Likelihood-based classification of single trees in hemi-boreal forests

Simon Vallin

Umeå University

Supervisors: Konrad Abramowicz and Åke Brännström

Master thesis: 30 ECTS

Simon Vallin: *Likelihood-based classification of single trees in hemi-boreal forests*.
Master's thesis in mathematical statistics, November 2014

Department:
Mathematics and mathematical statistics

Supervisors:
Konrad Abramowicz and Åke Brännström

Examiner:
Sara Sjöstedt de Luna

Location:
Umeå, Sweden

Time frame:
February 2013 - November 2014

Abstract

Determining species of individual trees is important for forest management. In this thesis we investigate if it is possible to discriminate between Norway spruce, Scots pine and deciduous trees from airborne laser scanning data by using unique probability density functions estimated for each specie. We estimate the probability density functions in three different ways: by fitting a beta distribution, histogram density estimation and kernel density estimation. All these methods classifies single laser returns (and not segments of laser returns). The resulting classification is compared with a reference method based on features extracted from airborne laser scanning data. We measure how well a method performs by using the overall accuracy,that is the proportion of correctly predicted trees. The highest overall accuracy obtained by the methods we developed in this thesis is obtained by using histogram-density estimation where an overall accuracy of 83.4 percent is achieved. This result can be compared with the best result from the reference method that produced an overall accuracy of 84.1 percent. The fact that we achieve a high level of correctly classified trees indicates that it is possible to use these types of methods for identification of tree species.

Sammanfattning

Att kunna artbestämma enskilda träd är viktigt inom skogsbruket. I denna uppsats undersöker vi om det är möjligt att skilja mellan gran, tall och lövträd med data från en flygburen laserskanner genom att skatta en unik täthetsfunktion för varje trädslag. Täthetsfunktionerna skattas på tre olika sätt: genom att anpassa en beta-fördelning, skatta täthetsfunktionen med histogram samt skatta täthetsfunktionen med en kernel täthetsskattning. Alla dessa metoder klassificerar varje enskild laserretur (och inte segment av laserreturer). Resultaten från vår klassificering jämförs sedan med en referensmetod som bygger på särdrag från laserskanner data. Vi mäter hur väl metoderna presterar genom att jämföra den totala precisionen, vilket är andelen korrekt klassificerade träd. Den högsta totala precisionen för de framtagna metoderna i denna uppsats erhölls med metoden som bygger på täthetsskattning med histogram. Precisionen för denna metod var $83,4$ procent rättklassificerade träd. Detta kan jämföras med en rättklassificering på $84,1$ procent vilket är det bästa resultatet för referensmetoderna. Att vi erhåller en så pass hög grad av rättklassificerade träd tyder på att de metoder som vi använder oss av är användbara för trädslagsklassificering.

Acknowledgements

First of all, I wish to thank my supervisors at Umeå University, Konrad Abramowicz and Åke Brännström for their time, help and patience. I also wish to thank Johan Holmgren and Eva Lindberg at the Swedish University of Agricultural Sciences in Umeå for providing the data for this thesis, answering all my questions and for all their advice. Without these people I would never have finished this thesis. Special thanks to Nicklas Persson and Erik Liljas for their time spent reading this thesis and providing constructive criticism.

Contents

1 Introduction **1**

2 Theory **4**

2.1 Probability Distributions 4

2.1.1 Normal Distribution 4

2.1.2 Beta-distribution 6

2.2 Non-parametric Density Estimation 7

2.2.1 Histogram Density Estimation 7

2.2.2 Kernel Density Estimation 9

2.3 Parametric Density Estimation 11

2.3.1 Maximum Likelihood Estimation 11

2.4 Discriminant Analysis 12

2.4.1 Naive Bayesian Classifier 13

2.4.2 Linear and Quadratic Discriminant Analysis 14

2.5 Cluster Analysis 16

2.5.1 k-means Clustering 16

2.6 Cross-validation 17

2.6.1 k-fold Cross-validation 17

3 Material and Methods **19**

3.1 The Data 19

3.1.1 Transformations 21

3.1.2 Height Classes 21

3.1.3 Data Used For The Reference Method 23

3.2 Methods for Classification 23

3.2.1 Accuracy Measures 25

3.2.2 Naive Bayesian Classifier 25

3.2.3 Reference Method 28
3.2.4 Tuning the Methods 28
3.2.5 Validating the Methods 29
3.2.6 Cluster Analysis With Beta-distribution 29

4 Results **32**

5 Discussion **39**
5.1 Results and Data 39
5.2 Further Studies 40
5.3 Conclusions 41

A Result Model Training Histogram **44**

B Result Model Training KDE **56**

C Results Model Training Classification Based On Beta-distribution **60**

D Supplementary Results **62**

Chapter 1

Introduction

Being able to determine species of individual trees is an important part of forest management. Having reliable information is crucial for both economic and ecological reasons. Information about tree species is important for estimating the economic values of the timber, since timber from different trees are used for different purposes and hence have different values. The ecological value of a forest is determined by which type of trees grows in the forest, since some species of animals require specific types of trees to survive. Errors made in tree species identification can lead to large errors in the calculation of stem biomass and stem dimensions as the allometric (also known as biological scaling, e.g the proportion of the tree crown) dependencies are species specific. Species inaccuracy in the inventory data may also result in non-optimal management decisions (Korpela and Tokola, 2006). Since traditional forest inventory methods are associated with high costs (Wezyk et al., 2007) it is important to find a non-expensive method to classify trees. One approach to this problem adapted during the past two decades is to use airborne laser scanning techniques.

Airborne Laser Scanning (ALS), also referred to as Light Detection And Ranging (LiDAR) has proved to provide information good enough to accurately determine species of individual trees (Næsset et al., 2004). LiDAR is an optical remote sensing technology that allows measures of relative heights of objects on the ground. Because it allows large surfaces to be surveyed at relatively low cost, LiDAR is widely applied for gathering data for forest inventory. A LiDAR system consists of the following components: an airborne platform, a GPS, an inertial navigation system and a laser scanner system. The distance to targets is calculated by emitting a laser light and measuring the light reflected back, the position of the object is calculated by using the first and last peak of the returning pulse.

A conceptual illustration of how the phase of light can be used to calculate the distance to an object on the ground can be seen in Figure 1.1. For further details see Lindberg (2012). In order to identify and extract individual trees from the laser data the data are segmented by some segmentation method e.g. Holmgren et al. (2002) or Vauhkonen et al. (2009). In the segmentation process the outline of the tree crowns are extracted. LiDAR derived canopy height models are often used to perform individual tree segmentation (e.g., Holmgren et al., 2002; Solberg et al., 2006). When the outline of the tree crown is derived laser returns inside the segment can be connected to a specific tree and detailed information about that tree can bee derived (e.g. Holmgren et al., 2002; Solberg et al., 2006).

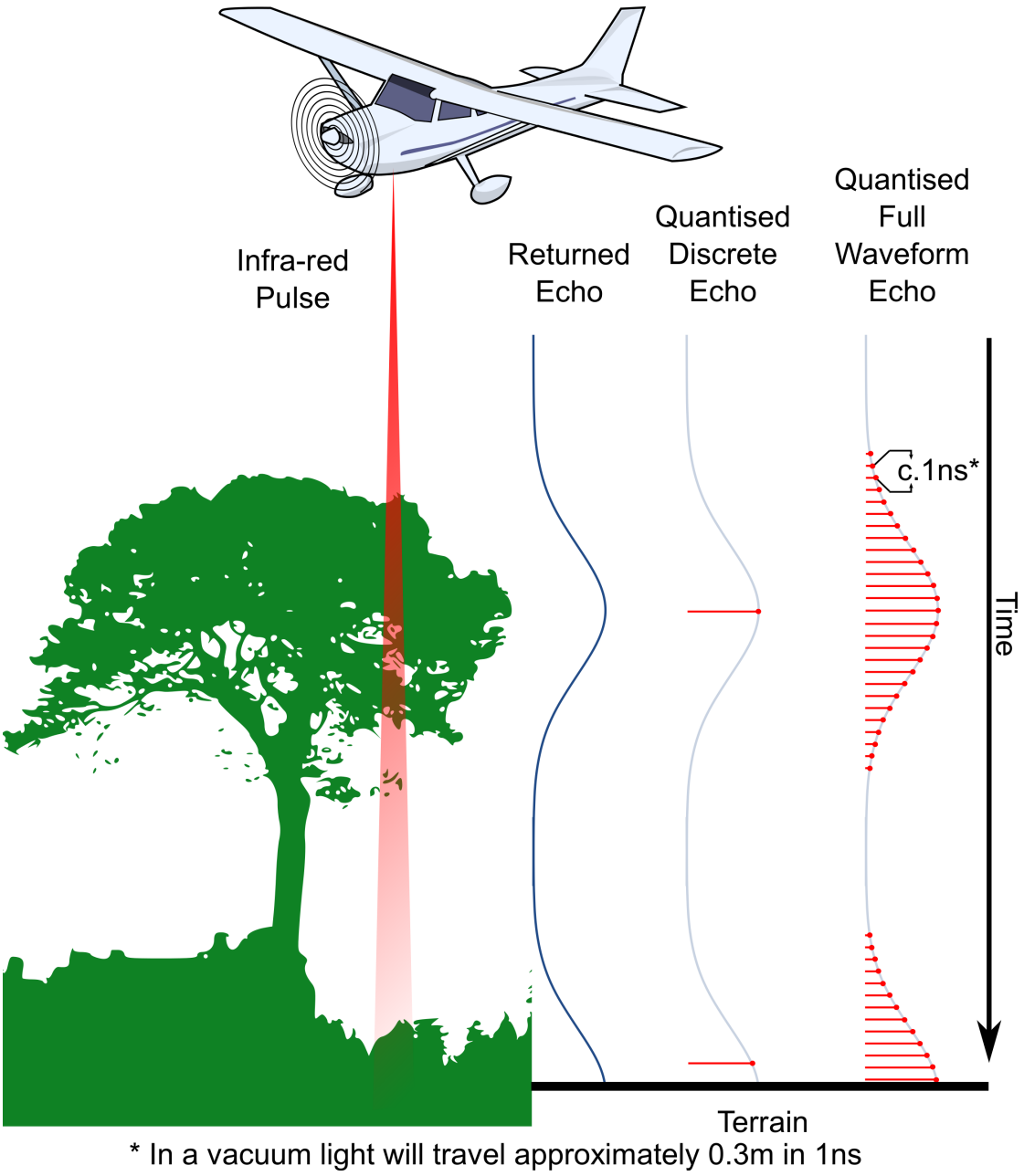

Figure 1.1: Conceptual illustration of airborne light detection and ranging (LiDAR) for forestry applications. The figure illustrates how the phase of the light is measured in order to calculate the distance to object on the ground. Source: Anthony Beck, via Wikimedia Commons

Over the last decade many papers addressing the issue of tree classification using LiDAR have been published (e.g. Liang et al., 2007; Heinzel et al., 2008; Holmgren and Persson, 2004; Ørka et al., 2007). Ørka et al. (2007) use two intensity metrics of mean intensity and standard deviation of intensity. These values are computed for laser return categories in each tree. Six variables

are thereafter derived. The classification of trees is performed with linear discriminant analysis. Heinzel et al. (2008) study if laser scanning data and colour infra-red aerial photographs can be used to classify trees. They separate the spectral bands into near infrared, red, green and further transformed into hue, intensity and saturation channels. Single tree polygons are derived from LiDAR data and these are fitted to spectral data. The classification is divided into two steps. In the first step, classes are separated into two groups. In the second step the global maximum position of the near infrared-band within each tree polygon is calculated. From these values a function that describes the frequency of the maximum distribution of all trees is calculated. The discrimination threshold is obtained from local minimum of the automatically smoothed function.

The identification of tree species in boreal forests in Scandinavia using LiDAR has been studied in previous articles by Holmgren and Persson (2004); Liang et al. (2007). Liang et al. (2007) use leaf-off data in their study. They assume that in coniferous trees, first and last pulse signals are reflected by tree tops whereas in deciduous trees, first pulse should hit the treetops and the last signal should reflect of the ground. This fact is used in the classification of the trees.

In Holmgren and Persson (2004), the authors study how features extracted from airborne laser scanning data can be used to discriminate between Scots pine and Norway spruce. The features that are used are related to the characteristics of the crown shape and structure, as well as non-shape measures derived from the laser data such as, the return intensity and proportions of different types of laser returns. In order to classify the trees Holmgren and Persson use linear and quadratic discriminant analysis based on variables derived from laser data.

Since different trees have different shapes the spread of laser returns in the tree crown should be different for different tree species. If this is the case it should be possible to fit a unique probability density function for the spread of laser returns for every tree species (using single laser returns instead of segments of laser returns which is used in all the reports described above). These density functions could then be used to discriminate between different tree species. In this thesis we investigate if it is possible to obtain higher precision by estimating a probability density function for the spread of laser returns in the tree crown for every tree species in order to discriminate between Norway spruce, Scots pine and deciduous trees. We also implement Holmgrens and Perssons method as reference method to our methods.

Chapter 2

Theory

In this Chapter we introduce the theory that underpins the tree-classification methods presented in Chapter 3. We start by introducing some recurring probability density functions in Section 2.1, how to estimate probability density functions needed for some of our classifiers is done in the Section 2.2 and 2.3. In Section 2.4 we introduce the different classifiers. In Chapter 2.5 we introduce k-means clustering, which is later used as an alternative method to classifiers for identification of tree species. We end the Chapter by explaining cross validation, a technique we use to estimate the performance of the methods.

2.1 Probability Distributions

In this section we introduce probability distributions that are recurring throughout this thesis. These distributions are the normal distribution (for one, two and higher dimensions) and the beta-distribution.

2.1.1 Normal Distribution

We use the normal distribution for two purposes in this thesis. These purposes are described later on in this chapter. The normal distribution is a continuous probability distribution defined for all real numbers.

One-dimensional Normal Distribution

A normal distribution with mean μ and variance σ^2 has the probability density function (pdf) :

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{(x-\mu)^2}{2\sigma^2}\right). \tag{2.1}$$

The normal distribution is symmetric around the expected value μ. The shape of the distribution is only dependent on the standard deviation σ. A high value produces a flat shape and a low value a more acute shape.

Bivariate Normal Distribution

The bivariate normal distribution with covariance matrix

$$\mathbf{\Sigma} = \begin{pmatrix} \sigma_1 & \sigma_{12} \\ \sigma_{12} & \sigma_2 \end{pmatrix},$$

is defined by the following pdf:

$$f(\boldsymbol{x}) = \frac{1}{2\pi|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right), \tag{2.2}$$

where $|.|$ represents the determinant. This can also be expressed as:

$$f(x_1, x_2) = \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} \exp\left(-\frac{Z}{2(1-\rho^2)}\right), \tag{2.3}$$

where

$$Z = \frac{(x_1-\mu_1)^2}{\sigma_1^2} - \frac{2\rho(x_1-\mu_1)(x_2-\mu_2)}{\sigma_1\sigma_2} + \frac{(x_2-\mu_2)^2}{\sigma_2^2}$$

and ρ is the correlation coefficient between X_1 and X_2, defined as $\sigma_{12}/(\sigma_1\sigma_2)$. In Figure 2.1 we can see a bivariate normal distribution.

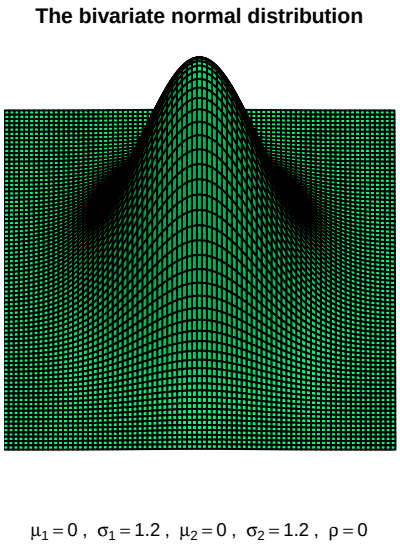

Figure 2.1: The density function for the bivariate normal distribution, with $\sigma_1 = \sigma_2 = 1.2$, $\rho = 0$ and centre of mass at $\mu_1 = \mu_2 = 0$.

Multivariate Normal Distribution

The multivariate normal distribution is a generalization of the one-dimensional normal distribution to d dimensions. The pdf of a multivariate normal distribution of a d-dimensional random vector $\boldsymbol{X} = (X_1, ..., X_d)$ is defined as:

$$f(\boldsymbol{x}) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right). \tag{2.4}$$

The expected value is estimated by the sample mean, that is $\hat{\boldsymbol{\mu}} = \bar{\boldsymbol{x}}$. The sample covariance Σ for n observations is estimated by:

$$\frac{1}{n-1}\sum_{i=1}^{n}(\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^T.$$

This is an unbiased estimator of the covariance matrix. When the normal distribution is used, this is how $\boldsymbol{\mu}$ and Σ are estimated.

2.1.2 Beta-distribution

In this section we describe the properties of the beta-distribution. The beta-distribution is a continuous probability distribution defined on the interval $[0, 1]$. It is parameterized by two shape parameters α and β. The pdf of the beta-distribution is defined as:

$$f(x) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}x^{\alpha-1}(1-x)^{\beta-1}, \tag{2.5}$$

where $\alpha>0$, $\beta>0$, $0 \leq x \leq 1$ and $\Gamma(x)$ is the Gamma function, defined as:

$$\Gamma(z) = \int_0^\infty t^{z-1}e^{-t}dt.$$

For more information about the beta-distribution and the Gamma function we refer to Hogg and Tannis (2009).

In Figure 2.2 we can see some examples of how the pdf of the beta-distribution is affected by changing the α and β parameters. By changing the α and β parameters it is possible to achieve shapes that resembles the shape of a tree. By choosing α=β we get a symmetric shape of the pdf and by letting one of the parameters be greater than the other we get a skewed shape of the pdf.

The α and β parameters are estimated by direct optimisation of the log-likelihood using Nelder-Mead's method on the training data. For a description of the Nelder-Mead method see Gao and Han (2012) and Wouk (1987, p.117-120). Often times we want to estimate the distributions from data. We will now introduce such a method.

