LAPPEENRANTA UNIVERSITY OF TECHNOLOGY
Department of Information Technology
Laboratory of Applied Mathematics

Sapna S. Sharma

Bayesian classification of forest ecological type from satellite data

The topic of this Master's thesis was approved by the department council of the Department of Information Technology on 16 April 2007.

The examiners of the thesis were Professor Heikki Haario and PhD Tuomo Kauranne. The thesis was supervised by PhD Tuomo Kauranne.

Lappeenranta, August 12, 2007

Sapna Sharma
15 A 9 Orioninkatu
53850 Lappeenranta
+358 509 270220
sapna.sharma@lut.fi

ABSTRACT

Lappeenranta University of Technology
Department of Information Technology

Sapna S Sharma

Bayesian classification of forest ecological type from satellite data

Master's Thesis

2007

51 pages, 33 figures, 2 tables and 4 appendices

Examiners: Professor Heikki Haario

Dr Tuomo Kauranne

Keywords: Bayesian Networks, Bayesian Parameter estimation, Feature extraction, Support Vector Machine(SVM), Relevance vector Machine.

The main objective of this study was to do a statistical analysis of ecological type from optical satellite data, using Tipping's sparse Bayesian algorithm. This thesis uses "the Relevence Vector Machine" algorithm in ecological classification between forestland and wetland. Further this bi-classification technique was used to do classification of many other different species of trees and produces hierarchical classification of entire sub-classes given as a target class.

Also, we carried out an attempt to use airborne image of same forest area. Combining it with image analysis, using different image processing operation, we tried to extract good features and later used them to perform classification of forestland and wetland.

ACKNOWLEDGEMENTS

I am very grateful to my respected supervisor Dr. Tuomo Kauranne, Prof Heikki Haario, Dr Matti Heiliö, and the entire research group for providing me with such an ambitious project in order to fulfil my master thesis at LUT. This work has widened my knowledge from basic regular courses taught in class to an advanced application. Tuomo Kauranne taught me many things to deal with in this project; his direction starts from how to start till end.

I heartily wish to thank the Department of Information technology; especially the branch of techno-mathematics, for providing me advanced mathematical knowledge along with sophisticated computational skills, without which to solve any real world problem seems hard. Also I wish to thank them for their kind help in terms of financial, social and emotional support for my entire stay and survival in Finland. I pay my salute to the branch of Information processing to give me introduction to image processing and machine learning.

Last but not least, my mummy, papa, brother, elders and entire family, they were and are my backbone. They supported me from every dimension, we think of. I respect their sacrifices, deep emotions (especially unspoken love of my dad) and love for me.

Few of best persons I had with me through out my entire studies work here in Finland at LUT, they are very close to me and they tried to fill a parental gap during my stay. I wish all of them best wishes, love, success and happiness to my beloved Arjun and to both of my best friends Paritosh and Srujal. I cannot forget the people who are far away yet close to my heart.

August 12, 2007

Contents

VOCABULARY

LAMF	The Lake Abitibi Model Forest
SAR	Synthetic Aperture Radar
CPT	Conditional Probability Table
SVM	Support Vector Machine
RVM	Relevance Vector Machine
CON	Types Coniferous forest
HWOOD	Types Hardwood forest
TREESOIL	Types of Treesoil in forest
OPENSOIL	Types of Opensoil in forest
Data1	Data obtained on date 23th Feb 2005
Data2	Data obtained on date 30th Jan 2005

NOTATIONS

General

$\equiv$	equivalent to (or defined to be)
$\propto$	proportional to
$\log(x)$	logarithm base 10 of x
e^x	exponential of x-that is, *e* raise to the power of x
∂ f(x)/∂ x	partial derivative of f with respect to x
$\int_a^b f(x)$	the integral of f with respect to x
θ	Unknown parameters
F(x;θ)	function of x, with implied dependence upon θ

Mathematical Operations

$\overline{x}$	mean or average value of x
$\sum_{i=1}^{n} a_i$	the sum from i=1 to n that is, $a_1 + a_2 + \ldots + a_n$
$\prod_{i=1}^{n} a_i$	the product from i=1 to n that is, $a_1 \times a_2 \times \ldots \times a_n$

Vectors and Matrices

I	Identity matrix, a square matrix having 1's on the diagonal and 0 everywhere else.
diag$(a_1, a_2, \ldots, a_d)$	matrix whose diagonal elements are $a_1, a_2, \ldots, a_d$ and off-diagonal elements are 0.
$\mathbf{x}^t$	transpose of vector **x**
$\|x\|$	Euclidean norm of vector **x**
$\sum$	covariance matrix
A^{-1}	the inverse of matrix **A**
A'	pseudoinverse of matrix **A**

$\|A\|$	determinant of **A**
λ	eigenvalue
e	eigenvector
x$\in$D	**x** is an element of set D
x$\notin$ D	**x**is not a element of set D

Distributions

w	state of nature
P(.)	probability mass
p(.)	probability density
P(a,b)	the joint probability-that is, the probability of having both a and b
p(a,b)	the joint probability density-that is, the probability density of having both a and b
p(**x**|θ)	the conditional probability of a desity of **x** given θ
w	weight vector
$\lambda(.,.)$	loss function
$\nabla_\theta = \left(\frac{\partial}{\partial\theta_1}\frac{\partial}{\partial\theta_2}\cdots\frac{\partial}{\partial\theta_d}\right)$	gradiant operator in θ coordinates
$\hat{\theta}$	maximum-likelihood estimate of θ
	"has the distribution"-for example, $p(x)$ $N(\mu,\sigma^2)$ means that the density of x is normal, with mean μ and variance σ^2
$N(\mu,\sigma^2)$	normal or Gaussian distribution with mean μ and variance σ^2
$N(\mu,\sum)$	multidimentional normal or Gaussian distribution with mean μ and variance σ^2
$\Gamma(.)$	Gamma function
n!	*n* factorial-that is, n$\times$(n-1)$\times$(n-2)$\times\ldots\times$1
$\delta(x)$	Dirac delta function, which had value 0 for x$\neq$0, and integrates to unity

1 INTRODUCTION

1.1 Background

The present Master thesis is part of a study of the Lake Abitibi Model Forest (LAMF) in Canada, established in 1992. It is one of the ten original Canadian Model Forest sites established by the Canadian Forest Service. Its aim is to generate more ideas for forest management with help of local, national and international partners.

The LAMF is located in the clay belt section of the boreal forest. It covers 1.2 million hectares of boreal forest. The soil is predominantly clay and silt-clay with glacial features of out washes, eskers and moraines. The forest is dominated by the black spruce, balsam fir, white birch, trembling aspen, cedar, tamarack and Jack pine type of tree species. In addition there are water bodies and streams. The LAMF is a habitat of protected species of wildlife including Golden Eagle, Bald Eagle, Eastern cougar, Woodland caribou, Short-eared owl, Great Gray owl, Black Term and Monarch butterfly. The study area has been influenced by fire and several human activities such as timber harvesting, mining and agricultural activities. These activities change the landscape and consequently affect forest inhabitants.

1.2 Objective and Structure of Thesis

In this Master Thesis, I am applying a sparse Bayesian classifier developed by Tipping to analyzing many different types of satellite images as to their capability to separate forest land from wetlands in boreal forests.

Tipping's sparse Bayesian algorithm has been successfully applied to forest measurement tasks from airborne laser scanned data, In the current thesis the target is to study algorithms for ecological classification using cheap but coarse grained satellite images. Tipping's Bayesian method appears well suited to the present task, but it will need to be provided with very different feature extraction from those used in airborne imagery.

Our department of Lappeenranta University of Technology has obtained images from five different satellites, including optical, hyper spectral and synthetic aperture microwave radar (SAR) satellite instruments. All images are captured from the same area near Lake

Abitibi in northern Ontario, where also a field classification of over 80 sample plots has been carried out to provide ground truth.

I am trying to develop appropriate feature extraction procedures for several different satellite instrument types, modify Tipping's method accordingly and produce classification results that will tell which satellite instruments and algorithms are able to provide reliable separation between forestlands and wetlands, and possibly even between the classes in a more refined ecological classification.

2 BAYESIAN NETWORKS

The motivation behind the development of Bayesian networks has its roots in the regular study of Bayesian probabilistic theory, which is a branch of mathematical probability and allows us to model uncertainty about the aim and outcome of interest by combining experimental knowledge and observational evidences. The following chapter will gives us a structure to develop any Bayesian network for any kind of problem. In order to get an entire over view, from basic to advanced application, by considering an example of type of data or observation and different classification technique which we are dealing with in a project.

2.1 Structure of a Bayesian belief network

The Bayesian network or The Bayesian belief network is graphically shown in Figure 2.1, Bayesian belief networks are models in which each variable is represented by a NODE and the NODE is related to other nodes by using an EDGE, denoted by an arrow.

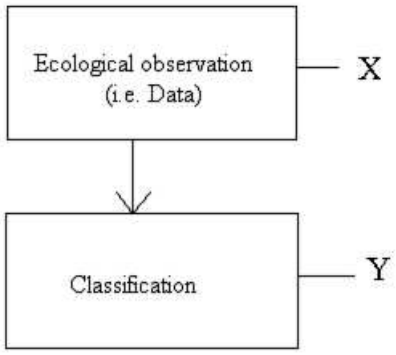


Figure 2.1: Bayesian belief network's very simplest example is two nodes and an edge.

2.1.1 Nodes

A variable which is to be modelled is called a node. Graphically it is shown by a labelled box. The example in Figure 2.1 shows two nodes representing two variables of ecological

data and classification.

2.1.2 Edges

A relationship between two nodes is defined by an edge. It is represented graphically by an arrow between nodes, whereas the direction of the arrow shows the direction of effect. Drawing an edge from node X to node Y indicates that node X has a direct influence on node Y. For example, in Figure 1, the edge shows that the type of Ecological Observation data directly influences the classification.

How nodes influence one another is defined by conditional probability, when two nodes are joined by an edge, the causal node is called the parent of the other node.

In this example, Ecological observation is a parent of Classification, and Classification is the child of Ecological observation. Child nodes are conditionally dependent upon their parent nodes. Graphical representation is shown in Figure 2.2

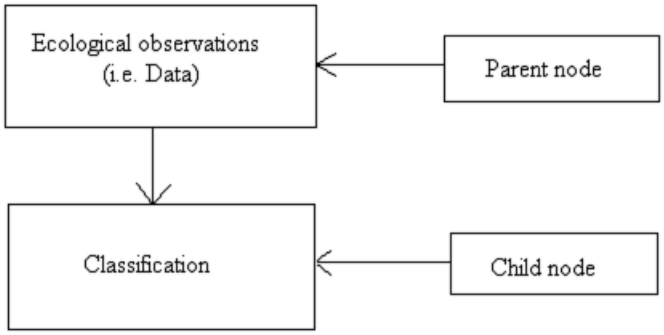


Figure 2.2: An edge indicates effect and conditional dependence.

2.1.3 States

The values taken on by a variable (represented by a node) are referred to as states. For example, two important states of the Dataset variable are linearly separable case and non-linearly separable case. In the linearly seperable case, there exist a hyperplane that separates the points that belong to two different classes. In the unknown separable case, separation must be carried out with a manifold that is not linear. We know that later, a

type of classes, dataset gives results for classification between Wetland and Dryland. The Wetland and Dryland are the states of the Classification nodes, as shown in Figure 2.3.

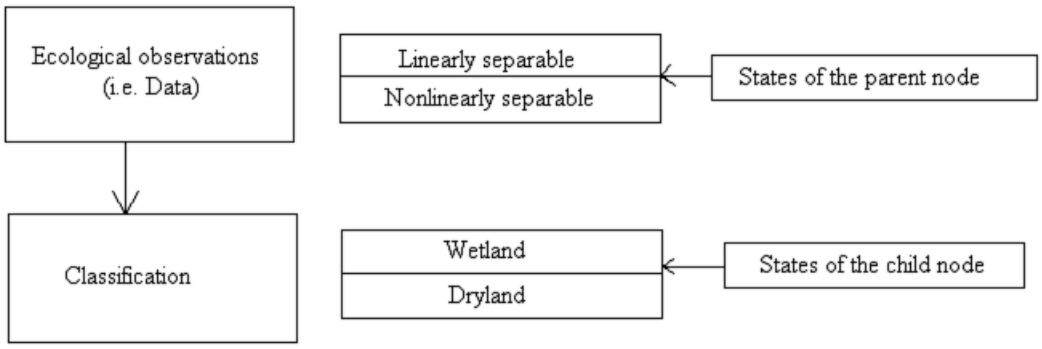


Figure 2.3: States are values that can be taken on by a node.

2.2 Conditional probability tables

Every node is associated with a conditional probability table(CPT) with it. Conditional probabilities represent likelihoods based on prior information or past experience.

A conditional probability table is graphically represented in table form as shown in Figure 2.4 and 2.5. The left most column represents Parent node, directly below that are the names of all posibilities. In this case, the node only has one parent, so there is only one column on that side of the table, but there may be more. On the right hand side of the table, we list columns of CHILD node.[1] Directly below this the state names for the child node are shown. The rest of the table holds the probabilities.

The CPT[1] shown in Figure 2.4 gives detailed information about classification techniques such as "Given Data set classified is wetland or forestland, what is high probability that the Classification will be either of them?" That question corresponds to the last row in the table, and the answer is Ecological dataset with linearly separable cases gives better results for e.g. 0.98 or 98%, whereas on the other hand a nonlinearly separable case gives results of 70%. Information about the probabilities for each state is given in Figure 2.5 which is CPT for a parent node, like in this case Ecological data containing 0.56% of wetland and 0.44% of dryland.

[1] In case of Nodes with no parents, then the CPTs are simpler and consist only of the probabilities for each state of the node under consideration.

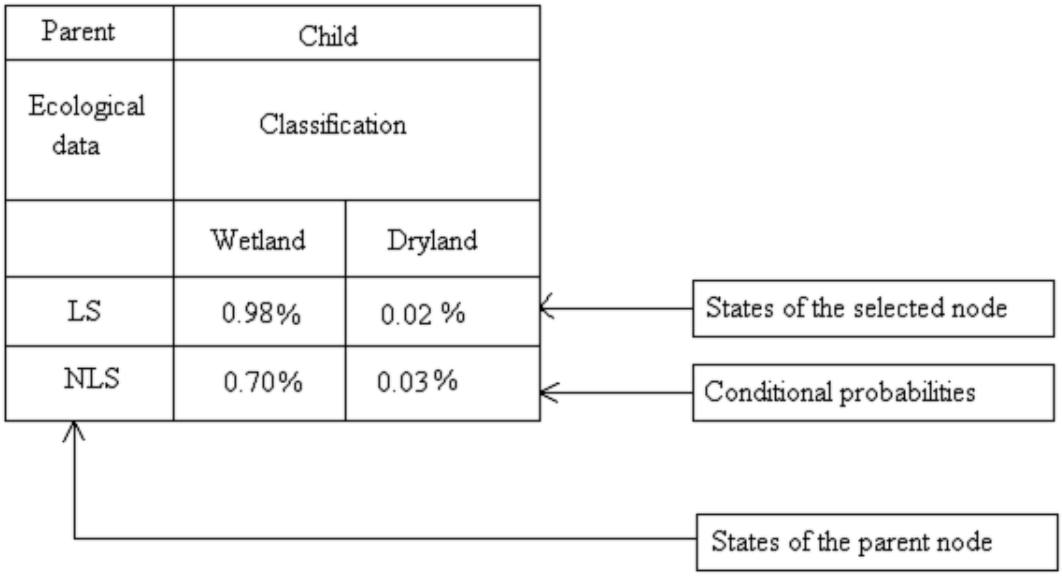


Figure 2.4: Conditional probability tables for a child node

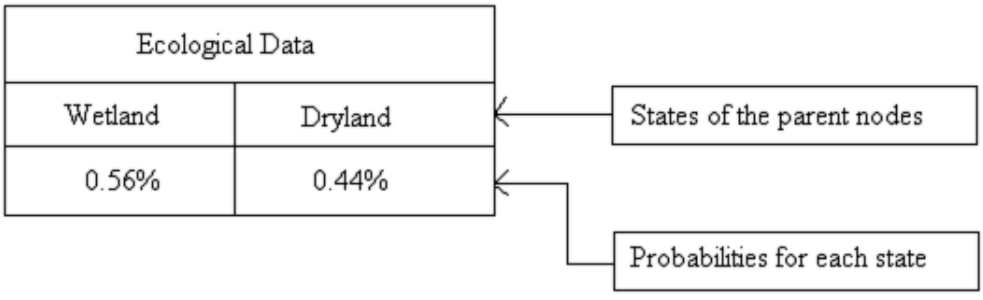


Figure 2.5: Conditional probabiity tables for a child node

2.3 Beliefs and evidence

Beliefs are the probability that a variable will be in a certain state based on the evidence in a current situation. Whereas A-priori beliefs are a special case of beliefs that are based only on prior information, A-priori beliefs are determined only by the information stored in the belief network's CPTs.[1]

Evidence gives knowledge about present situation. For example, in the simple Bayesian belief network shown in Figure 2.6, we have evidence that there is currently a Data set of linearly separable case. The effect of this evidence on current beliefs is reflected in the Beliefs column of the Data Set node: we are 100% sure that we have a correctly classified dataset.

