Statistical Methods for Classification of Wooden Boards

Johan Lindell

Master's thesis
2014:E60

Lund University

Faculty of Engineering
Centre for Mathematical Sciences
Mathematical Statistics

Centrum Scientiarum Mathematicarum

Statistical Classification of Wooden Boards

Johan Lindell

Recent technological advances have created new opportunities in the sawmill industry. The quality inspection of sawn boards has traditionally been performed manually, but it is increasingly common to use cameras and laser technology to determine board quality. In addition to reducing the workload on manual quality inspectors, an automatic inspection system can potentially be used as a replacement for other machines. One such is a strength classification machine. In this project it has been investigated how the information from a quality inspection system, consisting of cameras and point lasers, can be used to perform the classification otherwise done by a strength grading machine.

An automatic inspection system can be used to identify many different properties on a board. The information from laser scanning provides the direction of fibre angles, and cameras are used to capture color images of the boards. By combining these, board characteristics and defects such as knots, cracks etc. are identified. From this information the board is classified into one of several quality classes. Figure 1 shows a color image of a board, and figure 2 shows a laser measurement of the same board.

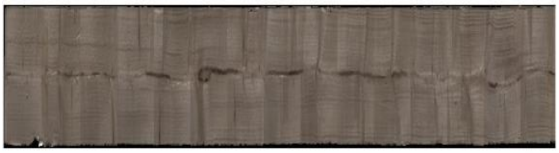

Figure 1: Color image of a board, obtained from an automatic inspection system.

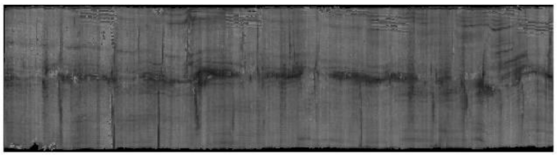

Figure 2: Visualisation of laser point area. In the lighter regions the laser point spreads over a larger area.

A new approach for board classification has been attempted in this project. Statistical classification methods have been applied to the board data, producing a classifier that attempts to divide boards into different strength classes. The classifier takes board characteristics, or features, from each board and makes a classification decision based on that information. Figure 3 shows a scatter plot of boards from two different strength classes. The red data points correspond to boards from a higher strength class compared to the green points. The black line is a decision boundary created by a classification model. The class belonging of a new observation is determined by which side of the boundary it is on.

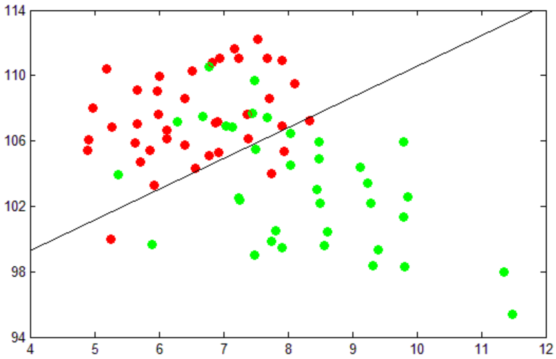

Figure 3: Boards from two strength classes. On the horizontal axis is standard deviation of fibre angles, and on the vertical axis is the mean color brightness.

It seems that a lower value of standard deviation of fibre angles combined with a higher value of mean brightness results in a higher strength class. Even though only two features have been used in this case the classes can be separated so that most of the data points are on the correct side of the decision boundary.

For this project data from 1000 boards from three different strength classes were collected, and two main types of classification methods were tested, Support Vector Machines and Multinomial Logistic Regression. The best classification was obtained by a version of the Multinomial Logistic Regression model. On a test set of 250 previously unseen boards from three strength classes this classifier managed to classify 191 correctly.

Apart from strength classification, another interesting application would be to create a classifier that can distinguish between different wood species.

Abstract

The quality inspection of wooden boards is experiencing a large change. By the use of camera and laser technology board characteristics and defects can be instantly identified and measured. This thesis investigates how the information from a quality inspection system can be used to classify boards into different quality classes, by the use of statistical classification models. Two types of classification models have been tested, Logistic Regression and Support Vector Machines. To deal with potential overfitting a regularized version of Logistic Regression is implemented, and to deal with ordinal dependent variables a logistic regression model for ordinal variables has been implemented. The classification models have been tested against board strength classes, and similar results have been obtained by most models. It is concluded that the regularized logistic regression is the model that manages to classify most boards correctly, but the Support Vector Machine produces a better result on classes where training data is scarce.

The thesis was done on behalf of RemaSawco AB, a company that manufactures measurement systems and inspection systems for the sawmill industry.

Acknowledgements

I would like to thank my supervisor at LTH, Johan Lindström, for his valuable guidance and advice throughout this project. I would also like to thank Erik Johansson at RemaSawco, for helping me with the collection of data, and for providing me with useful information on the quality inspection of wooden boards.

Contents

1 Introduction

1.1 Background

Over the last decades the sawmill industry has gone through significant changes. Many of the processes on a sawmill have become increasingly automated, due to the rapid development of computers and measuring devices. It is now possible to measure each log and board with high precision allowing for faster and more accurate cutting and sorting decisions. Tasks that traditionally have been performed by hand are being replaced by computer controlled systems.

The function of a sawmill can shortly be described as the processing of logs into boards. The logs are cut into boards, which are trimmed and sorted according to quality and dimension. A main goal of a sawmill is to get as much profit as possible from each log. This can be achieved both by converting a large volume of the log into boards and by producing high quality boards.

An area where there has been a lot of progress is in the quality inspection and sorting of boards. This is one of the final steps in a sawmill and is done at a so-called sorting mill. Here the boards are transversely transported on a chain conveyor and sorted into different bins. The boards can also be trimmed lengthwise to remove bad parts and thereby increasing the board quality.

Usually the boards are sorted into a number of different visual quality classes depending on the appearance of defects and other characteristics, such as the size of knots or the length of cracks. Traditionally this has been done by a person who determines the quality of each board as it passes by on the conveyor, but it is getting more common using a camera system to analyse the boards.

The sawmills in Sweden often base their board qualities on the standards in [1]. The boards can also be sorted into strength classes according to the European standard [2]. A strength grading requires the combination of a machine grading and a visual grading of the boards. The machine grading is typically done by measuring the elasticity module of the boards, and the visual grading is done by a person or a camera system.

1.2 RemaSawco

This thesis was done at RemaSawco AB, a company that manufactures measurement systems and optimisation systems for the sawmill industry. One of RemaSawco's main products is their system for quality inspection and sorting of boards and planks.

1.3 The Quality Inspection System

The quality inspection system uses cameras and point lasers to collect data from the boards, as they pass through the scanner on a conveyor. The data is then translated into board features such as dimension, fibre angles or the sizes and positions of knots. Also many types of defects, such as cracks or rot can be detected. These features are used when sorting the boards into different quality classes. Typically the system sorts boards into 2-5 different classes at a time, depending on the tree species and the intended usage of the boards. Classifying a board into a quality class is done by checking if the board meets the requirements of the class. Requirements for a class could for example be that the number of knots must be less than some value, or that the presence of a specific defect is not allowed. The board is then classified as belonging to the highest possible quality class where the requirements are met.

1.4 Aim of the thesis

In this thesis it will be investigated how the board features can be used for classification without setting specific rules and requirements, and instead applying statistical classification methods. A potential use for this type of classification is the ability to predict the classification of a strength grading machine, or to make the system distinguish between wood species. In this project the classification objective will be to predict the classification of a strength grading machine. The goal is thus to find a classification method that can resemble the classification of a strength grading machine, based on features given by the quality inspection system.

1.5 Overview of the thesis

A more detailed description regarding information available from the quality inspection system is given in Chapter 2. Here is also a discussion about which information to use in the classification models. Chapter 3 describes the tools and classification methods used. It contains a description of Support Vector Machines, Logistic Regression, and the Lasso. Chapter 4 contains a description of the software used and implementation of the classification models. In Chapter 5 results and comparisons from testing of the classification methods on real data are presented, and Chapter 6 contains conclusions and suggestions for future work.

2 Data

2.1 Available Data

Data from the cameras and lasers is collected from all sides of a board except the ends. The data is divided into seven channels. Three of the channels correspond to the color channels of the cameras - red, green and blue. The other four are different types of measurements from the point lasers – called Profile, Area, Scatter and Angle. The Profile measures the shape of the board using triangulation. The Area and Scatter are different types of measures of how the laser point spreads on the board, and the Angle corresponds to the angle of fibers on the wood surface. The color, area and scatter channels range from 0 to 255 and the fibre angles range from -90° to 90°, where 0° is the direction along the board.

