Decision Trees for Classification of Repeated Measurements

Department of Mathematics, Linköping University

Julianna Holmberg

LiTH-MAT-EX–2024/01–SE

Credits: **16 hp**

Level: **G2**

Supervisor: **Martin Singull**,
Department of Mathematics, Linköping University

Examiner: **Fredrik Berntsson**,
Department of Mathematics, Linköping University

Linköping: **January 2024**

Abstract

Classification of data from repeated measurements is useful in various disciplines, for example that of medicine. This thesis explores how classification trees (CART) can be used for classifying repeated measures data. The reader is introduced to variations of the CART algorithm which can be used for classifying the data set and tests the performance of these algorithms on a data set that can be modelled using bilinear regression. The performance is compared with that of a classification rule based on linear discriminant analysis. It is found that while the performance of the CART algorithm can be satisfactory, using linear discriminant analysis is more reliable for achieving good results.

Sammanfattning

Klassificering av data från upprepade mätningar är användbart inom olika discipliner, till exempel medicin. Denna uppsats undersöker hur klassificeringsträd (CART) kan användas för att klassificera upprepade mätningar. Läsaren introduceras till varianter av CART-algoritmen som kan användas för att klassificera datamängden och testar prestandan för dessa algoritmer på en datamängd som kan modelleras med hjälp av bilinjär regression. Prestandan jämförs med en klassificeringsregel baserad på linjär diskriminantanalys. Det har visar sig att även om prestandan för CART-algoritmen kan vara tillfredsställande, är användning av linjär diskriminantanalys mer tillförlitlig för att uppnå goda resultat.

Acknowledgements

I would like to thank my supervisor Martin Singull for suggesting the topic for this thesis and his continuous feedback throughout my work. I also want to thank Samuel Mossberg for his work as my opponent. Lastly, I thank my examiner Fredrik Berntsson for his comments which helped with the final adjustments of the thesis.

Contents

1 Introduction **1**

2 Theory **3**
2.1 Modelling the data 3
2.2 Decision Trees 5
2.2.1 Impurity function 5
2.2.2 Finding the best split of a node 6
2.2.3 Using a tree for classification 8
2.2.4 Finding the right sized tree 8

3 Results **11**
3.1 Simulation of data and CART algorithm using 20 samples 11
3.2 CART algorithm using 100 samples 15
3.3 CART algorithm on a higher number of samples 18
3.4 CART algorithm using more features 19

4 Discussion **21**

Chapter 1

Introduction

The classification problem refers to the problem of finding to which of a set of two classes an input vector belongs to [3]. A problem of this kind can be solved by constructing a function g, such that $g(\mathbf{x}) \in S$, where $\mathbf{x}$ is an observation vector of a sample with unknown class label and $S = \{1, 2\}$ is a set of class labels [1]. This function is called a classifier and is constructed based on training data, which is a set of measurements $(\mathbf{x}, y)$, where y is the class label of a sample $\mathbf{x}$.

The decision tree algorithm (CART), first introduced in 1984 by Breiman et al. [6], works by identifying a series of binary split points in a data set that are used to separate the data into different classes. Due to its simplicity and being easy to interpret, the algorithm is one of the most commonly used classification models [6]. The traditional CART algorithm as presented in [1] does not consider dependence between the variables, which in cases where the data consists of multiple measurements collected from the same samples or objects at different time stamps, and consequently does not capture this property. Collection of this kind of data, also referred to as repeated measurements data, is common in various disciplines, including medicine, finance and environmental studies. An example of this in the field of medicine is to classify tumours as benign or malignant based on its growth in size, as the classification of the tumour affects the required treatment for the patient.

Repeated measures data of multiple samples within the same class can be modelled according to the Growth Curve model, also known as the bilinear regression model. This model describes the change of a dependent variable over time [5].

The focus of this thesis is to explore how decision trees can be used to classify repeated measures data that follow the Growth Curve model, as well as estimating how well it can perform on a given data set. The suggested solutions

for this problem will be compared with a method which uses linear discriminant analysis to classify the data.

Chapter 2

Theory

The following chapter describes the necessary theory to understand the results presented in Chapter 3.