The belief network also shows how the evidence changes the current beliefs about the states. The states are "Classify" and "D/C(does not classify)". The current beliefs shown in column of classification node shows likelihood of "Classify" is 98%, only when we had evidence that there was linearly separable Data set.

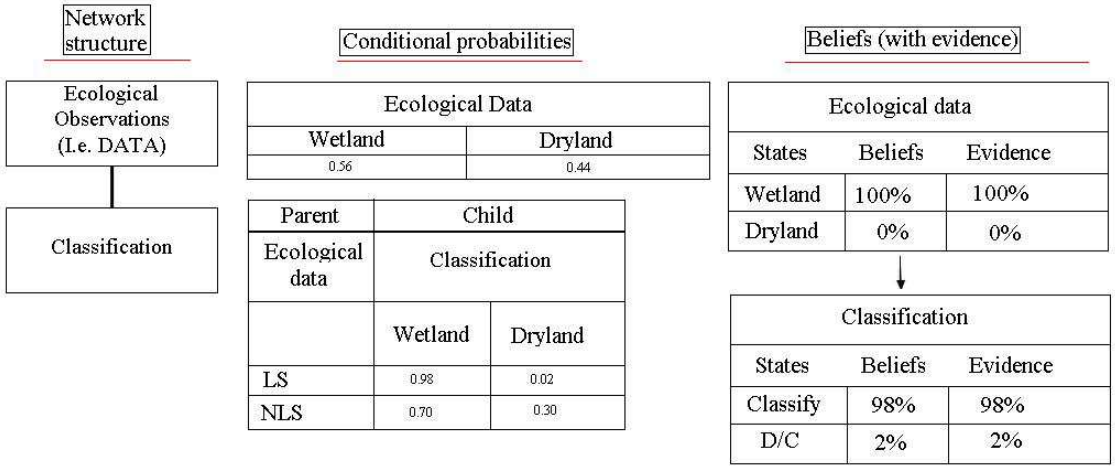


Figure 2.6: Belief network showing the effect of Data set on classification.

- Hard evidence
  The evidence is that a node is 100% in one state, and 0% in all other states. (Hard evidence was posted to the belief network shown in Figure 6.)

- Soft evidence
  The evidence is not hard evidence, in other words, evidence that a node is less than 100% in one state, and/or greater than 0% in other states. Soft evidence is

often used for information about which there is some uncertainty, such as from conflicting reports or an unreliable source.

By posting evidence to a Bayesian belief network, one can do analyze of current beliefs to predict a result or to diagnose a cause.

3 BAYESIAN PARAMETER ESTIMATION

3.1 Bayesian method

Bayesian methods take the parameters as random variables with known prior distribution. Observation of samples converts this to a posterior density, thereby revising opinion about the true values of the parameter. In the Bayesian case we see a typical effect of observing additional samples, which is to sharpen the a posteriori density function. This phenomenon is known as Bayesian learning.

Since we have knowledge of the state of nature or class label for each sample, this learning becomes supervised learning. Sample **x** are assumed to be obtained by selecting a state of nature, w_i with probability $P(w_i)$ and then independently selecting **x** according to the probability law $p(X|w_i)$.

3.2 Bayesian Estimation

Bayesian estimation or Bayesian learning approaches are used in pattern classification problems. In Bayesian learning the parameter θ is a random variable and training data allow us to convert a distribution on this variable into a posterior probability density.

3.2.1 The Class-Conditional densities

The computation of Posterior probabilities $P(w_i|X)$ is the core part of Bayesian Classification. With the help of the Bayes formula, we can compute these probabilities from the prior probabilities $P(w_i)$ and the class-conditional densities $p(X|w_i)$. But a question arises, how can we proceed when these quantities are unknown? The general answer to this question is that we can compute $P(w_i|X)$ using all of the information at our disposal. Part of this information might be prior knowledge, such as knowledge of the functional forms for unknown densities and ranges for the values of unknown parameters. Part of this information might reside in a set of training samples. If we let D denote the set of samples, then we can emphasize the role of the samples by saying that our goal is to compute the posterior probabilities $P(w_i|x, D)$. From these probabilities we can obtain the Bayes Classifier. [2][3]

Given the samples D, Bayes formula then becomes.

$$P(w_i|\mathbf{x}, D) = \frac{p(\mathbf{x}|w_i, D)P(w_i|D)}{\sum_{j=1}^{c} p(\mathbf{x}|w_i, D)P(w_i|D)} \tag{3.1}$$

The above equation suggests that we can use the information provided by the training samples to determine both the class-conditional densities and the prior probabilities.3.1

3.2.2 The Parameter Distribution

Knowing that the desired probability density p(**x**) is unknown, we will assume that the probability density has a known parametric form. The only thing assumed unknown is the value of a parameter vector θ. We express the fact that p(**x**) is unknown but it has a known parametric form by saying that the function $p(\mathbf{x}|\theta)$ is completely known. Any information we might have prior to observing the samples is assumed to be contained in a known prior density $p(\theta)$. Observation of the samples converts this to a posterior density $p(\theta|D)$, which, we hope is sharply peaked about the true value of θ.

Up till now we have converted our problem of learning a probability density function to one of estimating a parameter vector. Now our basic goal is to compute P(**x**|D), which is as close as we can come to obtain the unknown p(x). We do this by integrating the joint density $p(x, \theta|D)$ over θ. That is,

$$p(x|D) = \int p(\mathbf{x}, \theta|D)d\theta, \tag{3.2}$$

Above integration is extends over the entire parameter space. Now we can always write $p(\mathbf{x}, \theta|D)$ as the product $p(\mathbf{x}|\theta, D)p(\theta|D)$. Because the selection of x and that of the training samples in D is done independently, the first factor is merely $p(x|\theta)$. That is, the distribution of x is known completely once we know the value of the parameter vector. Thus 3.2 can be rewritten as

$$p(x|D) = \int p(\mathbf{x}|\theta)p(\theta|D)d\theta, \tag{3.3}$$

The most important equation above links the desired class-conditional density $p(\mathbf{x}|D)$ to the posterior density $p(\theta|D)$ for the unknown parameter vector. If $p(\theta|D)$ peaks very sharply about some value of θ, we obtained $p(\mathbf{x}|D)(\mathbf{x}|\widehat{\theta})$i.e. the results we would ob-

tain by substituting the estimation for the true parameter vector, this results rests on the assumption that $p(\mathbf{x}|\theta)$ is smooth, and that the tails of the integral are not important.

3.3 Bayesian Parameter estimation: GAUSSIAN CASE

Here in this section we use Bayesian estimation techniques to calculate the a posteriori density $p(\theta|D)$ and the desired probability density $p(\mathbf{x}|D)$, for the case where

$$p(\mathbf{x}|\mu) \approx N(\mu, \Sigma)$$

3.3.1 The Univariate Case: $p(\mu|D)$

Consider the case where μ is the only unknown parameter. For simplicity we treat first the univariate case, this is,

$$p(x|\mu) \approx N(\mu, \sigma^2), \tag{3.4}$$

When the only unknown quantity is the mean μ, we assume that whatever prior knowledge we might have about μ can be expressed by known prior density $p(\mu)$. Later we shall make the further assumption that

$$p(\mu) \approx N(\mu_0, \sigma_0^2), \tag{3.5}$$

When both μ_0 and σ_0^2 are known. Roughly speaking, μ_0 represents our best prior guess for μ and σ_0^2 measures our uncertainty about this guess. The assumption that the prior distribution for μ is normal will simplify the subsequent mathematics. However the crucial assumption is not so much that the prior distribution for μ is normal, but that it is know.

After selecting the prior density for μ, we can proceed as follows, suppose that a value is drawn for μ from a population governed by the probability law $p(\mu)$. Once this value is drawn, it becomes the true value of μ and completely determines the density for x. Suppose now that *n* samples $x_1, x_2, .., x_n$ are independently drawn from the resulting population. Letting $D = x_1, x_2, .., x_n$, we use Bayes formula to obtain,

$$p(\mu|D) = \frac{p(D|\mu)p(\mu)}{\int p(D|\mu)p(\mu)}$$

$$= \alpha \prod_{n}^{k=1} p(x_k|\mu)p(\mu), \tag{3.6}$$

Where α is a normalization factor that depends on D but is independent of μ, this equation shows how the observation of a set of training samples affects our ideas about the true value of μ, it relates the prior density $p(\mu)$ to an a posteriori density $p(\mu|D)$. Because $p(x_k|\mu) \approx N(\mu|\sigma^2)$ and $p(\mu) \approx N(\mu_0, \sigma_0^2)$, we have

$$p(\mu|D) = \alpha \prod_{k=1}^{n} \overbrace{\frac{1}{\sqrt{2\pi}\sigma} exp\left[-\frac{1}{2}\left(\frac{x_k - \mu}{\sigma}\right)^2\right]}^{p(x_k|\mu)} \overbrace{\frac{1}{\sqrt{2\pi}\sigma_0} exp\left[-\frac{1}{2}\left(\frac{\mu - \mu_0}{\sigma_0}\right)^2\right]}^{p(\mu)}$$

$$= \alpha^{'} exp\left[-\frac{1}{2}\left(\sum_{k=1}^{n}\left(\frac{\mu - x_k}{\sigma}\right)^2 + \frac{\mu - \mu_0}{\sigma_0}\right)^2\right)$$

$$= \alpha^{''} exp\left[-\frac{1}{2}\left[\left(\frac{n}{\sigma^2} + \frac{1}{\sigma_0^2}\right)\mu^2 - 2\left(\frac{1}{\sigma^2}\sum_{k=1}^{n} x_k + \frac{\mu_0}{\sigma_0^2}\right)\mu\right]\right], \tag{3.7}$$

Where factors that do not depend on μ have been absorbed into the constants $\alpha, \alpha^{'}$ and $\alpha^{''}$. Thus, $p(\mu|D)$ is an exponential function of a quadratic function of μ, i.e. again a normal density. Because this is true for any number of training samples, $p(\mu|D)$ remains normal as the number *n* of samples is increased, and $p(\mu|D)$ is said to be *a reproducing density* and $p(\mu)$ is said to be *a conjugate prior*. If we write $p(\mu|D) \approx N(\mu_n, \sigma_n^2)$, then μ_n and σ_n^2 can be found by equating coefficients in 3.7 with corresponding coefficients in the generic Gaussian of the form

$$p(\mu|D) = \frac{1}{\sqrt{2\pi}\sigma_n} exp\left[-\frac{1}{2}\left(\frac{\mu - \mu_n}{\sigma_n}\right)^2\right] \tag{3.8}$$

Identifying coefficient in this way yields

$$\frac{1}{\sigma_n^2} = \frac{n}{\sigma^2} + \frac{1}{\sigma_0^2} \tag{3.9}$$

and

$$\frac{\mu_n}{\sigma_n^2} = \frac{n}{\sigma^2}\hat{\mu_n}\frac{\mu_0}{\sigma_0^2}, \tag{3.10}$$

where $\hat{\mu_n}$ is the sample mean

$$\hat{\mu_n} = \frac{1}{n}\sum_{k=1}^{n} x_k, \tag{3.11}$$

We solve explicitly for μ_n and σ_n^2 and obtain

$$\mu_n = \left(\frac{n\sigma_0^2}{n\sigma_0^2 + \sigma^2}\right)\hat{\mu}_n + \frac{\sigma^2}{n\sigma_0^2 + \sigma^2}\mu_0 \tag{3.12}$$

and

$$\sigma_n^2 = \frac{\sigma_0^2\sigma}{n\sigma_0^2 + \sigma^2}. \tag{3.13}$$

This equation shows how prior information is combined with the empirical information in the samples to obtain the a *posteriori* density $p(\mu|D)$.[2] Roughly speaking, μ_n represents our best guess for μ after observing *n* samples and σ_n^2 measures our uncertainty about this guess. Because σ_n^2 decreases monotonically proportional to σ^2/n as *n* approaches infinity—each additional observation decreases our uncertainty about the true value of μ. As *n* increases, $p(\mu|D)$ becomes more and more sharply peaked, approaching a Dirac delta function as n approaches infinity. This behavior is commonly known as *Bayesian Learning*.

3.3.2 The Univariate Case: $p(x|D)$

After obtaining $p(\mu|D)$ the a priori density for the mean, we would like to calculate "the class conditional" density for $p(x|D)$ from 3.3,3.4 and 3.8.

$$p(x|D) = (x|\mu)p(\mu|D)d\mu$$
$$= \int \frac{1}{\sqrt{2\pi}\sigma} exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right] \frac{1}{\sqrt{2\pi}\sigma_n} exp\left[-\frac{1}{2}\left(\frac{\mu-\mu_n}{\sigma_n}\right)^2\right] d\mu$$

$$= \frac{1}{2\pi\sigma\sigma_n} exp\left[-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2+\sigma_n^2}\right] f(\sigma,\sigma_n), \tag{3.14}$$

where

$$f(\sigma,\sigma_n) = \left[-\frac{1}{2}\frac{\sigma^2+\sigma_n^2}{\sigma^2\sigma_n^2}\left(\mu - \frac{\sigma_n^2 x + \sigma^2\mu_n}{\sigma^2+\sigma_n^2}\right)^2\right] d\mu.$$

The above equation is a function of x, $p(x|D)$ is proportional to $exp[-(1/2)(x-\mu_n)^2/(\sigma^2+\sigma_n^2)]$, and hence$p(x|D)$ is normally distributed with mean μ_n and variance $\sigma^2+\sigma_n^2$.

$$p(x|D) \approx N(\mu_n, \sigma^2+\sigma_n^2).$$

3.3.3 The multivariate Case

In the multivariate case, if σ is known but not μ we obtain a straight generalization of the univariate case. So in this case we will assume that,

$$p(x|\mu) \approx N(\mu,\sigma) and p(\mu) \approx N(\mu_0,\sigma_0), \tag{3.15}$$

Where σ, σ_0, and μ are assumed to be known. When the set D consist of observations of *n* independent samples $x_1, \ldots, x_n$, we can use Bayes formula to obtain $p(\mu|D)$

$$p(\mu|D) = \alpha \prod_{k=1}^{n} p(x_k|\mu)p(\mu) \tag{3.16}$$

$$= \alpha^{'} exp\left[-\frac{1}{2}\left(\mu^t(n\Sigma^{-1} + \Sigma_0^{-1})\mu - 2\mu^t\left(\Sigma^{-1}\sum_{k=1}^{n} x_k + \Sigma_0^{-1}\mu_0\right)\right)\right],$$

Which has the form

$$p(\mu|D) = \alpha^{''} exp\left[-\frac{1}{2}(\mu - \mu_n)^t\Sigma_n^{-1}(\mu - \mu_n)\right] \tag{3.17}$$

Hence $p(\mu|D) \approx N(\mu_n, \Sigma_n)$, and we have obtained a reproducing density. Equating coefficents, we obtain the analogs of 3.12 and 3.13

$$\Sigma_n^{-1} = n\Sigma^{-1} + \Sigma_0^{-1} \tag{3.18}$$

and

$$\Sigma_n^{-1}\mu_n = n\Sigma^{-1}\hat{\mu_n} + \Sigma_0^{-1}\mu_0 \tag{3.19}$$

where $\hat{\mu_n}$ is the simple mean,

$$\hat{\mu_n} = \frac{1}{n}\sum_{k=1}^{n} x_k \tag{3.20}$$

μ_n and Σ_n can be solved by using knowledge of matrix identity.

$$(A^{-1} + B^{-1})^{-1} = A(A + B)^{-1} = B(A + B)^{-1}A \tag{3.21}$$

The above expression is valid for any pair of nonsingular,[2] (d$\times$ d) matrices A and B.

After a little manipulation, we obtain the final results:

$$\mu_n = \Sigma_0 \left(\Sigma_0 + \frac{1}{n}\Sigma \right)^{-1} \hat{\mu_n} + \frac{1}{n}\Sigma \left(\Sigma_0 + \frac{1}{n}\Sigma \right)^{-1} \mu_0$$

$$\Sigma_n = \Sigma_0 \left(\Sigma_0 + \frac{1}{n}\Sigma \right)^{-1} \frac{1}{n}\Sigma$$

From μ_n and Σ_n, we can observe that the Mean μ_n in this case is also the linear combination of $\hat{\mu_n}$ and μ_0, same as in the univariate case.

4 THE SUPPORT VECTOR MACHINE

In the previous chapter we assumed that we know probability densities, and described the way, we will use training samples to estimate the values of the parameters of those distributions. In this chapter, we shall instead assume that we know the proper forms for the discriminant functions, and use the samples to estimate the values of parameters of the classifier. We shall examine various procedures for determining discriminant functions, some of which are statistical and some of which are not.

4.1 Linear Discriminant Functions and decision surfaces

A discriminant function that is a linear combination of the components of x can be written as

$$g(x) = w^t x + w_0$$

Where w is the weights vectors and w0 are threshold weights[4].

In the simplest case there are only two categories.