Figure 2.1 shows the color channels from one side of a board converted into an RGB image. Figure 2.2 shows the laser point area, and figures 2.3-2.4 shows the fibre angles and scatter measurement.

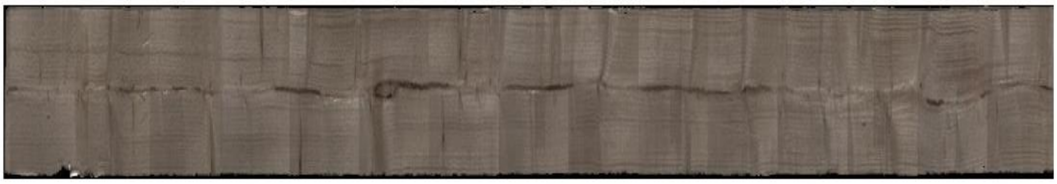

Figure 2.1: RGB image.

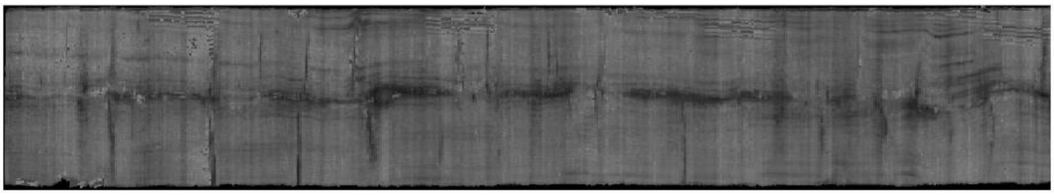

Figure 2.2: Laser point area.

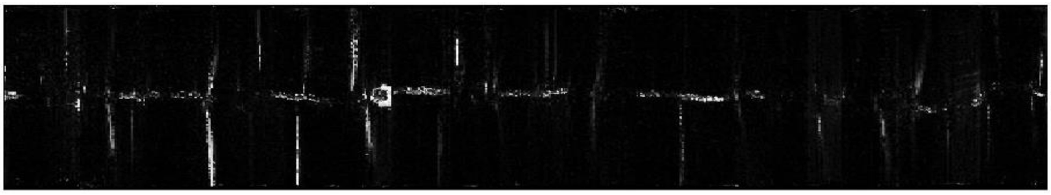

Figure 2.3: Absolute value of fiber angles.

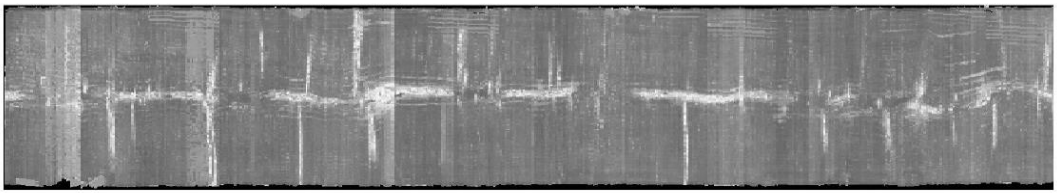

Figure 2.4: Laser point scatter.

Defects are detected on these images using image processing. Up to 20 different types of defects can be detected by combining information from the images. The first step to identify defects is to find the defect regions on a board. These regions are found from thresholding, by finding values in each channel that exceeds a predetermined level. Image processing is then used to merge regions and to remove small or insignificant regions. The regions corresponding to the RGB channels are merged creating one "map" of regions, and the regions corresponding to the Scatter, Angle and Area measurements are merged into a different map. There is also a defect region map for the profile channel. Thus, there are in total three defect region maps for a board – visual (RGB), laser point and profile. Shown in figure 2.5 is the visual (RGB) defect region map for the board in figure 2.1-2.4.

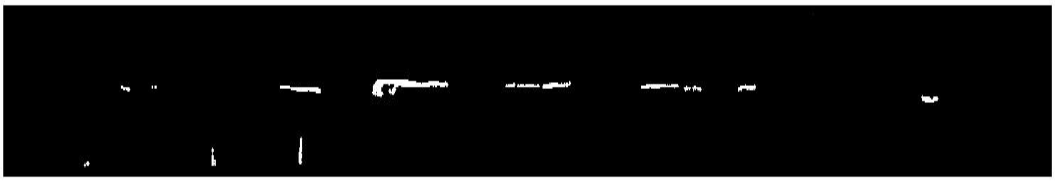

Figure 2.5: Visual defect regions

The dimension (width and height) and faults in the shape are also detected, using the profile measurement. There are four types of shape faults that are measured; twist, spring, bow and cup. These are given in deviation from 0 (in mm).

The quality inspection system processes information from the measurement devices using a program built in C#. The implementation and testing of classification models has been done in Matlab, so all the necessary information from each board has been exported from the quality inspection program into Matlab. Some of the board features have been imported directly from the quality inspection program, and some have been calculated from the different data channels, after importing the data to Matlab.

2.2 Features for Board Classification

The intention is to use as much of the information already available in the system as possible. Primarily because all additional computation will take extra time from the system, but also because the implementation of new features is time consuming for this project.

Some new features will however be taken from the data. Since the classification is based on strength grades, features that correlate with wood strength would be useful here. Wood strength has been proven to be dependent on the fibre angles, among others. In [3], the standard deviation of fibre angles is estimated to have a correlation to wood strength (Modulus of Rupture) of r=-0.53. Also the mean fibre angle has an estimated correlation of r=0.28 and the mean laser area a correlation of r=0.29 with the strength. Other properties that have an effect on wood strength are annual ring width and density. The density is not available in the quality inspection system so it cannot be used. The annual ring width could possibly be found by image analysis, but this will not be attempted in this project.

Considering the strength correlations and what features that easily can be extracted from the data – the mean, standard deviation and median from the different data channels will be computed. For the profile channel the mean and median gives nearly the same information as the width and height measurements, so only the standard deviation will be used. For the fibre angle channel the mean of the absolute values will be used instead of the mean. A merging of the color channels into a gray scale channel will also be added.

Apart from computing these new features some of the already available data will need to be adjusted before being used in a classification model. Since the boards can be of different sizes and dimensions, some of the information will need to be scaled so that the measurements are independent of the size of the board. The dimension of the board will not be taken directly as a feature. The ratio between the measured dimension and the nominal dimension will be used instead, to get a measure of how much the board differs from its nominal dimension.

The defect detection is customized for specific sawmills, so the identified defects are not very good to use in a statistical model that is supposed to work at different sawmills. The defect regions are however found in the same way at all sawmills and are therefore more appropriate. The number of defect regions and the total size of these regions will be scaled so that they are equivalent for all sizes of boards, i.e. they will be divided by the surface size of the board. The shape faults can be used as they are, as they are not dependent on the dimension.

To summarize, the following features are taken from each board to be used in a statistical classification model:

- Shape properties; twist, spring, bow, cup.
- Width and height deviation.
- Number of defect regions found for visual, laser point and profile maps, along with the mean size of these regions.
- Mean, standard deviation and median for the 7 channels.
- Standard deviation for the profile channel.

In total this gives 34 features that could be used as explanatory variables in a classification model.

2.2.1 Extracting Features

For every board there is one matrix of data for each side of the board and each data channel. This sums up to 4·8=32 matrices of data for each board. To reduce the number of features the data matrices from each data type have been merged together, creating one matrix of data for each data type.

Some modifications to the data were made before extracting the features. One thing that was noticed was that the laser measurements near the edges of the board often showed a much higher deviation than on the rest of the board. Table 2.1 shows part of the data for the fibre angles near an upper edge.

---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---
---	-12	-12	-5	71	71	---	---	---	-31	-31	---	---	---	---	---	---
-2	0	0	2	16	16	2	2	5	5	5	-24	-24	-22	21	21	-2
0	4	4	-4	5	5	-1	-1	2	0	0	-24	-24	-11	21	21	-5
2	4	4	-5	-8	-8	0	0	1	6	6	1	1	-1	-9	-9	2
-1	-1	-1	0	-1	-1	-1	-1	0	-1	-1	1	1	-1	-2	-2	1
-1	-1	-1	0	-1	-1	-1	-1	0	-1	-1	1	1	-1	-2	-2	1
-2	0	0	0	-1	-1	-3	-3	-2	-3	-3	0	0	0	0	0	4
1	1	1	-5	0	0	-1	-1	0	-1	-1	-1	-1	-1	-2	-2	-1
0	0	0	-1	0	0	1	1	-2	1	1	-1	-1	-1	-1	-1	1
1	0	0	0	-2	-2	-1	-1	-1	-3	-3	1	1	0	-1	-1	0
-1	0	0	4	-1	-1	0	0	-1	1	1	-2	-2	-4	-1	-1	0
0	0	0	0	1	1	0	0	-1	1	1	1	1	1	-1	-1	0

Table 2.1: Fibre angle values on the edge of a board.