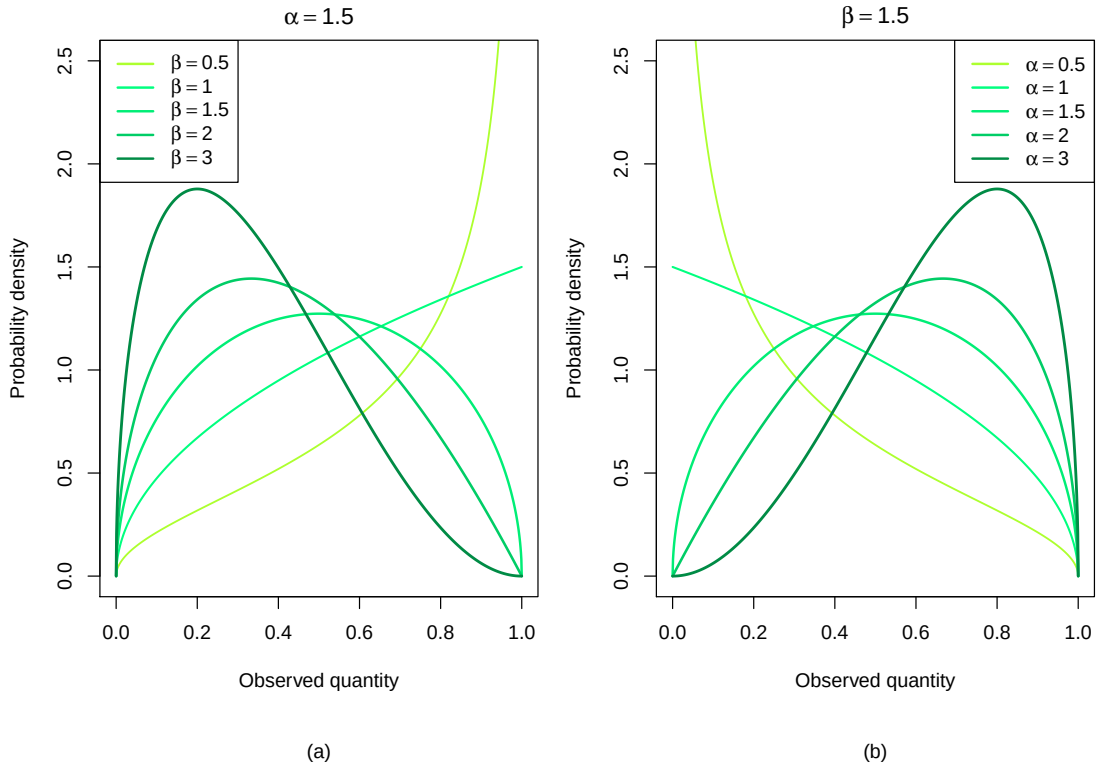

Figure 2.2: Illustration of how the density function for the beta-distribution is affected by changing the α and β parameters. In Figure (a) we see the effect of changing the α parameter and in (b) we see the effect of changing the β parameter. Thicker lines illustrates higher values of the changeable parameter.

2.2 Non-parametric Density Estimation

Let $X_1, X_2, ..., X_n$ be independent and identically distributed (i.i.d.) random variables with a continuous probability density function (pdf) $f(x)$. Non-parametric density estimation is a way to estimate $f(x)$ under minimal assumptions.

2.2.1 Histogram Density Estimation

A simple example of a non-parametric density estimator is the histogram density estimator. We start by explaining how the one-dimensional histogram estimator works. Assume f has its support on the interval $S = [a, b]$, let k be the number of bins (an integer) and let h be the binwidth defined as $h = (b - a)/k$. Then S is partitioned into bins according to:

$$B_1 = [a, a + h), B_2 = [a + h, a + 2h), ..., B_k = [a + (k - 1)h, a + kh].$$

Now, let $Y_i \in \mathbb{N}$ be the number of observations in B_i. Let $p_i = \int_{B_i} f(u)du$, thus p_i represents the probability of an observation in B_i. We can estimate p_i with the relative frequency $\hat{p}_i = Y_i/n$

where n is the total number of observations. The histogram estimator is defined as:

$$\hat{f}(x) = \sum_{i=1}^{k} \frac{\hat{p}_i}{h} I_{B_i}(x), \tag{2.6}$$

Where $I_A(x)$ represents the indicator function, defined as:

$$I_A(x) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{if } x \notin A. \end{cases}$$

If we denote the centre of a bin m_i, all observations in $B_i = [m_i - \frac{h}{2}, m_i + \frac{h}{2})$ are assigned the estimate $\hat{f}(m_i)$. The probability for a random variable X to be assigned to a bin B_i is:

$$P(X \in B_i) = \int_{B_i} f(u)du \approx f(m_i) \cdot h.$$

The estimator is asymptotically unbiased in the centres of bins (Wasserman, 2010), that is an estimator $\hat{f}_n(x)$ is asymptotically unbiased if:

$$\lim_{n\to\infty} E[\hat{f}_n(x)] = f(x).$$

The histogram naturally extends to higher dimensions. Without loss of generality we can consider f to have its support on $S_d = [a, b]^d$ in d dimensions. Here S_d is divided into d-dimensional cubes $B_1, ..., B_N$, where each d-dimensional cube has sides of length h. Hence the number of cubes is $N = ((b-a)/h)^d$. Than in d dimensions the histogram estimator is defined as:

$$\hat{f}(\boldsymbol{x}) = \sum_{j=1}^{N} \frac{\hat{p}_j}{h^d} I_{B_i}(\boldsymbol{x}). \tag{2.7}$$

If we allow different binwidths h_l, that is we allow the sides of the d-dimensional cubes $B_1, ..., B_M$ to be of different length. Every cube side is now divided into $k_l = (b-a)/h_l$ bins and $M = \prod_{l=1}^{d} k_l$ is the number of cubes. We now obtain a slightly different histogram estimator namely:

$$\hat{f}(\boldsymbol{x}) = \sum_{l=1}^{M} \frac{\hat{p}_l}{\prod_{l=1}^{d} h_l} I_{B_l}(\boldsymbol{x}). \tag{2.8}$$

The major weakness with histogram density estimation is that the choice of binwidth is imperative. A bad choice of binwidth will result in a poor estimation of the density function. Some of the rules for choosing number of bins (which is the same as to choose the bandwidth) are: Sturge's formula $k = \log_2(n) + 1$ and Scott's rule $h = 7\hat{\sigma}/2n^{1/3}$ (Scott, 1992, pp. 47-55) where n is the number of observations and $\hat{\sigma}$ is the sample standard deviation. Sturge's formula is based on the fact that the binomial distribution can be used as a model of an optimally constructed histogram with scaled normal data (Scott, 1992, pp. 47-55). The explanation of Scott's rule would require us to introduce statistical concepts that are beyond the scope of this thesis, for an explanation of Scott's rule we refer to Scott (1992, pp. 49-55). Another weakness is that histograms are not smooth. With this in mind we introduce kernel density estimators in the next section.

2.2.2 Kernel Density Estimation

Kernel density estimation (KDE) can be viewed as a generalization of histogram density estimation. Unlike the histogram estimators, most kernel estimators are smooth and tend to converge to the true density faster. Generally kernel density estimators will assign more weight to observations close to the point of estimation.

We start by defining the kernel concept. A kernel is a weighting function. When we use the word kernel we refer to a smooth function $K(\boldsymbol{x})$ where $\boldsymbol{x} = (x_1, ..., x_d)$ that satisfy the following conditions (Wasserman, 2010):

$$K(\boldsymbol{x}) \geq 0, \quad \int_{\mathbb{R}^d} K(\boldsymbol{x})d\boldsymbol{x} = 1 \quad \text{and} \quad \int_{\mathbb{R}^d} \boldsymbol{x}K(\boldsymbol{x})d\boldsymbol{x} = \boldsymbol{0},$$

which corresponds to a density function with centre of mass at the origin.

Examples of kernel functions in one-dimension are:

$$\begin{aligned}
\text{Uniform kernel}: \quad & K(x) = \frac{1}{2} I_{\{|x|\leq 1\}}(x) \\
\text{Triangular kernel}: \quad & K(x) = (1 - |x|) I_{\{|x|\leq 1\}}(x) \\
\text{Gaussian kernel}: \quad & K(x) = \frac{1}{\sqrt{2\pi}} \exp\left(\frac{-x^2}{2}\right).
\end{aligned}$$

Different kernels will result in different weighting. The uniform kernel implies all observations x_i within a closed ball of radius one to x are given the same weight. Using a smooth kernel like the Gaussian will result in locally weighted observations.

We now introduce the kernel density estimator, we start by defining the one-dimensional kernel density estimator. Let $\boldsymbol{x} = (x_1, ..., x_n)$ be an independent and identically distributed sample drawn from an unknown continuous pdf $f(x)$. Given a kernel K and a positive number h (which we refer to as the bandwidth) the kernel density estimation of $f(x)$ at a point x is defined as:

$$\hat{f}(x) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{h} K\left(\frac{x - x_i}{h}\right). \tag{2.9}$$

In Figure 2.3 (a) we can see a one-dimensional histogram density estimator and in Figure 2.3 (b) a one-dimensional kernel density estimator.

We now move on to multidimensional kernel density estimation. In d dimensions, consider a d-dimensional independent and identically distributed random sample $\boldsymbol{x_1}, ..., \boldsymbol{x_n}$ from an unknown density function $f(\boldsymbol{x})$, where $\boldsymbol{x}_i = (x_{i1}, ..., x_{id})^T$. Following the notation of Wand and Jones (1993), the kernel density estimator of $f(\boldsymbol{x})$ at $\boldsymbol{x} = (x_1, ..., x_d)^T$ is defined as:

$$\hat{f}(\boldsymbol{x}; \boldsymbol{H}) = \frac{1}{n} \sum_{i=1}^{n} K_{\boldsymbol{H}}(\boldsymbol{x} - \boldsymbol{x}_i). \tag{2.10}$$

Here $K(\boldsymbol{x})$ is a d-dimensional kernel function, $K_{\boldsymbol{H}}(\boldsymbol{x}) = |\boldsymbol{H}|^{-1/2} K(\boldsymbol{H}^{-1/2}\boldsymbol{x})$ and $\boldsymbol{H}$ is a $d \times d$ matrix called the bandwidth matrix where $|\boldsymbol{H}|$ denote the determinant of $\boldsymbol{H}$. The bandwidth

matrix contains the smoothing parameters. These values are important for the performance of the estimator. The bandwidth matrix $\boldsymbol{H}$ is a real-valued, positive, symmetric ($\boldsymbol{H}_{ij} = \boldsymbol{H}_{ji}$) and semi-positive definite matrix, meaning that for any real valued non-zero vector $\boldsymbol{b}$, $\boldsymbol{b}^T\boldsymbol{H}\boldsymbol{b} \geq 0$. The bandwidth matrix is defined as:

$$\boldsymbol{H} = \begin{pmatrix} h_{11}^2 & \cdots & h_{1d} \\ \vdots & \ddots & \vdots \\ h_{d1} & \cdots & h_{dd}^2 \end{pmatrix},$$

where we require that:

$$h_{1,1} \cdot h_{2,2} > |h_{1,2}|,\ h_{1,1} \cdot h_{3,3} > |h_{1,3}|\ ,...,\ h_{d-1,d-1} \cdot h_{d,d} > |h_{d-1,d}| \tag{2.11}$$

(Wand and Jones, 1993). This requirement (along with the requirement on $\boldsymbol{H}$ introduced earlier) ensures that $H^{-1/2}$ is real-valued and positive. Please notice the resemblance between $\boldsymbol{H}$ and a covariance matrix.

For a symmetric matrix $\boldsymbol{A}$ the operation $\boldsymbol{A}^{-1/2}$ can be calculated by $\boldsymbol{A}^{-1/2} = \boldsymbol{P}\boldsymbol{D}^{-1/2}\boldsymbol{P}^{-1}$ (Anton and Rorres, 2005, p. 377). Here $\boldsymbol{P}$ is a matrix made up of the orthonormal eigenvectors $\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_d$ of $\boldsymbol{A}$ so that:

$$\boldsymbol{P} = \begin{pmatrix} v_{11} & \cdots & v_{1d} \\ \vdots & \ddots & \vdots \\ v_{d1} & \cdots & v_{dd} \end{pmatrix},$$

and $\boldsymbol{D}$ is calculated by $\boldsymbol{D} = \boldsymbol{P}^{-1}\boldsymbol{A}\boldsymbol{P}$. This means that $\boldsymbol{D}$ is a diagonal matrix:

$$\boldsymbol{D} = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_d \end{pmatrix}.$$

$\boldsymbol{D}^{-1/2}$ will also be real valued, and is defined as:

$$\boldsymbol{D}^{-1/2} = \begin{pmatrix} \frac{1}{\sqrt{\lambda_1}} & 0 & \cdots & 0 \\ 0 & \frac{1}{\sqrt{\lambda_2}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{\sqrt{\lambda_d}} \end{pmatrix},$$

hence $\boldsymbol{A}^{-1/2} = \boldsymbol{P}\boldsymbol{D}^{-1/2}\boldsymbol{P}^{-1}$.

As far as we are concerned $\boldsymbol{H}$ is a positive, semi-positive definite and satisfies Eq. (2.11) so $\boldsymbol{H}^{-1/2}$ is guarantied to be real-valued and positive. Since $1/\sqrt{\lambda_1}, ..., 1/\sqrt{\lambda_d}$ are eigenvalues to $H^{-1/2}$ we are guaranteed that this is the only positive-semidefinite solution.

The Gaussian Kernel

Now that we have introduced kernel density estimators we introduce the kernel density estimator that we use throughout this thesis. In this thesis we use the bivariate Gaussian kernel, in other words we use the bivariate normal distribution as our kernel, recall Eq.(2.2). Since the data are two-dimensional, $\boldsymbol{x} = (x_1, x_2)^T$ and $\boldsymbol{H}$ is a 2×2 matrix, we obtain:

$$K_{\boldsymbol{H}}(\boldsymbol{x}) = \frac{1}{2\pi|\boldsymbol{H}|^{1/2}} \exp\left(-\frac{1}{2}\boldsymbol{x}^T\boldsymbol{H}^{-1}\boldsymbol{x}\right) \tag{2.12}$$

Using Eq.(2.10) with Eq.(2.12) we obtain the following density estimator:

$$\hat{f}(x_1, x_2) = \frac{1}{2\pi n|\boldsymbol{H}|^{1/2}} \sum_{i=1}^{n} \exp Z_i, \tag{2.13}$$

where

$$Z_i = -\frac{1}{2}\left(\left(\frac{x_1 - x_{i1}}{h_{11}}\right)^2 - \left(\frac{2h_{12}}{h_{11}^2 h_{22}^2}(x_1 - x_{i1})(x_2 - x_{i2})\right) + \left(\frac{x_2 - x_{i2}}{h_{22}}\right)^2\right). \tag{2.14}$$

Choosing a good bandwidth is of great importance, a small bandwidth will result in a rough estimate while a large bandwidth gives a smoother estimate (Ramsy and Silwerman, 1997, pp. 131-132). How to find the theoretical optimal bandwidth is beyond the scope of this thesis. For a discussion on this topic we refer to Wasserman (2010, pp. 133-135). In Chapter 3.2 we describe how the bandwidth is chosen for this thesis. The drawback with non-parametric methods is that it often takes quite a lot of time to estimate pdfs. If we already know something about the distribution from which the data comes from, we can use this fact to reduce computational times. With this in mind we introduce parametric density estimation in the next chapter.

2.3 Parametric Density Estimation

Let $X_1, X_2, ..., X_n$ be independent and identically distributed (i.i.d.) continuous random variables with probability density function (pdf) f. Parametric density estimation assumes a given form for the density function (in our case beta-distribution) and the parameters of the function are optimized to fit the data.

2.3.1 Maximum Likelihood Estimation

Maximum likelihood estimation is probably the best known, most widely used and most important method for estimation of parameters in statistical models. Assume we have a sample $\boldsymbol{x} = (x_1, ..., x_n)$ of i.i.d. observations from an unknown distribution with pdf $f(\boldsymbol{x}|\theta) = f(x_1|\theta) \cdot ... \cdot f(x_n|\theta)$. If we fix $\boldsymbol{x}$ and let θ be a variable the likelihood function is expressed as:

$$L(\theta|\boldsymbol{x}) = \prod_{i=1}^{n} f(x_i|\theta). \tag{2.15}$$

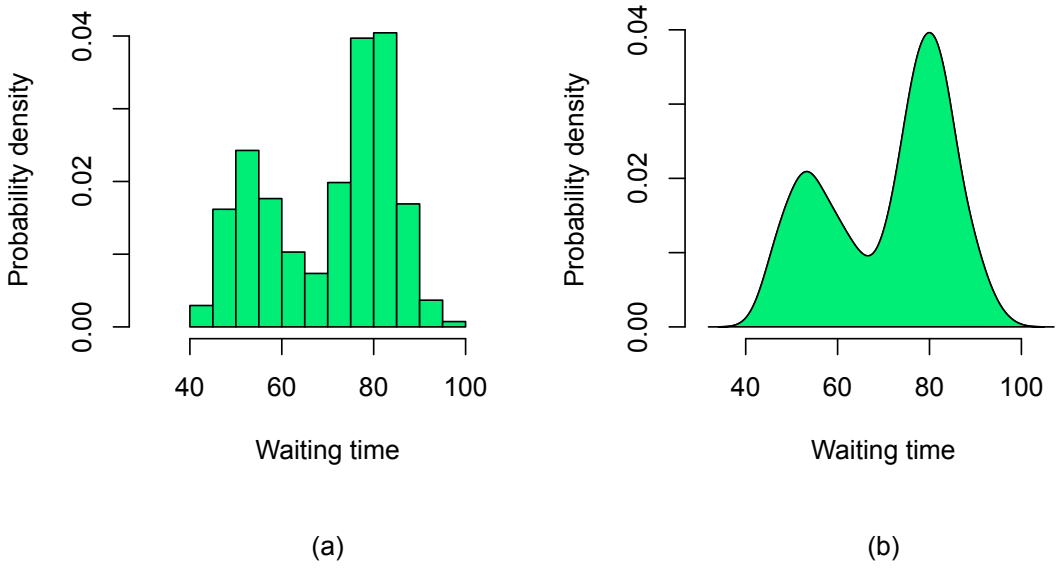

Figure 2.3: In Figure (a) we see a histogram density estimation with 12 bins and in Figure (b) a kernel density estimation based on a Gaussian kernel with bandwidth $h = 3$. The density estimation is based on waiting time between eruptions for the Old Faithful geyser in Yellowstone National Park, Wyoming, USA A.Azzalini and A.W.Bowman (1990)

The maximum likelihood estimator is the value $\hat{\theta}$ that maximises $L(\theta|\boldsymbol{x})$ (Garthwaite et al., 1995, p. 41). A possible (and popular) alternative formulation builds on the fact that the logarithm is an increasing function. To maximise the log-likelihood:

$$\log(L(\theta|\boldsymbol{x})) = \log\left(\prod_{i=1}^{n} f(x_i|\theta)\right) = \sum_{i=1}^{n} \log(f(x_i|\theta)) \tag{2.16}$$

is equivalent to maximise Eq. (2.15).