The Two Category Case

For a discriminant function of the above form a two category classifier implements the following decision rule: Decide

$$w_1 \ if g(x) > 0$$

$$w_2 \ if g(x) < 0$$

Thus, x is assigned to w_1 if the inner product $w^{'}x$ exceeds the threshold $-w_0$ and to w_2 otherwise. If g(x) =0, x can usually be assigned to either class, but in this chapter we shall leave the assignment undefined. The figure below shows a typical implementation of a clear example of the general structure of a pattern recognition system. The equation

$g(x) = 0$ defines the decision surface that separates points assigned to w_1 from points assigned to w_2. When $g(x)$ is linear, this decision surface is a hyperplane. If x_1 and x_2 are both on the decision surface, then

$$w^t x_1 + w_0 = w^t x_2 + w_0$$

or

$$w^t(x_1 - x_2) = 0$$

and this shows that w is normal to any vector lying in the hyperplane, In general the hyperplane H divides the feature space into two half-spaces; decision region $R1$ for w_1 and region R2 for w_2. Because $g(x) > 0$ if x is in R1, it follows that the normal vector w points into R1. It is sometimes said that any x in R1 is on the positive side of H, and any x in R2 is on the negative side.

The discriminant function g(x) gives an algebraic measure of the distance from x to the hyperplane. Perhaps the easiest way to see this is to express x as,

$$x = x_p + r\frac{w}{\|w\|},$$

Where x_p is the normal projection of x onto H, and r is the desired algebraic distance. positive if x is on the positive side and negative if x is on the negative side. Then, because $g(x_p) = 0$,

$$g(x) = w^t x + w^0 = r\,\|w\|$$

or

$$r = \frac{g(x)}{\|w\|}$$

To summarize, a linear discriminant function divides the feature space by hyperplanes as a decision surface, the orientation of the surface is determined by the normal vector w, and the location of the surface is determined by the bias w_0. The discriminant function $g(x)$ is proportional to the signed distance from x to the hyperplane with $g(x) > 0$ when x is on the positive side and $g(x) < 0$ when x is on the negative side.

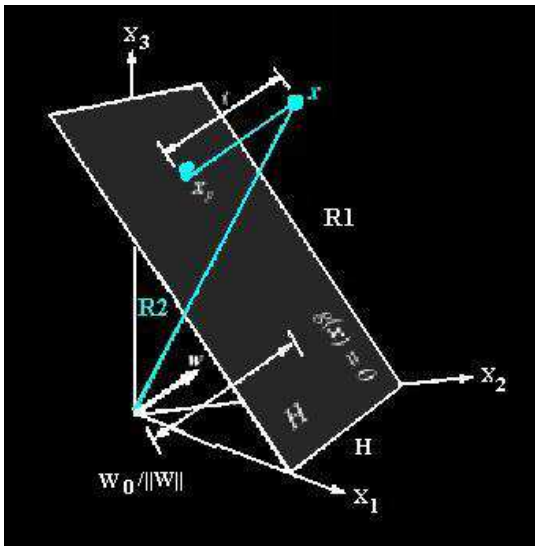


Figure 4.1: The linear decision boundary H, where g(x)=$w^t x + w_0$, separates the feature space into two half-spaces R_1 where g(x)>0) and R_2 where g(x)<0.

4.2 Support Vector Machine

The Support Vector Machine is motivated by many trained linear machines with margins, but relies on pre-processing of the data to represent patterns in a high dimensional space, typically much higher than the original feature space. With an appropriate nonlinear mapping $\varphi(.)$ to a sufficiently high dimension, data from two categories can always be separated by a hyperplane[5]and [6].

Here we assume each pattern x_k has been transformed to $y_k = \varphi(x_k)$. We shall return to the choice of $\varphi(.)$. For each of the n patterns, $k = 1, 2, \ldots, n$, we let $z_k = \pm 1$, according to whether pattern k is in w_1 or w_2. A linear discriminant in an augmented y space is

$$g(y) = a^t y. \tag{4.1}$$

where both the weight vector and the transformed pattern vector are augmented by a=w_0 and y_0=1, respectively.

$$z_k g(y_k) \geq 1, \;\; k = 1, \ldots, n \tag{4.2}$$

The margin is any positive distance from the decision hyperplane. The goal in training a Support Vector Machine is to find the separating hyperplane with the largest margin; we expect that the larger the margin, the better generalization of the classifier. As illustrated in Figure 4.1, the distance from any hyperplane to a (transformed) pattern y is $g(y)/\left\|a\right\|$,and

assuming that a positive margin b exists implies

$$\frac{z_k g(y_k)}{\|a\|},\ k = 1, \ldots, n; \tag{4.3}$$

The goal is to find the weight vector *a* that maximizes *b*. Of course the solution vector can be scaled arbitrarily and still preserve the hyperplane, and thus to ensure uniqueness we impose the constraint $b\,\|a\| = 1$; that is we demand the solution to 4.1 and 4.2 also minimize $\|a\|^2$.

The support vectors are the transformed training patterns for which 4.2 represents an equality-that is the support vectors are equally close to the hyperplane. The support vectors are the training samples that define the optimal separating hyperplane and are the most difficult patterns to classify. Informally speaking they are the patterns most informative for the classification task.

4.2.1 SVM algorithm for the linearly separable case

- Let $X = x_n, y_n$ be a set of feature vectors
  where $n = 1 : N$, x_n is the input vector and y_n is the target vector,
  such that $x_n \in w_1(w_2)$ then $y_n = 1(-1)$
- Thus, a linear discriminant function is defined by the weight vector w and the threshold w_0.
- The goal of the Support Vector Machine is to find the hyperplane with the largest margin to both classes.
- The separating hyperplane must fulfill

  $$y(w_T x + w_0) - 1 > 0, \tag{4.4}$$

  for every k.
- The distance is defined by $2/\|w\|$
- Thus our goal is to determine w and w_0 in such a way that the marginal is maximal, in other words, $\|w\|^2$is minimal and the conditions in above equation 4.4 are fulfilled.

- We solve the optimization problem by using Lagrange multipliers see Appendix B. By defining the Lagrange multiplier λ_k for each of the constraints, we can establish the Lagrange equation as follows.

$$L = \frac{1}{2}\|w\|^2 - \sum_{k=1}^{N} \lambda_k (y_k(w^T x + w_0) - 1), eq4.5 \tag{4.5}$$

Now we can find the minimum by taking gradient of L to be equal to zero. On the other hand it can be also shown to maximize the dual problem to solve λ.

$$Q(\lambda) = \sum_{k=1}^{N} \lambda_k - \frac{1}{2} \sum_{k=1}^{N} \sum_{j=1}^{N} \lambda_k \lambda_j y_k y_j x_k^T x_j, eq4.6 \tag{4.6}$$

such that

$$\sum_{k=1}^{N} \lambda_k y_k = 0, \lambda_k \geq 0 \forall k eq4.7 \tag{4.7}$$

- The solution can be found e.g. by using quadratic programming. After a solution λ to the dual problem has been found, optimal w and w_0 can be calculated from

$$w = \sum_{k=1}^{N} \lambda_k y_k x_k, eq4.8 \tag{4.8}$$

$$w_0 = 1 - x_k^T w, \;\; x_k \in w_1, \lambda_k > 0$$

- The only samples whose Lagrange multiplier $\lambda_k > 0$, affect the discriminant, are called *the support vectors*.

- Decision rule: Choose $w_1 (w_2)$ if $w^T x + w_0 > 0 (> 0)$

4.2.2 SVM algorithm for the nonlinearly separable case

- A nonlinear generalization can be done by replacing x by $\Phi(x)$.
  Where $\Phi : R^l \rightarrow R^m$ is a nonlinear mapping.[2]

[2] usually m»l

- We can find a *kernel function* $K(.,.)$, such that $K(x_i, x_j) = \Phi(x_i^T \Phi(x_j))$, and we do not have to use Φ directly.
- For example, the Gaussian kernel function is defined below.

$$K(x_i, x_j) = e^{-\|x_i - x_j\|^2 / 2\sigma^2}$$

  or the Polynomial kernel

$$K(x_i, x_j) = (x_i^T x_j + 1)^p$$

  where p is the degree of the polynomial kernel.
- Thus, if the values of the kernel function can be calculated efficiently, the nonlinear SVM will train roughly as fast as the linear SVM.
- For example: XOR problems(i.e Nonlinearly separable examples) can be solved using nonlinear SVM method.

4.3 Limitations

Disadvantages of SVM

- Although SVM is relatively sparse, the number of Support Vectors grows linearly with the size of the training set.
- Prediction is not probabilistic, i.e. The SVM classification is based on a hard binary decision.
- It is necessary to find error margin parameters, whereas in regression it is an epsilon term.
- The Kernel function must be a continuous, symmetric kernel of a positive integral operator.

These limitations are removed by"The Relevance vector machine (RVM)". RVM is a Bayesian treatment of SVM.

5 THE RELEVANCE VECTOR MACHINE

The present chapter introduces a general Bayesian framework to achieve a sparse solution to regression and classification tasks utilising models linear in the parameters. Tipping's approach, known as the "Relevance vector machine (RVM)", is a model of identical functional form to the "Support Vector Machine (SVM)". The Relevance vector machine was introduced by Tipping [7]. By exploiting a probabilistic Bayesian learning framework we can derive an accurate prediction model which utilises dramatically fewer basic function [7].

5.1 Sparse Bayesian Learning

Some drawbacks mentioned at the end of previous chapter forces us to think about other probabilistic models. The sparse Bayesian regression model and inference procedure is described in the following two sections.

5.1.1 Model Specification

Given input data set and target set,

$$\{x_n, t_n\},$$

we follow the standard probabilistic formulation and assume that the targets are samples from the model with additive noise:

$$t_n = y(x_n; w) + \epsilon_n$$

Where ϵ_n are independent samples from some noise process which is further assumed to be zero mean Gaussian distribution with variance σ^2.

Thus $p(t_n|x) = N(t_n|y(x_n), \sigma^2)$, where the notation specifies a Gaussian distribution over t_n with mean $y(x_n)$ and variance σ^2. The function $y(x)$ is defined as

$$y(x; w) = \sum_{k=1}^{N} w_i K(x, x_i) + w_0;$$

for SVM, to identify basis functions with kernel $K(x, x_i)$ as parameterised by the training vectors: $\phi_i = K(x, x_i)$. Making the assumption that y_n are independent, the likelihood of the complete data set can be written as

$$p(t|w, \sigma^2) = (2\pi\sigma^2)^{-N/2} exp - \frac{1}{2\sigma^2} \|t - \Phi w\|^2, \tag{5.1}$$

where $t = t_1, t_2, \ldots, t_N{}^T$, $w = (w_0, \ldots, w_N)^T$;
Φ is the $N \times (N + 1)$ 'design' matrix.
$\Phi = [\Phi(x_1), \Phi(x_2), \ldots, \Phi(x_n)]^T$,
$\Phi(x_n) = [1, K(x_n, x_1), K(x_n, x_2), \ldots, K(x_n, x_N)]^T$.

After defining maximum likelihood, we would like to estimate weight w and variance σ^2 from the above equation 5.1. This might lead us to severe over-fitting. In order to avoid this we will impose some additional constraints on the parameter, for example, through addition of a 'complexity' penalty term to the likelihood or error function.

We encode a preference for less complex functions by making the popular choice of a zero-mean Gaussian prior distribution over w.

$$p(w|\alpha) = \prod_{i=0}^{N} N(w_i|0, \alpha_i^{-1}), \tag{5.2}$$

with α a vector of $N + 1$ hyperperameters.

$$p(\alpha) = \prod_{i=0}^{N} Gamma(\alpha_i|a, b),$$

$$p(\beta) = Gamma(\beta|c, d),$$

with $\beta = \sigma^{-2}$ and where

$$Gamma(\alpha|a, b) = \Gamma(a)^{-1} b^a \alpha^{a-1} e^{-b\alpha}, \tag{5.3}$$

with $\Gamma(a) = \int_0^\infty t^{a-1} e^{-t} dt$, the 'gamma function'.In order to make priors non-informative we can fix their parameters to small values: e.g.a=b=c=d=10^{-4}.

5.1.2 Inference

In the above section we have defined the prior. We can now proceed with Bayesian inference by computing the posterior over all unknowns from Bayes's rule[8].

$$p(w, \alpha, \sigma^2|t) = \frac{p(t|w, \alpha, \sigma^2)p(w, \alpha, \sigma^2)}{p(t)}, \tag{5.4}$$

Now for given set of new point x_* we can make predictions for the corresponding target t_*, in terms of the predictive distribution:

$$p(t_*|t) = \int p(t_*|w, \alpha, \sigma^2)p(w, \alpha, \sigma^2|t)dw\ d\alpha\ d\sigma^2. \tag{5.5}$$

But we cannot compute the posterior $p(w, \alpha, \sigma^2|y)$ in 5.4 above directly, hence we cannot calculate the normalising integral on the right hand side,

$$p(t) = \int p(t|w, \alpha, \sigma^2)p(w, \alpha, \sigma^2)dw\ d\alpha\ d\sigma^2.$$

However, we can decompose the posterior as:

$$p(w, \alpha, \sigma^2|t) = p(w|t, \alpha, \sigma^2)p(\alpha, \sigma^2|t) \tag{5.6}$$

since we can compute the posterior distribution analytically over the weights since its normalising integral, $p(t|\alpha, \sigma^2) = \int p(t|w, \sigma^2)p(w|\alpha)dw$ is nothing but a convolution of Gaussians.
The posterior distribution over the weights is thus given by:

$$p(w|t, \alpha, \sigma^2) = \frac{p(t|w, \sigma^2)p(w|\alpha)}{p(t|w, \sigma^2)} \tag{5.7}$$

$$= (2\pi)^{-(N+1)/2}|\Sigma|^{-1/2}exp\left\{-\frac{1}{2}(w-\mu)^T\Sigma^{-1}(w-\mu)\right\}, \tag{5.8}$$

Here the posterior covariance and mean are given below respectively.

$$\Sigma = (\sigma^{-2}\Phi^T\Phi + A)^{-1}, \tag{5.9}$$

$$\mu = \sigma^2\Sigma\Phi^T t, \tag{5.10}$$

with A=diag$\sigma_0, \sigma_1, \ldots, \sigma_N$

Approximating the second term in right-hand side of 5.6, the hyperparameter posterior $p(\alpha, \sigma^2|t)$, by a delta-function at its mode, i.e. at its most probable values $\alpha_{MP}, \sigma^2_{MP}$. By doing so the basis that point estimate is representative of the posterior in the sense that functions generated utilising the posterior mode values are near-identical to those obtained by sampling from the full posterior distribution. It is important to realise that this does not necessitate that the entire mass of the posterior be accurately approximated by the delta-function. For predictive purpose, rather than requiring $p(\alpha, \sigma^2|t) \approx \delta(\alpha_{MP}, \sigma^2_{MP})$.

$$\int p(t_*|\alpha, \sigma^2)\delta(\alpha_{MP}, \sigma^2_{MP})d\alpha d\sigma^2 \approx \int p(t_*|\alpha, \sigma^2)p(\alpha, \sigma^2|t)d\alpha d\sigma^2, \tag{5.11}$$

Our desire is to get good optimization.

Relevance Vector 'Learning' thus becomes the search for the hyperparameter posterior mode, i.e. the maximisation of $p(\alpha, \sigma^2|t) \propto p(t|\alpha, \sigma^2)p(\alpha)p(\sigma^2)$ with respect to α and β. In case of uniform hyperpriors we need to get only maximum of $p(t|\alpha, \sigma^2)$, which is computable and given by:

$$p(t|\alpha, \sigma^2) = \int p(t|w, \sigma^2)p(w|\alpha)dw$$

$$= (2\pi)^{-N/2} \left|\sigma^2 I + \Phi A^{-1}\Phi^T\right|^{-1/2} exp\left\{-\frac{1}{2}t^T(\sigma^2 I + \Phi A^{-1}\Phi^T)^{-1}t\right\}, \tag{5.12}$$

In Bayesian Models this quantity is known as the *marginal likelihood*, and its maximisation is known as the t*ype-II maximum likelihood* method. The marginal likelihood is also referred as the "evidence for the hyperparameters" and its maximisation is known as "evidence procedure".

5.1.3 Optimizing the hyperparameters

Estimation of values α and σ^2 is carried out using an iterative re-estimation approach. Details about hyperparameter estimation, including alternative expectation-maximisation-based re-estimates, are given in Appendix C. For α, differentiation of 5.12, equating to zero and rearranging, following the approach of MacKay gives:

$$\alpha_i^{new} = \frac{\gamma_i}{\mu_i^2}, \tag{5.13}$$

where μ_i is the *i*-th posterior mean weight from 5.10 and we have defined the quantities γ_i by:

$$\gamma_i \equiv 1 - \alpha_i N_{ij}$$

With N_{ij} the *i*-th diagonal element of the posterior weight covariance from 5.9 computed with current α and σ^2 values. Each $\gamma_i \in [0, 1]$ can be interpreted as a measure of how 'well-determined' its corresponding parameter weight w_i is by the data. For α_i large wehere w_i is highly constrained by the prior, $N_{ii} \approx \alpha_i^{-1}$ and it follows that $\gamma_i \approx 0$. Conversely, when α_i is small and w_i fits the data, $\gamma_i \approx 1$.