For the fibre angles the measurements vary a lot near the edges, but a bit further in the board it stays near 0 unless there is a defect. This could lead to inaccurate estimates of the features that are to be calculated. To solve this potential issue the 5 pixels that are closest to the edges have been removed from all data matrices belonging to the laser measurements.

Another thing that was noted was that there often were errors in the measurements, possibly because of dirt on the measurement devices. Figure 2.6 shows the laser area measurement from one side of a board.

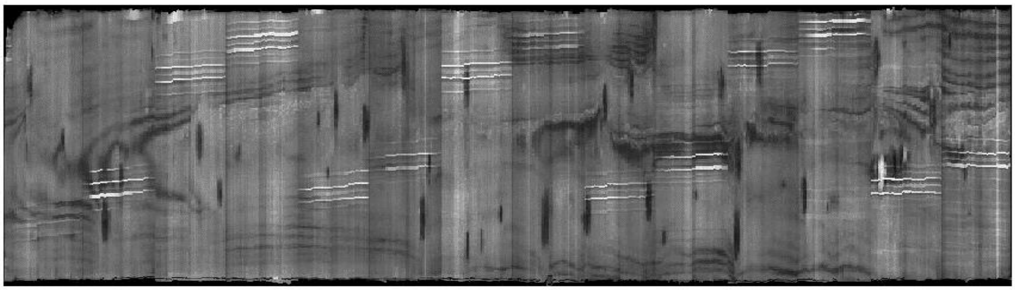

Figure 2.6: Error on laser area measurement.

The pixels corresponding to the white lines have much higher values than the rest of the data, possibly resulting in inaccurate calculations of the mean and standard deviation. To deal with this problem all pixels that are above a certain level have been set equal to the median of that data channel. Figure 2.7 shows the result from this.

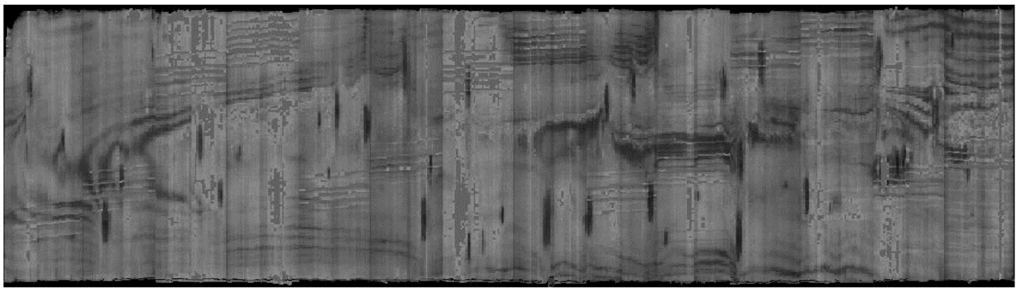

Figure 2.7: Error on laser area measurement reduced.

2.3 Strength Classes

As mentioned in Chapter 1 the classification methods will be tested on board strength classes. These classes are obtained from a strength grading machine, which measures the resonance frequency of the board combined with the density. This gives an estimate of the modulus of elasticity of the boards, which is used for classification by setting limits for minimum value for each class. For example, one class may contain all boards with an estimated modulus of elasticity between 10000 and 15000 MPa. It is thus not actually the strength (i.e. modulus of rupture) that is measured, but the stiffness. This since measuring the actual strength would require breaking the boards. The boards in this project were strength classified according to the classification system in the European Standard EN-14081. Two strength classes were obtained – C30 and C24, where boards in the C30 class have a higher quality than those in the C24 class. There is also a Reject class, which contains boards that do not meet the requirements of the other classes.

3 Theory

This chapter contains the theory used in this thesis. The first section describes basic classification theory and methods for comparing and selecting classification models. The second section describes the classification models used, as well as methods for fitting the models to data.

3.1 Classification

Classification in statistics means constructing a model based on past observations which can make decisions regarding the class belonging of new observations. A simple classification example is given in figure 3.1, which shows data points from two classes together with a linear decision boundary.

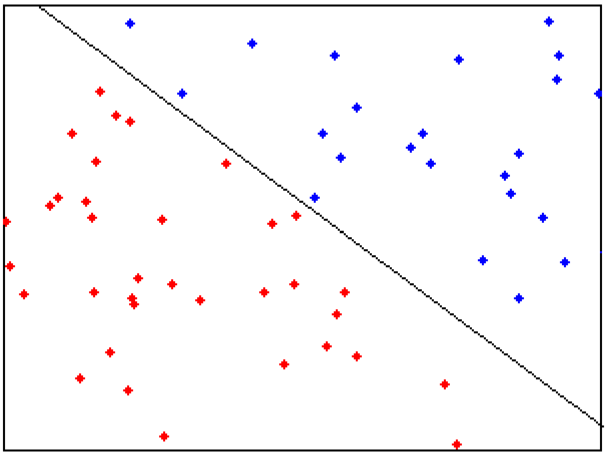

Figure 3.1: Two classes of data separated by a linear decision boundary.

The decision boundary is created so that it separates the data points from the two classes. A new observation is then classified based on which side of the decision boundary it is on. More formally the class belonging can be expressed as a categorical variable taking values in {1,2,3,...,K}, where K is the number of classes. Let Y be the categorical variable and **X** be a vector of explanatory variables, i.e.

$$Y \in \{1,2,3,\dots,K\},$$
$$\mathbf{X} = \left[x_1, x_2, x_3, \dots, x_p\right]$$

where p is the number of variables. Classification uses the feature vector, **X,** to predict the class, Y_0. There are a variety of different models that can be used to solve classification problems. The following sections describe methods for training models and for comparing the classification outcome of different models.

3.1.1 Evaluating Classification Models

A simple way of comparing the performance of different classification models is by calculating the misclassification error, i.e. the proportion of data points classified incorrectly. However, to analyse the classification outcome more thoroughly a confusion matrix can be used. It displays the classification outcome in a matrix where the columns represent the predicted classes and the rows represent the actual classes. Fig. 3.2 shows a confusion matrix for three classes; C1, C2, C3. Ideally the diagonal is the only one containing non-zero elements, i.e. none of the observations are incorrectly classified.

		Predicted class		
		C1	C2	C3
Actual class	C1			
	C2			
	C3			

Figure 3.2: Confusion matrix for the classes C1, C2 and C3.

3.1.2 Cross-validation

When dealing with limited amount of data and many explanatory variables there is a risk of overfitting the model, i.e. the model fits well to the training data but not very well when applied to unseen data. A way to avoid overfitting is to perform cross-validation. When training a model the data is divided into training data and test data, where the model is trained using the training data and the test data is used to determine if the model performs well on data not seen in training. K-fold cross-validation means dividing the data into K groups of equal size, and using one of the groups as test data and the other K-1 as training data. The test group is then alternated among the K groups giving K different estimates of the classification error for unseen data. The training error can be defined as

$$\overline{err} = \frac{1}{N}\sum_{i=1}^{N} L\left(Y,\hat{Y}(\boldsymbol{X})\right)$$

where $L\left(Y,\hat{Y}(\boldsymbol{X})\right)$ is a loss function and N is the number of observations. An example of a loss function is

$$L\left(Y,\hat{Y}(\boldsymbol{X})\right) = I\left(Y \neq \hat{Y}(\boldsymbol{X})\right)$$

which gives 0 if a classification is correct and 1 if it is wrong. The training error is thus the loss function averaged over the number of observations. The test error can be defined as

$$Err_{\tau} = E\left[L\left(Y,\hat{Y}(\boldsymbol{X})\right)|\tau\right]$$

i.e. the expected value of the loss function, given τ, the specific training data used. As the complexity of a model increases, the training error will always decrease. However, the test error will not always decrease. As model complexity becomes sufficiently large overfitting will start to increase the test error. Choosing a model can therefore be done by finding the model that minimizes the test error. Fig. 3.3 illustrates the relationship between model complexity and test- or training error.