2.4 Discriminant Analysis

The purpose of Discriminant Analysis (DA) is to allocate attributes (an attribute is a measurable quantity, e.g. length, weight, density, etc.) from an unknown class (a discrete category such as gender, species, substances, etc.) to one of the in advance known k classes (Pawlowsky-Glahn and Buccianti, 2011, pp. 65-66). There are two types of DA: Bayesian DA and Canonical DA, also known as Fisher's DA. In this thesis we use Bayesian DA.

Since Bayesian DA is based on Bayes' theorem, we start by introducing Bayes' theorem. Bayes' theorem is used to determine conditional probability, which is the probability for an outcome given another outcome. In the most simple form: for two events A and B, provided $P(B) > 0$, Bayes' theorem states:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}. \tag{2.17}$$

Generally, let $A_1, ..., A_n$ be n disjoint events. Further assuming that the events cover the event space, Bayes' theorem states:

$$P(A_i|B) = \frac{P(B|A_i)P(A_i)}{P(B)}, \tag{2.18}$$

where $P(B) = \sum_{j=1}^{n} P(B|A_j)P(A_j)$.

Assume we have a random sample of $\boldsymbol{X}$, consisting of n attributes. These attributes comes from a continuous pdf $f_{\boldsymbol{X}}(\boldsymbol{x})$. Further we assume that the distribution of $\boldsymbol{X}$ is dependent of the class $C = c$ where $C \in (1, ..., k)$ and k specifies the number of classes. That is given a class $C = c$ we get the pdf $f_{\boldsymbol{X}|C=c}(\boldsymbol{x})$. Since we use continuous pdfs here Eq. (2.18) will be somewhat different. Strictly speaking we get likelihoods rather than probabilities. We denote likelihoods by $L(.)$ and the following relation holds:

$$L(C = c|\boldsymbol{X} = x) = \frac{f_{\boldsymbol{X}|C=c}(\boldsymbol{x})P(C = c)}{f_{\boldsymbol{X}}(\boldsymbol{x})}. \tag{2.19}$$

This implies that we can calculate how likely it is that we have an observation from the class $C = c$ given a set of attributes $\boldsymbol{X}$. Of course in reality $f_{\boldsymbol{X}|C=c}(\boldsymbol{x})$ is seldom known and often has to be estimated from data.

Now we have explained the foundation of Bayesian DA and proceed to explain the different methods used for DA in this thesis.

2.4.1 Naive Bayesian Classifier

The naive Bayesian classifier is the most important classifier for this thesis. It is used for three methods. The naive Bayesian classifier is an efficient and non-complicated method for classification.

When we refer to a classifier we refer to a function that maps a set of attributes to a number of classes. The Naive part of the name comes from the fact that we assume conditional independence between different attributes. Conditional independence between different attributes means that the presence of an attribute in a class is unrelated to any other attributes. Two events A and B are conditionally independent if:

$$P(A \cap B|D) = P(A|D)P(B|D).$$

In order to use Eq. (2.19) we need to know $f_{\boldsymbol{X}|C=c}(\boldsymbol{x})$, $f(x)$ and $P(C = c)$.

Because of the assumption that all the attributes are conditionally independent we get:

$$f_{\boldsymbol{X}=x|C=c}(\boldsymbol{x}) = \prod_{i=1}^{n} f_{X_i|C=c}(x_i). \tag{2.20}$$

Combining Eq. (2.19) and Eq. (2.20) we get:

$$L(C = c|\boldsymbol{X} = \boldsymbol{x}) = \frac{P(C = c)\prod_{i=1}^{n} f_{X_i|C=c}(x_i)}{f_{\boldsymbol{X}}(\boldsymbol{x})}. \tag{2.21}$$

Finally, we assign the attribute $\boldsymbol{x}$ to the class c which maximises Eq. (2.21).

In Eq. (2.21) $f_{\boldsymbol{X}}(\boldsymbol{x})$ can be ignored since it is independent of class. Neglecting $f_{\boldsymbol{X}}(\boldsymbol{x})$ we obtain:

$$P(C = c) \prod_{i=1}^{n} f_{X_i|C=c}(x_i). \tag{2.22}$$

Maximising this product can sometimes produce a product that is too small or too large to be used in practice; to avoid this problem we use the fact that the logarithm of a product is the same as the sum of the logarithm of the factors. Since the logarithm is an increasing function, maximising

$$\log\left(P(C = c) \prod_{i=1}^{n} f_{X_i|C=c}(x_i)\right) \tag{2.23}$$

is the same as maximising Eq. (2.22) and we obtain the following classifier:

$$\arg\max_{c} \left(\log\left(P(C = c)\right) + \sum_{i=1}^{n} \log\left(f_{X_i|C=c}(x_i)\right)\right). \tag{2.24}$$

This is the classifier we use when referring to naive Bayesian classifier in this thesis. Estimating the pdfs needed for classification can be both difficult and time consuming. If we can assume that the data comes from a multivariate normal distribution we can use linear and quadratic discriminant analysis instead.

2.4.2 Linear and Quadratic Discriminant Analysis

The reference methods we use to compare our results with in this thesis are based on linear and quadratic discriminant analysis. It is therefore important to understand how linear and quadratic discriminant analyses are performed.

The difference between discriminant analysis based on the naive Bayesian classifier and linear and quadratic discriminant analysis is that for linear and quadratic discriminant analysis we do not need to assume that attributes are conditionally independent of each other, instead we assume

$$f_{\boldsymbol{X}|C=c}(\boldsymbol{x}) \sim N(\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c),$$

for these methods. That is, we assume the attributes from the k classes comes from multivariate normal distributions when we use linear and quadratic discriminant analysis. We estimate the expected value $\boldsymbol{\mu}_c$ and the covariance matrix $\boldsymbol{\Sigma}_c$ for every class $C = c$ from some training data, thus we obtain the distributions of the k classes.

For a fixed $\boldsymbol{x}$, the conditional likelihoods are proportional to:

$$P(C = c) f_{\boldsymbol{X}|C=c}(\boldsymbol{x}), \tag{2.25}$$

and the classification problem reduces to a maximisation over c.

Notice that this is Eq.(2.19) where $f_{\boldsymbol{X}}(\boldsymbol{x})$ has been omitted. Because of the assumption that the data from the k classes comes from multivariate normal distributions we have:

$$P(C = c) f_{\boldsymbol{X}|C=c}(\boldsymbol{x}) = P(C = c)(2\pi)^{-\frac{n}{2}} |\boldsymbol{\Sigma}_c|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_c)^T \, \boldsymbol{\Sigma}_c^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_c)\right\}. \tag{2.26}$$

We want to maximise this expression with respect to c, taking the logarithm of Eq. (2.26) gives:

$$\begin{aligned}&\log\left(P(C=c)f_{\boldsymbol{X}|C=c}(\boldsymbol{x})\right) = \\ &\log\left(P(C=c)(2\pi)^{-\frac{n}{2}}|\boldsymbol{\Sigma}_c|^{-\frac{1}{2}}\exp\left\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_c)^T\ \boldsymbol{\Sigma}_c^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_c)\right\}\right).\end{aligned} \tag{2.27}$$

The factor $(2\pi)^{-\frac{n}{2}}$ can be ignored in Eq. (2.27) since it is independent of c. Ignoring this factor, the right hand side of Eq. (2.27) can be rewritten as:

$$-\frac{1}{2}\left((\boldsymbol{x}-\boldsymbol{\mu}_c)^T\ \boldsymbol{\Sigma}_c^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_c)+\log|\Sigma_c|-2\log\left(P(C=c)\right)\right). \tag{2.28}$$

Our goal is to maximise this expression. In other words given an attribute we want to find the class $C=c$ for which Eq. (2.28) is as large as possible. Formally we write this as:

$$\arg\max_c\left(-\frac{1}{2}\left((\boldsymbol{x}-\boldsymbol{\mu}_c)^T\ \boldsymbol{\Sigma}_c^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_c)+\log|\Sigma_c|-2\log\left(P(C=c)\right)\right)\right). \tag{2.29}$$

Now that we have introduced the basic concepts we can move on to explain the difference between linear and quadratic discriminant analysis.

Linear Discriminant Analysis

In linear discriminant analysis (LDA) we assume the covariance $\boldsymbol{\Sigma}_c$ of attributes is the same for all classes $C=c$ (Pawlowsky-Glahn and Buccianti, 2011, pp. 65-66). If it is not reasonable to assume that $\boldsymbol{\Sigma}_c$ is equal for all classes, we should use quadratic DA instead.

Letting $\Sigma_c \equiv \Sigma$ and neglecting the constant term $\frac{1}{2}\log|\Sigma|$ which is independent of the class, Eq. (2.28) reduces to:

$$\begin{aligned}\log\left(P(C=c)\right)-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_c)^T\ \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_c) = \\ \log(P(C=c))-\frac{1}{2}\boldsymbol{x}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{x}-\frac{1}{2}{\boldsymbol{\mu}_c}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_c+{\boldsymbol{\mu}_c}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{x}.\end{aligned} \tag{2.30}$$

Remember that $\boldsymbol{x}$ is fixed, thus $\frac{1}{2}\boldsymbol{x}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{x}$ does not depend on c and can therefore be neglected, since we want to maximise overe c. The remaining expression is linear in $\boldsymbol{x}$ and we are left with the expression:

$$L_c(\boldsymbol{x})=\log\left(P(C=c)\right)-\frac{1}{2}{\boldsymbol{\mu}_c}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_c+{\boldsymbol{\mu}_c}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{x}, \tag{2.31}$$

$L_c(\boldsymbol{x})$ is called the discriminant function.

Given a fixed $\boldsymbol{x}$, $L_c(\boldsymbol{x})$ is maximised over c. That is we find the class for which Eq. (2.31) is maximised. Formally we write:

$$\arg\max_c\left(L_c\right), \tag{2.32}$$

This is called linear discriminant analysis since $L_c(\boldsymbol{x})$ is linear.

Quadratic Discriminant Analysis

Quadratic discriminant analysis (QDA) is very similar to LDA except that we assume the covariance matrix $\mathbf{\Sigma}_c$ can be different for each class and so, the covariance matrix $\mathbf{\Sigma}_c$ is estimated separately for each class.

Since the covariance matrix $\mathbf{\Sigma}_c$ is not equal for all classes it is not possible to simplify Eq. (2.28) any further. Thus the quadratic discriminant function for a fixed attribute $\boldsymbol{x}$ becomes:

$$Q_c(\boldsymbol{x}) = -\frac{1}{2}\left((\boldsymbol{x}-\boldsymbol{\mu}_c)^T\ \mathbf{\Sigma}_c^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_c) + \log|\Sigma_c| - 2\log\left(P(C=c)\right)\right). \tag{2.33}$$

Given a fixed $\boldsymbol{x}$, $Q_c(\boldsymbol{x})$ is maximised over c. That is we find the class for which Eq. (2.33) is maximised. Following the notation from previous sections, we express this as:

$$\arg\max_c\left(Q_c\right). \tag{2.34}$$

Since $Q_c(\boldsymbol{x})$ is quadratic this is called quadratic discriminant analysis.

2.5 Cluster Analysis

We use cluster analysis for one of or methods in this thesis, therefore it is important that the reader has a basic understanding of the concept.

The purpose of cluster analysis is to cluster objects into groups (clusters) so similar objects are put in the same cluster. In other words, in cluster analysis our aim is to sort different groups of data in a way that objects in the same cluster have a maximal degree of similarity and objects in different clusters have a minimal degree of similarity. Cluster analysis can discover structures in the data without explaining these structures. Cluster analysis is not one specific method. There are several types of clusters analysis such as hierarchical clustering, non-hierarchical clustering and model based clustering. In this thesis we will use k-means clustering which is a non-hierarchical clustering method (Johnson and Wichern, 2007, p. 773).

2.5.1 k-means Clustering

The basic idea with k-means clustering is: given a desired number of k clusters, observations should be assigned to clusters so the means across clusters are as different from each other as possible. Let $\boldsymbol{x}_1, ..., \boldsymbol{x}_n$ be a set of observations where each observation is a d dimensional vector. We partition the n observations into k sets $\boldsymbol{S} = \{S_1, ..., S_k\}$, where $k \leq n$, so that the within-cluster sum of squares is minimized. That is, we seek:

$$\arg\min_{\boldsymbol{S}} \sum_{i=1}^{k} \sum_{x_j \in \boldsymbol{S_i}} ||\boldsymbol{x_j} - \boldsymbol{\mu_i}||^2, \tag{2.35}$$

where $\boldsymbol{\mu_i}$ is the mean of observations in $\boldsymbol{S_i}$ (Frahling and Sohler, 2008, pp. 605-625) and $||.||$ denotes the Euclidean norm. In Figure 2.4 we can see an example of a k-mean clustering.

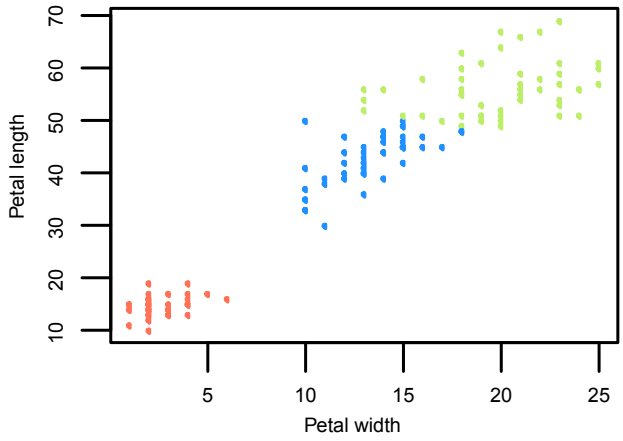

Figure 2.4: Result of k-means clustering with $k = 3$ using information about petal length and petal width from the Iris flower data set. Different colours represent different clusters. The data are known as Fishers Iris data and was published in Fisher (1936).

2.6 Cross-validation

Cross-validation is a statistical technique of evaluating models by dividing the data into two segments: one to train the model and the other to validate the model. Cross-validation is used to assess how the results of a statistical model will generalize to an independent data set. There are different types of cross-validation, perhaps two of the most common are k-fold cross-validation and leave-one-out cross-validation (leave-one-out cross-validation is a special case of k-fold cross-validation where k is the number of observations).

2.6.1 k-fold Cross-validation

When testing a model with k-fold cross-validation the data are divided into k roughly equal sized data sets. The model is tested on one of the k data sets, this is used as the validation data for testing our model. The remaining $k - 1$ data sets are used as training data. This process is repeated k times, with each of the k data sets used one time. In Figure 2.5 we can see an illustration of a 3-fold cross validation.

To facilitate understanding of cross validation we present an algorithm for k-fold cross validation below.
Step 1. Partition the data into k approximately equal sized data sets.
Step 2. Hold out on one data set and train the model on the remaining $k - 1$ datasets.
Step 3. Validate the model on the dataset not used to train the model and calculate the measure of interest.
Step 4. Repeat step 2-3 until all data sets have been used as validation sets.
Step 5. Calculate the mean value of the iterations from step 3.

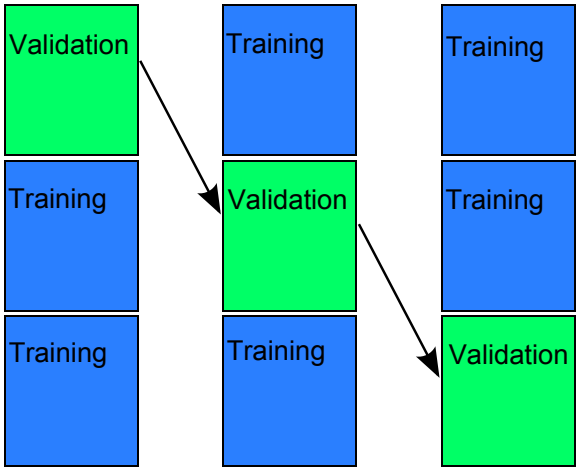

Figure 2.5: Illustration of k-fold cross validation with $k = 3$. The boxes illustrate the 3 datasets and the green box represents the validation data. The blue boxes represent the training data.

Chapter 3

Material and Methods

In this chapter we present the data we use, the transformations of the data and the different methods of classification. We start by explaining the properties of the data and how it was obtained. We then proceed to explain the transformations of our data (the transformed data is called attributes). In the last part of this chapter we describe our classification methods, every method has its own section.

3.1 The Data

The data was collected on behalf of the Swedish University of Agricultural Sciences in 2010 between the 29th of August to the 9th of September in Remingstorp, Sweden (lat. $58°27'48''$ N, long.$13°37'50''$ E), situated between Skara and Skövde in Västergötaland. The data comes from 25 circular field plots; each field plot has a diameter of 20 meters. The data set contains 1432 Norway spruces, 229 Scots pines and 170 deciduous trees (this is field data). The field data includes the position of each segment (tree).

Our data set (ALS data) contains 30 point returns per m^2. We have a segmented data set with X, Y and Z coordinates for each laser return, where X represent the longitude, Y represent the latitude and Z represents the elevation (height above a local minimum, that is height above ground) . Even though some segments may contain more than one tree we will assume that all segments represent only a single tree. The segments have been provided with information about the tree species found in that segment. This information is used to evaluate our methods.

In Figure 3.1, we can see a conceptual picture of how the segmentation and classification is performed.