For the noise variance σ^2, differentiation leads to the re-estimate:

$$(\sigma^2)^{new} = \frac{\|t - \Phi\mu\|^2}{N - \Sigma_i \gamma_i}, \tag{5.14}$$

Where $'N'$ in the denominator refers to the number of data examples and not to the number of basis functions. Thus the learning algorithm proceeds by repeated application of the above two 5.13 and 5.14, concurrent with updated posterior statistics Σ and μ from 5.9 and 5.10, until we have obtained some suitable convergence. Whereas in practical case during re-estimating we generally find that many of the α_i tend to infinity. From 5.8 this implies $p(w_i|t, \alpha, \sigma^2)$ becomes highly peaked at zero. i.e we are a *posteriori* 'certain' that those w_i are zero. The corresponding basis functions can thus be 'pruned' and sparsity is realised.

5.1.4 Making predictions

Predictions based on the posterior distribution over the weights were made at convergence of the hyperparameter estimation procedure. Conditionally on the maximising values α_{MP} and σ^2_{MP}. For the new datum x_* using 5.8 we can compute the predictive distribution from 5.5

$$p(t_*|t, \alpha_{MP}, \sigma^2_{MP}) = \int p(t_*|w, \sigma^2_{MP}) p(w|t, \alpha_{MP}, \sigma^2_{MP}) dw \tag{5.15}$$

As both terms in the integrand are Gaussian, this computed and gives

$$p(t_*|t, \alpha_{MP}, \sigma^2_{MP}) = N(t_*|y_*, \sigma^2)$$

with

$$y_* = \mu^T \phi(x_*), \tag{5.16}$$

$$\sigma^2_* = \sigma^2_{MP} + \phi(x_*)^T \Sigma \phi(x_*), \tag{5.17}$$

So intuitively the predictive mean is $y(x_*, \mu)$, or in other words the basis function weighted by the posterior weights, many of which will typically be zero. The predictive variance comprises the sum of two variance components: the estimated noise on the data and other due to the uncertainty in the prediction of the weights. In the practice, then, we may thus choose to set our parameter **w** to fixed values μ for the purpose of point prediciton and retain Σ if required for computation of error bars.

5.2 Sparse Bayesian Classification

As for regression in the previous section, Relevance vector classification follows an essentially identical framework. We simply adapt the target conditional distribution and the link functional to account for the change in the target quantities. As a consequence, we must introduce an additional approximation step in the algorithm.

In any two-class classification, it is important to predict the posterior probability of membership of one of the classes given the input x. By following statistical convention and generalise the linear model by applying the logistic sigmoid link function $\sigma(y) = 1/(1 - e^{-y})$ to $y(x)$ and adopting the Bernoulli distribution for $P(t|x)$, we can write the likelihood function as follows

$$p(t|w) = \prod_{n=1}^{N} \sigma\left\{y(x_n; w)\right\}^{t_n} \left[1 - \sigma\left\{y(x_n; w)\right\}\right]^{1-t_n} \tag{5.18}$$

Note:- We cannot integrate out the weights analytically, unlike the regression case and so are denied closed-form expression for either the weights posterior $p(w|t;\alpha)$ or the marginal likelihood $P(t|\alpha)$. We thus choose the utilise the following approximation procedure as used by MacKay (1992b), which is based on Laplace's Method.

1. At present for fixed values of α, the 'most probable' weights we got are w_{MP}, giving the location of the mode of the posterior distribution. Since $p(w|t,\alpha)\alpha P(t|w)p(w|\alpha)$ is equivalent to finding the maximum over w, of

$$log\left\{P(t|w)p(w|\alpha)\right\} = \sum_{n=1}^{N}[t_n log y_n + (1-t_n)log(1-y_n)] - \frac{1}{2}w^T Aw \tag{5.19}$$

   with $y_n = \sigma\left\{y(w_n; w)\right\}$. This is a standard procedure, since 5.19 is a penalised logistic log-likelihood function, and necessitates iterative maximisation. Second-order Newton methods may be effectively applied, since the Hessian of 5.19, required next in step-2, is exactly computed. We adopted the efficient 'iteratively-reweighted least-squares' algorithm to find w_{MP}.

2. Laplace's Method is simply a quadratic approximation to the log-posterior around its mode. The quantity 5.19 is differential twice to give:

$$\nabla_w \nabla_w logp(w|t,\alpha)|_{w_{MP}} = -(\Phi^T B\Phi + A) \tag{5.20}$$

   Where B=diag$\beta_1, \beta_2, \cdots, \beta_n$ is a diagonal matrix with $\beta_n = \sigma\left\{y(x_n)\right\}\left[1-\sigma\left\{y(x_n)\right\}\right]$. This is then negated and inverted to give the covariance Σ for a Gaussian approximation to the posterior over weights centred at w_{MP}.

3. Using the statistic Σ and w_{MP} of the Gaussian approximation, the hyperparameters α are updated using 5.13 in identiacl fashion to the regression case.

At the mode of $p(w|t,\alpha)$, using 5.20 and the fact that $\nabla_w logp(w|t,\alpha)|_{W_{MP}} = 0$, we can write

$$\Sigma = (\Phi^T B\Phi + A)^{-1}, \tag{5.21}$$

$$w_{MP} = \Sigma \Phi^T B t \tag{5.22}$$

The above 5.21 and 5.22 are equivalent to the solution to a 'generalised least square' problem when compared to 5.9 and 5.10

Stepwise presentation of the RVM algorithm

- Standardized variables X≈N(0,1)
- target vector are the classes t_n,$t_n \in \{0, 1\}$
- Classification model of Tipping [7]:$y(x_n; w) = w' x_n$ applied to the logistic sigmoid link function (Appendix A): $\sigma(y) = 1/(1 + e^{-y})$
- Likelihood is Bernoulli distribution:

$$P(t|w) = \prod_{n=1}^{N} \sigma((y(x_n; w))^{t_n} (1 - \sigma(y(x_n; w)))^{1-tn}$$

- Find the maximum over w of $p(w|t, \alpha) \approx P(t|w)p(w|\alpha)$:

$$logP(t|w)p(w|\alpha) = \sum_{n=1}^{N} [t_n \log y_n + (1 - t_n) \log(1 - y_n)] - \frac{1}{2} w^T A w$$

  where $y_n = \sigma(y(x_n; w))$ is the predicted class of the plot n.

- Likelihood $p(w|\alpha) = N(w|0, 1/\alpha)$ makes the model sparse, forcing part of the weights w_m to zero ($\alpha_m \to \infty \Rightarrow w_m \to 0$).

6 FEATURE EXTRACTION

Image processing is a kind of information processing in which the input is an image, but output need not necessarily be an image but it can be in any needed form[9]. A feature extraction technique describes the best way of getting information out of an image for desired analysis.

The following are the fundamental steps in a Machine Vision system and these steps combined to make a system. It starts from image capture and proceeds till classification. Classification itself comprises, recognition of forestland and wetland from given radarsat and an airborne image.

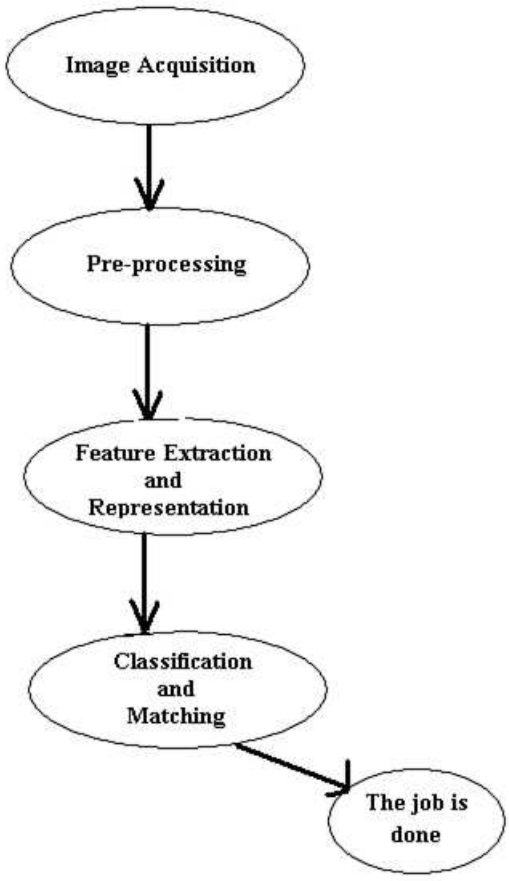


Figure 6.1: An image processing system

- *Image acquisition* is the first process of an image processing system. It starts from an image that is already in digital form. For e.g. an airborne image, it is an image captured from a lower altitude than to the satellited based Radarsat image.

- *Pre-processing* is generally applied after the image acquisition stage and involves functions, such as scaling and enhancement. It may involve changing the size of a digital image. Image enhancement is among the simplest and most appealing areas of digital image processing. Basically the idea behind enhancement techniques is to bring out detail that is obscured or simply to highlight certain features of interest in an image.

- *Representation and description*, The first decision that must be made is whether the data should be represented as a boundary or as a complete region. Boundary representation is appropriate when the focus is on external shape characteristics, such as a corner or an edge. Regional representation is appropriate when the focus is on internal properties, such as texture or area. But choosing a representation is only part of the solution for transforming raw data into a form suitable for subsequent computer processing. A method must also be specified for describing the data so that features of interest are highlighted. Description is also known as *feature selection* and it deals with extracting attributes that result in some quantitative information of interest or are basic for differentiating one class of objects from another.

- *Classification* is the last stage in entire process, it assigns a label to a target on an image based on the descriptors of that target.

There are many different image processing operations which we can use to do transformation, enhancement, restoration, encoding, segmentation, representation and description on an image. These can be combined with algorithms to detect and isolate various desired portions or shapes (features) of a digitized image.

6.1 Feature extraction from an airborne image

An airborne image is an image captured from a low altitude compared to the altitudes of satellites. The aircraft used are helicopters, planes or unmanned aircarft, but not satellites. The image is being processed using basic image processing operation in order to remove noise. Feature selection and extraction is the next step after pre-processing. It plays an important role in both pattern recognition and image processing; it is a special form of

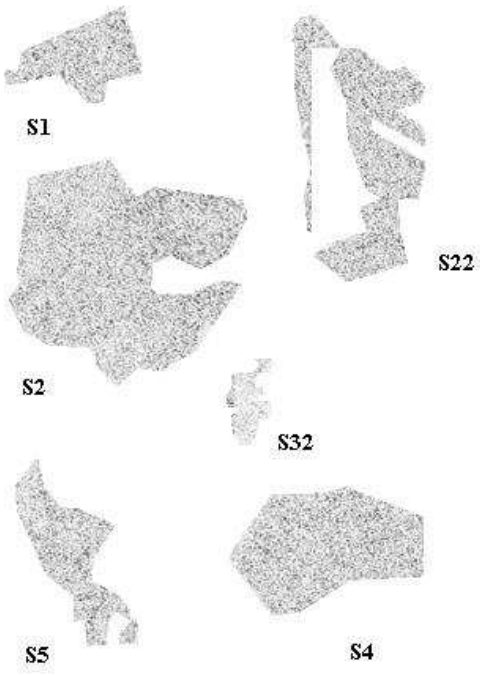


Figure 6.2: The figure shows labeled sample plots taken from an airbone optical image

dimensionality reduction. It involves the amount of resources required to describe a large set of data accurately in a simple form, but it has a lot of complexity involved in it, like the number of variables needed. On the other hand the analysis of a large number of variables requires a large amount of memory and computation power. We may also face a classification algorithm which overfits the training samples and generalizes poorly to new samples. To answer and solve these problem is the main task of feature extraction, while still describing the data with sufficient accuracy[12],[13],[10]and[11]. The classification results can be improved for example by using the following techniques:

- Principal component analysis
- Kernel PCA
- Isomap

In our case we focus on one of the simplest features of an image which is the histogram. We use histogram values as features in both classification techniques that we shall test. We begin by applying appropriate image processing operations on an image. After reading an image we will proceed to conversion of an RGB image to greyscale, then after

doing smoothing by taking convolution, we use different masks like average, prewitt or Gaussian. Using these filters for noise reduction smoothens the image.

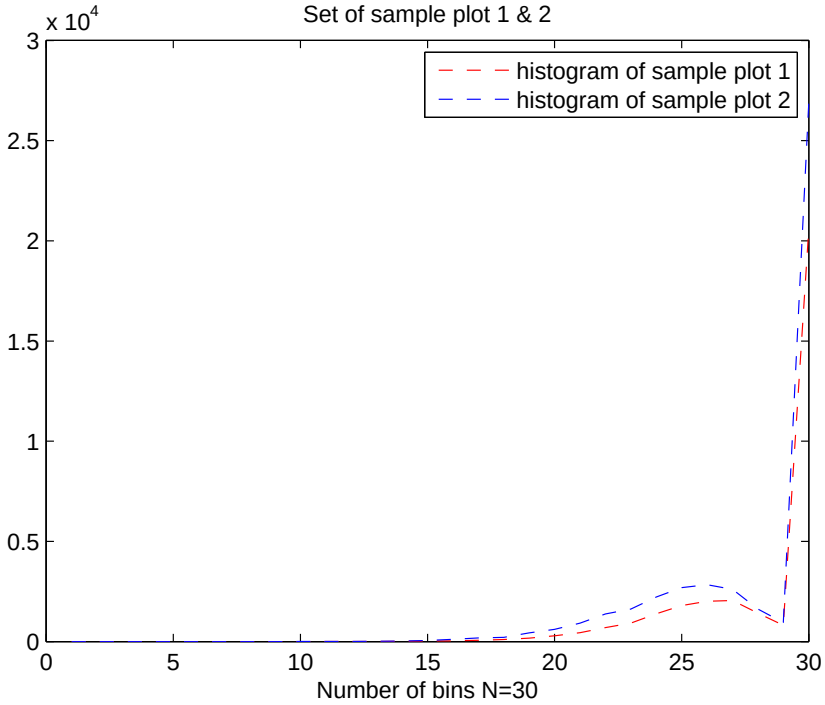


Figure 6.3: The figure shows the histogram of each sample plot 1 and 2

The figure 6.3 represents the greyscale histogram of the sample plots 1 and 2. In a similar manner the histograms from other sample plots are presented in Appendix D.

6.2 Feature extraction from Radarsat image

Microwave based Radarsat images are the type of imagery that can be obtained from satellite. The data provides height and intensity measurements only, represented in the form of a large matrix. Each column represents measurements from the corresponding sample plot. A detailed description of the sample plots is given in chapter 7.

Feature extraction for Radarsat data was done by using methods from statistical analysis. We have 87 plots. Each of them containts Radar measurements of 2 individual bands divided by the measurements for each of the points by the sum of all measurements of the same point. We quantized the measurements to seven histogram bins and the percentile parts within these limits are the features defined in an algorithm. The percentile points in this case are the percentages of the ordered standard measurements, compared to the

total sum of the standardized measurements. The percentile points used in this case are 20%, 50% and 80%. The average mean and variance, along with percentual data, is also a feature.

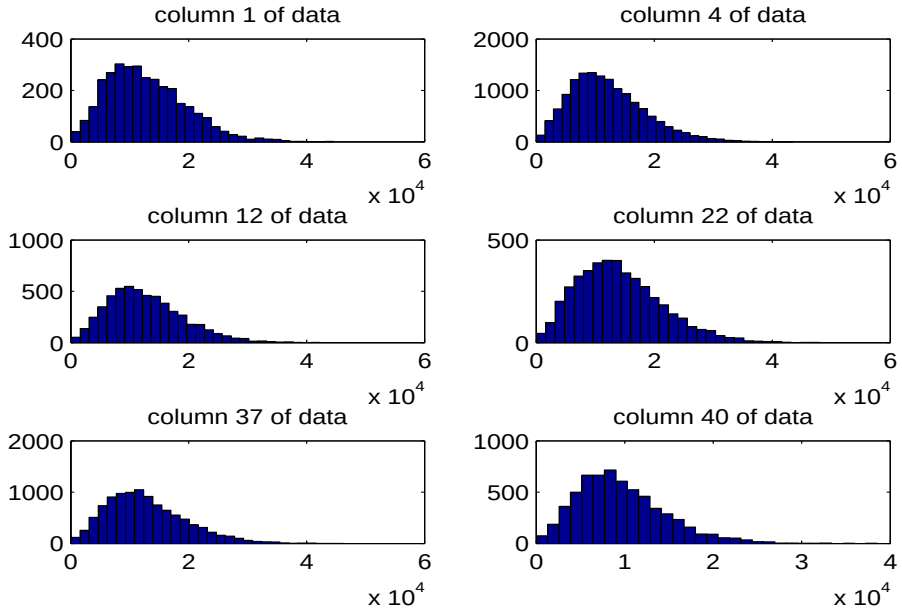


Figure 6.4: Sample of forestland

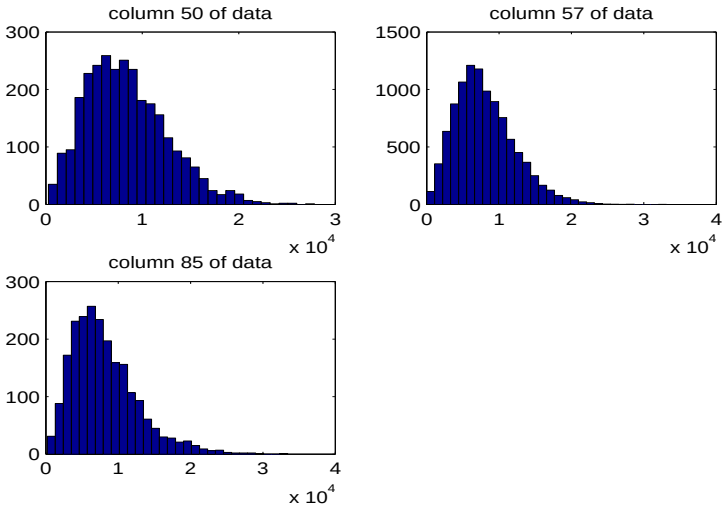


Figure 6.5: Sample of wetland

7 ECOLOGICAL CLASSIFICATION OF FOREST LAND BASED ON SATELLITE AND AIRBORNE IMAGES

7.1 About Radarsat measurements

The advanced Earth observation satellite called "RADARSAT", was developed by the Canadian Space Agency, building on history of achievements in remote sensing and space technologies. The satellite was launched in November, 1995 and is designed for a five-year lifetime. RADARSAT is equipped with a C-band, horizontally polarized Synthetic Aperture Radar (SAR) system. The satellite's orbit is repeated every 24 days and provides complete global coverage with the flexibility to support specific requirements. Canada and the world now have access to the first radar satellite system capable of large scale production and timely delivery of data, it meets the need of commercial, government and scientific programs and provides a new source of reliable and cost-efficient data for environmental and resource professionals worldwide. The users are getting RADARSAT data products in well structured form with a choice of resolutions and formats.

