System for Video Technology*, vol. 43, pp. 888–897, July 2005.

[9] F. Roli and G. Fumera, "Support vector machines for remote-sensing image classification.," in *SPIE*, 2001.

[10] M. Brown, H. G. Lewis, and S. R. Gunn, "Linear spectral mixture models and support vector machines for remote sensing.," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 38, pp. 2346–2360, September 2000.

[11] M. Pal and P. M. Mather, "Support vector machines for classification in remote sensing," *Internatinal Journal of Remote Sensing*, vol. 26, no. 5, pp. 1007–1011, 2005.

[12] L. Hermes, D. Frieauff, J. Puzicha, and J. M. Buhmann, "Support vector machines for land usage classification in landsat tm imagery.," pp. 348–350, 1999.

[13] V. N. Vapnik, *Statistical Learning Theory*. New York: John Wiley, 1998.

[14] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data mining and knowledge discovery*, vol. 2, pp. 121–167, June 1998.

[15] D. M. J. Tax, *One Class Classification*. PhD thesis, Technische Universities Delft, The Netherlands, 2001.

[16] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural computation*, vol. 13, no. 7, pp. 1443–1471, 2001.

[17] Y. Hamamoto, S. Uchimura, and S. Tomita., "A bootstrap technique for nearest neighbor classifier design.," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 73–79, Jan 1997.

[18] P. E. Hart, "The condensed nearest neighbor rule," *IEEE transactions on Information Theory*, vol. IT, no. 14, pp. 515–516, 1968.

[19] P. A. Devijver and J. Kittler, "On the edited nearest neighbor rule," pp. 72–80, Proc. 5th international conference on Pattern Recognition, 1980.

[20] G. Cherian, X. Song, G. Fan, and M. Rao., "Application of support vector machines for automatic compliance monitoring of the conservation reserve program (crp) tracts.," in *IEEE International Geosciences and Remote Sensing Symposium (IGARSS2004)*, September 2004.

[21] X. Song, G. Cherian, and G. Fan, "A ν-insensitive svm approach for compliance monitoring of the conservation reserve program.," *IEEE Geosciences and Remote Sensing Letters*, vol. 2, pp. 99–103, April 2005.

[22] M. Simard, S. S. Saatchi, and G. D. Grandi, "The use of decision tree and multiscale texture for classification of jers-1 sar data over tropical forest," *IEEE transactions on Geoscience and Remote Sensing*, vol. 38, pp. 2310–2321, September 2000.

[23] S. L. Egbert, R. Y. Lee, K. P. Price, M. D. Nellis, and R. Boyce, "Mapping conservation reserve program (crp) lands using multi-seasonal thematic mapper imagery.," *GeoCarto International*, vol. 13, no. 4, pp. 17–24, 1998.

[24] S. L. Egbert, S. Park, K. P. Price, R. Y. Lee, J. Wu, and M. D. Nellis, "Using conservation reserve program maps derived from satellite imagery to characterize landscape structure.," *Computers and Electronics in Agriculture*, vol. 37, pp. 141–156, December 2002.

[25] K. I. Kim, K. Jung, S. H. Park, and H. J. Kim, "Support vector machines for texture classification," *IEEE Transactions on Pattern Analysis And Machine Intelligence*, vol. 24, pp. 1542–1550, November 2002.

[26] S. Venkataraman, D. Metaxas, D. Fradkin, and C. Kulikowski, "Distinguishing mislabeled data from correctly labeled data in classifier design," in *16th IEEE International Conference on Tools with Artificial Intelligence*, IEEE Computer Society, 2004.

[27] V. N. Vapnik, "An overview of statistical learning theory," *IEEE Transactions on Neural Networks*, vol. 10, pp. 988–999, September 1999.

[28] D. M. J. Tax, *One-class Classification.* PhD thesis, Technische Universiteit Delft, The Netherlands, 2001.

[29] K. I. Kim, C. S. Shin, M. H. Park, and H. J. Kim, "Support vector machine-based text detection in digital video," vol. 2, pp. 634–641, Proc. of the 2000 IEEE Signal Processing Society workshop, December 2000.

[30] F. Melgani and L. Bruzzone, "Classification of hyperspectral remote sensing images with support vector machines," *IEEE transactions on Geoscience and Remote Sensing*, vol. 42, pp. 1778–1790, August 2004.

[31] L. Hermes, D. Frieauff, J. Puzicha, and J. M. Buhmann, "Support vector machines for land usage classification in landsat tm imagery," in *Geoscience and Remote Sensing Symposium*, vol. 1, pp. 348–350, 1999.

[32] B. Efron, "Bootstrap methods: Another look at the jackknife," vol. 7, pp. 1–26, 1979.

[33] G. W. Gates, "The reduced nearest neighbor rule," *IEEE transactions on Information Theory*, vol. IT, no. 18, pp. 431–433, 1972.

[34] D. L. Wilson, "Asymptotic properties of nearest neighbor rules using edited data," *IEEE transactions on Systems, Man, and Cybernetics*, vol. SMC, pp. 408–421, May/June 1972.

[35] L. I. Kuncheva, "Editing for k-nearest neighbors rule by a henetic algorithm," *Pattern Recognition Letters 16*, pp. 809–814, March 1995.

[36] X. Li, Y. Zhu, and E. Sung, "Sequential bootstrapped support vector machines - a svm accelerator," vol. 3, pp. 1437–1442, Proc. International joint conference on Neural Networks, July 2005.

[37] V. V. Saradhi and M. N. Murthy, "Bootstrapping for efficient handwritten digit recognition," *Pattern Recognition*, vol. 34, no. 5, pp. 1047–1056, 2001.

[38] K. I. Kim, K. Sung, S. H. Park, and H. J. Kim, "Support vector machines for texture classification," *IEEE transactions Pattern Analysis and Machine Intelligence*, vol. 24, pp. 1542–1550, November 2002.

[39] "Libsvm: a library for support vector machines.," 2003. http://www.csie.ntu.edu.tw/cjlin/libsvm.

VITA

Anne Krishna Sravanthi

Candidate for the degree of

Master of science

EDITED-BOOTSTRAPPED SUPPORT VECTOR MACHINES FOR ONE-CLASS REMOTE SENSING DATA ANALYSIS

Major Field: Electrical and Computer Engineering

Biographical:

Born in Andhra Pradesh, India, on July 12, 1981, Son of Anne Samba Siva Rao and Anne Lakshmi.

Received a Bachelor of Technology degree from Nagarjuna University in Electronics and Communication Engineering in May 2003. Completed the requirements for the Master of Science Degree in with a major in Electrical and Computer Engineering at Oklahoma State University in July, 2006.

Worked as a lecturer in Koneru Lakshmaiah College of Engineering, Department of Electronics and Communication Engineering, Andhra Pradesh, India from August 2003 till June 2004. Presently working as a Research Assistant in the department of Electrical and Computer Engineering at Oklahoma State University since February 2005.