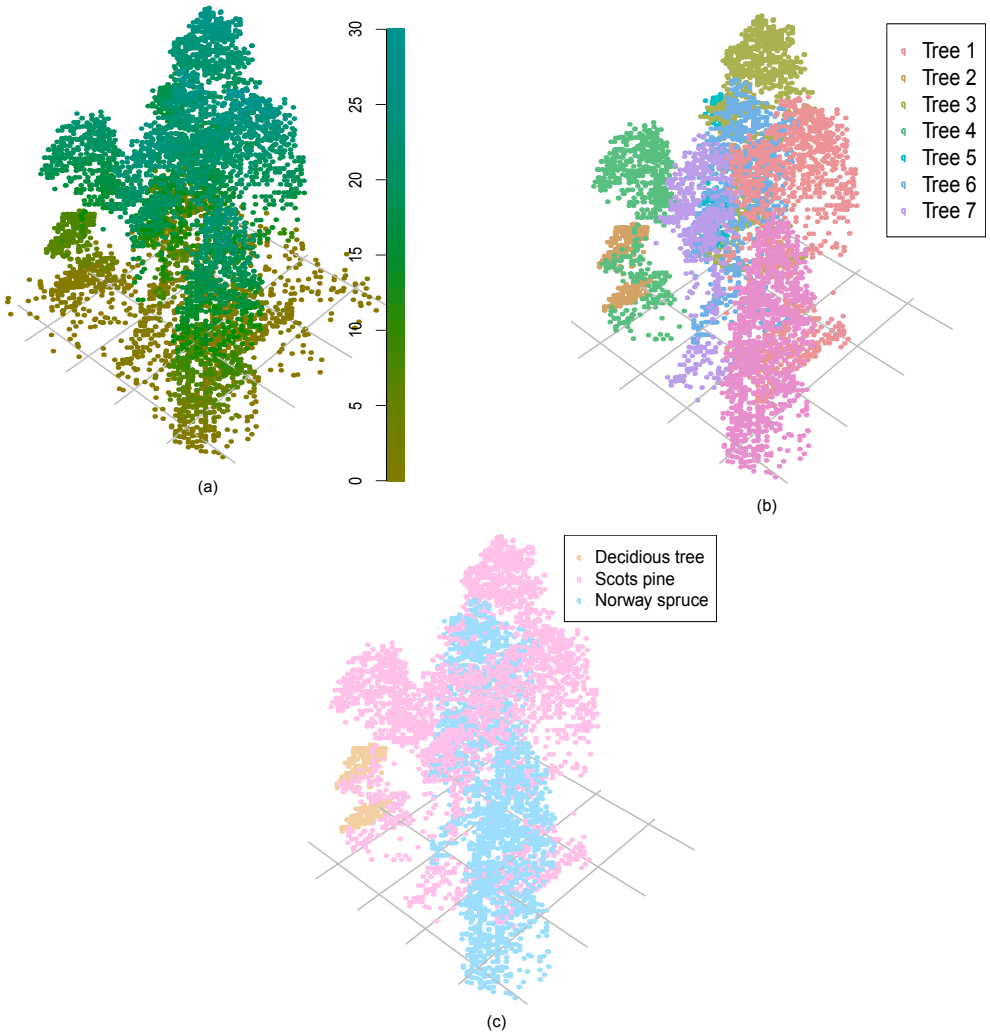

Figure 3.1: Conceptual pictures of segmentation of the data. Figure (a) raw data, no segmentation has been performed; all trees have the same colour. Points are coloured according to their height, points close to the ground are brown and points far from the ground are green. A legend with the colour scale and height in meters is included. Points that do not belong to any of the trees are kept in this data. Figure (b) segmented data, points that do not belong to any of the trees are removed. All trees are marked with a unique colour. Figure (c) classified trees, trees from the same species are marked with the same colour where yellow represent deciduous trees, light blue Norway spruces and pink Scots pines.

We have a lot of ground hits in our data, that is, laser beams that have been reflected of the ground and not of a tree. We want to find the crown of the tree and remove as much vegetation under the tree crown as possible. A solution to these problems is presented in Holmgren et al. (2002), we will implement this solution. We start by defining all laser points below two meters as ground hits, that is all points below two meters are removed. This means that we will only investigate points above two meters. Now, the issue is to find the crown base height, the crown base height is calculated using 0.5 m height layers. Each layer which contains less than one percent of the total number of non-ground laser points within the segment is set to zero, the

rest to one. To reduce the influence of laser points from low vegetation and neighbour trees, a one-dimensional median filter with a sliding window of size 9 is applied on the array of height layers. The crown base is set as the distance from the ground to the lowest laser data point above the highest 0-layer. We referee to this data as filtered data.

3.1.1 Transformations

We will need to derive the following attributes in order to apply our methods: the stem position, the height of the tree, the relative height of the tree and the standardized distance to the stem. In order to calculate the stem position we follow Holmgren and Persson (2004): The stem position is taken as the X, Y-value of the point with the highest Z-value. The height of the tree is taken as the highest Z-value, which we will denote $Z_{\max}$. The distance to the stem for a point X_p, Y_p is calculated according to the Euclidean distance:

$$d_s = \sqrt{(X_p - X_s)^2 + (Y_p - Y_s)^2},$$

where d_s represents the distance to the stem and X_s and Y_s represents the position of the stem. The relative height or altitude a (this notation is used to not confuse this variable with the bandwidth h) at a certain X, Y, Z point is determined by dividing the measured height Z by the height of the tree, thus:

$$a = \frac{Z}{Z_{\max}}.$$

The standardized distance to the stem d_{sr} is calculated according to:

$$d_{sr} = \frac{d_s}{Z_{\max}}.$$

We chose to use the standardized distance to the stem rather than to use the relative distance to the stem (which would be defined as $r = d_{sr}/d_{\max}$). This is because we lose some information about the shape by using the relative distance to the stem; differences between wide and slim trees are lost. For example a tall tree with slim crown would look like a tall tree with a wide crown, by using the standardized distance where these differences are kept.

The variables we will use to predict which tree species a set of points comes from are: relative height a and the squared standardized distance to stem $d = d_{sr}^2$. The standardized distance to stem is squared since using only the standardized distance to stem will result in a measure that depends on the distance to the stem (the crowns radius). By using standardized distance to stem we would get more point returns for higher values of the standardized distance to stem. This is not desirable. By looking at Figure 3.2 we see that we get more points for a higher value on the x-axis in (a) than in (b). Thus we conclude that the squared standardized distance to stem is a better measure.

3.1.2 Height Classes

In order to obtain more correctly classified trees, we will divide the tress into height classes. The idea with dividing the trees into height classes is that trees of different sizes have different allometrics (also known as biological scaling, e.g the proportion of the tree crown). This could mean that the difference between smaller and larger tress within a species is greater than between trees from different species. If this is the case we are likely to obtain a higher level of classification

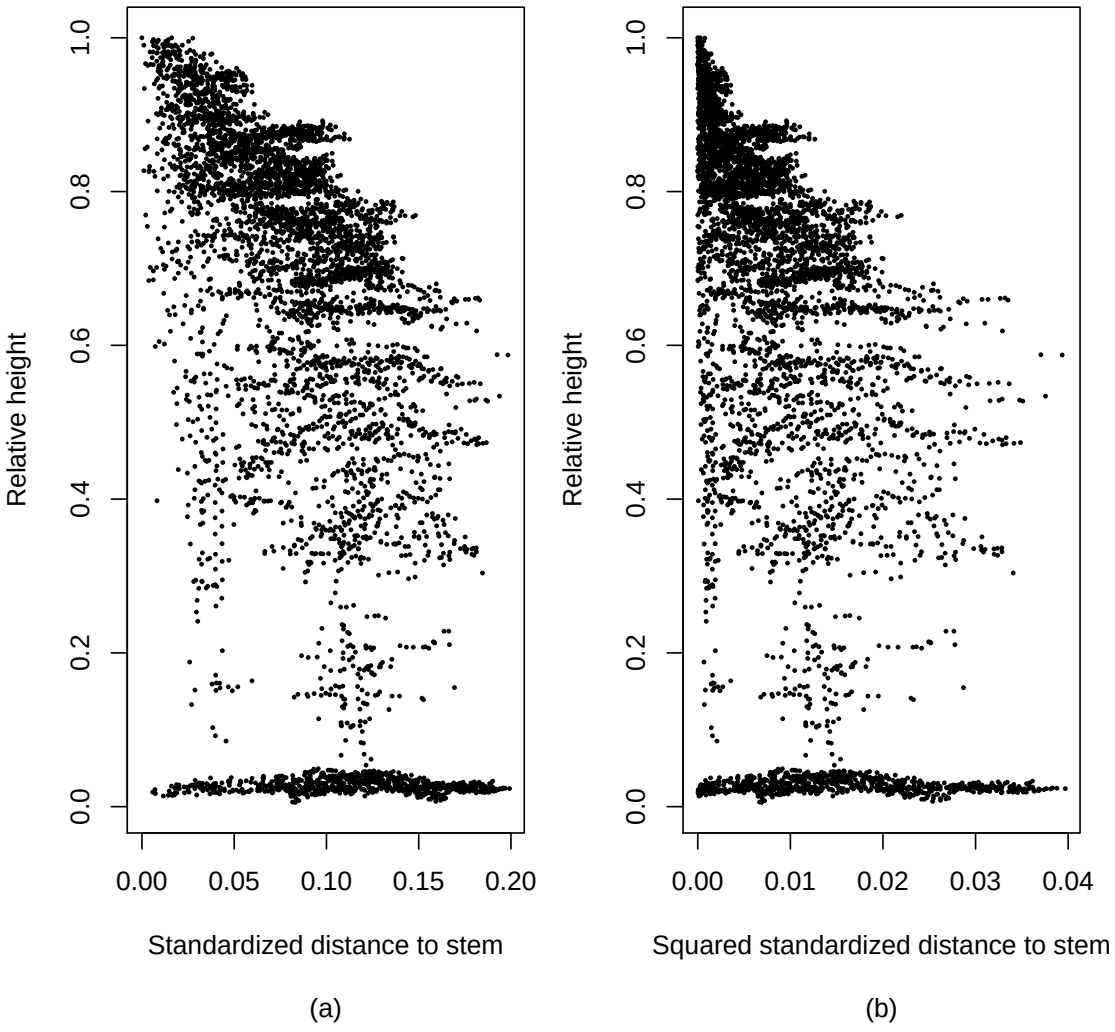

Figure 3.2: Plot of the different distance measures for one tree: standardized distance to stem $\sqrt{d}$ and squared standardized distance d. Standardized distance to stem in (a) and squared standardized distance to stem in (b)

if we compare trees with similar heights to each other. We will use the following partitions of heights:

g_1: one height class, no partition of trees are done.

g_2: two height classes, the trees are partitioned into two equally sized subsets according to the heights of the trees.

g_{31}: three height classes, the trees are partitioned into three equally sized subsets according to the heights of the trees.

g_{32}: three height classes. Trees with a height greater or equal to the 80th percentile in one class. Trees with a height greater or equal to the 20th percentile but less than the 80th percentile in one class and trees with a height less than the 20th percentile are put in a class.

g_4: four height classes, the trees are partitioned into four equally sized subsets according to the heights of the trees.

g_5: five height classes, the trees are partitioned into five equally sized subsets according to the heights of the trees.

3.1.3 Data Used For The Reference Method

For the reference method we use another type of information as input to classification then the transformations used for the methods we develop in this thesis. To be able to easily distinguish between the attributes we derive in this thesis and the attributes used for the reference method we henceforth refer to these latter attributes as features. In order to implement the method used in Holmgren and Persson (2004), a list of features were provided by Johan Holmgren at the Swedish University of Agricultural Sciences Umeå. Every segment has one value for each feature. The features we use are:

- Relative crown base height
- Relative 90th height percentile
- Relative 80th height percentile
- Relative 60th height percentile
- Fraction of returns two meter above the ground
- Fraction of returns on the convex hull
- Mean value of two parameters for a fitted parabolic surface.

We have excluded features that are based on laser intensity measures since this data have not been available for the other methods, thus it would not be fair to compare methods based on laser intensity measures with methods without these measures. For a description of how these features are derived see Holmgren and Persson (2004).

3.2 Methods for Classification

We start this chapter by presenting Figure 3.3. This is done to make the relations of the different classifiers more evident to the reader.

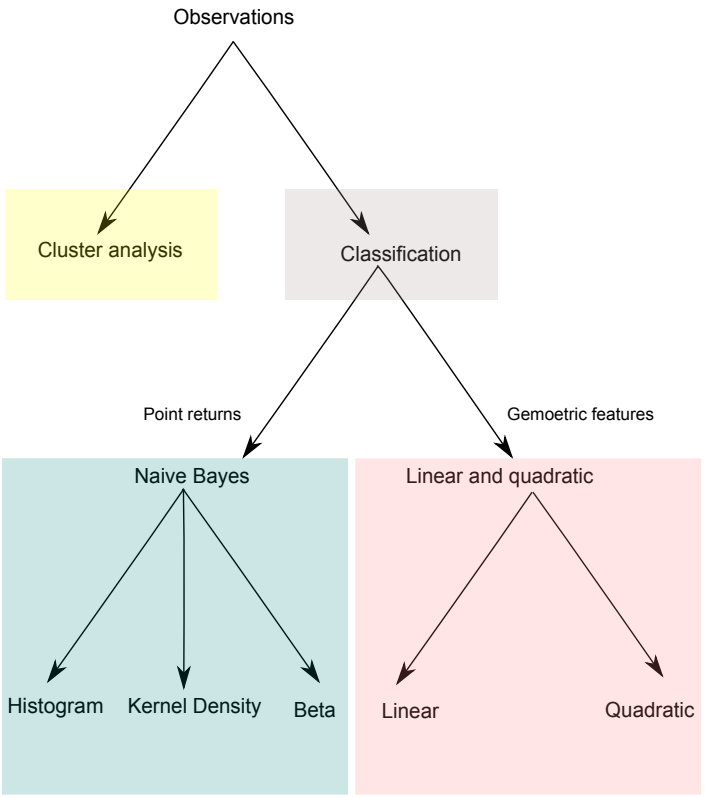

Figure 3.3: Overview of classifications methods and their relations. Different colours are used to highlight differences . Observe that the difference between features and attributes are explained in Chapter 3.1.

We use three methods for classification of trees and implement two methods by Holmgren and Persson (2004) as reference to our methods described in Chapter 3.2.3. The methods we use are: naive Bayesian approach based on the beta-distribution, naive Bayesian approach based on histogram-density estimation and naive Bayesian approach based on kernel-density estimation (KDE). How these methods relate to each other can bee seen in Figure 3.3. It should be clear to the reader that the data used for the reference method are different from the data used for the methods developed in this thesis. Hence we use different names to stress that different data types is used (features for the reference method and attributes for our methods).

We also investigate if it is possible to use cluster analysis based on the α and β parameters from the beta-distribution to discriminate between different species.

We use k-fold cross validation with $k = 5$ to tune our models and to validate our results, we describe this in detail in Chapter 3.2.4. Cross validation serves two purposes. On the training data set, the purpose is to optimise our methods. That is, we choose the parameters that result in the best classification on the training data set. On the validation data set cross validation is used to estimate the performance of our methods. It should be clear to the reader that we use two separate datasets, one training data set for choosing the parameters and one validating data set used to validate our methods. No trees in one set is included in the other, that is the have no trees in common.

3.2.1 Accuracy Measures

We measure the performance of our methods with user accuracy, producer accuracy and the overall accuracy. User accuracy is computed by dividing the number of correctly classified trees in each species by the total number of trees that were classified as that tree species. The producer accuracy is the result from dividing the number of correctly classified trees for each tree species by the number of true trees in that species. Overall accuracy is the sum of all correctly predicted trees divided by the sum of all trees.

3.2.2 Naive Bayesian Classifier

The naive Bayesian classifier is used in three different methods in this thesis. The difference between the methods is that the probability density functions (pdfs) are estimated in different ways. In this section we describe the general approach for this method.

Let $(d_1, a_1), (d_2, a_2), \ldots, (d_p, a_p)$ be p pairs of observations of squared standardized distance to the stem (d) and relative height (a) respectively from the tree we want to classify. Where given a species c we consider (d_i, a_i) to be a pair of conditionally independent observations. Recall Eq. 2.24, the tree is allocated to a species according to:

$$\arg\max_c \left(\log(P(C = c)) + \sum_{i=1}^{p} \log(\hat{f}_{D,A|C=c}(d_i, a_i)) \right), \tag{3.1}$$

where $P(C = c)$ represent the prior probability of $C = c$. Since we have three tree species we need to estimate a pdf for every tree species. The pdfs are estimated from a random sample where we know which tree species the sample comes from (in other words from a training data set) $\boldsymbol{d}_\mathrm{t} = (d_{\mathrm{t}1}, ..., d_{\mathrm{t}n})^T$ and $\boldsymbol{a}_\mathrm{t} = (a_{\mathrm{t}1}, ..., a_{\mathrm{t}n})^T$ representing observations of squared standardized distance to stem and relative height respectively. From the same data set we used to estimate the pdfs we estimate $P(C = c)$ with the proportion of trees from species c. We refer to the data with a subscripted t as training data. The procedure of finding and optimising the pdfs is described in section 3.2.4.

Naive Bayesian Classifier Based on Histogram Density Estimation

Since our data is two-dimensional, we want to estimate two-dimensional pdfs $f_{D,A|C=c}(d, a)$. That is, we want to use a two-dimensional histogram density estimator. Here we calculate $\hat{p}_i$ from the training data. Following the notation from Chapter 2.2.1, the histogram density estimation at a point d, a is defined as:

$$\hat{f}_{D,A|C=c}(d, a) = \sum_{i=1}^{N} \frac{\hat{p}_i}{h_1 h_2} I_{B_i}(d, a), \tag{3.2}$$

where $\hat{p}_i$ is estimated from training data.

By changing the number of bins $k_1 \times k_2$, we change the binwidths h_1 and h_2 and hence we obtain different estimates of $f_{D,A|C=c}(d, a)$. We use the same number of bins of bins $k_1 = k_2$ in order to reduce the number of iterations necessary to find a good setting for the number of bins. In Figure 3.4 we see how a two-dimensional histogram density estimator works, and in Figure 3.6 (b) (c)

and (d) we see how the two-dimensional histogram density estimator is affected by changing the number of bins. We will hereafter refer to this as the histogram-based classifier.

Naive Bayesian Classifier Based on Kernel Density Estimation.