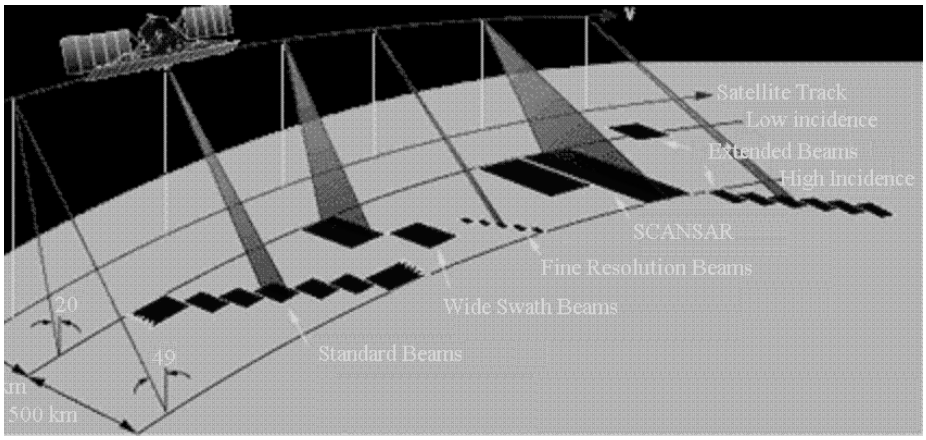


Figure 7.1: Radarsat SAR Operating Modes

In our case, we use the Radarsat-data for forest characteristic figure estimation. Radarsat data suffers from a lack of a model that fits the measurements to the target vector of forest parameters. The Radarsat-data must be processed in a statistical manner to produce plot specific[3] data. However, the connection between Radarsat-data based height histogram

[3]Description about the plot specific is given in the next section on variables.

and e.g stem number is not clear. So, many variables are extracted from the Radarsat-data, and in conventional modeling, the best variables to describe the needed target vector values are estimated with cross-validation. In sparse Bayesian regression the selection of best variables is performed using a numerical algorithm.

7.2 Variables based on Radarsat image

The Variables defined in this section are of satellite image Radarsat. In total there are maximum $i = 1, \ldots, N = 18,700$ measurements and $k = 1, \ldots, K = 87$ plots in the given data. In each plot, a set of first and last pulse radarsat-measurements concerning the heights and intensity at a certain point are given. The height and intensity histograms are the only information we are using at present.

The variables are defined from some given information:

Very Shallow Soils over Bedroack (mainly dry, locally moist)

- $x_{1,\ldots,3} \in$ Sample type 1 Coniferous (total conifer $\geq 70\%$)
- $x_{4,\ldots,5} \in$ Sample type 2 Mixed or hardwood (total hardwood $\geq 30\%$)

Course-textures Minearal Soils: Glaciofluvial and Lacustrine Material

- $x_{6,\ldots,11} \in$ Sample type 3 Coniferous (total conifer $\geq 65\%$)
- $x_{12,\ldots,15} \in$ Sample type 4 Mixed or hardwood (total hardwood $\geq 25\%$ and $< 65\%$)
- $x_{16,\ldots,19} \in$ Sample type 5 Hardwood (total hardwood $> 75\%$)
- $x_{20,\ldots,21} \in$ Sample type 6 Coniferous (total conifer $\geq 65\%$)
- $x_{22,\ldots,24} \in$ Sample type 7 Mixed or hardwood (total hardwood $\geq 25\%$ and $< 65\%$)
- $x_{25} \in$ Sample type 8 Hardwood (total hardwood $> 75\%$)

Fine-textured Minearal Soils: Lacustrine plains and Morainal Materials

- $x_{26,\ldots,28} \in$ Sample type 16 Mixed or hardwood (total hardwood $\geq 25\%$ and $< 65\%$)

- $x_{29,...,30} \in$ Sample type 17 Hardwood (total hardwood $> 75\%$)
- $x_{31,...,34} \in$ Sample type 18 Coniferous (total conifer $\geq 65\%$)
- $x_{35,...,36} \in$ Sample type 19 Mixed or hardwood (total hardwood $\geq 25\%$ and $< 65\%$)
- $x_{37,...,39} \in$ Sample type 20 Hardwood (total hardwood $> 75\%$)

Productive forested areas

- $x_{40,...,43} \in$ Sample type 21 "Bogs" - black spruce dominated.
- $x_{44,...,48} \in$ Sample type 22 "Fens" - mix of black spruce, larch, alder
- $x_{49} \in$ Sample type 23 "Rich fens" - white cedar and larch are indicator, Ce+L cover $\geq 30\%$

Non-Productive forested areas

- $x_{50,...,53} \in$ Sample type 24 Treed blogs
- $x_{54,...,56} \in$ Sample type 25 Treed fens

Open wetlands

- $x_{57,...,62} \in$ Sample type 26 Open bog
- $x_{63,...,68} \in$ Sample type 27 Open poor fens
- $x_{69,...,71} \in$ Sample type 28 Open rich fen
- $x_{72,...,74} \in$ Sample type 29 Shore fen
- $x_{75,...,77} \in$ Sample type 30 Thicket Swamp
- $x_{78,...,82} \in$ Sample type 31 Meadow march
- $x_{83,...,84} \in$ Sample type 32 Marches

Others

- $x_{85,...,87} \in$ Sample type 33 Recently cutover area

7.3 Classification approaches

Using the Relevance Vector Machine(RVM) and the Support Vector Machine(SVM) respectively, the following results were derived. Figure 7.2 represents classification results using the RVM method and figure 7.3 represents classification results using the SVM method.

The RVM algorithm is a binary classification method, therefore forest areas were labeled as class 1 and wetland areas were labeled as class 0. The classifier was trained using the leave-one-out method. In this iterative method, each plot was considered as a verification plot and the rest were considered as a teaching set. The classification rule was set in such a way that plot n is said to be classified as class 1 (forest) only if $y_n \geq 0.5$, otherwise it belongs to class 0 (wetland).

The classification error is defined as follows:

$$\sum_{n=1}^{87}(t_n \neq round(y_n))/(87 \times 100) \tag{7.1}$$

The figure 7.2 presents the classification result of radarsat data1, by using the RVM method. In the plot, as shown in figure 7.2 green color represents the correct class of the data and blue dot represents the estimated values of the corresponding test data. Based on the classification condition, the classes were assigned the test data. The error percentage defines the classification error. It is the percentage of misclassified data, calculated by using equation (7.1).

A Support Vector Machine (SVM) performs classification by constructing a 2-dimensional hyperplane that optimally separates the data into two categories. These two categories are forestland and wetland. In the algorithm, it was labeled as class 1 to represent the forest area and class 2 for the wetland. But in the plot shown in figure 7.3 these categories were distinguished by using two colors i.e. blue and red. The features belonging to class 1 were marked blue and rest belong to class 2 were marked by red.

The SVM analysis finds the line or hyperplane that is oriented so that the margin between the support vectors is maximized. The figure 7.3 shows the classification results of radarsat data1 with an error percentage of 20.93%. The error percentage gives the percentage of misclassified features. It was also calulated using equation (7.1)

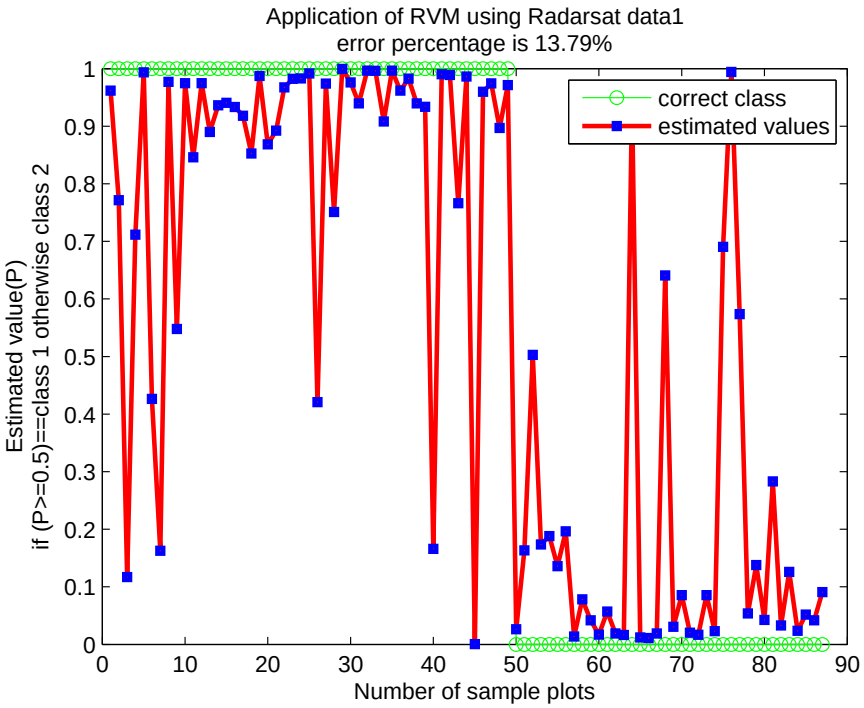


Figure 7.2: The figure is obtained by using the RVM algorithm on Radarsat data case 1 showing error percentage of 13.79%

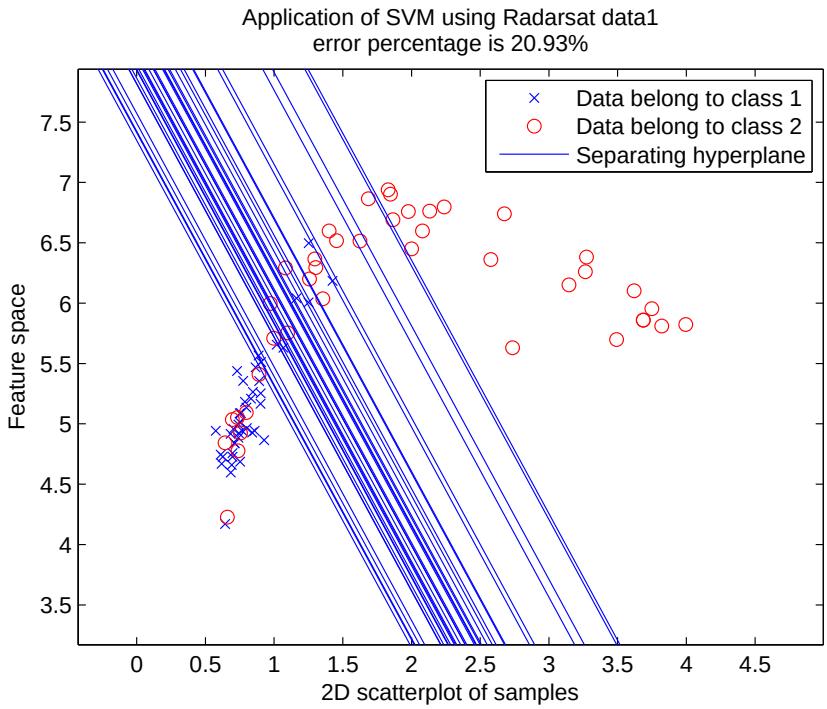


Figure 7.3: The figure is obtained by using the SVM algorithm on Radarset data1, with an error percentage of 20.93%

7.4 Variables based on airborne images

The Variables defined in this section are based on an airborne image. Each plot from an airborne image gives grayscale intensity value pixel by pixel. In total there are maximum $k = 1, \ldots, K = 76$ plots, because of some missing plots including recently cutover stands. These plots were defined using following 76 variables. Each variable describes the forest type, which were same as described in section 7.2, but this image have different amount of plots for each forest type on compare to radarsat image.

Very Shallow Soils over Bedroack (mainly dry, locally moist)

- $x_{1,\ldots,3} \in$ Sample type 1 Coniferous (total conifer $\geq 70\%$)
- $x_{4,\ldots,5} \in$ Sample type 2 Mixed or hardwood (total hardwood $\geq 30\%$)

Course-textures Minearal Soils: Glaciofluvial and Lacustrine Material

- $x_{6,\ldots,10} \in$ Sample type 3 Coniferous (total conifer $\geq 65\%$)
- $x_{11,\ldots,12} \in$ Sample type 4 Mixed or hardwood (total hardwood $\geq 25\%$ and $< 65\%$)
- $x_{13,\ldots,16} \in$ Sample type 5 Hardwood (total hardwood $> 75\%$)
- $x_{17,\ldots,18} \in$ Sample type 6 Coniferous (total conifer $\geq 65\%$)
- $x_{19,\ldots,21} \in$ Sample type 7 Mixed or hardwood (total hardwood $\geq 25\%$ and $< 65\%$)
- $x_{22} \in$ Sample type 8 Hardwood (total hardwood $> 75\%$)

Fine-textured Minearal Soils: Lacustrine plains and Morainal Materials

- $x_{24,\ldots,25} \in$ Sample type 16 Mixed or hardwood (total hardwood $\geq 25\%$ and $< 65\%$)
- $x_{26,\ldots,28} \in$ Sample type 17 Hardwood (total hardwood $> 75\%$)
- $x_{29,\ldots,32} \in$ Sample type 18 Coniferous (total conifer $\geq 65\%$)
- $x_{33} \in$ Sample type 19 Mixed or hardwood (total hardwood $\geq 25\%$ and $< 65\%$)
- $x_{34,\ldots,36} \in$ Sample type 20 Hardwood (total hardwood $> 75\%$)

Productive forested areas

- $x_{37,...,40} \in$ Sample type 21 "Bogs" - black spruce dominated.
- $x_{41,...,44} \in$ Sample type 22 "Fens" - mix of black spruce, larch, alder
- $x_{45} \in$ Sample type 23 "Rich fens" - white cedar and larch are indicator, Ce+L cover $\geq 30\%$

Non-Productive forested areas

- $x_{46,...,48} \in$ Sample type 24 Treed blogs
- $x_{49,...,51} \in$ Sample type 25 Treed fens

Open wetlands

- $x_{52,...,57} \in$ Sample type 26 Open bog
- $x_{58,...,62} \in$ Sample type 27 Open poor fens
- $x_{63,...,65} \in$ Sample type 28 Open rich fen
- $x_{66,...,68} \in$ Sample type 29 Shore fen
- $x_{69,...,70} \in$ Sample type 30 Thicket Swamp
- $x_{71,...,74} \in$ Sample type 31 Meadow march
- $x_{75,...,76} \in$ Sample type 32 Marches

7.5 Classification approaches

Using same classification methods of the Relevance Vector Machine(RVM) and the Support Vector Machine(SVM), for an airborne image. The following two classification results were drawn as shown in figure 7.4 and 7.5.