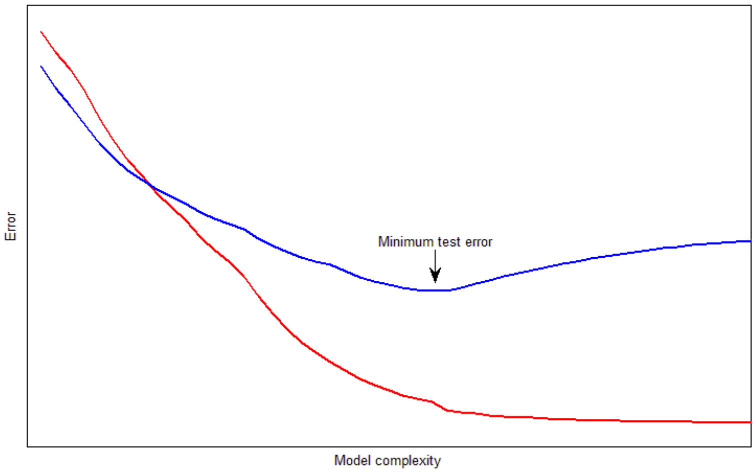

Figure 3.3: Training error in red and test error in blue. On the horizontal axis is model complexity and on the vertical axis is error.

As described in Chapter 5 the data in this project is divided into three sets. Cross-validation is only performed on the data called “Training set”. The training and test data described in this section should not be confused with the “Training set” and “Test set” mentioned in Chapter 5.

3.2 Classification Models

3.2.1 Logistic regression

Logistic regression is a classification model that uses logistic functions to produce the probabilities of different outcomes. A logistic function has the form

$$f(x) = \frac{1}{1+e^{-x}}$$

For values in $-\infty < x < \infty$, the logistic function produces a value between 0 and 1, which can be used to express a probability. The following description of logistic regression and multinomial logistic regression follows the outline in [4]. When distinguishing between two classes, i.e. a binary classification, the logistic regression model has the form

$$\ln\frac{P(Y=1|X=x)}{P(Y=2|X=x)}=\beta_0+\beta^T x$$

To calculate the odds of a specific outcome, this can be re-written into the logistic functions (using the fact that $P(Y=1)+P(Y=2)=1$).

$$P(Y=1|X=x)=\frac{1}{1+e^{-(\beta+\beta^T x)}}$$

$$P(Y=2|X=x)=\frac{1}{1+e^{\beta+\beta^T x}}$$

For $K>2$ classes the logistic regression model can be replaced by a multinomial logistic regression model. Using the K:th class as denominator when calculating the probabilities of the other classes the model becomes

$$\ln\frac{P(Y=1|X=x)}{P(Y=K|X=x)}=\beta_{10}+\beta_1^T x$$

$$\ln\frac{P(Y=2|X=x)}{P(Y=K|X=x)}=\beta_{20}+\beta_2^T x$$

$$\vdots$$

$$\ln\frac{P(Y=K-1|X=x)}{P(Y=K|X=x)}=\beta_{(K-1)0}+\beta_{K-1}^T x$$

The probability of any class $k<K$ is then

$$P(Y=k|X=x)=P(Y=K|X=x)e^{\beta_{k0}+\beta_k^T}$$

Since all class probabilities should sum up to 1, the probability of class K is given by

$$P(Y=K|X=x)=\frac{1}{1+\sum_{i=1}^{K-1}e^{\beta_{i0}+\beta_i^T x}}$$

Thus the probabilities for a class $k<K$ can be expressed as

$$P(Y=k|X=x)=\frac{e^{\beta_{k0}+\beta_k^T x}}{1+\sum_{i=1}^{K-1}e^{\beta_{i0}+\beta_i^T x}}$$

For the K classes this results in K-1 sets of coefficients,
$\beta_{10}+\beta_1^T, \beta_{20}+\beta_2^T, \ldots, \beta_{(K-1)0}+\beta_{K-1}^T$.

3.2.2 Proportional Odds Model

The proportional odds model can be used when there is a specific ordering between the classes (e.g. small, medium, large). It models the probability of an observation being less than or equal to a certain category. Denoting the classes as Y_i, where $Y_1 < Y_2 < \cdots < Y_k$, the model is

$$ln\left(\frac{P(y \le Y_1)}{P(y > Y_1)}\right) = \beta_{01} + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p$$

$$ln\left(\frac{P(y \le Y_2)}{P(y > Y_2)}\right) = \beta_{02} + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p$$

$$\vdots$$

$$ln\left(\frac{P(y \le Y_{k-1})}{P(y > Y_{k-1})}\right) = \beta_{0(k-1)} + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p$$

The model uses the same β-coefficients throughout, except for the β_0:s (intercepts). This results in K-1 parallel separating hyperplanes. Figure 3.4 shows a separation of 100 data points into three classes by two decision boundaries created by a proportional odds model.

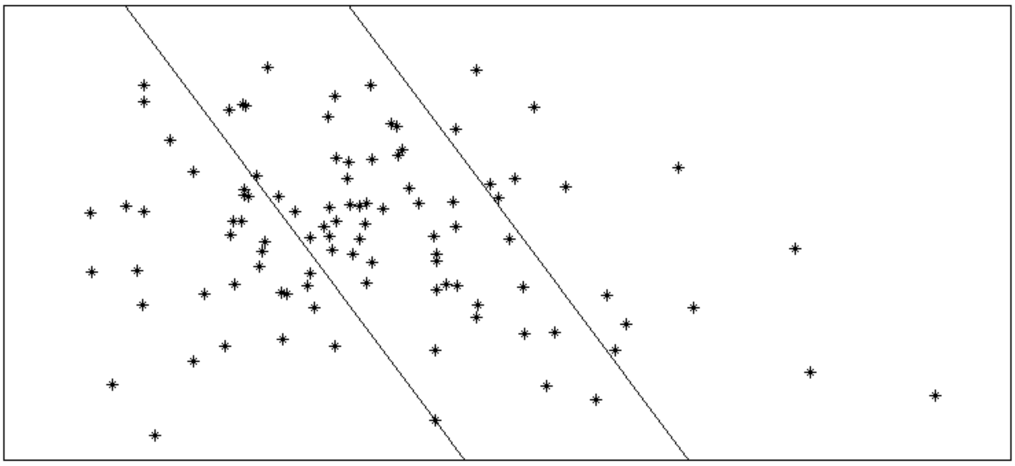

Figure 3.4: Two-dimensional separation into three classes by proportional odds model.

3.2.3 Lasso

To deal with overfitting on a logistic regression model, regularization can be applied, i.e. a penalty that increases with the complexity of the model. Lasso is a method for regularization introduced in [5], and stands for “Least Absolute Shrinkage and Selection Operator”. The idea of lasso is to constrain the sum of the absolute values of the coefficients in the model. Depending on how much the model is constrained, some of the coefficients will be forced to zero, and the corresponding variables are thereby removed from the model. This is also called L_1-regularization.

For an Ordinary Least Squares model the Lasso is defined as

$$(\widehat{\beta_0}, \hat{\beta}) = \arg\min \left\{ \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_j \beta_{ij} x_{ij} \right)^2 \right\} \qquad \text{subject to } \sum_{j=1}^{p} |\beta_j| \leq t$$

where β is the vector of coefficients, β_0 is the intercept, y_i are the responses from the underlying process, x_i are the explanatory variables and t is a constraint. The above expression is equivalent to, see [4] for details,

$$(\widehat{\beta_0}, \hat{\beta}) = \arg\min \left\{ \frac{1}{2} \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_j \beta_{ij} x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}$$

where λ is a penalty term on the sum of the absolute values of the coefficients. When applying lasso on a logistic regression model the probabilities for $K > 2$ classes can be written as

$$P(Y = k | X = x) = \frac{e^{\beta_{k0} + \beta_k^T x}}{\sum_{i=1}^{K} e^{\beta_{i0} + \beta_i^T x}}$$

This expression is however not estimable without constraints [6]. The log-likelihood function to be maximized for fitting the coefficients is then

$$l(\theta) = \sum_{i=1}^{N} ln \left(\frac{e^{\beta_{y_i0} + \beta_{y_i}^T x_i}}{\sum_{l=1}^{K} e^{\beta_{l0} + \beta_l^T x_i}} \right) - \lambda \sum_{j=1}^{p} |\beta_j|$$

The parameter λ can be tuned using cross-validation, by finding the value of λ that minimizes the error of the model. In figure 3.5 is a plot of the cross-validated mean error at different values of λ. The corresponding standard deviation is also shown.