As the data we are working with is two-dimensional, we will use a two-dimensional kernel density function. In Figure 3.6 (e),(f) and (g) we can see how a two-dimensional kernel density estimator is affected by different bandwidths. The pdfs are estimated using the two-dimensional Gaussian kernel introduced in Chapter 2.2.2. In order to simplify our calculations and reduce the number of possible combinations we set $h_{12} = h_{21} = 0$, make another simplification by setting $h = h_{11} = h_{22}$, by doing this we reduce the number of possible combinations and hence we reduce the number of iterations needed and we obtain the bandwidth matrix:

$$\boldsymbol{H} = \begin{pmatrix} h^2 & 0 \\ 0 & h^2 \end{pmatrix}.$$

We evaluate the density function at d and a. With the simplification of $\boldsymbol{H}$ the kernel density estimator in Eq. (2.14) reduces to:

$$\hat{f}_{D,A|C=c}(d,a) = \frac{1}{2\pi n h^2} \sum_{i=1}^{n} \exp\left(-\frac{1}{2}\left(\left(\frac{d-d_{\mathrm{t}i}}{h}\right)^2 + \left(\frac{a-a_{\mathrm{t}i}}{h}\right)^2\right)\right). \tag{3.3}$$

By changing the smoothing parameters h we obtain different estimates of $f_{D,A|C=c}(d,a)$. Hence forth we will refer to this as the KDE-based classifier.

Naive Bayesian Classifier Based on Beta-distribution

Since the beta-distribution is defined for one-dimension we cannot use pairs of observations. Instead we use the observations squared standardized distance to the stem (d) and the relative height (a) one at the time. That is, we get two cases where given a class c we consider $(d_1, ..., d_n)$ and $(a_1, ..., a_n)$ to conditionally independent observations. Thus we estimate different pdfs $f_{A|C=c}(a)$ and $f_{D|C=c}(d)$. In other words we get the two pdfs:

$$\hat{f}_{A|C=c}(a) = \frac{\Gamma(\hat{\alpha}+\hat{\beta})}{\Gamma(\hat{\alpha})\Gamma(\hat{\beta})} a^{\hat{\alpha}-1}(1-a)^{\hat{\beta}-1}, \tag{3.4}$$

calculated using the training data $\boldsymbol{a}_{\mathrm{t}} = (a_{\mathrm{t}1}, ..., a_{\mathrm{t}n})^T$ and

$$\tilde{f}_{D|C=c}(d) = \frac{\Gamma(\tilde{\alpha}+\tilde{\beta})}{\Gamma(\tilde{\alpha})\Gamma(\tilde{\beta})} d^{\tilde{\alpha}-1}(1-d)^{\tilde{\beta}-1}, \tag{3.5}$$

which is calculated using the training data $\boldsymbol{d}_{\mathrm{t}} = (d_{\mathrm{t}1}, ..., d_{\mathrm{t}n})^T$.

From now on we will refer to this method as the beta-based classifier.

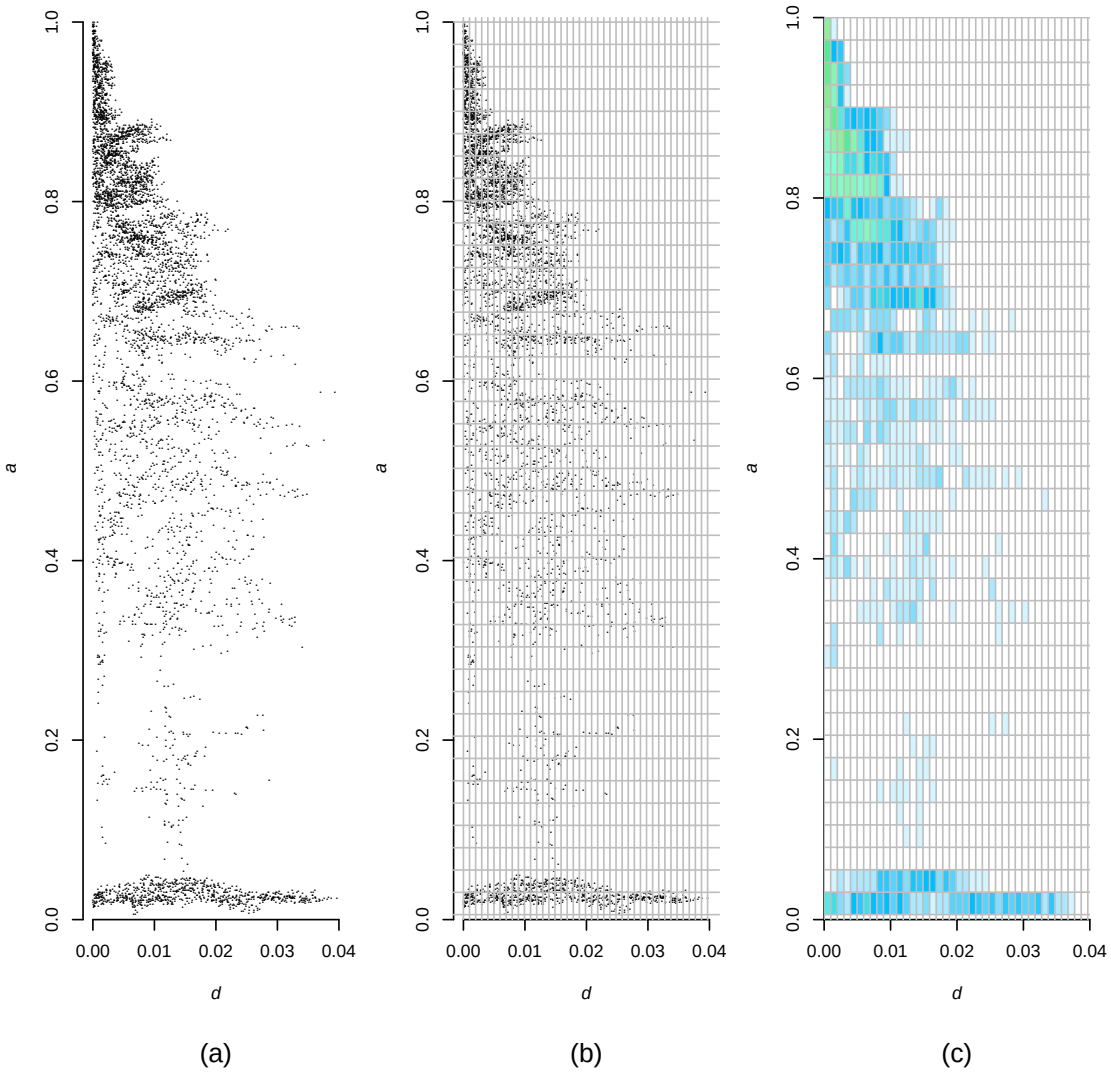

Figure 3.4: Description of how a two-dimensional histogram with 40×40 bins is created. (a): data for the histogram, (b) grid, representing the bins in the histogram. (c) the final histogram. The grids (bins) in Figure (b) with more observations are now coloured in green and grids with less observations are coloured in cyan.

Figure 3.5: Partition of data. The blue data is used as training data. Observe that we only use training data for methods based on the naive Bayes classifiers. The green part is validation data, this is available for all methods.

3.2.3 Reference Method

The features associated with each data segment that we require to implement the reference method are introduced in Chapter 3.1.3. The classification is done with both linear discriminant analysis Eq. (2.31) and quadratic discriminant analysis Eq. (2.33). Prior probabilities $P(C = c)$, mean vectors $\boldsymbol{\mu}$ and the covariance matrices $\boldsymbol{\Sigma}$ are calculated using training data, this is described in detail in Chapter 3.2.4. As mentioned in 2.4.2 for the linear classification, one common covariance matrix $\boldsymbol{\Sigma}$ is computed. For the quadratic classification one covariance matrix $\boldsymbol{\Sigma_c}$ is calculated for each tree species.

When linear classification is used an individual tree is assigned to the species for which Eq. (2.31) is maximised and for quadratic classification to the species for which Eq. (2.33) is maximised.

3.2.4 Tuning the Methods

In order to find the optimal settings (e.g. bandwidth, binwidth and number of height classes) for the methods based on naive Bayesian classifiers we need to tune them. This is accomplished by using cross 5-fold validation on the training data set. The training data (the blue part of Figure 3.5) set consists of 1459 tress (roughly 80 percent of the data). There are 180 Scots pine, 1146 Norway spruce and 133 deciduous trees in the data set.

For the reference method based on linear discriminant analysis (LDA) and quadratic discriminant analysis (QDA) we do not need to optimise the parameters of the model. This is because we have all the features of the model to start with. Thus we do not need to train the model. However we still need to find $\boldsymbol{\mu}_c$ and $\boldsymbol{\Sigma}_c$ and $P(C = c)$. This can be done at the same time as we validate the method. This means that we do not use the blue data from Figure 3.5 for this method.

Tuning the Naive Bayesian Classifiers.

The tuning routine is described here. For the tuning of the parameters we use the training data set (the blue part of Figure 3.5). First we choose the setting we want to test (e.g. for the KDE-based classifier we test number of height classes and bandwidth). We use 5-fold cross validation

hence the blue data are partitioned into 5 roughly equal sized data sets. Each of these are then split into new subsets according to the number of height classes used. We estimate unique pdfs for every height class and following the notation from previous chapters we obtain:

$$f_{D,A|C=1}(d,a), \quad f_{D,A|C=2}(d,a) \quad \text{and} \quad f_{D,A|C=3}(d,a). \tag{3.6}$$

These pdfs are then used in the classifier (recall Eq.(3.1)). After each iteration the user, producer and the overall accuracy is calculated. This is rotated until all 5 subsets have been used as validation data. After all iterations, we calculate the mean value of user, producer and the overall accuracy. This is our estimate of that accuracy measure.

On the training data set, this process is repeated until we have tried all settings that we are interested to optimise (e.g. bandwidth and number of height classes). In other words if we are interested in testing five different binwidths and all the six ways of partitioning trees into height classes we would need to repeat this $5 \cdot 6 = 30$ times. We then choose the setting that produces the highest overall accuracy for our final model.

For the histogram-based classifier the parameters we strive to optimise are binwidth and number of height classes, for the KDE-based classifier we optimise bandwidth and number of height classes. For the beta-based classifier the process is somewhat different.

Since we can not use pairs of observations for the beta distribution, we have to consider the measures squared standardized distance to the stem (d) and the relative height (a) as a parameter for this method. That is, we estimate two different pdfs $f_{A|C=c}(a)$ and $f_{D|C=c}(d)$. The other parameter we consider for this method are the same as for the other two Bayesian classifier methods, namely number of height classes. We then choose the setting (d or a and number of height classes) that produces the highest overall accuracy for our final model.

3.2.5 Validating the Methods

The validation of our methods are accomplished with 5-fold cross validation as well, both the reference methods and the naive Bayesian classifiers are validated. The methods are validated on the validation data set (the green part of Figure 3.5), we only run the procedure from previous section once for every method (Chapter 3.2.4). Here the purpose is only to estimate the accurucy of the method and not to find the paramters of the model. We calculate the mean value of user, producer and the overall accuracy for each method.

The validation process is a bit different for the naive Bayesian classifiers and the reference methods. For the naive Bayesian classifiers we use the optimal settings found in the tuning process. Then 5-fold validation is used on the validation data set in order to estimate the accuracy of the models. For the reference method we validate the method with the features given in Chapter 3.2.3 using 5-fold validation. During the validation process we calculate the covariance matrices and the mean vectors needed to perform the discriminant analysis.

3.2.6 Cluster Analysis With Beta-distribution

When we use cluster analysis with Beta-distribution, we fit two unique beta-distributions for each tree, one is fitted using the squared standardized distance to the stem and fitted using the relative height. The α and β parameters are then used as coordinates and plotted against each other. The idea is that different tree species may form unique clusters.

If we find three separate clusters we can use the α and β parameters as observations in k-means clustering to classify trees. Observations from a single tree consist of one α and one β parameter. These will be collected in vectors $\boldsymbol{x_1}, ..., \boldsymbol{x_n}$ such that α_i and β_i corresponds to observations from the ith tree and thus placed in vector $\boldsymbol{x_i}$. Recall Eq. (2.35), here $\boldsymbol{x_1}, ..., \boldsymbol{x_n}$ are sets of 2-dimensional vectors. Since we have three species we get $\boldsymbol{S} = (S_1, S_2, S_3)$, where S_i represent one species. Remember that $\boldsymbol{\mu_i}$ is the mean of the observations in $\boldsymbol{S_i}$ or in other words, $\boldsymbol{\mu_i}$ is the cluster centroid. Trees are assigned to a species according to:

$$\arg\min_{\boldsymbol{S}} \sum_{i=1}^{3} \sum_{x_j \in \boldsymbol{S_i}} ||\boldsymbol{x_j} - \boldsymbol{\mu_i}||^2. \tag{3.7}$$

In other words we expect the α and β parameters to be different for each species, so that the distance between different cluster centroids are large.

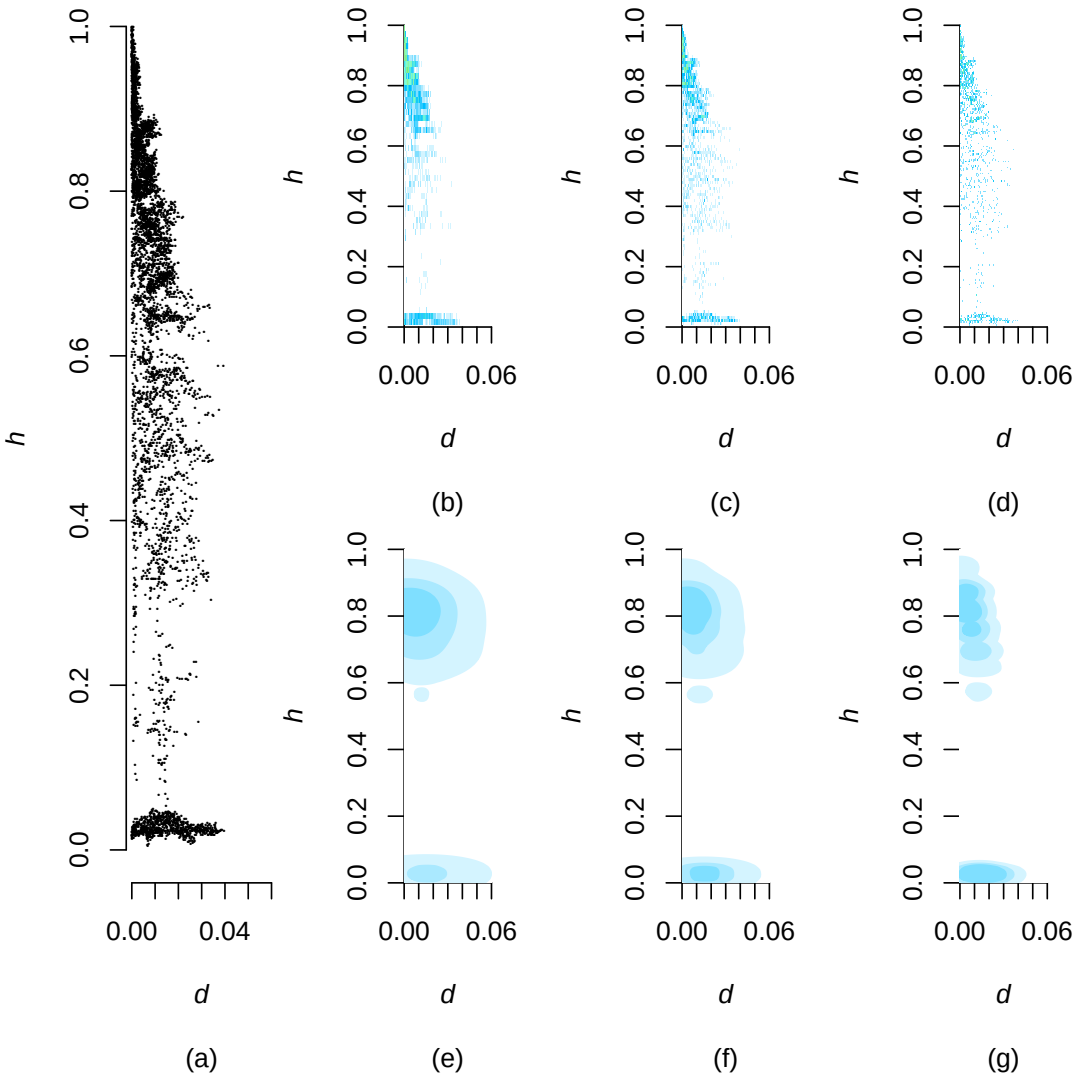

Figure 3.6: Histograms and kernel density estimators with different number of bins and bandwidths respectively. Figure (a) the data (points from one tree) used to create the histograms and KDEs. In (b),(c) and (d) we can see how a histogram is affected by changing the number of bins. In (b) there are 50 bins, in (c) 100 bins and 200 in (d). In (e),(f) and (g) we can see how a kernel density estimators is effected by changing the bandwidths h where $h_{11} = h_{22}$. In (e) h= 0.001, in (f) $h = 0.0005$ and in (g) h= 0.0001. Higher probability densities are coloured in green and lower densities are coloured in blue.

Chapter 4

Results

We present user accuracy, producer accuracy for each tree species and the overall accuracy (these measures are defined in chapter 3.2.1). The results are presented in bar plots, see Figure 4.1 and 4.2.

We obtained the same highest overall accuracy for two settings for the histogram method. Both 200×200 bins and 250×250 bins give an overall accuracy of 82.4 percent using unfiltered data and by partitioning the trees into three height classes according to g_{31} introduced in Chapter 3.1.2. We use 200×200 since this setting predicts a proportion of the tree species that is closer to the true proportion. The result from the model training for this setting can be found in Appendix A.5.

For the classifier based on kernel density estimation (KDE) the highest accuracy is obtained by using the bandwidth $h_{11} = h_{22} = 0.09$, unfiltered data and by partitioning the trees into tree height classes according to g_{32}. The result from the model training for this setting can be found in Appendix B.5.

The Beta method did not provide any good results, and choosing the best settings for this method is difficult. We use filtered data, the variable relative distance to the stem and five height classes according to the partition into height classes chapter3.1.2 . This is not the setting that obtains the highest overall accuracy; however the setting that obtains a higher accuracy seems rather unstable. By looking at Table C.2 we can see that choosing one height class instead of five provides a slightly higher overall accuracy but with this setting we fail to classify any Scots pine.

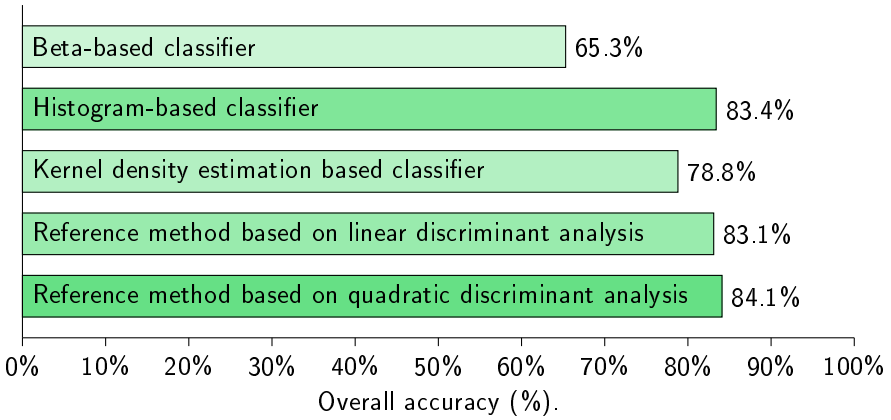

Figure 4.1: Overall accuracy of all tress.