2.1 Modelling the data

A data set used for this thesis can be described as a $p \times n$ matrix $\mathbf{X} = (\mathbf{x}_1, \cdots, \mathbf{x}_n)$, where n is the number of individuals and p is the number of repeated measurements of a certain characteristic of each individual. Let each measurement in $t = (t_1, t_2, ..., t_p)$ be a variable of the data used for constructing a decision tree. For each individual in $\mathbf{X}$ there will then be a set of measurements $\mathbf{x}_i^T = (x_{i,t_1}, ..., x_{i,t_p}) = (x_{i,1}, ..., x_{i,p})$ which will be referred to as the measurement vector of an individual. In addition, x_j is the acquired measurement for the sample at a time j.

Assume that the p measurements on each individual are multivariate normal distributed with a covariance matrix $\mathbf{\Sigma}$ and that the measurements between different individuals are independent [5]. Each individual belongs to one of two groups, and the growth of the measured property of each group can be modeled as a polynomial of degree $q-1$. Then the mean for each group can be written as

$$\boldsymbol{\mu}_i = \beta_{0i} + \beta_{1i}t + \beta_{2i}t^2 + \ldots + \beta_{(q-1)i}t^{q-1},\ i = 1, 2. \tag{2.1}$$

The Growth Curve model is for this set of data then given by

$$\mathbf{X} = \mathbf{ABC} + \mathbf{E}, \quad \mathbf{E} \sim \mathcal{N}_{p,n}(\mathbf{0}, \mathbf{\Sigma}, \mathbf{I}),$$

where the first n_1 rows of $\mathbf{X}$ correspond to the measurements of samples belonging to group one and the rest of the rows belonging to group two, $\mathbf{A}$ is a $p \times q$ within individual design matrix and $\mathbf{C}$ is a $2 \times n$ between individual design matrix and are defined as

$$\mathbf{A} = \begin{pmatrix} 1 & t_1 & \cdots & t_1^{q-1} \\ 1 & t_2 & \cdots & t_2^{q-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & t_p & \cdots & t_p^{q-1} \end{pmatrix}, \quad \mathbf{C} = \begin{pmatrix} \mathbf{1}'_{n_1} & \mathbf{0}'_{n_2} \\ \mathbf{0}'_{n_1} & \mathbf{1}'_{n_2} \end{pmatrix}, \tag{2.2}$$

where n_1 and n_2 are the number of samples from group one and two, respectively. In matrix $\mathbf{C}$, $\mathbf{1}'_{n_i}$ and $\mathbf{0}'_{n_i}$ are vectors of ones and zeroes with length n_i.

The matrix $\mathbf{B}$ is a $q \times 2$ parameter matrix such that

$$\mathbf{B} = \begin{pmatrix} \boldsymbol{\beta}_1 & \boldsymbol{\beta}_2 \end{pmatrix} = \begin{pmatrix} \beta_{01} & \beta_{02} \\ \beta_{11} & \beta_{12} \\ \vdots & \vdots \\ \beta_{q1} & \beta_{q2} \end{pmatrix}.$$

The parameter matrix $\mathbf{B}$ and the covariance matrix $\boldsymbol{\Sigma}$ are not always known and must be estimated by using for example the maximum likelihood estimator [4]. If $\mathbf{A}$ and $\mathbf{C}$ have full rank, the maximum likelihood estimators for $\mathbf{B}$ and $\boldsymbol{\Sigma}$ are given by

$$\hat{\mathbf{B}} = (\mathbf{A}'\mathbf{S}^{-1}\mathbf{A})^{-1}\mathbf{A}'\mathbf{S}^{-1}\mathbf{C}'(\mathbf{C}\mathbf{C}')^{-1},$$

and

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{n}(\mathbf{X} - \mathbf{A}\hat{\mathbf{B}}\mathbf{C})(\mathbf{X} - \mathbf{A}\hat{\mathbf{B}}\mathbf{C})',$$

where $\mathbf{S} = \mathbf{I} - \mathbf{C}(\mathbf{C}\mathbf{C}')^{-1}\mathbf{C}$ [4].