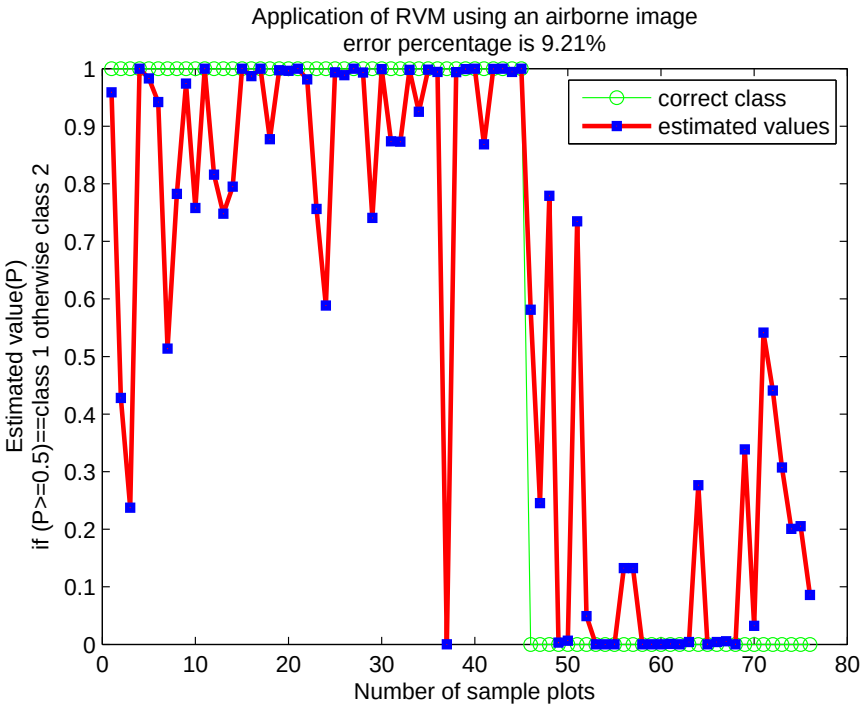


Figure 7.4: The result above gives an application of the RVM algorithm using an airborne image, with an error percentage of 9.21%

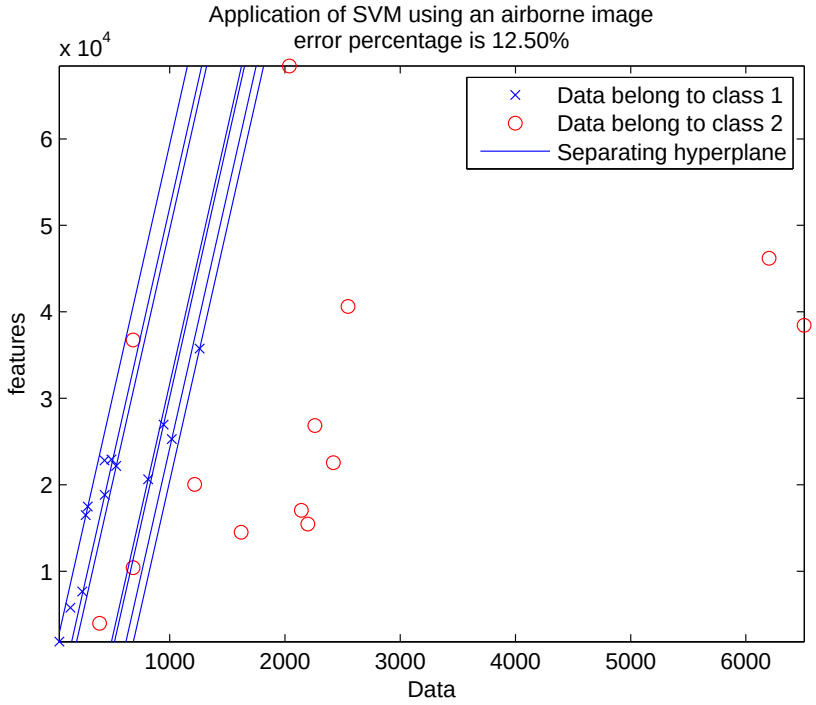


Figure 7.5: The result above gives an application of the SVM algorithm using an airborne image, with an error percentage of 12.50%

8 RESULTS AND DISCUSSION

8.1 Experimental results

Classification between forestland and wetland has been shown in figure 8.1 and 8.2 using Radarsat measurements. Since these measurements were taken on 23-02-2005 and 30-01-05 date, therefore two set of Radarsat measurements were denoted as data1 and data2 respectively.

The forest types were initially grouped into five different types which were coniferous, hardwood, tree soil, open soil and cutover as follows. Respectively these forest types were labeled as 1,2,3,4 and 5. The features were extracted from the data by using histogram variables and perceptual points method. After feature extraction these features were passed on for classification and following classification results were drawn by using the RVM method, as shown in figure 8.1 and 8.2.

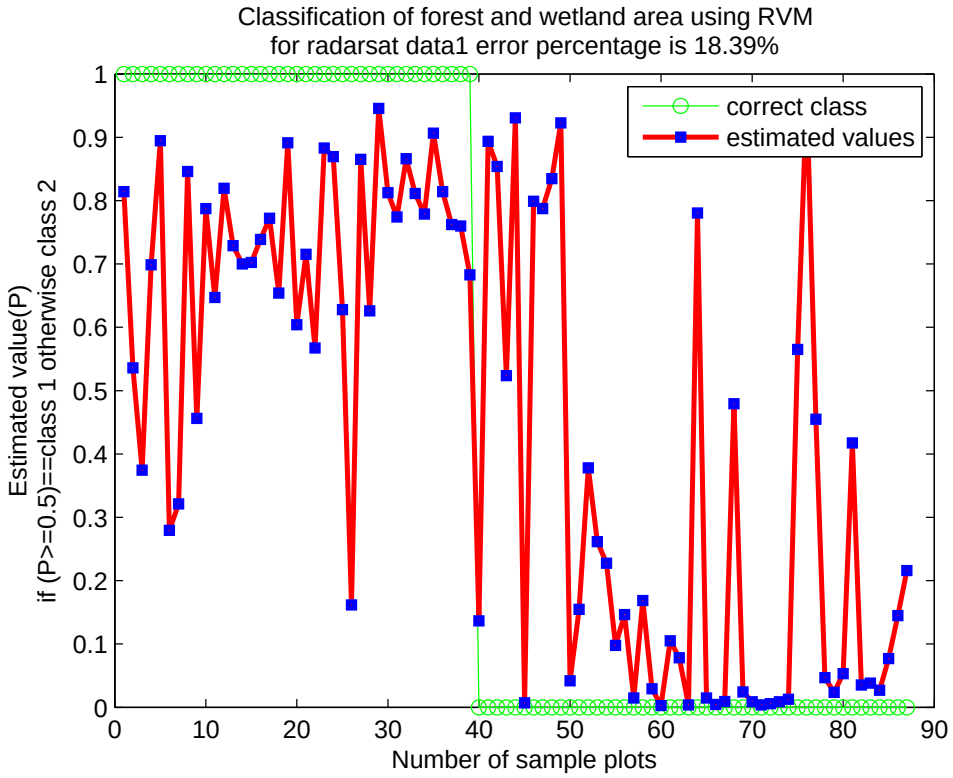


Figure 8.1: For data1 the error percentage is 18.39%

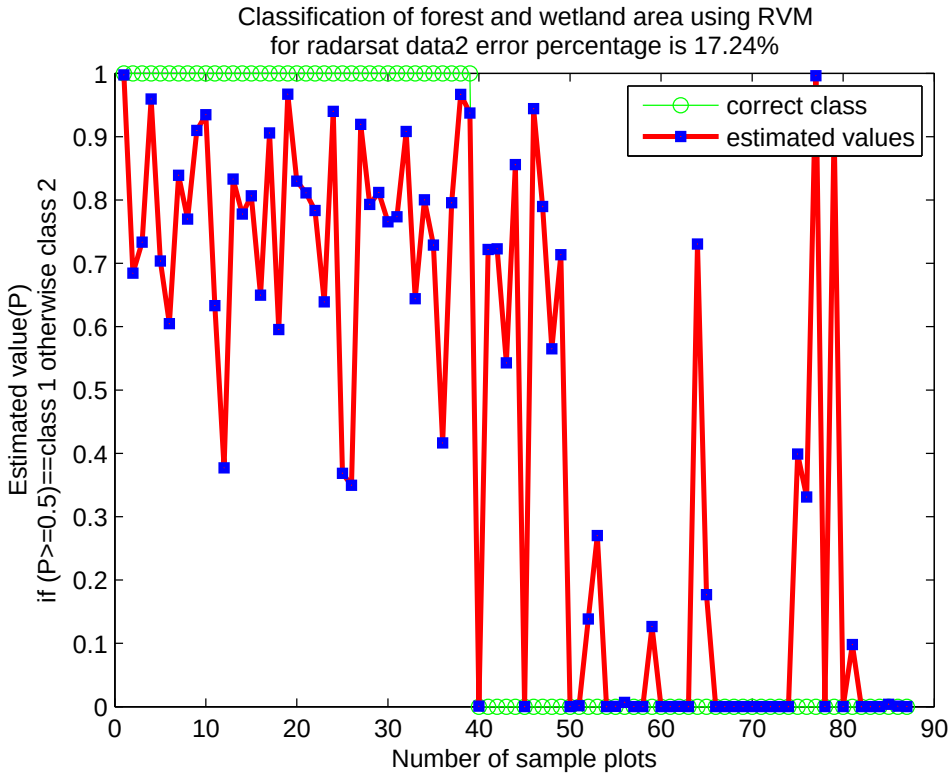


Figure 8.2: For data2 the error percentage is 17.24%

8.2 Classification of forestland and wetland using radarsat data

The forest type distribution presented below gives information about the variable distribution according to the type of forest they belong. Only difference to be noticed here is that the variable from productive forested areas types 21, 22 and 23 now belongs to forest type Hardwood. Feature were extracted using perceptual point and histogram values from the data. Similarly, the productive forested area was labeled '0' stating forestland in the binary classification algorithm which works in similar way as described in previous section.

CON = [1 2 3 4 6 7 15 16 18 19];
HWOOD = [5 8 17 20 21 22 23]
TREESOIL = [24:25]
OPENSOIL = [26:33]
CUTOVER = [33]

Thus the small change of variables labeling in radarsat data1 and radarsat data2, gives new results with less error percentage compare to experimental done in previous section.

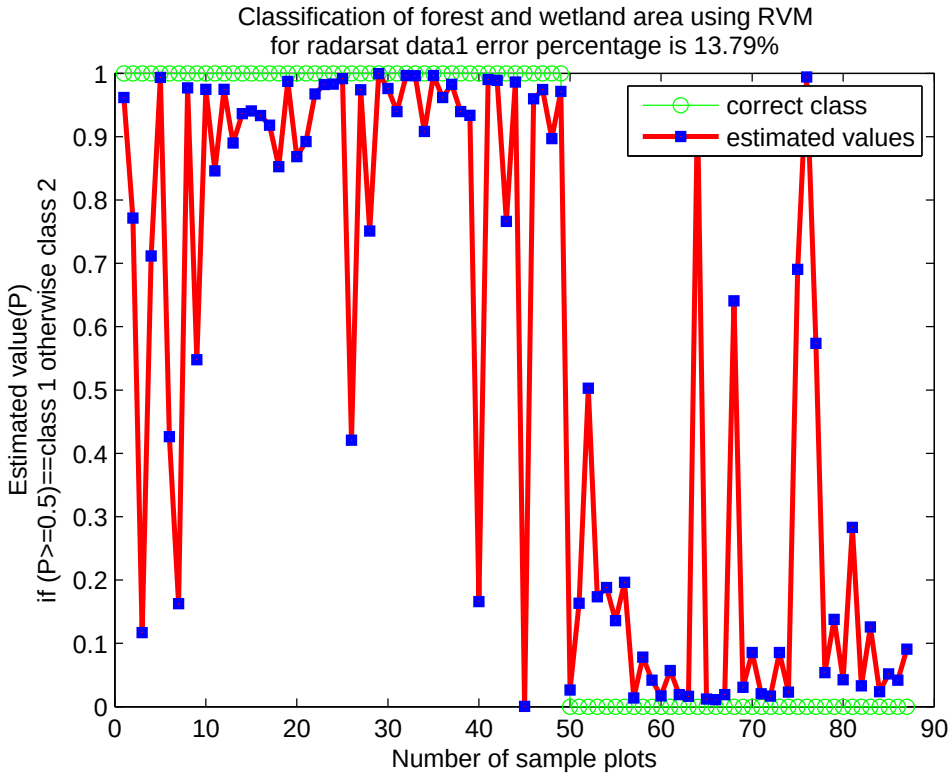


Figure 8.3: For data1 (i.e. of date 23-02-2005), the error percentage is 13.79%

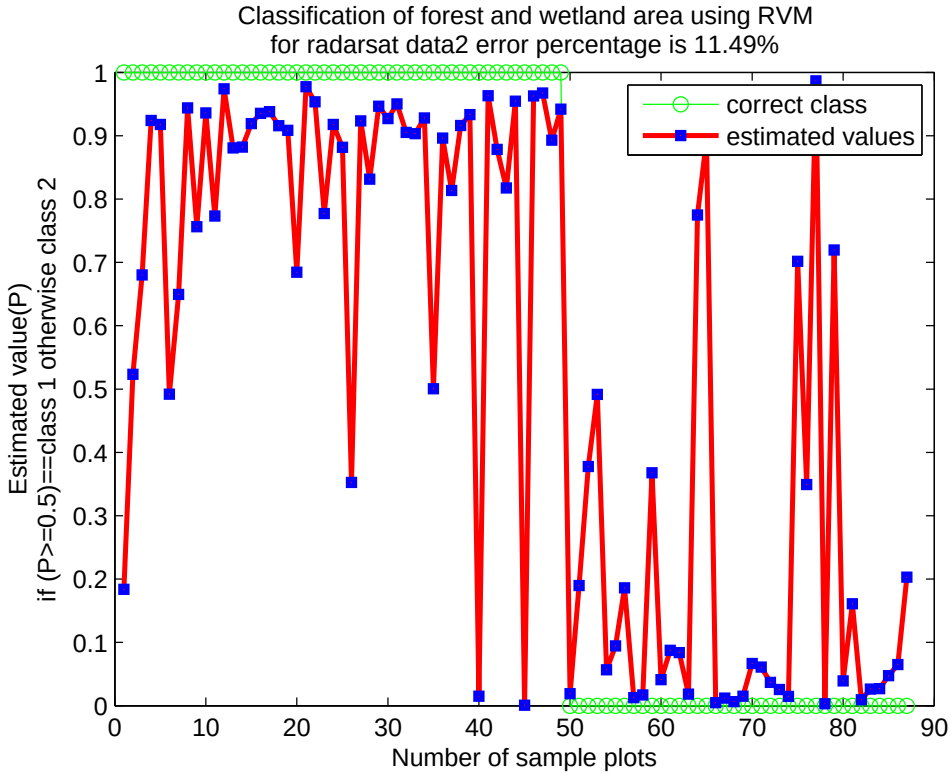


Figure 8.4: For data2 (i.e. of date 30-01-2005), the error percentage is 11.49%

8.3 Classification of forestland species using radarsat data

The Following two experimental results as shown in figure 8.5 and 8.6 gives classification between two types of trees from forestland i.e. Coniferous and Hardwood type forest trees. The variables used in this part of experiments were of forest types only. Since knowledge about only forest type from entire area was obtained from previous experiment. Thus data set containing forest type was obtained. The next step is to classify two different species of tree types. For classification, the same feature extraction technique and binary classifier was used.

Figure 8.5 and 8.6 gives the classification results for data1 and data2 respectively by using RVM. The error percentage provides accuracy rate of an algorithm, in case of data1 error percentage is 46.74%. Similarly for data2 error percentage is 38.78%.

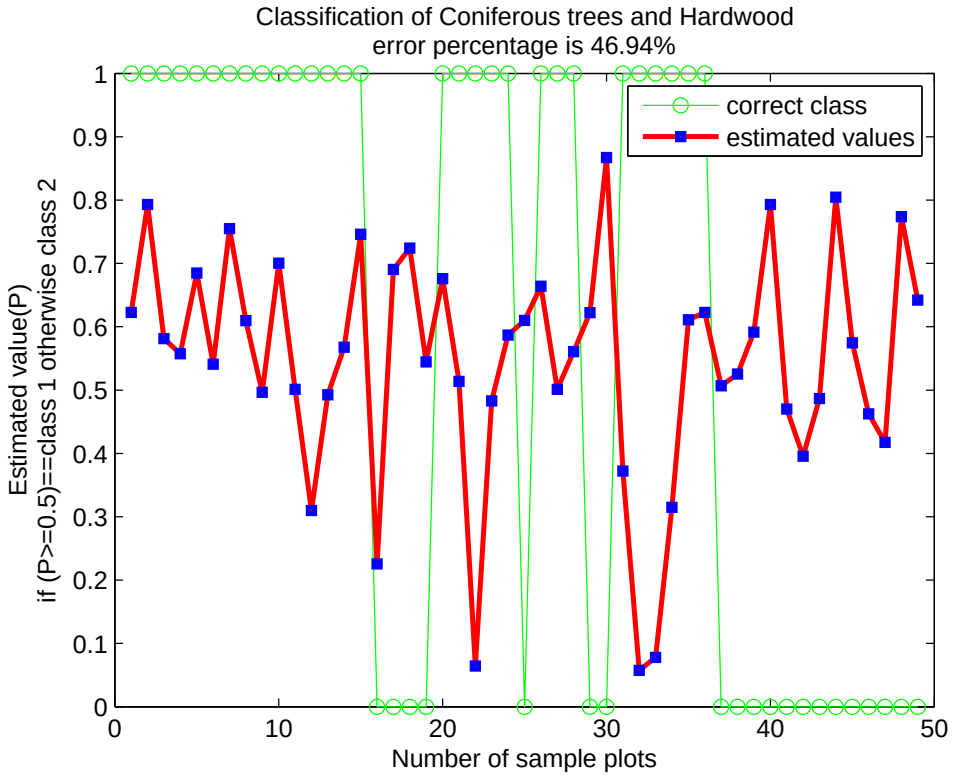


Figure 8.5: For data1 (i.e. of date 23-02-2005), the error percentage is 46.94%

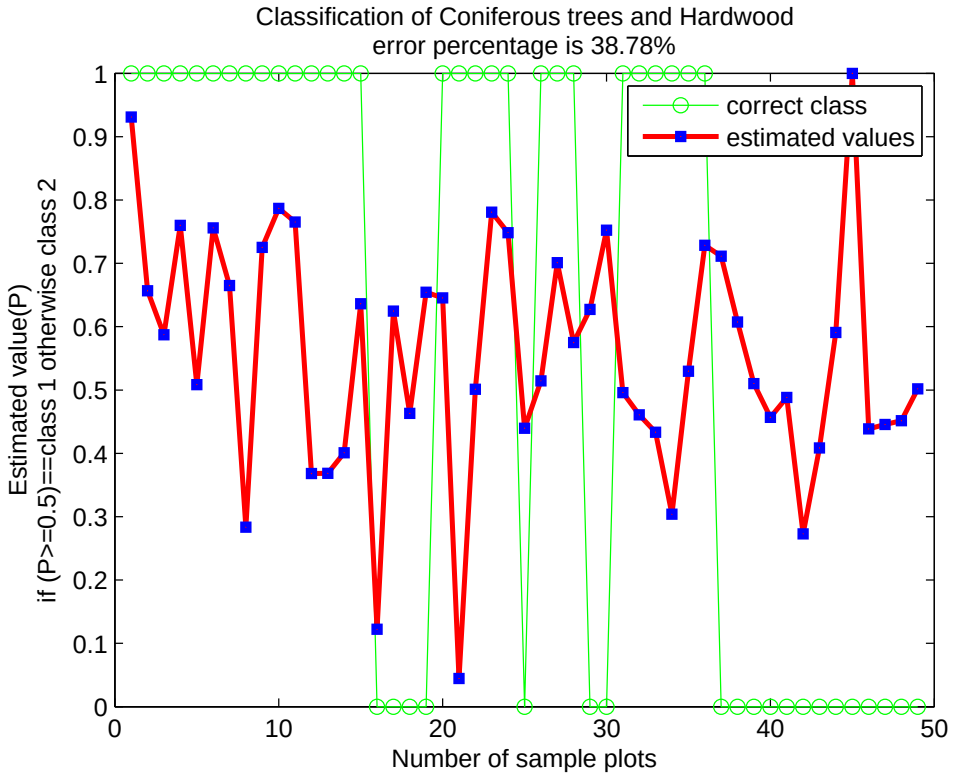


Figure 8.6: For data2 (i.e. of date 30-01-2005), the error percentage is 38.78%

8.4 Classification of wetland species using radarsat data

In this section the following figures represents the classification of two different types of soil from wetland area i.e. nonproductive area and opensoil. The variables describing the wetland species was used to form data structure. It was used for feature extraction method and later the extracted features were used by the RVM to classify nonproductive area and opensoil. The figure 8.7 provides classification results. The error defines the accuracy of an algorithm. For data1 as shown in figure 8.7 error percentage is 17.14% and figure 8.8 represents same classification result for data2 with an error of 14.29%