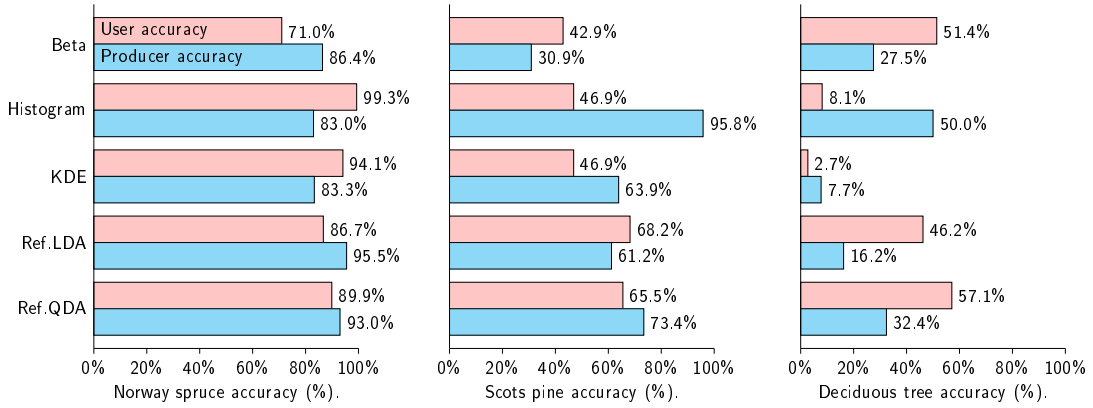

Figure 4.2: User and producer accuracy for Scots pine, Norway spruce and deciduous trees. The user accuracy is pink and producer accuracy is cyan. The following labels are used to denote the methods: Beta represent the beta-based classifier, Histogram represent the histogram-based classifier, KDE represent the KDE-based classifier, Ref. LDA represent the reference model with linear classification and Ref. QDA represent the reference model with quadratic classification.

If we look at Figure 4.1 we can see that the histogram-based classifier and the reference method based on QDA performed best. Of these two, the reference method based on QDA performed best, with an overall accuracy of 84.1 percent compared to an overall accuracy of 83.4 percent for the histogram-based classifier. If we look at user and producer accuracy in Figure 4.2 we can see that all methods except the method based on the beta-distribution obtains a user and producer accuracy greater than 80 percent for Norway spruce. All methods performs poorly when it comes to classifying the deciduous trees. Our results can bee compared to a random classification of the trees, that is, if we were to classify all trees at random. If we do this without considering the prior probabilities we manage to classify approximately 1/3 of the trees correct. If we use the prior probabilities $P(C = c)$ (the proportion of $C = c$ in training data set) as the probability to classify a tree as a species, we classify approximately 62.9 percent of the trees correctly. This means that all classifiers are better then both random classifiers, however the

beta based classifier is barely better than the random classifier that uses the prior probabilities.

The result from plotting the α and β parameters from the beta-distribution in order to investigate if it is possible to use these as clustering parameters can be seen in Figure 4.3. We see that no clear clusters are formed, and thus we do not expect this method to bee successful for classifying trees. The result of a 3-mean clustering is seen in Figure 4.4. In Figure 4.4 (a) we see no apparent pattern in the clusters, however we only get a few deciduous trees in cluster B. It seams as if cluster C contains a mixture of all trees and cluster A contains mostly spruce, however there are a fair number of pines in this cluster as well. As seen in Figure 4.4(b) these clusters do not contain any clear patterns either. It is therefore hard to use this as a basis for classification of trees.

The spatial distribution is an interesting aspect when classifying tress since trees tend to have a spatial correlation. That is, trees from the same species tend to grow alongside each other. We are especially interested to see the spatial distribution of the histogram estimator, since this is the method that obtains the highest overall accuracy of the methods developed in this thesis. In Figure 4.5 and 4.6 the true spatial distribution of the trees in the validation data set can bee seen next to the predicted spatial distribution obtained by using the histogram-based classifier. We can see the histogram method tend to overestimate the number of spruces and underestimate the number of deciduous trees in the field plots. This is an expected result since the histogram-based classifier does not consider the neighbouring trees and therefore is not designed to predict the spatial distribution of the trees. Still, this is of some interest since this is an area of improvement for this classifier.

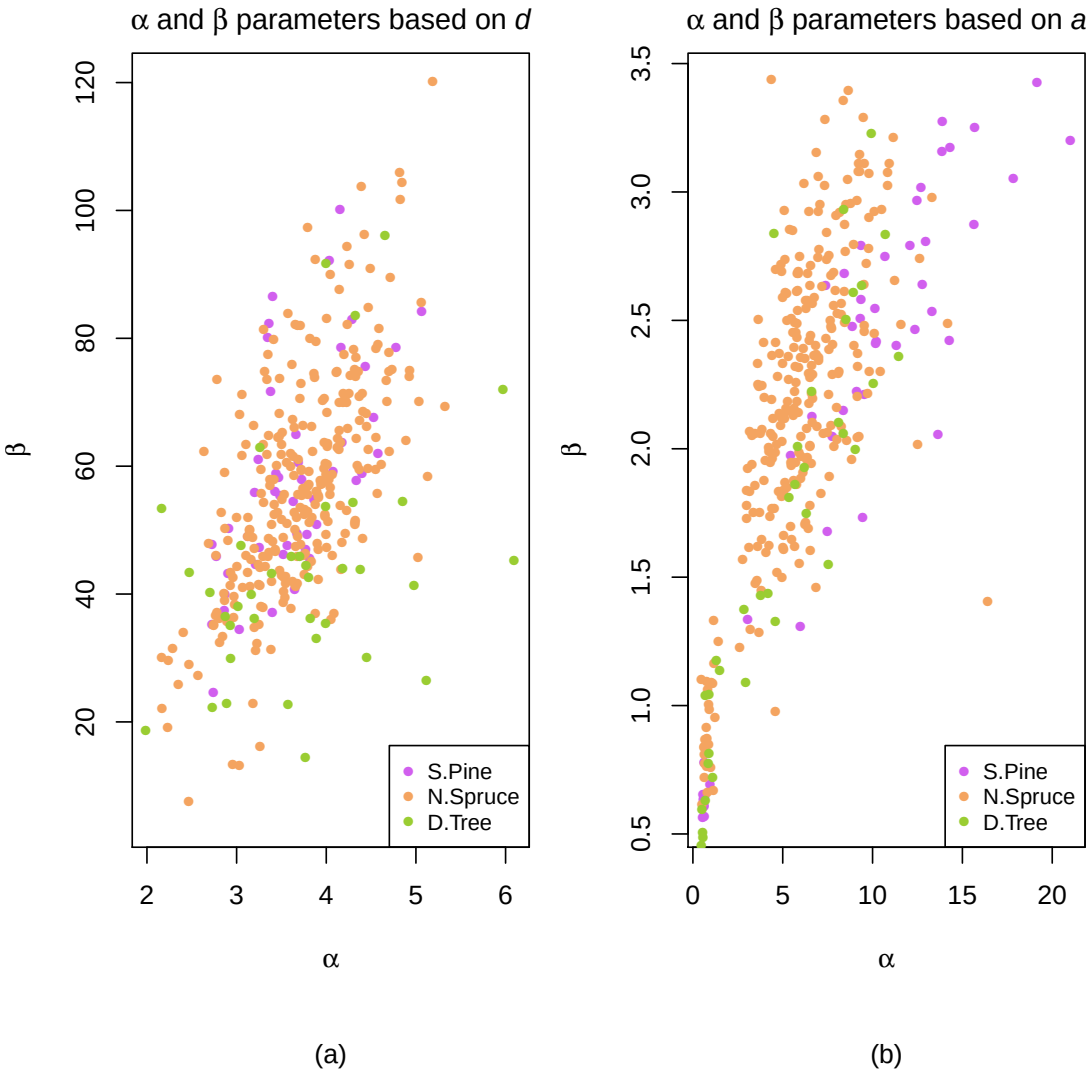

Figure 4.3: α and β parameters from a Beta-distribution based on the filtered data from the validation data set. In Figure (a), the α and β parameters are calculated using the squared standardized distance to the stem and in Figure (b) the α and β parameters are calculated using the relative height.

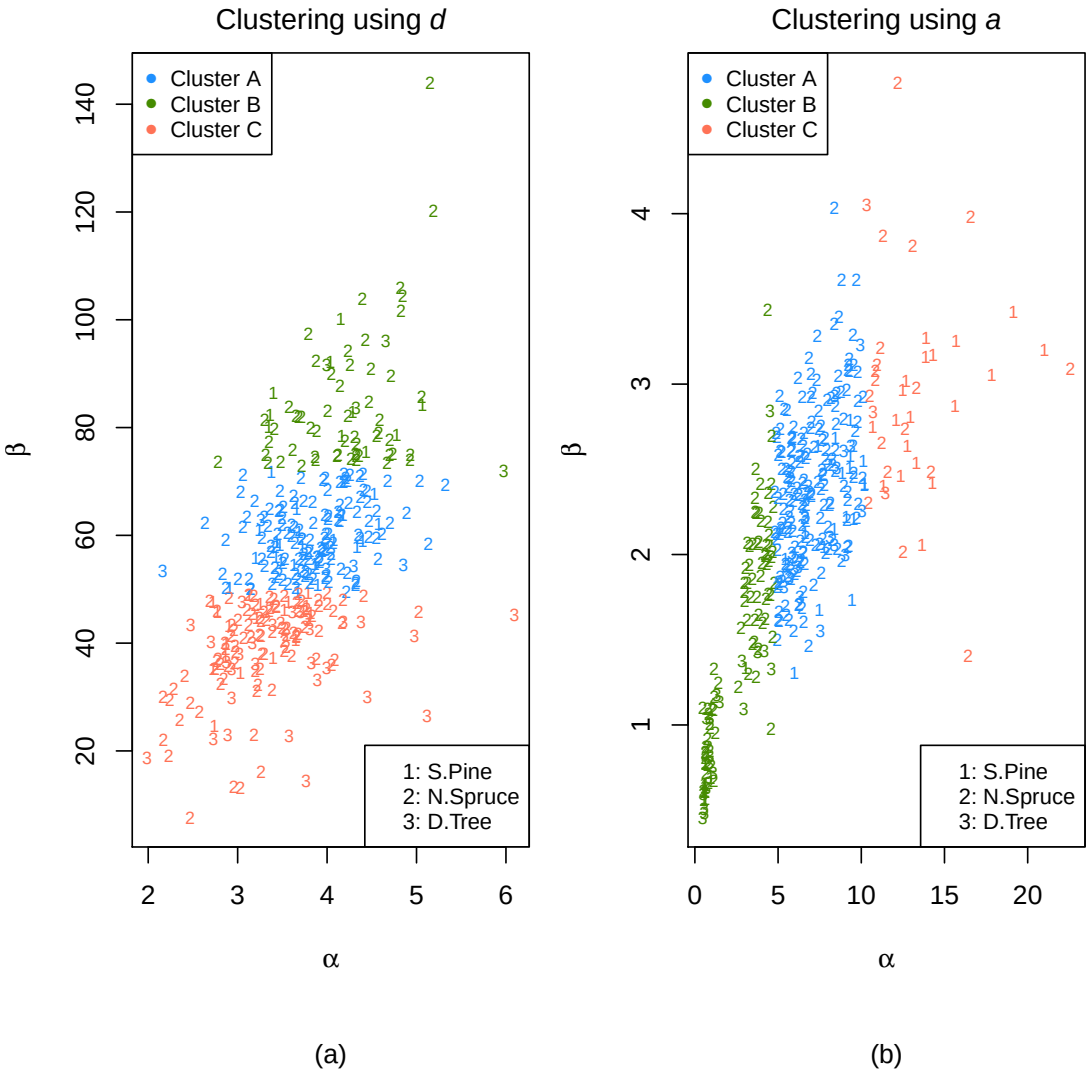

Figure 4.4: 3-means clustering where the α and β parameters from the Beta-distribution are used as clustering parameters. This is based on the filtered data from the validation data set. In Figure (a), the α and β parameters are calculated using the squared standardized distance to the stem and in Figure (b), the α and β parameters are calculated using the relative height. Different clusters are coloured in different colours and observations from the different trees are given numbers representing the species of that tree.

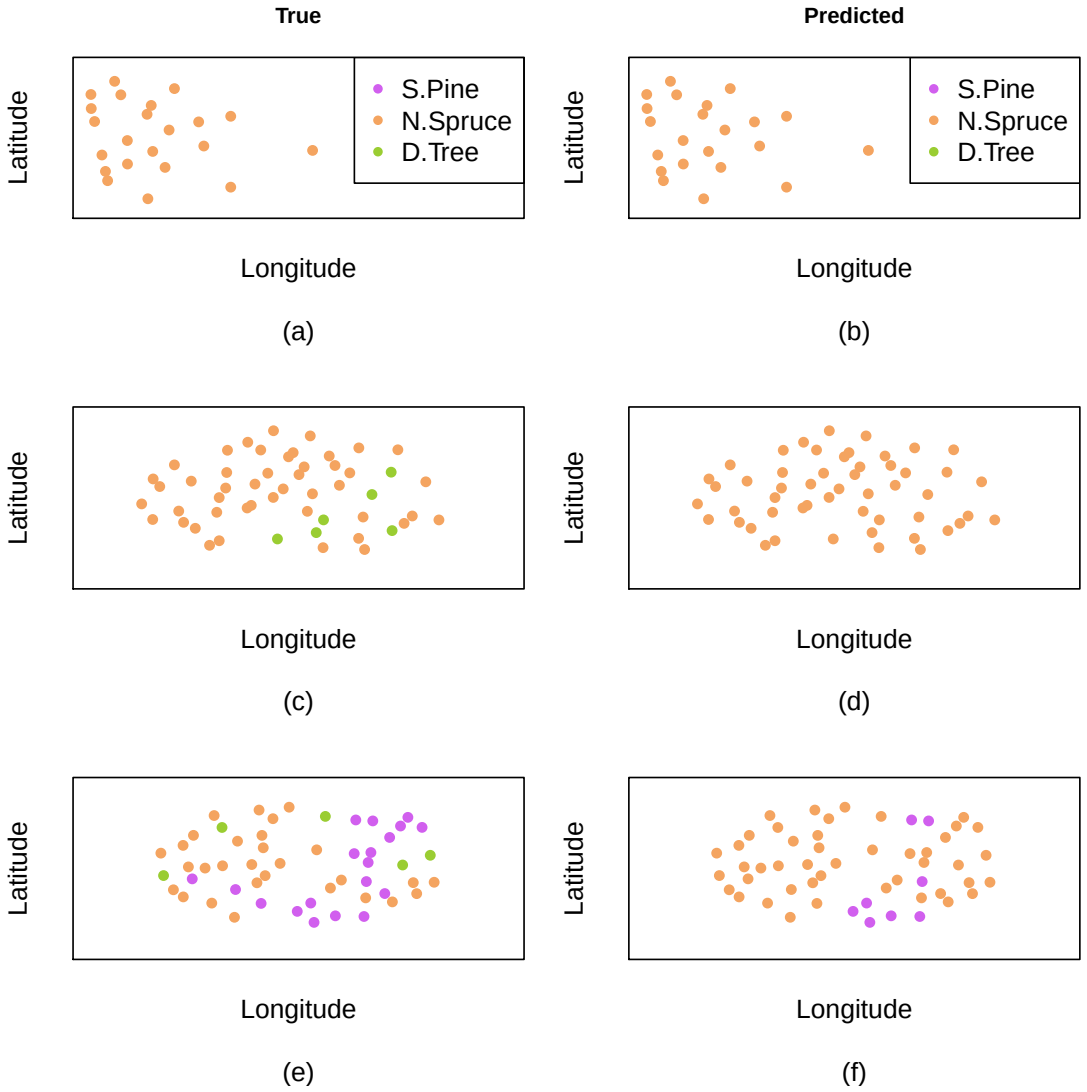

Figure 4.5: Spatial distribution of the trees in the validation data set. The true spatial distribution is found on the left side, that is, in (a),(c) and (e). The spatial distribution of the trees classified with the histogram-based classifier is found on the right side, that is, in (b),(d) and (f). Plots next to each other are based on the same sample, that is the data are taken from the same field plot. Together with Figure 4.6(a),(c) and (e),(a), (c) and (e) make up the whole validation set.