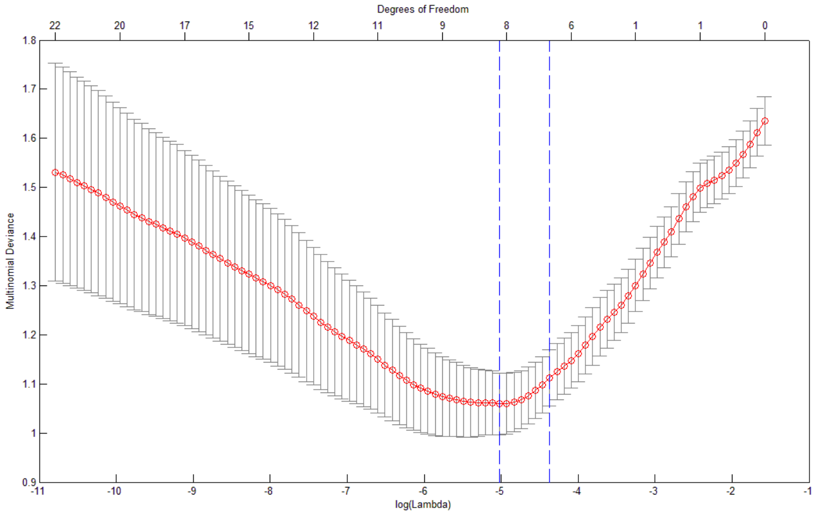

Figure 3.5: Cross-validated error for a sequence of λ-values. On the x-axis is model complexity and on the y-axis is model error.

A larger value of λ (to the right) means the model is more regularized, i.e. smaller. The left dotted vertical line corresponds to the model with the minimum error, and the right dotted line corresponds to the model chosen using the "one standard error"-rule. The idea of this rule is to choose the smallest model where the mean error is within one standard deviation of the minimum error [4]. Since there is an uncertainty in the error estimates, choosing a less complex model is a safer approach.

3.2.4 Support Vector Machines

Support Vector Machines (SVM) is a method of classification introduced in [7]. It is used to separate data points from two classes by finding a hyperplane that makes the margin between the two classes as large as possible. The hyperplane can be defined as

$$\boldsymbol{w} \cdot \boldsymbol{x} - b = 0$$

where $\boldsymbol{x} = \{x_1, x_2, .., x_p\}$ is the feature vector and $\boldsymbol{w}$ is a vector of weights. Denoting the class as $y_i \in \{-1,1\}$ for the data points $(\boldsymbol{x}_i, y_i)$, a new observation is classified based on the decision function

$$\begin{array}{c} -1 \text{ if } \boldsymbol{w} \cdot \boldsymbol{x} - b < 0 \\ 1 \text{ if } \boldsymbol{w} \cdot \boldsymbol{x} - b > 0 \end{array}$$

The size of the vector $\boldsymbol{w}$ is chosen so that the distance from the hyperplane to the nearest data points is $\frac{1}{\|\boldsymbol{w}\|}$. The size of the margin is thus $\frac{2}{\|\boldsymbol{w}\|}$. Figure 3.6 [8] shows a separation of two classes in two dimensions.

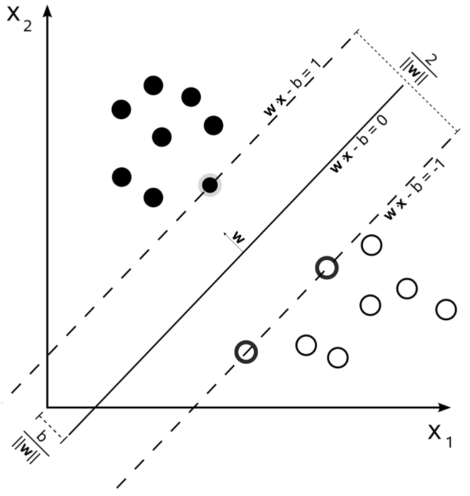

Figure 3.6: Separation of two classes by maximizing the margin.

Only some of the data points influence the positioning of the decision boundary here. These are called the support vectors, and lie on the margin in this case.

If the classes are not separable another approach was developed in [7]. The idea is to allow some of the training data to be on the wrong side of the margin, and instead add a penalty for these misclassifications. This is done by introducing the slack variables $\xi = \{\xi_1, \xi_2, \dots, \xi_N\}$, which represent how much on the wrong side of the margin each training point is. The object is then to minimize the sum of the errors $\sum_{i=1}^{N} \xi_i$ subject to

$$y_i(\boldsymbol{w} \cdot \boldsymbol{x}_\mathrm{i} + b) \geq 1 - \xi_i \qquad \xi_i \geq 0$$

So in this case not only the points on the margin have an effect on the decision boundary, but also the points that are on the wrong side of the margin. The optimal separating hyperplane is chosen to minimize

$$\frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{N} \xi_i \qquad (1)$$

where C is a constant defining the penalty on misclassified training data. When optimizing the expression in (1) for a given dataset the Lagrange primal function

$$L_P = \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{N} \xi_i - \sum_{i=1}^{N} \alpha_i\big[y_i\big(x_i^T w + b\big) - (1 - \xi_i)\big] - \sum_{i=1}^{N} \mu_i\, \xi_i$$

can be used [8], where α are the Lagrange multipliers. This expression is minimized with respect to w, b and ξ_i. However, to make the optimization simpler this can be reformulated into maximizing the dual Lagrange function

$$L_D = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j x_i x_j^T \qquad (2)$$

subject to $\sum_{i=1}^{N} \alpha_i y_i = 0$ and $0 \leq \alpha_i \leq C$. The expression in (2) uses the fact that

$$\boldsymbol{w} = \sum_{i=1}^{N} \alpha_i y_i x_i$$

$$0 = \sum_{i=1}^{N} \alpha_i y_i$$

$$\alpha_i = C - \mu_i$$

when the derivatives for w, b and ξ_i are set to zero in the primal Lagrange function.

3.2.4.1 Kernels

The class boundaries created with SVM are linear in the input feature space. Using linear boundaries makes calculations simpler and decreases the computation time. There may, however, be cases where a good separation of the classes cannot be obtained with a linear boundary. SVM deals with this by transforming the input space into a new feature space, where the usual SVM approach for finding an optimal separating hyperplane can be used. Using linear class boundaries in the new feature space then results in non-linear boundaries when transformed back to the original feature space. The new feature space is created by transforming the input vector by an N-dimensional function $\phi(\boldsymbol{x}) = \phi_1(\boldsymbol{x}), \phi_2(\boldsymbol{x}), \dots, \emptyset_N(\boldsymbol{x})$. A new observation is then classified as

$$-1 \text{ if } \boldsymbol{w} \cdot \phi(\boldsymbol{x}) - b < 0$$
$$1 \text{ if } \boldsymbol{w} \cdot \phi(\boldsymbol{x}) - b > 0$$

Finding the optimal separating hyperplane is done by maximizing (2) as before, but replacing $x_i x_j^T$ with a “kernel function” - $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$. A kernel function is a dot product of input data transformed into the new feature space, i.e. $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \phi(\boldsymbol{x}_i) \cdot \phi(\boldsymbol{x}_j)$. The function to be maximized is then

$$L_D = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) \qquad (3)$$

Three examples of kernel functions are

Linear kernel function: $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i \boldsymbol{x}_j^T$

Quadratic kernel function: $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = (1 + (\boldsymbol{x}_i \boldsymbol{x}_j^T))^2$

Radial basis kernel function: $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = exp(-\gamma \|\boldsymbol{x}_i - \boldsymbol{x}_j^T\|^2)$

3.2.4.2 Tuning Parameters

The parameters C and γ can be tuned by minimizing (3) over a search grid, which means searching through all the combinations of C and γ from given sets of possible values (for example C=1,2,..,10, γ=1,2,..,10). The values that produce the smallest test-error can then be found by cross-validation. In [9] it is recommended that exponentially growing sequences are used for the grid search, e.g. $\{2^{-5},2^{-4},...,2^{5}\}$.

3.2.4.3 SVM for Multiple Classes

SVM is constructed as a binary classifier, but classifying into more than two classes can be done by combining several binary classifiers. One way is a “one-vs-all”-approach, where K classifiers are created (K=number of classes), and for each classifier one of the classes is given one label and the rest of the classes the other label. A new observation is assigned to the class with the largest value in the decision function. Another approach is a “one-vs-one”-classification, where $K(K$-1)/2 classifiers are trained, one for each pair of classes. Classifying a new observation is then done by counting the number of times the observation is assigned into each class. The final assigned class is the one that gets the most “votes”. Other approaches to classify into multiple classes have also been suggested, such as the DAGSVM [10].