The two groups can be described by their probability density functions $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$. A likelihood based decision rule is then to assign a sample $\mathbf{x}$ to Group 1 if

$$p_1 f_1(\mathbf{x}) > p_2 f_2(\mathbf{x}) \tag{2.3}$$

holds true and to Group 2 otherwise. Here, p_1 and p_2 are the prior probabilities that an individual belongs to groups 1 and 2, respectively. For the purpose of this thesis, assume $p_1 = p_2$. The functions $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$ are functions given by

$$f_i(\mathbf{x}) = (2\pi)^{\frac{p}{2}} |\boldsymbol{\Sigma}|^{-\frac{1}{2}} \exp\{\frac{1}{2} \operatorname{tr}\{\boldsymbol{\Sigma}^{-1}(\mathbf{x} - \mathbf{A}\boldsymbol{\beta}_i)(\mathbf{x} - \mathbf{A}\boldsymbol{\beta}_i)\}\}. \tag{2.4}$$

Given (2.3) and (2.4) the linear classification function is given by

$$L(\mathbf{x}; \boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \boldsymbol{\Sigma}) = (\boldsymbol{\beta}_1 - \boldsymbol{\beta}_2)'\mathbf{A}'\boldsymbol{\Sigma}^{-1}\mathbf{x} - \frac{1}{2}(\boldsymbol{\beta}_1 - \boldsymbol{\beta}_2)'\mathbf{A}'\boldsymbol{\Sigma}^{-1}\mathbf{A}(\boldsymbol{\beta}_1 + \boldsymbol{\beta}_2).$$

Hence, the plug in classification rule is that $\mathbf{x}$ is classified to group 1 if $L(\mathbf{x}; \hat{\boldsymbol{\beta}}_1, \hat{\boldsymbol{\beta}}_2, \hat{\boldsymbol{\Sigma}}) > 0$ and to group 2 otherwise. For more details see [4].

2.2 Decision Trees

Decision trees are classifiers constructed by repeatedly splitting a training data set into two subsets, which both contain at least one sample [2]. These subsets will be referred to as internal nodes and the training data as the root node. The aim of each split is to make the samples in the consecutive subsets more similar than in the parent subset. After enough splits, all samples in a node will belong to the same class, and the algorithm will not do any more splits. Each split of a node corresponds to a rule where one or a combination of the features x_t is compared to a threshold τ.

2.2.1 Impurity function

The empirical probability that a random sample $(\mathbf{x}, y)$ in node k containing n samples, belongs to class 1 and 2 can be given by

$$\pi_{km} = P(y = m, \mathbf{x} \in k) = \frac{1}{n} \sum_{i=1}^{n} 1(y_i = m) \text{ where } m \in \{1, 2\}.$$

Using these probabilities, it is possible to define the impurity of a node as a measurement of the number of distinct classes the samples in a node belong to and with what proportions [1]. A function that measures the impurity of a node $Q(k)$ is therefore such that $Q(k)$ is the largest when all classes 1 and 2 have equal proportions in k and 0 when all samples in k belong to the same class.

There are several possible functions for $Q(k)$. One possible function to use in classification problems is the the gini index [2] given as

$$Q(k) = \sum_{m \in \{1,2\}} \pi_{km}(1 - \pi_{km}).$$

There are other possible impurity functions, but it is not necessarily the case that the choice of impurity function has a great impact on the final tree [2].

Considering a node k, which has a possible split s that divides it into k_l and k_r. A given proportion p_l of the samples in k will go to k_l and a proportion p_r into k_r. It is then possible to define a measurement for how good the split is as the decrease in impurity,

$$\Delta Q(s, k) = Q(k) - p_l Q(k_l) - p_r Q(k_r).$$

2.2.2 Finding the best split of a node

The best split s for a node k is one that yields the greatest decrease in impurity [1]. To find s, define a set K of possible binary splits. In general K consists of thresholds corresponding to all distinct values for all variables. This means that for each variable t in a data set, the algorithm checks the value for each sample. Every value that does not already exist in K for the specified variable is added to K.

For each sample, the algorithm checks if the value x at the given variable t is greater or equal to the threshold c, where c is a value in K corresponding to t. If the inequality holds true for given sample, the sample will belong to a subset k_r and else to subset k_l. For each possible split s, the decrease in impurity is calculated and the split with the greatest decrease in impurity is chosen as the best split for a given node.