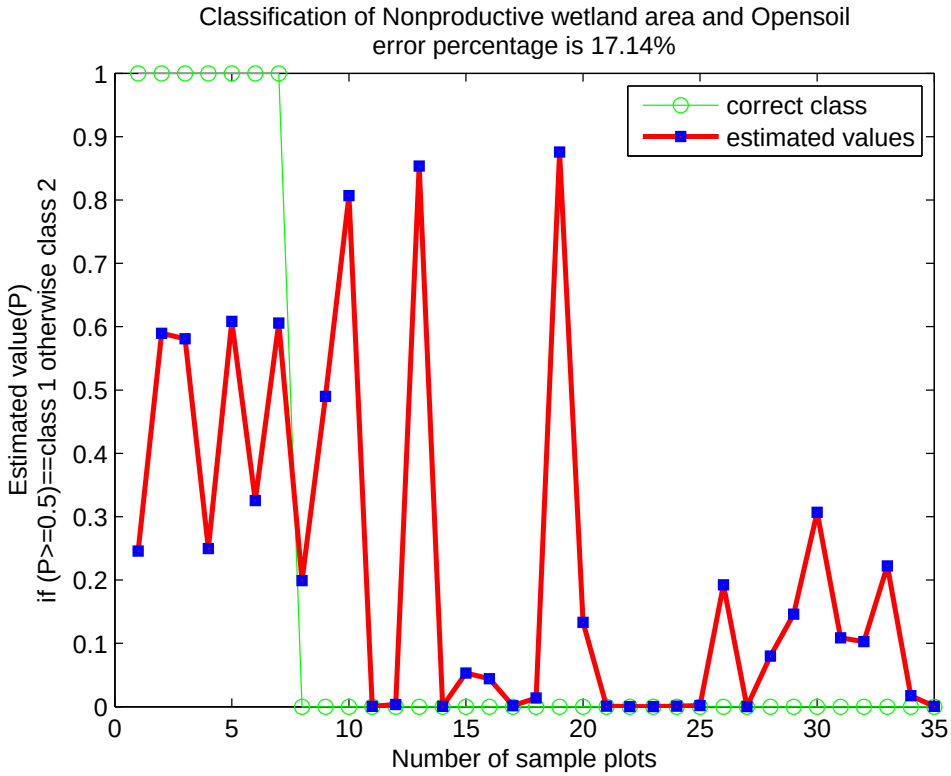


Figure 8.7: For data1 (i.e. of date 23-02-2005), the error percentage is 17.14%

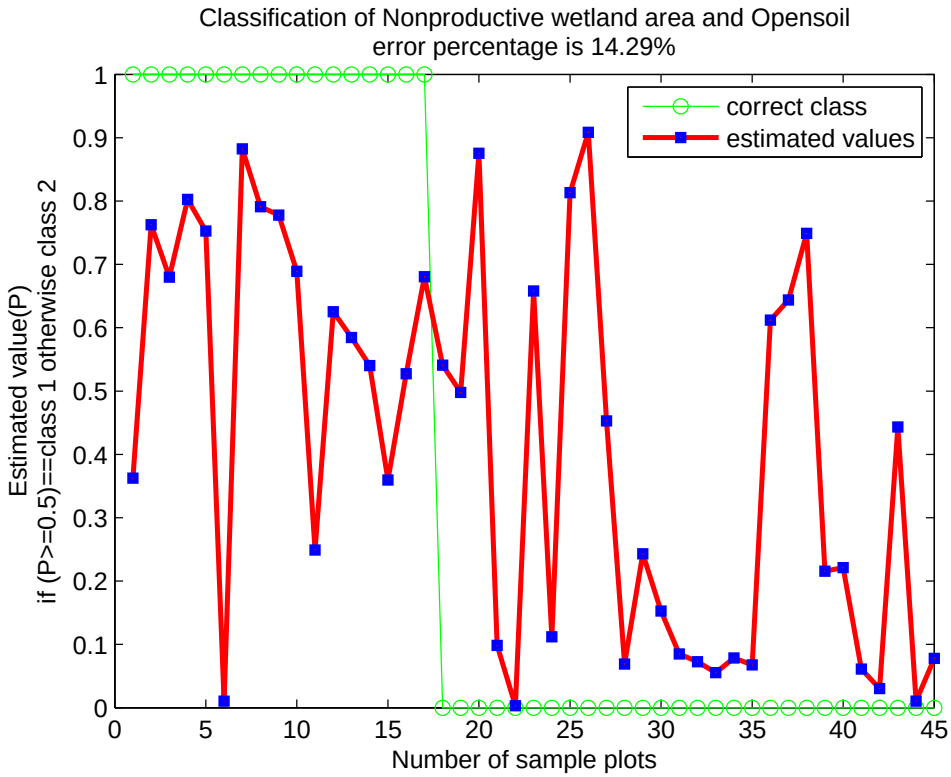


Figure 8.8: For data2 (i.e. of date 30-01-2005), the error percentage is 14.29%

8.5 Conclusions

After performing the entire experiment from modeling till estimation results, some conclusions can be drawn about the relation between the raw data, feature extraction and classification techniques. These three aspects are much dependent on each other. If the raw data is good extracted features will be more informative and hence can be classified easily. This is not always the case in real world problems and experiments, where high nonlinearities might be observed. In such cases this complexity needs to be addressed either by using feature extraction methods or by using an accurate and efficient classification algorithm.

The RVM algorithm has been shown to perform well on the current classification task, even if the feature space is high dimensional, the data interdependencies are complex and there are strong nonlinearities. The table below briefly represents the relative classification error of the RVM and the SVM method observed in the current experiments and application of the SVM method using radarsat data2 is presented in Appendix D.

	Radarsat image	
	RVM	**SVM**
Data1	13.79%	20.93%
Data2	11.49%	30.23%

Table 1: Comparing the RVM and the SVM algorithm using radarsat image

From table 1 it can be observed that in the case of Radarsat measurements the RVM method work more efficiently than the SVM.

We have also performed the corresponding classification using airborne optical images for comparison. In this case it is necessary to perform some image processing operations on the image before feature extraction. In image processing, initially we went through the basic operations, in order to get a better image with more informative features. Then after such features were extracted using histogram values, the processed images display the corresponding intensity pixel by pixel. After collecting histograms of sample plots, we quantized the range of frequencies as features. They were extracted and passed on for classification analysis.

Airborne image	
RVM	**SVM**
9.21%	11.53%

Table 2: Comparing the RVM and the SVM algorithm using data from an airborne image

Table 2 represents the efficiency of both algorithms i.e. the RVM algorithm and the SVM algorithm, in terms of relative error with error percentage 9.21% and 11.53% respectively. As from 1, from 2 it can be observed that even in the case of airborne images, the RVM algorithm provides better results than the SVM method.

Thus after applying both algorithms on two different kinds of data sets extracted from a radarsat image and an airborne image, it can be concluded that the RVM algorithm overcomes all the limitations stated previously in section 4.4 and gives better classification results when compare to the SVM.

On the other side, we can also conclude from a comparison between tables 1 and 2 that airborne images provides better classification than radarsat images. This is to be expected, because airborne images have a much higher resolution than Radarsat images.

However, the similar performance rations between the RVM and the SVM methods indicates that spectral histograms of both imaging modalities do capture relevant information of similar kind for the task of ecological classification. This is a remarkable observation because optical and microwave wavelengths are separated by some six orders of magnitude.

References

[1] Ghahramani, Z. 1997. *Learning Dynamic Bayesian Network*. University of Toronto. Toronto, ON M5S 3H5 Canada.

[2] Richard O. Duda, Peter E. Hart, David G. Stork. 2001. *Pattern Classification*. Second edition. New York, USA: Wiley-Interscience

[3] Sergios Theodoridis, Konstantinos Koutroumbas. 1999. *Pattern Recognition*. Academic Press Publications.

[4] Robert, C. and Georg. C 2004. *Monte Carlo Statistical Methods*. Springer Publications, (Vol. 48).

[5] Kyrki, Ville 2005. *Lecture notes from course Ti5216000 - PATTERN RECOGNITION* . Lappeenranta University of Technology, Finland.

[6] Mitchell, T. 1997. *Machine Learning* . McGraw-Hill publication.

[7] Michael E. Tipping. 2001. *Sparse Bayesian Learning and the Relevance Vector Machine*. Journal of Machine Learning Research 1 (211-244 p). Microsoft Research. Cambridge CB2 3NH, U.K.

[8] D.J.C MacKay. *Bayesian interpolation. Neural Computation*, 4(3):415447 , 1992a.

[9] Rafael C. Gonzalez, Paul Wintz. 1987. *Digital Image processing*. Second edition. Addison-Wesley publication.

[10] Haisong Gu, Qiang Ji. 2004 *Information extraction from image sequences of real-world facial expressions*. Springer Verlang.

[11] David Masip, Ludmila I., Kuncheva, Jordi Vitria. 2005 *An ensemble-based method for linear feature extraction for two-class problems*. Springer Verlang.

[12] Qi Li, Jieping Ye, Chandra Kambhamettu. 2006. *Spatial interest pixels (SIPs): useful low-level features of visual media data*. Western Kentucky University. Bowling Green, KY 42101, USA.

[13] Mark S. Nixon, Alberto S. Aguado. 2002. *Feature extraction Image processing*. First edition. Reed Educational and Professional Publishing Ltd.

Appendix A. Basics of probability and Statistics

Normal Distributions

Center limit theorem is one of the most important results of probability theory which stats that, under various conditions, the distribution for the sum of d independent random variable approaches a particular limiting form known as the normal distribution. As such, the normal or Gaussian probability density function is very important, both for theoretical and practical reasons. In one dimension, it is defined by,

$$P(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-1/2((x-\mu)^2/\sigma^2)} \tag{a.1}$$

The normal density is described as a "bell-shaped curve"; it is completely determined by the numerical values for two parameters, the mean μ and the variance σ^2. This is often emphasized by written $p(x)$ $N(\mu, \sigma^2)$, which is read as "x is distributed normally with mean μ and variance σ^2". The distribution is symmetrical about the mean, the peak occurring at $x = \mu$ and the width of the "bell" is proportional to the standard deviation σ. The parameters of a normal density in equation above satisfy the following equation:

Sigmoid function

A sigmoid function is a mathematical function that produces a sigmoid curve

$$P(t) = \frac{1}{1 + e^{-t}} \tag{a.2}$$

Appendix B. Linear Algebra

Matrix Inversion

The inverse of a $d \times d$ matrix M, denoted M^{-1}, is the $d \times d$ matrix such that,

$$MM^{-1} = 1, \tag{b.1}$$

We call the scalar $C_{ij} = (-1)^{i+j}|M_{i|j}|$ the i, j cofactor or, equivalently, the cofactor of the i, j entry of M. $M_{i|j}$ is the $(d-1) \times (d-1)$ matrix formed by deleting the ith row and jth column of M. The adjoint of M, written Adj[M], is the matrix, whose i, j entry is the j, i cofactor of M. Given these definitions, we can write the inverse of a matrix as,

$$M^{-1} = \frac{Adj[M]}{|M|}, \tag{b.2}$$

If M is not the square then in such case we typically use instead the pseudoinverse M'. If M^TM is nonsingular then the pseudoinverse is defined as,

$$M' = [M^T M]^{-1} M^t \tag{b.3}$$

The pseudoinverse ensure *M'M=I* and is very useful in solving least square problems.

Eigenvectors and Eigenvalues

The inverse of a product of two square matrices obeys $[MN]^{-1} = N^{-1}M^{-1}$, as can be verified by multiplying on the right or the left by MN. Given a d-by-d matrix M, a very important class of linear equations is of the term,

$$Mx = \lambda x$$

for scalar λ, which can be rewritten

$$(M - \lambda I)x = 0$$

Where I the identity matrix and 0 is the zero vector, the solution vector $x = e_i$ and corresponding scalar $\lambda = \lambda_i$ are called the eigenvector and associated eigen value, respectively.

Lagrange Optimization

In search of the an extremum position x_0 of a scalar-valued function $f(x)$, subject to the constraint. If a constraints can be expressed in the form $g(x) = 0$, then we can find the extremum of $f(x)$ as shown below. Firstly we will form a Lagrange function.

$$L(x, \lambda) = f(x) + \underbrace{\lambda g(x)}_{=0}, \tag{b.4}$$

Where λ is a scalar called the Lagrange undetermined multiplier. We convert this constrained optimization problem into an unconstrained problem by taking the derivative,

$$\frac{\partial L(x, \lambda)}{\partial x} = \frac{\partial f(x)}{\partial x} + \lambda \frac{\partial g(x)}{\partial x} = 0, \tag{b.5}$$

And using standard methods from calculus to solve the resulting equation for λ and the extremizing values of x. The solution gives the x position of the extremum, and it is simple matter of substitution to find the extreme values of $f(.)$ under the constraints.

Appendix C. Further Details about Relevance Vector Learning

Maximization of the product of the marginal likelihood and the priors over α and σ^2 (or $\beta = \sigma^{-2}$ for convenience) got involved in Relevance vector learning. Equivalently, and more straightforwardly, we maximize the log of this quality. In additional we are maximizing with respect to $\log\alpha$ and $\log\beta$ - this is convenient since in practice we assume uniform hyper priors over a logarithmic scale, and the derivatives of the prior terms vanish in this space. So, retaining for now general Gamma priors over α and β we maximize:

$$\log p(t|\log\alpha, \log\beta) + \sum_{i=0}^{N} \log p(\log\alpha_i) + p(\log\beta) \tag{c.1}$$

Which, ignoring terms independent of α and β and noting that $p(\log\alpha) = \alpha p(\alpha)$, gives the objective function:

$$L = -\frac{1}{2}[\log|\beta^{-1}I + \Phi A^{-1}\Phi^T| + t^T(\beta^{-1}I + \Phi A^{-1}\Phi^T)^{-1}t] + \sum_{i=0}^{N}(a\log\alpha_i - b\alpha_i) + c\log\beta - d\beta \tag{c.2}$$

Note that the later terms disappear with a, b, c, d set to zero.

Now our task is to consider most robust and efficient way of computing the objective L, considering outline the derivation of the hyperparameter updates and the numerical difficulties involved within algorithm.

A Computing the Log Objective Function

The given matrix $\beta^{-1}I + \Phi A^{-1}\Phi^T$ which appears in the first two terms in L, is of size N×N. However computation of both of the terms of interest may be written as a function of the posterior weight covariance $\Sigma = (A + \beta\Phi^T\Phi)^{-1}$. This matrix is M×M, where M is the number of basic functions in the model. While initially $M = N + 1$ in practice many basis functions will be 'deleted' during optimization (see shortly B.1) and M will decrease considerably giving significant computational advantages as optimization progresses.