4 Implementation

4.1 Multinomial Logistic Regression in Matlab

For the multinomial logistic regression and the proportional odds model the built-in Matlab function **mnrfit** has been used to find the coefficients in the models. In the case of three classes and a multinomial logistic regression **mnrfit** produces two sets of β-coefficients. The coefficients are used to compute the probabilities of a new observation belonging to each of the classes, as described in section 3.2.1. For the proportional odds model one set of β-coefficients and two intercepts are produced, as described in section 3.2.2. To predict the outcome of unseen data the function **mnrval** has been used. The regularized logistic regression models have been implemented using the Glmnet package [11] for Matlab, described below.

4.2 Glmnet

Glmnet is a package for Matlab and R which is used to produce penalized models based on the algorithms described in [11]. It can be used for a number of different models, e.g. linear regression, logistic regression and multinomial logistic regression. Glmnet produces a sequence of λ-values, and for each λ it computes the coefficients of β that produces the optimal model for that particular λ. For a multinomial logistic regression model it produces K sequences of β-coefficients, where K is the number of classes.

The value of λ that corresponds to the model with the minimum error is then found by cross-validation on the dataset, using the **cvglmnet** function. Cross-validation also produces the standard deviation of the model for each λ. The plots of cross-validated error in this thesis, such as fig. 3.5 were all produced with Glmnet.

4.3 SVM in Matlab

For the support vector machines the Matlab function **svmtrain** has been used for training and the function **svmclassify** for classification of unseen data. As described in chapter 3 SVM is a binary classifier, so when classifying into three classes the one-vs-one approach has been used, producing three binary classifiers. The possibility that all classes gets one “vote” has been taken into account by letting those cases be classified as the middle class, C24. Three types of kernels have been tested on the data, linear, radial basis and quadratic. Figure 4.1 shows an example of a separation of 100 data points into two classes using an SVM classifier with quadratic kernel function. The data comes from two board strength classes.

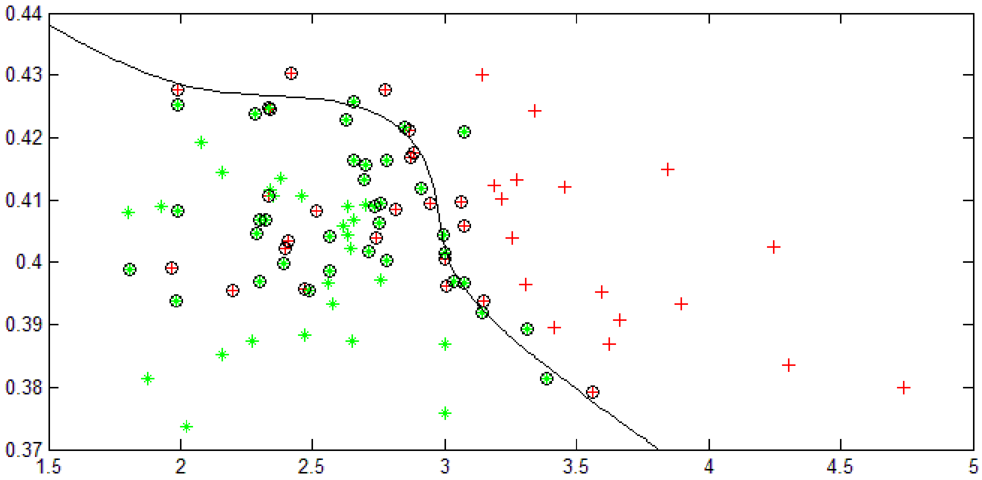

Figure 4.1: Two-dimensional SVM classification produced by the Matlab function svmtrain. On the x-axis is *mean absolute value of fibre angles* and on the y-axis is *mean gray*. The data points with a circle around them are the support vectors.

5 Results

5.1 Test Setup

Data from 1000 boards in two different dimensions have been collected. 500 boards of the dimension 50x100 mm (height x width), and 500 of the dimension 50x150 mm. All data comes from the same sawmill and the wood species is spruce. The data has been divided into three sets; training set, validation set and test set. The training set will be used in training the models, the validation set will be used to compare the outcome of different models, and the test set will be used to test the final model on unseen data. Table 5.1 shows the dimension of the boards in the different sets.

	50x100	**50x150**	**Total**
Training set	250	250	500
Validation set	125	125	250
Test set	125	125	250

Table 5.1: Dimension on collected boards divided into training, validation, and test set.

Shown in table 5.2 is the distribution of the boards among the three classes – C30, C24 and R (Reject).

	C30	**C24**	**R**	**Total**
Training set	222	256	22	500
Validation set	130	114	6	250
Test set	125	117	8	250
Whole set	477	487	36	1000

Table 5.2: Distribution of boards in the classes C30, C24 and R.

5.2 Choice of Classification Methods

When selecting classification methods for this project some aspects must be considered. A main requirement is that the classification of a board should be fast, since there is a limited time before a decision has to be made. It is therefore important that the actual classification is fast, but it is also good if the number of features can be reduced so that less computation has to be made. The features used now are not very time consuming to compute, or they are already being computed in the system for other purposes, so this will not be an issue here. However, if more time consuming features were added to the classification it would be very good to be able to choose a subset of those features. It is also important that there is a way to deal with possible overfitting, which is a risk here since the data is limited (especially for the Reject class) and the number of explanatory variables is quite large.

Two types of classification methods were chosen for this project – the logistic regression model and support vector machines. The logistic regression model was chosen because it is simple to implement and has good possibilities of regularization. It can also be modified into the proportional odds model, which will be tested here since there is a natural ordering between the classes. The fact that the Reject class has much fewer observations might also be a good reason to use the proportional odds model. This means the information from the classification between the C24 and C30 classes will help in creating the decision boundary between the Reject and C24 classes. The support vector machine was chosen because it is very adaptable. By using different kernels and different values of the penalty parameter C it can be adjusted to fit many types of data.

The results of the different models are compared using confusion matrices and misclassification error, i.e. the ratio of boards wrongly classified. Since there are more boards in some classes the overall misclassification error may be misleading, i.e. the misclassification error of a small class will not affect the overall misclassification error very much. This will be taken into account when comparing the models. Another possibly negative aspect of only using the misclassification error is that it is worse if a board from the C30 class is classified as a Reject than if a board from the C24 class is classified as a Reject, since the difference between the C30 and the Reject class should be greater. This will also be taken into account when comparing the models.

5.3 Test of Classification Models

5.3.1 Logistic Regression

Fitting a multinomial logistic regression model to the training set produced the following results.

Training set

	C30	C24	R
C30	189	33	0
C24	33	218	5
R	0	9	13

Misclassification error: 0.16

Validation set

	C30	C24	R
C30	91	39	0
C24	29	84	1
R	2	3	1

Misclassification error: 0.2960

The misclassification error is significantly larger on the validation set, which could imply on model overfit.

5.3.2 Regularized Logistic Regression

To reduce the model size and avoid overfitting, the Lasso method described in chapter 3.6 has been used to regularize the multinomial logistic regression model. A 20-fold cross-validation was applied to find the value of λ that minimizes the error of the model. The classification results for the model are

Training set

	C30	C24	R
C30	184	38	0
C24	34	222	0
R	0	16	6

Misclassification error: 0.1760

Validation set

	C30	C24	R
C30	91	39	0
C24	22	92	0
R	1	4	1

Misclassification error: 0.2640

Multinomial deviance is used as a measure of error on this model. The multinomial deviance is a measure on how much the fitted model differs from a perfectly fitted model. Figure 5.1 shows a plot of the cross-validated deviance at different values of λ. The expected deviance of the model is at its minimum when regularized. However, too much regularization gives a larger deviance.