Example 2.2.1. To illustrate, consider a node with the data shown i Table 2.1. The set of possible thresholds K would be each of the unique values for each of the variables j. Then to find the best split, the algorithm would, for each value c in K, check if $x_j \geq c$, where x_j is the value of a sample corresponding to a given variable. This threshold is then used to categorize the samples into two nodes. For example, from table 2.1, we can see that a value in K corresponding to variable x_1 is 4. The algorithm then asks if the value $x_1 \geq 4$ for a given sample. If it is true, the sample x belongs to k_l and to k_r otherwise.

Table 2.1: Example data consisting of labelled data with two variables.

Sample	1	2	3	4	5	6	7	8	9	10
$j = 1$	1	2	2	2	4	2	3	5	5	7
$j = 2$	3	1	2	5	3	4	6	3	4	1
Group	1	1	1	1	1	2	2	2	2	2

The resulting subsets the threshold 4 for variable 1 are $k_l = \{5, 8, 9, 10\}$ and $k_r = \{1, 2, 3, 4, 6, 7\}$. The decrease in impurity for this split is given by

$$\Delta Q(s, k) = \frac{1}{4} - \frac{4}{10} \cdot \frac{3}{16} - \frac{6}{10} \cdot \frac{8}{36} = \frac{1}{24}$$

This is repeated for each value in K until the largest possible $\Delta Q(s, k)$ is found.

Considering that the variables in repeated measures data are dependent, the general method of finding the best split is likely to not give the best results. Instead, a split on a combination of the variables might lead to a better classifier. Breiman et al. [1] suggest three methods of how to combine variables to make

such a split. Two of these; to search for a best linear combination split (CART-LC algorithm), and to add new variables, might be useful for classifying repeated measures data.

The procedure of finding a best linear combination is that for each non-pure node there is a set of coefficients $\boldsymbol{a} = (a_1, ..., a_T)$ such that the sum of squares of the coefficients is 1. The best split, that yields the greatest decrease in impurity, is assumed to be of the form

$$\sum_{i=1}^{T} a_i x_{t_i} \leq \tau,$$

where τ ranges over all possible values and $\boldsymbol{a}$ is the corresponding best set of coefficients. This method is suggested for data with a strong linear structure [1]. The algorithm for finding the best split of linear combination of a node is shown in Algorithm 1. A benefit of the decision tree algorithm is that it is

Algorithm 1 CART-LC algorithm

```
L = 0
while TRUE do
    L = L + 1
    for Repeated measurements t in X do
        Let the current split s_L be v ← Σ_{t=1}^{T} a_t x_t such that v ≤ τ
        for γ in {−0.25, 0, 0.25} do
            Search for the δ that maximizes the goodness of the split
            v − δ(a_t + γ) ≤ τ
        end for
        Let δ∗, γ∗ be the combination that resulted in the best split.
        Let a_t = a_t − δ∗, τ = τ − δ ∗ γ∗
    end for
    Change τ to maximize the goodness of s_L. Keep a_t constants.
    if |ΔQ(s_L) = Q(s_{L−1})| ≤ ε then
        Exit While-loop
    end if
end while
```

possible to introduce a large number of variables and select the best variables to classify the data. These variables can correspond to aspects or properties of the data that were difficult to detect using only the original variables. There is no obvious method of adding new features to the data, and it instead relies on previous understanding of the nature of the data [1].

2.2.3 Using a tree for classification

Once a tree has been constructed with a set of terminal nodes $\tilde{k}$, a class $m \in \{1, 2\}$ can be assigned to each terminal node $k \in \tilde{k}$. The class assigned to a given terminal node is the class m that maximizes the probability a random sample in a terminal node k belongs to the assigned class. In the case that the proportions in k are equal, an arbitrary class can be chosen.

The performance of a classifier is determined by how well it can predict the class of unlabeled samples. A simple way to estimate the performance is to calculate the accuracy for a set of samples that has not been used for constructing the classifier. The accuracy is given by the sum of correctly predicted samples divided by the total number of samples used for estimating the performance.

2.2.4 Finding the right sized tree

A decision tree that continues to split nodes until all terminal nodes are pure nodes will produce good results on the training data, but is unlikely to produce equally satisfying results when tested on a new set of samples. This happens as training data usually contains information which is not useful for predicting the class. Since the amount of samples in an impure node is limited, there can be cases where a split is found even if it does not hold true for the data set as a whole. To avoid such a split from occurring and possibly decreasing the accuracy of the model, the size of the tree should be limited.