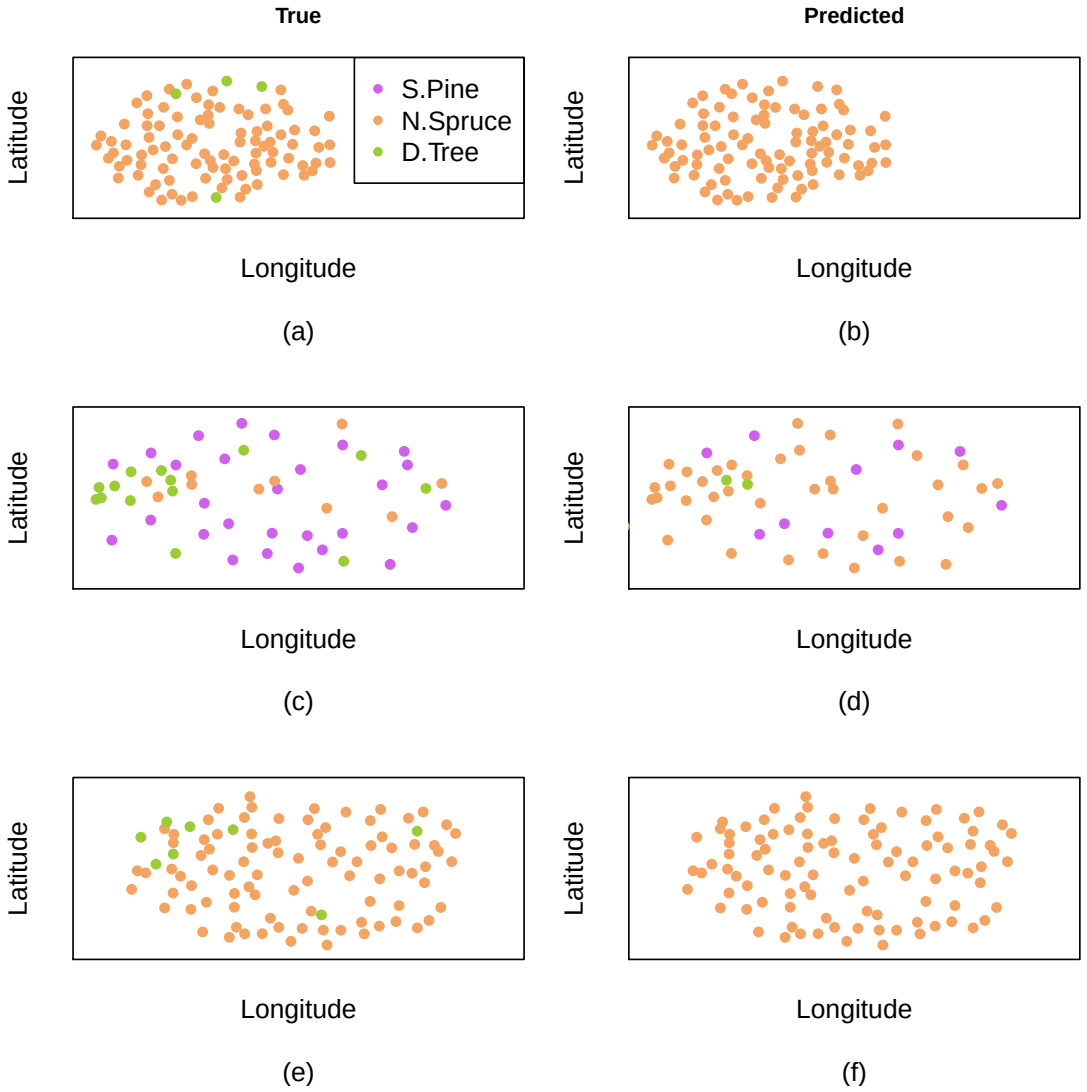

Figure 4.6: Spatial distribution of the trees in the validation data set. The true spatial distribution is found on the left side, that is, in (a),(c) and (e). The spatial distribution of the trees classified with the histogram-based classifier is found on the right side, that is, in (b),(d) and (f). Plots next to each other are based on the same sample, that is the data are taken from the same field plot. Together with Figure 4.5 (a),(c) and (e), (a), (c) and (e) make up the whole validation set.

Chapter 5

Discussion

5.1 Results and Data

Both the kernel density estimator (KDE) based classifier and the histogram-based classifier performs poorly when it comes to classifying deciduous trees correct. The main reason is probably that deciduous trees are rare. We only have 37 deciduous trees in the validation data set and when we partition the trees into height classes we run the risk of ending up with validation sets that do not contain a single deciduous tree. This is also a problem when we estimate the pdf; a small amount of information will not give a good estimate. Another difficulty with deciduous trees is that all deciduous trees in the test area are grouped together. The group deciduous trees contain trees of e.g. birch, aspen and oak. This will make it more complicated to classify deciduous trees since it is not a completely homogeneous group of trees. However it is necessary to group them together in order to get a large enough sample of deciduous tress.

Our data set mostly contains Norway spruce, about 78 percent of the trees are spruces. We should have a validation set from another forest, since we run the risk of over-fitting our models to a forest with many spruces. Unfortunately we do not have such a data set available.

We expected the kernel density based classifier to perform better than the histogram-based classifier since kernel density estimators tend to converge faster than the histogram density estimator. The fact that we did not obtain this result might indicate that we did not manage to find a good enough bandwidth matrix.

The computations of the results for the kernel density method are very time consuming, the kernel density method requires approximately 1450 times longer computational times than the histogram method. Of course this affects our chances to find good bandwidth matrices. As we can see in the appendix A and in B we have considerably less computations for the kernel density method compared to the histogram method. However this does not explain why the histogram method generally produces a higher accuracy.

It is no surprise that we do not obtain a good result for the method based on the beta-distribution. The data are reduced to one-dimension and hence we lose information. This is the only method in this thesis that almost performs as poorly as random classifier with prior probabilities. This indicates that it is not suitable to use this method to classifies trees. A better approach might have been to multiply the pdfs $\tilde{f}_{D|C=c}(d)$ and $\hat{f}_{A|C=c}(a)$ in order to get the two-dimensional

pdf $\hat{f}_{D,A|C=c}(d,a)$. This is possible due to the assumption of conditional independence between relative height and squared standardized distance to the stem given a class c. This would mean that we do not reduce the information to one-dimension.

The 3-mean clustering based on the α and β parameters from the beta-distribution did not prove to provide distinguishable clusters in two-dimensions for different species. It would have been possible to conduct a 3-mean clustering based on the α and β parameters in four dimensions using the α and β from the beta-distribution estimated on the relative height and the α and β from the beta-distribution estimated using the squared standardized distance to the stem. However this would mean that we would lose the visual interpretation of the result of the clustering and hence have not been conducted.

The idea to partition the trees into height classes seems to be a good approach. By inspecting the tables found in Appendix A and B we can see that we classify more trees correctly if we use more than one height class but using more than three height classes does not improve the accuracy.

One thing that might have affected our results is that we assumed that all trees were independent of each other when we partitioned the trees into the cross validation sets. It would have been better to put all trees from the same field plot in one in either the training set or the validation set, when performing the cross validation. However we have made sure that overlap of field plots in the data partitioning for training data set and validation data set (illustrated in 3.5) was as small as possible (we only have an overlap for one of 25 field plots).

5.2 Further Studies

How the partition of trees into height classes should be performed is a problem that is not addressed in this thesis. For all but one of the height classes the trees are partitioned into equally sized groups according to the height of the tree. This way of partitioning trees can probably be improved. At least when we use three height classes, we managed to obtain a higher accuracy for the KDE based classifier by using the partition into different height classes according to g_{32} compared to using g_{31}. That is we achieve a higher overall accuracy when we partition the trees into differently sized groups.

To use filtered data does not seem to be a good approach. When we use filtered data for the KDE based classifier and for the histogram-based classifier we obtain a lower overall accuracy. This is more evident for the KDE based classifier. Clearly the data we filter contains information that is important for the methods to be successful. Most of the data that is removed are ground hits. Ground hits contain information about how dense the tree crown is, so removing ground hits is not a good approach. One approach to avoid this problem is to to add the ground hits to the filtered data. We have made a few computations with filtered data with ground hits, the results look promising. Still only a few computations were performed so further studies are needed. By using filtered data with ground hits we obtain a higher overall accuracy compared to using filtered data. The results can be found in Appendix D. In future studies the filtering algorithm used for this thesis should not be used. Either another filtering method should be used or filtered data with ground hits should be used.

For the histogram method we would like to investigate if allowing unequal bin sizes can improve the results. We have made a few computations where we have allowed different number of bins. The results are found in Appendix D. These settings are interesting since we obtain a higher

accuracy than when we use bins of the same size. However we have only made a limited number of computations so further studies are needed before any conclusions can be drawn. Another interesting aspect would be to allow the binwidths to vary. That is, to use bins of different sizes so we make smaller bins where we have a lot of observations and larger bins where the data are limited.

In future studies it would also be interesting to investigate if allowing different values on the bandwidth parameters $h_{11}$and h_{22} would result in a better performance. Setting $h_{11} = h_{22}$ might not be optimal since the two variables in our data set squared standardized distance to stem (d) and real height (a) have different range. Hence, we might want to consider this when we perform our smoothing. Otherwise we will probably over-smooth in the direction of d or under-smooth in the direction of a. We have performed a few computations where we scaled one of the smoothing parameters with the maximal range of the other parameter. Since the range of a is $[0, 1]$ due to the definition, we will use the range of d, denoted d_r when we scale. This gave us the following bandwidth matrices:

$$\begin{pmatrix} d_r \cdot h_{11}^2 & 0 \\ 0 & h_{22}^2 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} h_{11}^2 & 0 \\ 0 & h_{22}^2/d_r \end{pmatrix}$$

The results are found in Appendix D. The results look promising since we obtain a higher accuracy when we scale the bandwidth matrix compared to using an un-scaled bandwidth matrix. The best result is achieved by using the following bandwidth matrix:

$$\begin{pmatrix} h_{11}^2 \approx 0.034 & 0 \\ 0 & h_{22}^2 = 0.090 \end{pmatrix}$$

which gave an overall accuracy of 81.5 percent. However we have only used this scaling in a few calculations and further studies are needed to obtain conclusive results.

5.3 Conclusions

The fact that we managed to achieve an overall accuracy of 83.4 percent with the histogram method shows the potential with likelihood based classification methods. However we would have expected the KDE based classifier to provide better results than the histogram method. With a good bandwidth matrix the KDE based classifier should ultimately perform better than the histogram-based classifier. Other methods of filtering the data should also be explored and the spatial distribution should be considered. With this in mind it should be possible to achieve better results for likelihood based classification methods in the future.

Bibliography

A.Azzalini, A.W.Bowman, 1990. A look at some data on the old faithful geyser. Applied Statistics 39 (3), 357–365.

Anton, H., Rorres, C., 2005. Elementary Linear Algebra with Applications. Vol. 9. Wiley.

Fisher, R. A., 1936. The use of multiple measurements in taxonomic problems. Annals of Eugenics 7, 179–188.

Frahling, G., Sohler, C., 2008. A fast k-means implementation using coresets. International Journal of Computational Geometry and Applications 18 (6), 605–625.

Gao, F., Han, L., 2012. Implementing the nelder-mead simplex algorithm. Springer Science+Business Media 51 (1), 259–277.

Garthwaite, P. H., Jolliffe, I. T., Jones, B., 1995. Statistical inference. Prentice Hall International (UK) Limited, London.

Heinzel, J. N., Weinacker, H., Koch, B., 2008. Full automatic detection of tree species based on delineated single tree crowns - a data fusion approach for airborne laser scanning data and aerial photographs. Proceedings of SilviLaser 2008: 8th international conference on LiDAR applications in forest assessment and inventory. Edinburgh, 2008.

Hogg, R. V., Tannis, E. A., 2009. Probability and Statistical Inference, 8th Edition. Pearson.

Holmgren, J., Persson, Å., 2004. Identifying species of individual trees using airborne laser scanner. Remote Sensing of Enviroment 90 (4), 415–423.

Holmgren, J., Persson, Å., Söderman, U., 2002. Detecting and measuring individual trees. Photogrammetric Engineering and Remote Sensing 68 (9), 925–932.

Johnson, R. A., Wichern, D. W., 2007. Applied multivariate statistical analysis, sixth Edition. Pearson Prentice Hall, Upper Saddle River, NJ.

Korpela, I., Tokola, T., 2006. Potential of aerial image-based monoscopic and multi-view single-tree forest inventory - a simulation approach. Forest Science 52 (2), 136–147.

Liang, X., Hyyppä, J., Matikainen, L., 2007. Deciduous-coniferous tree classification using difference between first and last pulse laser signatures. ISPRS Workshop on Laser Scanning 2007 and SilviLaser 2007 36, 253–257.

Lindberg, E., 2012. Estimation of canopy structure and individual trees from laser scanning data. Ph.D. thesis, Swedish University of Agricultural Sciences.

Næsset, E., Gobakken, T., Holmgren, J., Hyyppä, H., Hyyppä, J., Maltmo, M., Nilsson, M., Olsson, H., Persson, Å., Södreman, U., 2004. Laser scanning of forest resources: The nordic experience. Scandinavian Journal of Forest Research 19 (6), 482–499.

Pawlowsky-Glahn, V., Buccianti, A., 2011. Compositional data analysis. John Wiley & Sons Ltd., Chichester, theory and applications.

Ramsy, J., Silwerman, B., 1997. Functional Data Analysis. Springer.

Ørka, H. O., Naesset, E., Bollandsås, O. M., 2007. Utilizing airborne laser intensity for tree species classification. ISPRS Workshop on Laser Scanning and SilviLaser 36, 300–304.

Scott, D. W., 1992. Multivariate Density Estimation and Visualization, 1st Edition. Springer.

Solberg, S., Naesset, E., Bollandsas., O. M., 2006. Photogrammetric engineering and remote sensing. Single tree segmentation using airborne laser scanner data in a structurally heterogeneous spruce forest. 72, 1369–1378.

Vauhkonen, J., Tokola, T., Packalén, P., Matamo, M., 2009. Identification of scandinavian commercial species of individual trees from airborne laser scanning data using alpha shape metrics. Forest Science 55 (1), 37–47.

Wand, M. P., Jones, M. C., 1993. Comparison of smoothing parameterizations in bivariate kernel density estimation. J. Amer. Statist. Assoc. 88 (422), 520–528.

Wasserman, L., 2010. All of Nonparametric Statistics. Springer.

Wezyk, P., Koziol, K., Glista, M., Pierzchalski, M., 2007. Terrestrial laser scanning versus traditional forest inventory. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, ISPRS 3, 424–429.

Wouk, A., 1987. New Computer Environments: Microcomputers in Large Scale Computing. Society for Industrial and Applied Mathematics.

Appendix A

Result Model Training Histogram

The results for the model training are based on 1459 tress, there are 180 Scots pine, 1146 Norway spruce and 133 deciduous trees in the data set. In the tables in we use pred as acronym for predicted, S.Pine for Scots pine, N.Spruce for Norway spruce and deciduous for deciduous trees.

Table A.1: Table for correctly classified with 2-D histogram, with one height class. Using non-filter data.

$k \times k$	Pred S.pine	Pred N.spruce	Pred deciduous	Proportion correct
10	0	865	594	0.545
20	0	466	993	0.328
30	2	464	993	0.321
40	13	499	947	0.331
50	22	543	894	0.352
60	35	586	838	0.378
70	32	597	830	0.380
80	37	612	810	0.388
90	38	636	785	0.404
100	42	648	769	0.412
110	39	667	753	0.420
120	44	671	744	0.426
130	42	672	745	0.429
140	41	664	754	0.422
150	40	673	746	0.428
160	38	678	743	0.432
170	38	670	751	0.427
180	38	680	741	0.429
190	40	676	743	0.429
200	40	681	738	0.433
210	41	675	743	0.425
220	43	674	742	0.427
230	41	679	739	0.433
240	39	688	732	0.432
250	40	683	736	0.432
260	40	679	740	0.432
270	35	668	756	0.422
280	37	667	755	0.430
290	40	646	773	0.406
300	33	635	791	0.404

Table A.2: Table for correctly classified with 2-D histogram, with one height class. Using filter data.

$k \times k$	Pred S.pine	Pred N.spruce	Pred deciduous	Proportion correct
10	0	1052	407	0.645
20	0	960	499	0.611
30	16	1025	418	0.640
40	47	1038	374	0.640
50	61	1039	359	0.642
60	74	1041	344	0.644
70	75	1041	343	0.642
80	87	1039	333	0.639
90	85	1043	331	0.643
100	87	1043	329	0.648
110	92	1041	326	0.644
120	91	1037	331	0.639
130	89	1037	333	0.641
140	95	1025	339	0.637
150	86	1028	345	0.635
160	89	1011	359	0.622
170	105	965	389	0.597
180	99	965	395	0.599
190	83	961	415	0.595
200	91	920	448	0.566
210	88	901	470	0.559
220	83	889	487	0.553
230	91	837	531	0.513
240	85	818	556	0.498
250	78	826	555	0.510
260	78	789	592	0.487
270	80	795	584	0.493
280	75	779	605	0.477
300	85	751	623	0.466

Table A.3: Table for correctly classified with 2-D histogram, with two height classes, partition of trees according to g_2 . Using non-filter data.

$k \times k$	Pred S.pine	Pred N.spruce	Pred deciduous	Proportion correct
10	974	379	106	0.367
20	676	671	112	0.567
30	514	803	142	0.659
40	431	870	158	0.698
50	364	917	178	0.726
60	330	945	184	0.745
70	302	968	189	0.757
80	275	995	189	0.773
90	250	1020	189	0.781
100	224	1049	186	0.798
110	220	1061	178	0.801
120	215	1055	189	0.796
130	195	1092	172	0.807
140	179	1099	181	0.809
150	173	1112	174	0.816
160	172	1121	166	0.813
170	170	1123	166	0.811
180	155	1148	156	0.812
190	146	1155	158	0.809
200	149	1157	153	0.813
210	138	1165	156	0.812
220	138	1176	145	0.814
230	127	1181	151	0.817
240	132	1184	143	0.807
250	126	1192	141	0.804
260	121	1197	141	0.812
270	111	1218	130	0.812
280	108	1229	122	0.807

Table A.4: Table for correctly classified with 2-D histogram, with two height classes. Using filter data.

$k \times k$	Pred S.pine	Pred N.spruce	Pred deciduous	Proportion correct
10	822	513	124	0.434
20	594	759	106	0.611
30	500	834	125	0.665
40	447	869	143	0.683
50	416	892	151	0.698
60	385	921	153	0.717
70	357	934	168	0.724
80	339	952	168	0.733
90	327	970	162	0.748
100	312	982	165	0.751
110	305	985	169	0.753
120	287	992	180	0.753
130	275	1012	172	0.764
140	263	1018	178	0.770
150	247	1033	179	0.773
160	239	1034	186	0.770
170	224	1061	174	0.783
180	212	1066	181	0.780
190	202	1081	176	0.781
200	195	1087	177	0.785
210	185	1091	183	0.786
220	189	1100	170	0.785
230	170	1116	173	0.779
240	154	1127	178	0.791
250	149	1133	177	0.790
260	145	1140	174	0.793
270	136	1154	169	0.794
280	124	1162	173	0.801

Table A.5: Table for correctly classified with 2-D histogram, with tree height classes, partition of trees according to g_{31} . Using non-filter data.

$k \times k$	Pred S.pine	PredN.spruce	Pred deciduous	Proportion correct
10	625	632	202	0.446
20	338	944	177	0.631
30	278	1026	155	0.718
40	255	1046	158	0.747
50	252	1047	160	0.770
60	259	1055	145	0.790
70	253	1057	149	0.795
80	249	1079	131	0.803
90	238	1101	120	0.814
100	232	1107	120	0.809
110	228	1112	119	0.822
120	226	1119	114	0.814
130	221	1135	103	0.809
140	201	1146	112	0.814
150	193	1164	102	0.820
160	182	1177	100	0.818
170	176	1175	108	0.813
180	176	1182	101	0.814
190	160	1200	99	0.815
200	156	1209	94	0.824
210	152	1219	88	0.822
220	145	1222	92	0.814
230	140	1236	83	0.820
240	135	1237	87	0.820
250	135	1234	90	0.824
260	134	1240	85	0.821
270	128	1260	71	0.820
280	124	1246	89	0.819

Table A.6: Table for correctly classified with 2-D histogram, with tree height classes, partition of trees according to g_{31}. Using filter data.