We compute the first term by exploiting the determinant identity

$$|A||\beta^{-1}I + \Phi A^{-1}\Phi^T| = |\beta^{-1}I||A + \beta\Phi^T\Phi|,$$

giving

$$\log|\beta^{-1}I + \Phi A^{-1}\Phi^T| = -\log|\Sigma| - N\log\beta - \log|A|, \tag{c.3}$$

Using the Woodbury inversion identity

$$(\beta^{-1}I + \Phi A^{-1}\Phi^T)^{-1} = \beta I - \beta\Phi(A + \beta\Phi^T\Phi)^{-1}\Phi^T\beta,$$

the second, data-dependent, term may be expressed as

$$t^T(\beta^{-1}I + \Phi A^{-1}\Phi^T)^{-1}t = \beta t^T t - \beta t^T\Phi\Sigma\Phi^T\beta t$$

$$= \beta t^t(t - \Phi\mu) \tag{c.4}$$

With $\mu = \beta\Sigma\Phi^T t$, the posterior weights mean. Note that () may be also be re-expressed as:

$$\begin{aligned}\beta t^t(t - \Phi\mu) &= \beta\left\|t - \Phi\mu\right\|^2 + \beta t^T\Phi\mu - \beta\mu^T\Phi^T\Phi\mu,\\ &= \beta\left\|t - \Phi\mu\right\|^2 + \mu^T\Sigma^{-1}\mu - \beta\mu^T\Phi^T\Phi\mu\\ &= \beta||t - \Phi\mu||^2 + \mu^T A\mu\end{aligned} \tag{c.5}$$

Which corresponds to the penalized log likelihood evaluated using the posterior mean weights. The terms in (c.3) are sometimes referred to as the "Ockham factors".

Computation of Σ, its determinant and μ is achieved robustly through Cholesky decomposition of $A + \beta\Phi^T\Phi$ an $O(M^3)$ procedure.

Derivatives and Updates

The Hyperparameters

The deriative of (c.2) with respect to $\log\alpha$ are:

$$\frac{\partial L}{\partial \log \alpha_i} = \frac{1}{2}[1 - \alpha_i(\mu_i^2 + N_{ii})] + a - b\alpha_i \tag{c.6}$$

Equating above equation equals to zero and solving for α_i gives a re-estimation rule:

$$\alpha_i^{New} = \frac{1 + 2\alpha}{\mu_i^2 + N_{ii} + 2b} \tag{c.7}$$

This is same as an *expectation-maximisation (EM)* update and so is guaranteed to locally maximise L. However, setting eq (c.6) to zero, and following MacKay(1992a) in defining quantities $\gamma_i \equiv 1 - \alpha_i N_{ii}$, leads to the following update:

$$\alpha_i^{new} = \frac{\gamma_i + 2a}{\mu_i^2 + 2b} \tag{c.8}$$

which was observed to leaad to much faster convergence although it does not benifit from the guarantee of local maximisation of L.

The noise variance

Derivatives with respect to $\log\beta$ are:

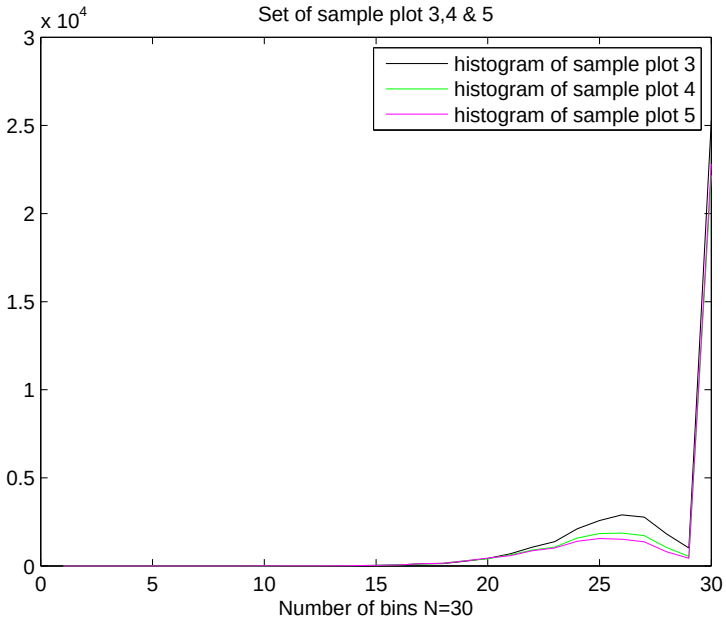
$$\frac{\partial L}{\log \beta} = \frac{1}{2}\left[\frac{N}{\beta} - \|t - \Phi\mu\|^2 - tr(\sum \Phi^T\Phi)\right] + c - d\beta, \tag{c.9}$$

and since $tr(\sum \Phi^T \Phi)$ can be written as $\beta^{-1} \sum_i \gamma_i$ setting the derivative to zero and rearranging and re-expressing in terms of σ^2 gives:

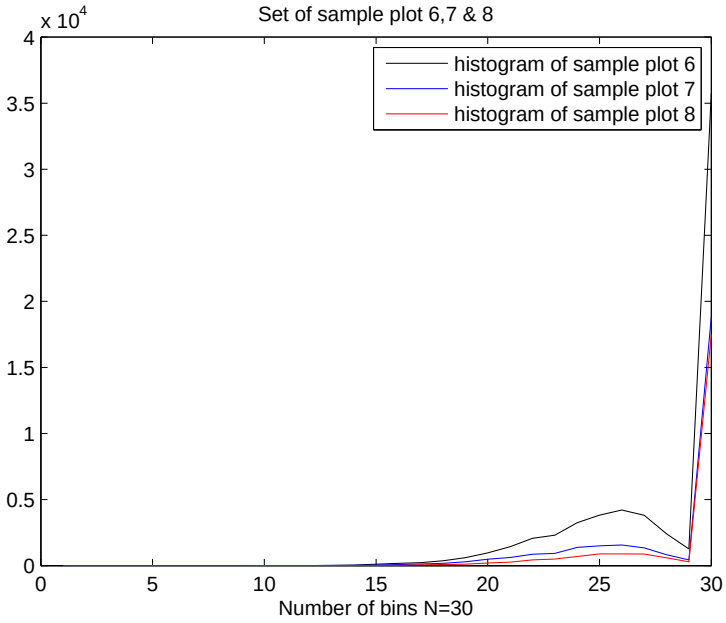
$$(\sigma^2)^{new} = \frac{\|t - \Phi\mu\|^2 + 2d}{N - \sum_i \gamma_i + 2c} \tag{c.10}$$

Appendix D. Histograms of sample plots

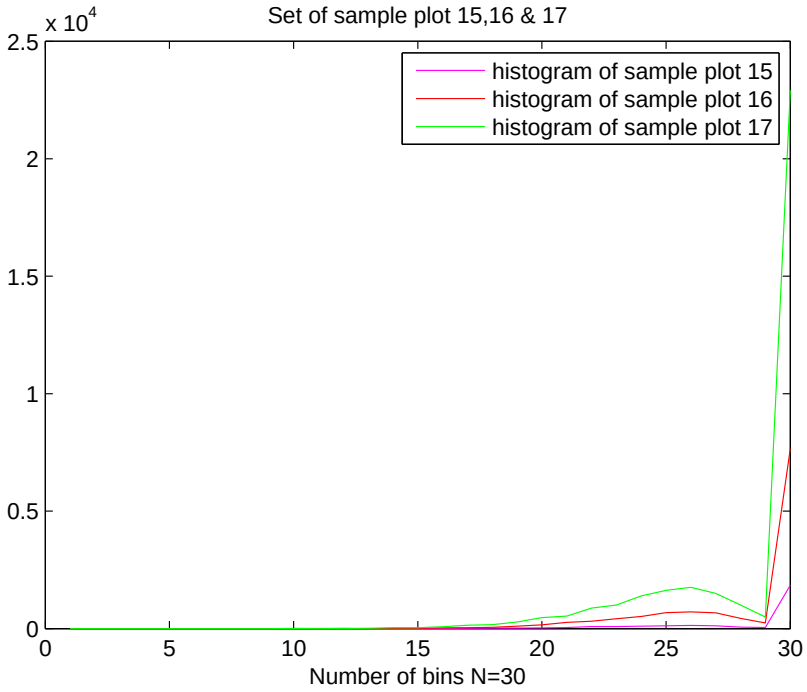
The below seven figures gives the representation of the greyscaled histogram of the sample plots, as explained in section 6.1.



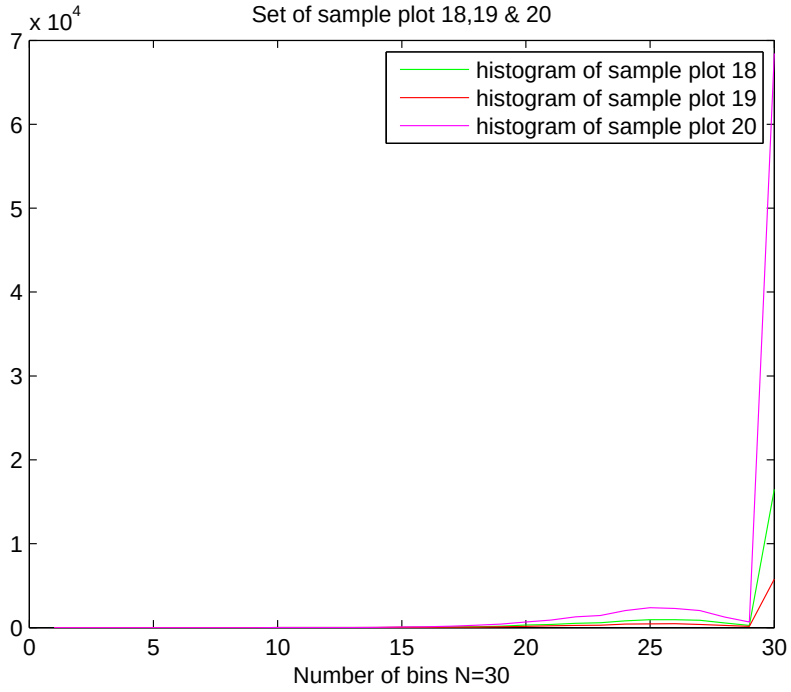
The figure shows the histogram of sample plot 3,4 and 5



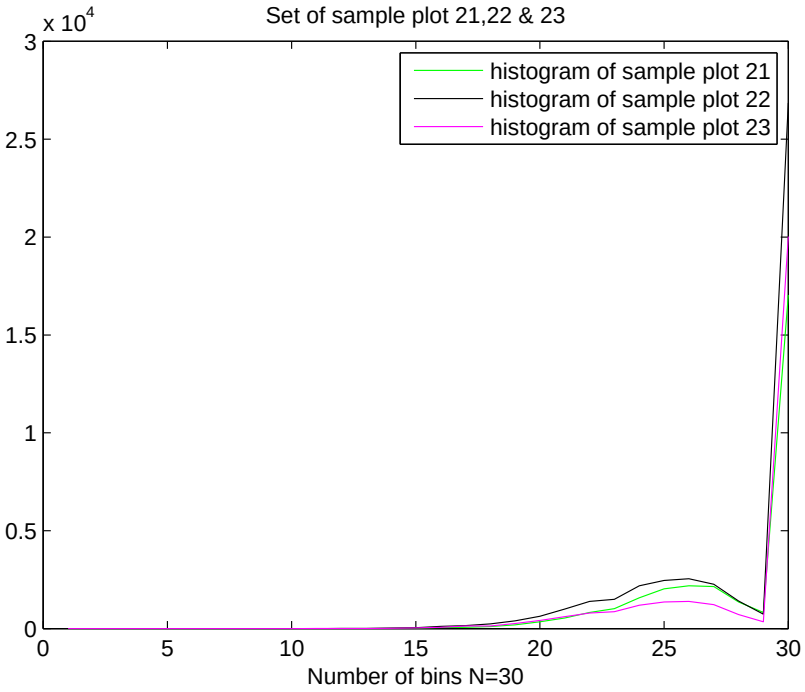
The figure shows the histogram of sample plot 6,7 and 8



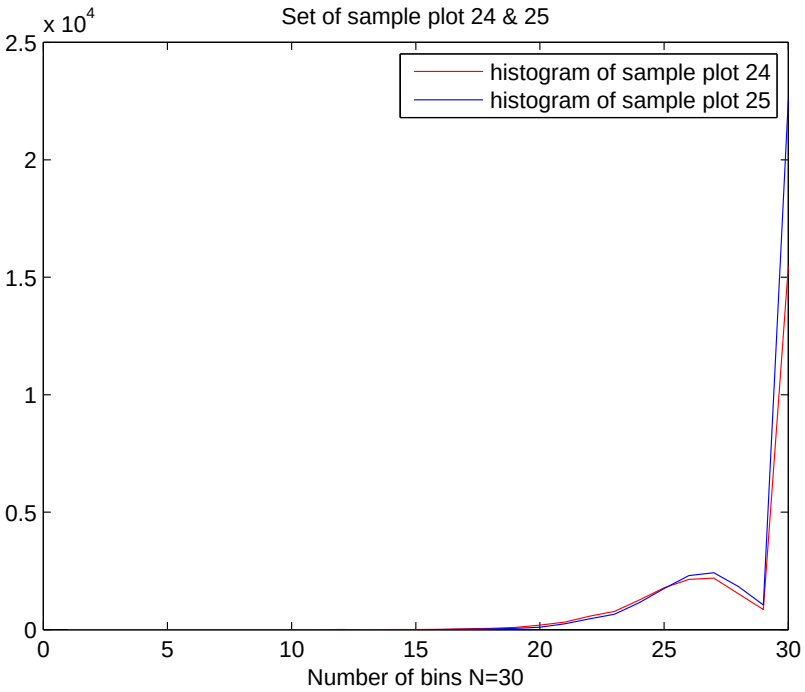
The figure shows the histogram of sample plot 15,16 and 17



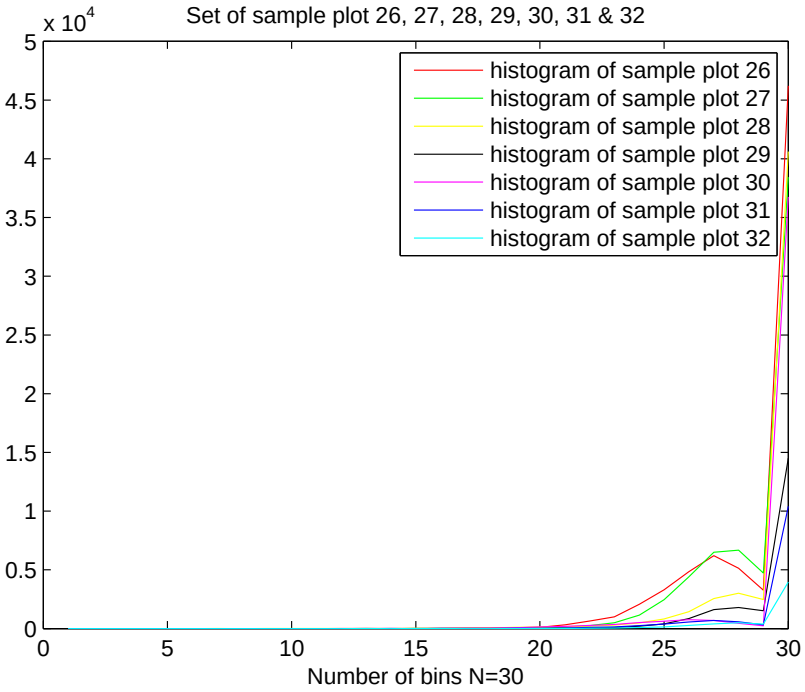
The figure shows the histogram of sample plot 18,19 and 20



The figure shows the histogram of sample plot 21,22 and 23

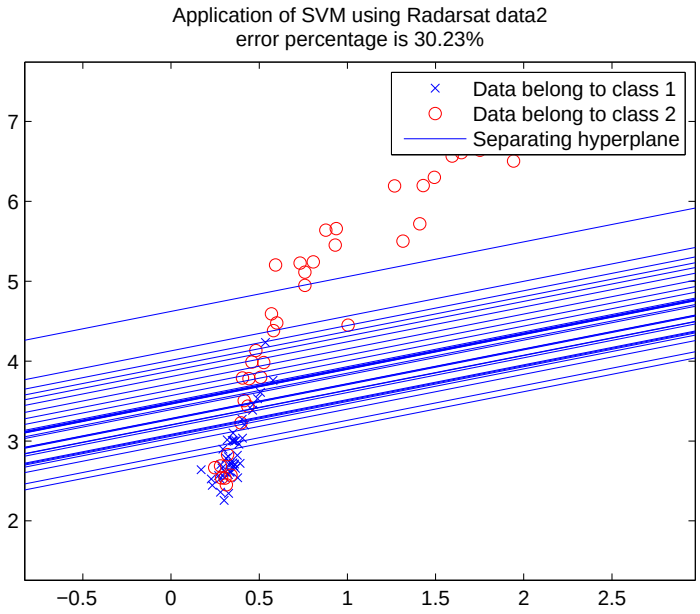


The figure shows the histogram of sample plot 24 and 25



The figure shows the histogram of sample plot 26,27,28,29,30,31 and 32

The figure below represent the SVM result for radarsat data2 and it was used in table 1 of section 8.5.



The figure shows application of the SVM method using radarsat data2