There are different suggestions for finding what sized tree will lead to the highest accuracy. A possible solution to avoid this problem is to set a threshold γ and deciding not to split a node if the decrease in impurity was less than γ. This solution is, however, unsatisfactory [1], as a split with a small decrease in impurity may be followed by a split that has a large decrease in impurity.

It can instead be argued that pruning is a satisfactory method to determine the right size of the tree [1]. The first step in pruning is to grow a sufficiently large tree T_{max}. For each set of parent node and descendant nodes, the decrease in impurity is calculated, and the nodes are merged for the set with the smallest decrease in impurity. This gives a new tree T^* with a set of nodes $\tilde{T}^*$ and an overall misclassification cost

$$R(T) = \sum_{k \in \tilde{T}*} n(k)Q(k),$$

where $n(k)$ is the number of samples in a node k and $Q(k)$ is the impurity of a node k. The process is repeated until the tree consists of 1 node and the tree with the lowest misclassification cost is selected as the best tree.

Bootstrap Aggregating

A suggested method for improving the accuracy of a tree structured classifier is to use ensemble methods [2]. The main idea of ensemble methods is to use the average of multiple classifiers in order to reduce the variance in the set of observed data. This thesis will use an ensemble method which is closely related to the CART algorithm; boostrap aggregating.

In boostrap aggregating, commonly known as bagging, classifiers are constructed using different training data sets. A given training data set used in the bagging algorithm is a sample from the original training data set with replacement [3]. The data sets are constructed using the bootstrap algorithm [2], in accordance to the pseudocode depicted in Algorithm 2 .

Algorithm 2 The Algorithm for Boostrap

Data: $\mathbf{X} = \{\mathbf{x}_i, y_i\}_{i=1}^n$
for i in $1, ...n$ **do**
  Sample l uniformly on the set of integers $1, ..., n$
  Set $\tilde{\mathbf{x}}_i = \mathbf{x}_l$ and $\tilde{y}_i = y_l$
end for
Return $\{\tilde{\mathbf{x}}_i, \tilde{y}_i\}_{i=1}^n$

On each of the data sets, a decision tree is trained and returned, which is seen in Algorithm 3. Finally, when predicting a new sample, the final prediction is the average of the predicition from all of the classifiers, as seen in Algorithm 4.

Algorithm 3 The Algorithm for Boostrap Aggregating

Data: $\mathbf{X} = \{\mathbf{x}_i, y_i\}_{i=1}^n$
Z is a user-defined constant.
for z in $1, ...Z$ **do**
  Run the algorithm for bootstrap to obtain a sample data set $\mathbf{X}^{(z)}$
  Learn a decision tree on data set $\mathbf{X}^{(z)}$
end for
Return Decision trees trained on Z data sets.

Algorithm 4 The Prediction using Boostrap Aggregating Algorithm

Data: Z tree models and test data $\mathbf{x}$
for z in $1, ...Z$ **do**
  Predict $\hat{y}^{(z)}(\mathbf{x})$ using tree z
end for
Return $\hat{y}(\mathbf{x})$ by averaging the values of $\hat{y}^{(z)}(\mathbf{x})$

This chapter presented the Growth Curve model and the theoretical background to variations of classification trees as well as the equation of the linear classification function. How well different classification trees perform when applied to a simulated data set will be evaluated by comparing the accuracy of different decision trees with the accuracy of the linear classification function.

Chapter 3

Results

This chapter presents the results from applying the theory described in Chapter 2 onto a simulated data set. In Section 3.1, the simulated data set and the results from constructing decision trees using 20 samples are presented. In Section 3.2, the same algorithms are applied to a data set consisting of 100 samples, and in Section 3.3 a greater number of samples is used. Section 3.4 presents the results based on using further derived variables to construct decision trees.