$k \times k$	Pred S.pine	Pred N.spruce	Pred deciduous	Proportion correct
10	568	719	172	0.506
20	374	939	146	0.655
30	320	988	151	0.716
40	305	994	160	0.733
50	297	1009	153	0.746
60	292	1024	143	0.763
70	282	1033	144	0.766
80	278	1039	142	0.767
90	274	1046	139	0.777
100	262	1063	134	0.777
110	260	1068	131	0.784
120	249	1078	132	0.788
130	256	1080	123	0.800
140	246	1088	125	0.798
150	238	1102	119	0.800
160	238	1105	116	0.798
170	234	1106	119	0.795
180	231	1113	115	0.802
190	228	1124	107	0.797
200	224	1124	111	0.806
210	220	1133	106	0.805
220	211	1145	103	0.803
230	203	1143	113	0.803
240	201	1158	100	0.804
250	202	1150	107	0.803
260	209	1151	99	0.805
270	196	1160	103	0.810
280	177	1187	95	0.809

Table A.7: Table for correctly classified with 2-D histogram, with tree height classes, partition of trees according to g_{32}. Using non-filter data.

$k \times k$	Pred S.pine	Pred N.spruce	Pred deciduous	Proportion correct
10	625	632	202	0.529
20	338	944	177	0.735
30	278	1026	155	0.790
40	255	1046	158	0.798
50	252	1047	160	0.807
60	259	1055	145	0.807
70	253	1057	149	0.807
80	249	1079	131	0.804
90	238	1101	120	0.807
100	232	1107	120	0.807
110	228	1112	119	0.808
120	226	1119	114	0.797
130	221	1135	103	0.808
140	201	1146	112	0.804
150	193	1164	102	0.805
160	182	1177	100	0.811
170	176	1175	108	0.798
180	176	1182	101	0.797
190	160	1200	99	0.799
200	156	1209	94	0.802
210	152	1219	88	0.801
220	145	1222	92	0.794
230	140	1236	83	0.797
240	135	1237	87	0.793
250	135	1234	90	0.790
260	134	1240	85	0.790
270	128	1260	71	0.802
280	124	1246	89	0.794
290	125	1251	83	0.792
300	119	1263	77	0.792

Table A.8: Table for correctly classified with 2-D histogram, with tree height classes, partition of trees according to g_{32}. Using filter data.

$k \times k$	Pred S.pine	Pred N.spruce	Pred deciduous	Proportion correct
10	568	719	172	0.576
20	374	939	146	0.724
30	320	988	151	0.757
40	305	994	160	0.764
50	297	1009	153	0.772
60	292	1024	143	0.777
70	282	1033	144	0.773
80	278	1039	142	0.777
90	274	1046	139	0.778
100	262	1063	134	0.778
110	260	1068	131	0.785
120	249	1078	132	0.784
130	256	1080	123	0.778
140	246	1088	125	0.781
150	238	1102	119	0.787
160	238	1105	116	0.788
170	234	1106	119	0.792
180	231	1113	115	0.786
190	228	1124	107	0.791
200	224	1124	111	0.786
210	220	1133	106	0.786
220	211	1145	103	0.796
230	203	1143	113	0.796
240	201	1158	100	0.797
250	202	1150	107	0.795
260	209	1151	99	0.795
270	196	1160	103	0.790
280	177	1187	95	0.800

Table A.9: Table for correctly classified with 2-D histogram, with four height classes, partition of trees according to g_4. Using non-filter data.

$k \times k$	Pred S.pine	Pred N.spruce	Pred deciduous	Proportion correct
10	526	730	203	0.520
20	389	936	134	0.732
30	321	995	143	0.795
40	284	1019	156	0.798
50	254	1045	160	0.812
60	249	1051	159	0.819
70	237	1073	149	0.821
80	245	1068	146	0.811
90	238	1071	150	0.810
100	226	1088	145	0.816
110	210	1108	141	0.818
120	205	1115	139	0.807
130	206	1112	141	0.807
140	191	1132	136	0.806
150	188	1135	136	0.810
160	186	1133	140	0.810
170	182	1147	130	0.796
180	175	1159	125	0.805
190	175	1159	125	0.808
200	164	1171	124	0.810
210	155	1174	130	0.803
220	158	1173	128	0.805
230	156	1186	117	0.803
240	154	1196	109	0.809
250	154	1187	118	0.798
260	146	1202	111	0.800
270	145	1204	110	0.798
280	141	1206	112	0.803

Table A.10: Table for correctly classified with 2-D histogram, with five height classes, partition of trees according to g_5. Using non-filter data.

$k \times k$	Pred S.pine	Pred N.spruce	Pred deciduous	Proportion correct
10	546	700	213	0.553
20	296	994	169	0.753
30	249	1050	160	0.796
40	217	1083	159	0.805
50	203	1097	159	0.812
60	203	1102	154	0.810
70	192	1117	150	0.817
80	186	1133	140	0.809
90	173	1165	121	0.811
100	163	1174	122	0.809
110	159	1184	116	0.815
120	149	1196	114	0.803
130	153	1203	103	0.815
140	141	1207	111	0.814
150	137	1220	102	0.811
160	138	1221	100	0.816
170	134	1219	106	0.807
180	143	1212	104	0.809
190	131	1229	99	0.815
200	132	1232	95	0.812
210	128	1241	90	0.812
220	132	1234	93	0.805
230	127	1247	85	0.805
240	130	1240	89	0.807
250	133	1236	90	0.804
260	126	1247	86	0.804
270	121	1267	71	0.811
280	118	1251	90	0.807
290	125	1250	84	0.803
300	120	1262	77	0.803

Table A.11: Table for correctly classified with 2-D histogram, with five height classes, partition of trees according to g_5. Using filter data.

$k \times k$	Pred S.pine	Pred N.spruce	Pred deciduous	Proportion correct
10	515	753	191	0.586
20	343	967	149	0.726
30	279	1032	148	0.770
40	256	1042	161	0.775
50	240	1062	157	0.785
60	236	1069	154	0.785
70	225	1083	151	0.790
80	218	1091	150	0.788
90	211	1097	151	0.791
100	208	1110	141	0.798
110	202	1112	145	0.792
120	194	1126	139	0.798
130	200	1132	127	0.796
140	180	1146	133	0.797
150	172	1165	122	0.807
160	166	1175	118	0.806
170	172	1169	118	0.807
180	163	1179	117	0.806
190	163	1185	111	0.809
200	163	1183	113	0.805
210	155	1195	109	0.805
220	149	1206	104	0.816
230	145	1202	112	0.814
240	146	1210	103	0.809
250	143	1206	110	0.814
260	146	1212	101	0.813
270	134	1220	105	0.802
280	133	1230	96	0.810
290	133	1227	99	0.804
300	131	1228	100	0.806

Appendix B

Result Model Training KDE

Table B.1: Table for correctly classified with 2-D KS, with one height classes. Using non-filter data.

σ	Pred S.pine	Pred N.spruce	Pred deciduous	Proportion correct
0.02	1	1450	8	0.783
0.04	1	1448	10	0.782
0.06	1	1448	10	0.782
0.08	1	1448	10	0.782

Table B.2: Table for correctly classified with 2-D KS, with two height classes, partition of trees according to g_2. Using non-filter data.

σ	Pred S.pine	Pred N.spruce	Pred deciduous	Proportion correct
0.02	675	649	135	0.532
0.04	629	706	124	0.567
0.06	493	878	88	0.644
0.08	314	1084	61	0.719
0.1	172	1252	35	0.740
0.12	80	1345	34	0.755

Table B.3: Table for correctly classified with 2-D KS, with three height classes, partition of trees according to g_{31}. Using non-filter data.

σ	Pred S.pine	Pred N.spruce	Pred deciduous	Proportion correct
0.02	635	637	187	0.528
0.04	566	685	208	0.546
0.06	400	837	222	0.609
0.08	258	1016	185	0.668
0.1	132	1142	185	0.667
0.12	64	1254	141	0.698

Table B.4: Table for correctly classified with 2-D KS, with three height classes, partition of trees according to g_{31}. Using non-filter data.

σ	Pred S.pine	Pred N.spruce	Pred deciduous	Proportion correct
0.2	8	1411	40	0.760
0.3	1	1412	46	0.761
0.4	1	1432	26	0.771
0.5	1	1438	20	0.775

Table B.5: Table for correctly classified with 2-D KS, with three height classes, partition of trees according to g_{32}. Using non-filter data.

σ	Pred S.pine	Pred N.spruce	Pred deciduous	Proportion correct
0.01	444	691	323	0.580
0.03	419	717	323	0.587
0.06	260	991	208	0.720
0.09	99	1291	68	0.805
0.12	4	1438	17	0.783
0.18	0	1458	1	0.786
0.27	0	1459	0	0.785

Table B.6: Table for correctly classified with 2-D KS, with three height classes, partition of trees according to g_{31}. Using filter data.

σ	Pred S.pine	Pred N.spruce	Pred deciduous	Proportion correct
0.02	635	637	187	0.528
0.04	583	698	178	0.559
0.06	405	832	222	0.605
0.08	781	594	84	0.446
0.1	792	580	87	0.426
0.12	794	578	87	0.414
0.14	794	575	90	0.409

Table B.7: Table for correctly classified with 2-D KS, with three height classes, partition of trees according to g_{31}. Using filter data.

σ	Pred S.pine	Pred N.spruce	Pred deciduous	Proportion correct
0.4	672	665	122	0.426
0.3	709	622	128	0.401
0.2	753	575	131	0.383

Table B.8: Table for correctly classified with 2-D KS, with three height classes, partition of trees according to g_{31}. Using filter data.

σ	Pred S.pine	Pred N.spruce	Pred deciduous	Proportion correct
1	472	900	87	0.565
2	323	1079	57	0.644
3	226	1198	35	0.709

Table B.9: Table for correctly classified with 2-D KS, with four height classes, partition of trees according to g_4. Using non-filter data.

σ	Pred S.pine	Pred N.spruce	Pred deciduous	Proportion correct
0.02	609	674	176	0.552
0.04	552	747	160	0.578
0.06	415	909	135	0.653
0.08	254	1078	127	0.705
0.1	132	1228	99	0.718
0.12	57	1310	92	0.726
0.14	34	1343	82	0.735

Table B.10: Table for correctly classified with 2-D KS, with four height classes, partition of trees according to g_4. Using non-filter data.

σ	Pred S.pine	Pred N.spruce	Pred deciduous	Proportion correct
0.2	8	1411	40	0.760
0.3	1	1412	46	0.761
0.4	1	1432	26	0.771
0.5	1	1438	20	0.775

Table B.11: Table for correctly classified with 2-D KS, with four height classes, partition of trees according to g_4. Using filter data.

σ	Pred S.pine	Pred N.spruce	Pred deciduous	Proportion correct
0.1	779	523	157	0.397
0.2	796	539	124	0.387
0.3	733	598	128	0.413
0.4	694	634	131	0.434
0.5	655	690	114	0.461
0.6	584	750	125	0.474
0.7	609	757	93	0.500
0.8	587	784	88	0.517
0.9	573	810	76	0.528
1	545	841	73	0.547

Table B.12: Table for correctly classified with 2-D KS, with four height classes, partition of trees according to g_4. Using filter data.

σ	Pred S.pine	Pred N.spruce	Pred deciduous	Proportion correct
1.5	426	973	60	0.620
2	361	1050	48	0.662
2.5	315	1115	29	0.696
3	279	1163	17	0.718
3.5	252	1191	16	0.745
4	224	1224	11	0.753
5	168	1289	2	0.779

Table B.13: Table for correctly classified with 2-D KS, with five height classes, partition of trees according to g_5. Using non-filter data.

σ	Pred S.pine	Pred N.spruce	Pred deciduous	Proportion correct
0.02	436	681	342	0.5633996
0.04	388	754	317	0.5887594
0.06	264	919	276	0.6710075
0.08	161	1104	194	0.7272104
0.1	74	1248	137	0.7278958
0.12	31	1300	128	0.7224126
0.14	28	1324	106	0.7290809

Table B.14: Table for correctly classified with 2-D KS, with five height classes, partition of trees according to g_5. Using non-filter data.

σ	Pred S.pine	Pred N.spruce	Pred deciduous	Proportion correct
0.2	9	1371	79	0.735
0.3	2	1389	68	0.745
0.4	0	1396	63	0.748
0.5	0	1404	55	0.753

Appendix C

Results Model Training Classification Based On Beta-distribution

Table C.1: Table for correctly classified with Beta-distribution . Using non-filter data and the squared standardized distance to stem.

Height class	Pred S.pine	Pred N.spruce	Pred deciduous	Proportion correct
1	4	23	1432	0.106
2	266	939	254	0.594
3	418	767	274	0.492
4	395	773	291	0.489
5	158	1011	290	0.626

Table C.2: Table for correctly classified with Beta-distribution . Using filter data and the squared standardized distance to stem.

Height class	Pred S.pine	Pred N.spruce	Pred deciduous	Proportion correct
1	0	1149	310	0.633
2	235	952	272	0.608
3	398	775	286	0.502
4	327	843	289	0.535
5	200	978	281	0.620

Table C.3: Table for correctly classified with Beta-distribution . Using non-filter data and the relative height.

Height class	Pred S.pine	Pred N.spruce	Pred deciduous	Proportion correct
1	400	810	249	0.622
2	343	728	388	0.604
3	343	728	388	0.576
4	369	767	323	0.598
5	321	777	361	0.587

Table C.4: Table for correctly classified with Beta-distribution . Using filter data and squared relative height.

Height class	Pred S.pine	Pred N.spruce	Pred deciduous	Proportion correct
1	204	555	700	0.425
2	507	700	252	0.508
3	454	677	328	0.500
4	467	661	331	0.496
5	417	642	400	0.515

Appendix D

Supplementary Results

Table D.1: Table for correctly classified with 2-D histogram, with two height classes, partition of trees according to g_2. Using filter data and ground hist.

$k \times k$	Pred S.pine	Pred N.spruce	Pred deciduous	Proportion correct
10	822	513	124	0.434
20	594	759	106	0.611
30	500	834	125	0.665
40	447	869	143	0.756
50	416	892	151	0.772
60	385	921	153	0.779
70	385	921	153	0.784
80	357	934	168	0.792
90	339	952	168	0.796
100	327	970	162	0.804
110	312	982	165	0.807
120	305	985	169	0.801
130	287	992	180	0.805
140	275	1012	172	0.800
150	263	1018	178	0.812
160	247	1033	179	0.806
170	239	1034	186	0.807
180	224	1061	174	0.811
190	212	1066	181	0.813
200	202	1081	176	0.821
210	195	1087	177	0.809
220	185	1091	183	0.812
230	189	1100	170	0.813
240	170	1116	173	0.821
250	154	1127	178	0.822
260	149	1133	177	0.818
270	145	1140	174	0.810
280	136	1154	169	0.817

Table D.2: Table for correctly classified with 2-D histogram, with tree height classes, partition of trees according to g_{31}. Using filter data and ground hits.

$k \times k$	Pred S.pine	Pred N.spruce	Pred deciduous	Proportion correct
10	568	719	172	0.576
20	374	939	146	0.724
30	320	988	151	0.757
40	305	994	160	0.750
50	297	1009	153	0.760
60	292	1024	143	0.778
70	282	1033	144	0.786
80	278	1039	142	0.790
90	274	1046	139	0.800
100	262	1063	134	0.796
110	260	1068	131	0.800
120	249	1078	132	0.803
130	256	1080	123	0.809
140	246	1088	125	0.806
150	238	1102	119	0.816
160	238	1105	116	0.812
170	234	1106	119	0.810
180	231	1113	115	0.810
190	228	1124	107	0.816
200	224	1124	111	0.820
210	220	1133	106	0.824
220	211	1145	103	0.818
230	203	1143	113	0.820
240	201	1158	100	0.825
250	202	1150	107	0.824
260	209	1151	99	0.820
270	196	1160	103	0.827
280	177	1187	95	0.825

Table D.3: Table for correctly classified with 2-D histogram, with four height classes, partition of trees according to g_4.Using filter data and ground hits.

$k \times k$	Pred S.pine	Pred N.spruce	Pred deciduous	Proportion correct
10	646	641	172	0.563
20	362	956	141	0.708
30	259	1050	150	0.754
40	235	1059	165	0.787
50	221	1074	164	0.800
60	210	1095	154	0.808
70	202	1092	165	0.800
80	192	1113	154	0.814
90	182	1138	139	0.807
100	173	1153	133	0.812
110	164	1162	133	0.813
120	162	1160	137	0.814
130	166	1170	123	0.820
140	150	1175	134	0.819
150	151	1186	122	0.817
160	146	1197	116	0.820
170	151	1190	118	0.818
180	148	1195	116	0.814
190	144	1207	108	0.811
200	145	1213	101	0.813
210	147	1203	109	0.814
220	140	1216	103	0.811
230	137	1226	96	0.818
240	136	1226	97	0.814
250	147	1217	95	0.816
260	144	1218	97	0.819
270	144	1225	90	0.814
280	144	1214	101	0.817

Using the following bandwidth matrix:

$$\begin{pmatrix} h_{11}^2 \approx 0.030 & 0 \\ 0 & h_{22}^2 = 0.080 \end{pmatrix}$$

gave an overall accuracy of 78.5 percent.

The scaled bandwidth matrix:

$$\begin{pmatrix} h_{11}^2 \approx 0.034 & 0 \\ 0 & h_{22}^2 = 0.090 \end{pmatrix}$$

gave an overall accuracy of 81.5 percent.

$$\begin{pmatrix} h_{11}^2 = 0.08 & 0 \\ 0 & h_{22}^2 \approx 0.030 \end{pmatrix}$$

gave an overall accuracy of 78.5 percent.

The scaled bandwidth matrix:

$$\begin{pmatrix} h_{11}^2 = 0.09 & 0 \\ 0 & h_{22}^2 \approx 0.034 \end{pmatrix}$$

gave an overall accuracy of 78.5 percent.