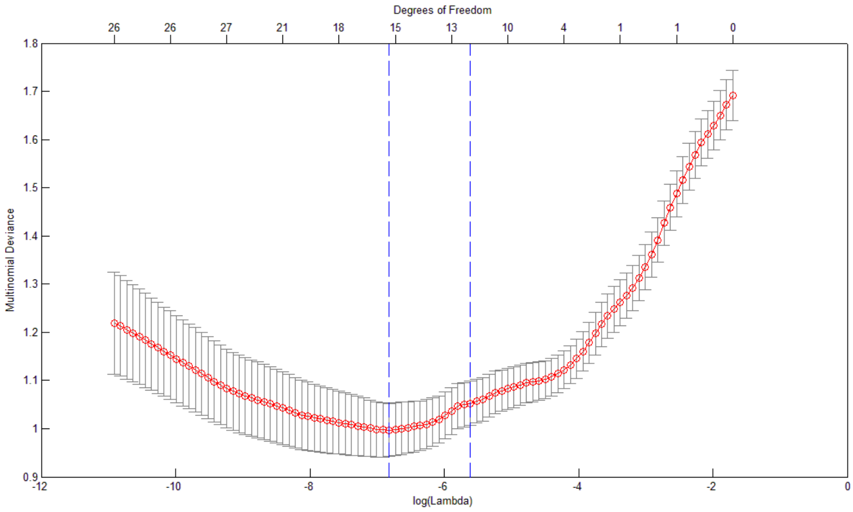

Figure 5.1: Regularization of a Multinomial Logistic Regression model. On the horizontal axis is the size of the penalty parameter λ, and on the vertical axis is Multinomial deviance, i.e. the error of the model.

The minimum deviance found was 0.9972 with the corresponding λ = 0.0011. The standard deviation for this deviance is 0.0555. Using the "one standard-error"-rule described in section 3.2.3 leads to a λ-value of 0.0037, corresponding to a deviance of 1.0526. The β-values for this model are non-zero for 18 of the 34 features for class C30, 13 of the features for class C24 and 9 of the features for the Reject class. This means the model uses 18 features for calculating the probability of class C30, 13 features for C24 and 9 features for R. The C30 class has 9 features in common with the C24 class, and the R class has only 3 features in common with both of the other classes. It thus seems that the C30 and C24 classes use quite similar features, but R uses mostly features that none of the other classes uses. Looking at the sequence of β-coefficients it can be seen that the first feature that is non-zero for the C30-class is *standard deviation of fibre angles*. The first feature that is non-zero for the C24-class is the *mean absolute value of fibre angles*. Both of these are expected since the fibre angles have a proven correlation with wood strength [3]. For the R class however, the first feature that is non-zero is the *number of visual defect regions*. The *standard deviation of fibre angles* and the *mean absolute value of fibre angles* are not among the 9 features used for the R class at all. An explanation to this could be found in fig 5.2, which shows the *standard deviation of fibre angles* plotted against the *mean laser scatter* of the training set. These features are among the first used for calculating the probabilities of C30 and C24, but not used at all for R.

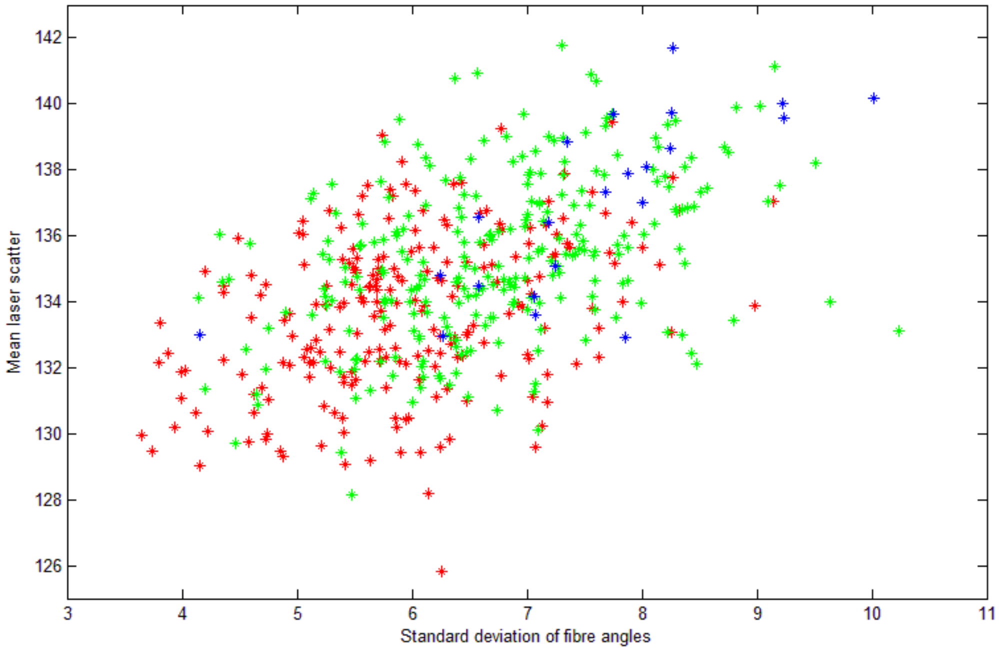

Figure 5.2: Standard deviation of fibre angles vs. mean laser scatter. C30 shown in red, C24 shown in green, and R shown in blue.

From the C30 and C24 classes in fig. 5.2 it seems that a lower value of *standard deviation of fibre angles*, and a lower value of *mean scatter* corresponds to a higher strength class, and vice versa. There is also a little tendency for the R class to have high *standard deviation of fibre angles* and a high *mean laser scatter*-value, but the data points are quite few and quite spread out. It is therefore probably difficult for a linear classifier such as logistic regression to distinguish the R class from the rest of the data based on these two features.

Another thing that can be noted in the β-value sequences is that when two features have high correlation one of them is often zero while the other is non-zero. This is probably because little extra information is gained from using both of them at the same time. For example, in the β-coefficients for the C30 class corresponding to the model chosen with the “one standard error”-rule, none of the median values for the color channels are used and none of the values for the green channel at all. As shown in table 5.1 the color channels are highly correlated, and the median values have very high correlation with the mean values.

	mR	sdR	mdR	mG	sdG	mdG	mB	sdB	mdB
mR	1	0.04	0.98	0.99	0.05	0.98	0.98	0.09	0.97
sdR	0.04	1	0.01	0.03	0.99	-0.02	-0.01	0.97	-0.06
mdR	0.98	0.01	1	0.97	0.02	0.98	0.96	0.06	0.97
mG	0.99	0.03	0.97	1	0.03	0.98	0.99	0.08	0.98
sdG	0.05	0.99	0.02	0.03	1	-0.01	0.01	0.99	-0.04
mdG	0.98	-0.02	0.98	0.98	-0.01	1	0.98	0.03	0.98
mB	0.98	-0.01	0.96	0.99	0.01	0.98	1	0.05	0.98
sdB	0.09	0.97	0.06	0.08	0.99	0.03	0.05	1	0.01
mdB	0.97	-0.06	0.97	0.98	-0.04	0.98	0.98	0.01	1

Table 5.1: Correlation of color parameters (m=mean, sd=standard deviation, md=median, R=red, G=green, B=blue).

Another approach for a regularized logistic regression model is to find the value of λ that minimizes the expected misclassification error of the model. Figure 5.3 shows the misclassification error at different values of λ, as before produced by a 20-fold cross-validation.

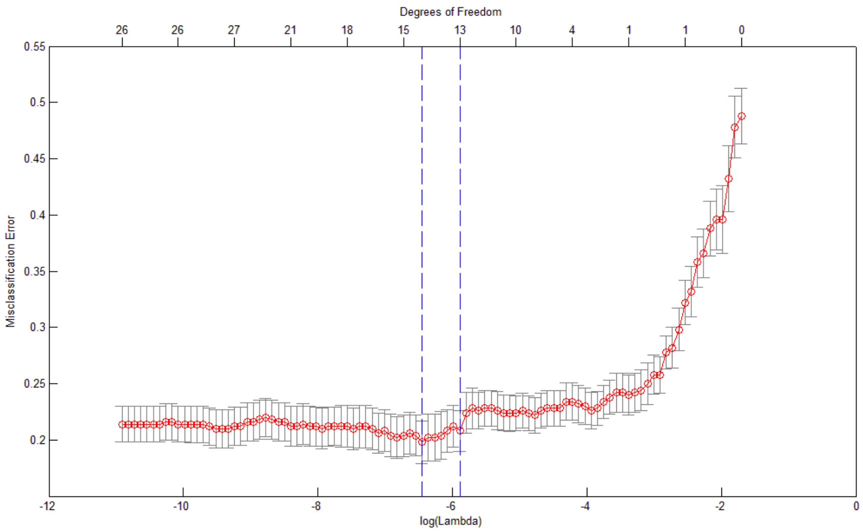

Figure 5.3: Regularization of a Multinomial Logistic Regression model. On the horizontal axis is the size of the penalty parameter λ, and on the vertical axis is the misclassification error.