3.1 Simulation of data and CART algorithm using 20 samples

The growths of two groups were simulated, where the growths of the means of each group are as described in (2.1) with the parameter matrix

$$\mathbf{B} = \begin{pmatrix} 2.7 & 3.8 \\ 1.0 & 0.3 \\ -0.2 & -0.1 \\ 0.01 & 0.01 \end{pmatrix}, \tag{3.1}$$

and for repeated measurements at $t = 1, 2, \cdots, 10$, the within individual design matrix $\mathbf{A}$ is given by

$$\mathbf{A} = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 10 & 100 & 1000 \end{pmatrix}. \tag{3.2}$$

The model of the two groups corresponding to the matrices in (3.1) and (3.2) is depicted in Figure 3.1.

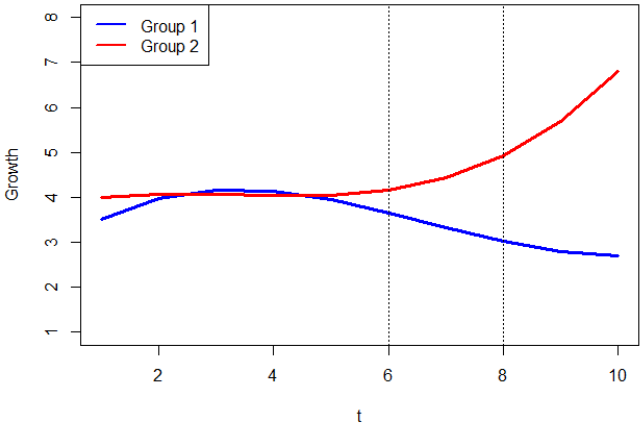

Figure 3.1: True models of simulated data with 10 consequtive measurements.

It is visible from Figure 3.1 that the groups are more similar in the interval $2 \leq t \leq 5$ and diverge at $t = 1$ and $t \geq 6$, with the groups diverging more as t increases.

For 10 individuals from each group, the between-individual design matrix $\mathbf{C}$ is given by

$$\mathbf{C} = \begin{pmatrix} \mathbf{1}'_{10} & \mathbf{0}'_{10} \\ \mathbf{0}'_{10} & \mathbf{1}'_{10} \end{pmatrix}.$$

For this model, let the covariance matrix be

$$\mathbf{\Sigma} = \begin{pmatrix} 1.0 & 0.5 & 0.5^2 & \cdots & 0.5^{p-1} \\ 0.5 & 1.0 & 0.5 & \cdots & 0.5^{p-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0.5^{p-1} & 0.5^{p-2} & 0.5^{p-3} & \cdots & 1.0 \end{pmatrix}$$

which corresponds to setting the covariance of two consecutive variables to $\text{cov}(t, t+i) = 0.5^i$ for $i = 0, \cdots, 9$. Finally, given the model above, the repeated measurements for 20 samples were simulated. The growth curves of the simulated samples are seen in Figure 3.2.

In Figure 3.2, it seen that while the recorded values at time t_9 and t_{10} differ between the two groups, the groups are not easily distinguishable at earlier values of t.

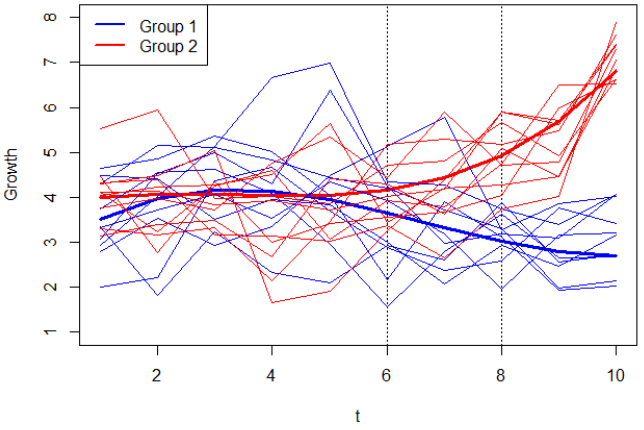

Figure 3.2: Simulated data using the true model in (3.1) and (3.2).

With the 20 simulated samples used as training data, it is possible to construct a decision tree for classifying the samples. The *rpart*() function from the *rpart* package [1] was used in Rstudio to construct a classification tree, with default values. Decision trees were constructed using the first 6, 8 and 10 consecutive measurements as variables.