The figure shows some sudden larger increases in the misclassification error as the model gets more regularized. This could be explained by the removal of explanatory variables that had a positive effect on the classification.

The minimum expected error is 0.19, found at λ=0.0019 with standard deviation 0.0161. The most regularized model within one standard deviation of this is the one found at λ=0.0028, with expected misclassification error 0.2040. The classification result for this model is

Training set

	C30	C24	R
C30	187	35	0
C24	34	220	2
R	0	16	6

Misclassification error: 0.1740

Validation set

	C30	C24	R
C30	90	40	0
C24	22	92	0
R	1	4	1

Misclassification error: 0.2680

5.3.3 Proportional Odds Model

Taking into account that C30 is the finest, C24 the medium and Reject is the worst quality, the proportional odds model might be more appropriate than the multinomial models above. The model produced the following results.

Training set

	C30	C24	R
C30	193	29	0
C24	32	220	4
R	0	13	9

Misclassification error: 0.1560

Validation set

	C30	C24	R
C30	94	36	0
C24	31	83	0
R	2	3	1

Misclassification error: 0.2880

The outcome of the Proportional Odds model is very similar to the other Logistic Regression models. There are no indications that the model produces a better result.

5.3.4 Support Vector Machines

Linear Kernel Function

Using a linear kernel function on the form $K(\boldsymbol{x}_i,\boldsymbol{x}_j) = \boldsymbol{x}_i^T\boldsymbol{x}_j$, performing a grid search on C in $\{2^{-15},2^{-14},\ldots,2^{15}\}$ and cross-validating on the training set the smallest error was found when $C = 2^{-2}$. The corresponding results are:

Training set

	C30	C24	R
C30	182	38	2
C24	33	184	39
R	0	1	21

Misclassification error: 0.1920

Validation set

	C30	C24	R
C30	89	38	3
C24	19	88	7
R	1	2	3

Misclassification error: 0.2800

Radial Basis Kernel Function

Using a radial basis kernel function on the form $K(\boldsymbol{x}_i,\boldsymbol{x}_j) = exp(-\gamma\left\|\boldsymbol{x}_i - \boldsymbol{x}_j^T\right\|^2)$, and performing a grid search on the parameters C and γ in $\{2^{-15},2^{-14},\ldots,2^{15}\}$ gives the smallest error when $C = 2^{10}$ and $\gamma = 2^5$. The corresponding classification results are

Training set

	C30	C24	R
C30	192	30	0
C24	20	215	21
R	0	1	22

Misclassification error: 0.1440

Validation set

	C30	C24	R
C30	92	37	1
C24	22	87	5
R	1	4	1

Misclassification error: 0.2800

Quadratic Kernel Function

Applying a quadratic kernel function, i.e. $K(\mathbf{x}_i,\mathbf{x}_j) = (1+(\mathbf{x}_i,\mathbf{x}_j^T))^2$, and using the same procedure as before produces the smallest error when $C=2^{-7}$.

Training set

	C30	C24	R
C30	196	25	1
C24	18	232	6
R	0	0	22

Misclassification error: 0.1000

Validation set

	C30	C24	R
C30	75	51	4
C24	28	83	3
R	1	5	0

Misclassification error: 0.3680

These results indicate an overfit since the training error is much smaller than the validation error. Testing on higher degrees of kernels produces almost perfect fits on the training set, but quite poor results when applied to unseen data.

5.4 Comparison of the results

There is not a significant difference between the results of the different models. The classification result on the Reject class is poor in most cases, possibly because of an insufficient amount of training data for that class. When comparing the misclassification error, the model that produced the best results on the validation data was the regularized logistic regression model. Applying the regularized logistic regression model on the "test set", which contains previously unseen data, gives the following results.

Regularized logistic regression – Test set

	C30	C24	R
C30	105	20	0
C24	31	83	3
R	0	5	3

Misclassification error: 0.2360

Taking into consideration the proportion of misclassifications of each class, the linear SVM was the best performing model. This was the only model that showed a tendency to classify the Reject class correctly.

Linear SVM – Test set

	C30	C24	R
C30	100	23	2
C24	31	70	16
R	0	3	5

Misclassification error: 0.30

Also on the test set the linear SVM classifies the Reject class better. It has a higher misclassification error, mainly because of more boards wrongly classified as Rejects.

6 Discussion

6.1 Conclusions

There is an uncertainty in the classification for all the models tested. There may be properties on a board which cannot be explained by the features from the quality inspection system, and there may also be an uncertainty in the strength grading machine from which the classes were obtained.

In the comparison of the models, the logistic regression models produced slightly better results on the misclassification error. However, most models do not give acceptable results classifying the Reject class. This is most likely due to the small amount of boards that belong to the class, limiting the training data. Making sure the Reject class has more data will probably make the classification of that class better. The linear SVM model classified the Reject class a bit better than the other models, at the expense of classifying more boards from the other classes as Rejects. The misclassification error is around 0.25-0.30 for almost all models. With this degree of accuracy, a classification from any of the models could be used as a decent prediction of strength grade, and probably help improve the interaction between the quality inspection system and a strength grader. It however seems to be too much uncertainty in the classification to be able to use the quality inspection system alone for strength grading.

The choice of method to use would ultimately be the Support Vector Machine classifier. The reason is that there are many ways to adjust the model to fit a particular data set. This makes it suitable for data sets where the class sizes differ a lot, and it also has a good potential to perform well on other types of classification problems, such as wood species.

6.2 Future work

To improve the classification results an initial approach is to make sure that the features that are extracted from the board data can be used to explain the type of classification that is made. In this case it is important that the features are dependent on the strength in some way. A way to make the classification better would therefore be to find out what other features can be extracted from the board data that has some correlation to wood strength.

The data set used comes from one sawmill and contains only two dimensions of the same wood species. For future work, it would be a good idea to use a more diversified data set to make sure that a classification model produces good results on different data. Using data from more than one sawmill is especially important since the properties of wood may differ a lot depending on the area of growth [12].

There are also possible improvements to make on the classification methods tested. A regularized version of the proportional odds model was not tested, and might lead to a good result. In the support vector machine classification the same parameters for λ and C were used for all three binary classifiers. It is possible that different values of these parameters for the different classifiers could produce better results, especially since the classes differ a lot in size.

Bibliography

[1] Föreningen Svenska Sågverksmän (FSS). *Nordiskt trä – Sorteringsregler*, 1994.

[2] Timber structures – Strength graded structural timber with rectangular cross section, EN-14081, European Committee for Standardization, 2012.

[3] M. Brännström, J. Manninen, J. Oja. Predicting the Strength of Sawn Wood by Tracheid Laser Scattering. *BioResources,* 3(2), 437-451.

[4] T. Hastie, R. Tibshirani and J. Friedman, *The Elements of Statistical Learning*. Springer, 2nd ed., 2009.

[5] R. Tibshirani, Regression Shrinkage and Selection via the Lasso, *Journal of the Royal Statistical Society*, Vol 58, Issue 1, 267-288, 1996.

[6] J. Friedman, T. Hastie and R. Tibshirani, Regularization Paths for Generalized Linear Models via Coordinate Descent. Stanford University, Department of Statistics, 2009.

[7] C. Cortes and V. Vapnik, Support-Vector Networks, *Machine Learning*, Vol 20, 273-297, 1995.

[8] http://en.wikipedia.org/wiki/Support_vector_machine#mediaviewer/ File:Svm_max_sep_hyperplane_with_margin.png, 2014-11-11.

[9] Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin, A Practical Guide to Support Vector Classification. National Taiwan University, Department of Computer Science, Taipei, 2003.

[10] J.C. Platt, N. Cristianini and J. Shawe-Taylor, Large Margin DAGs for Multiclass Classification. *Advances in Neural Information Processing Systems*, Vol 12, 547-553, MIT Press, 2000.

[11] J. Friedman et al, Package 'glmnet'. http://cran.r-project.org/web/packages/glmnet/glmnet.pdf, 2014-11-11.

[12] E. Saarman. *Träkunskap*. Sveriges Skogsindustriförbund, 1992.

Master’s Theses in Mathematical Sciences 2014:E60
ISSN 1404-6342

LUTFMS-3268-2014

Mathematical Statistics
Centre for Mathematical Sciences
Lund University
Box 118, SE-221 00 Lund, Sweden

http://www.maths.lth.se/