An example of a classification tree using all 10 measurements is shown in Figure 3.3. The tree in Figure 3.3 consists of a single split at $t = 9$. As the two groups are linearly separable at this time, the chosen variable is not unexpected. However, since the two groups differ more from each other at $t = 10$, it might have been a more desirable variable to choose for classifying the samples. The reason for why the split is not at the latter time is likely due to $t = 9$ and $t = 10$ resulting in equally good accuracy when tested on the provided training data and $t = 9$ simply being the first variable that was tested, or it was a random choice between the two variables.

The tree structure was similar for 8 and 6 measurements. In both cases, the tree consisted of a single split at times 8 and 6 respectively. Since the difference between the groups increases when t increases, it is reasonable that the split for each of the classification trees is when t is the latest. Furthermore, since the difference between the groups is smaller when t is smaller, it can be expected

[1] https://cran.r-project.org/web/packages/rpart/rpart.pdf

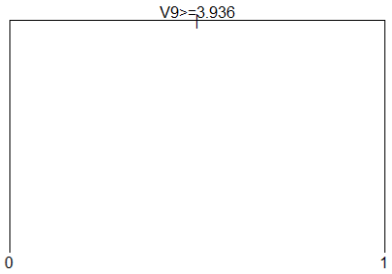

Figure 3.3: Classification tree constructed using 20 samples of training data and 10 consecutive measurements.

that the accuracy of the trees decreases as less repeated measurements are used.

The classification tree using the CART-LC algorithm was constructed using the $ODT()$ function from the $ODRF$ package [2]. The function uses the algorithm described in Algorithm 1 and results in a classification tree using linear combinations of variables. Classification trees using each of the bootstrap aggregating algorithms presented in Algorithms 2, 3 and 4 were also constructed, and the corresponding accuracy was estimated. The accuracy of each of the models was tested by simulating an additional 5000 samples from each group, predicting to which group the samples belong and measuring how many times the models predicted the correct group. The results of the accuracy for the different methods can be seen in Table 3.1

When simulating the data, it is possible to state the covariance between consecutive variables. This is represented by a covariance matrix and might have an impact on the accuracy of the classifier. To see how big of an impact the dependence has on the performance of the classifier, the decision trees were constructed and evaluated for training data sets simulated with different covariance

[2] https://cran.r-project.org/web/packages/ODRF/ODRF.pdf

matrices

$$\mathbf{\Sigma} = \begin{pmatrix} 1.0 & \rho & \cdots & \rho^{p-1} \\ \rho & 1.0 & \cdots & \rho^{p-2} \\ \vdots & \vdots & \ddots & \vdots \\ \rho^{p-1} & \rho^{p-2} & \cdots & 1.0 \end{pmatrix},$$

for different ρ.

The results in Table 3.1 show that using decision trees with a training sample size of 20 can produce models where the accuracy of the classification is almost comparable to that of a linear discriminant analysis. From these results, however, it does not appear to be the case that modifying the dependence between the variables has an impact on the accuracy of the decision tree. The number of repeated measurements used for the classification has an impact on the accuracy for the given models. This can be compared with the results of the linear discriminant analysis, in which the accuracy of the classification increases as the covariance between the variables increases.

Whether the decision tree is a good classifier is highly dependent on the training data, which would explain why some decision trees performed significantly worse when the covariance between the variable was changed. It can also explain why some decision trees have an accuracy below 0.5, as it is a possible outcome if the training data used for constructing the decision tree does not reflect the data accurately.

An additional issue was that due to the small number of training data samples, the classifier from the CART-LC algorithm was such that all samples were classified as the same class. This is reflected in the results seen in Table 3.1, as the estimated accuracy for the models using the CART-LC algorithm are around 0.5 regardless of the number of repeated measurements used.

3.2 CART algorithm using 100 samples

It is expected that if more training samples are used for constructing the model, the accuracy of the classifier will increase. When constructing a tree using a 100 as the training sample size, the decision tree overall performed better when classifying the data. This is mainly due to more training data providing a better estimate of the chosen threshold as well as decreasing the probability that the training data does not reflect the data that is measured. The results are seen in Table 3.2.

Figures 3.4 shows an example of what a CART-LC decision tree can look like graphically. In Figure 3.4, the variable proj1 is $-0.175t_2 + 0.033t_3 + 0.015t_6 + 0.069t_9 - 0.982t_{10}$. There were, however, cases when no combination of vari-

Table 3.1: Classification accuracy of different methods using 20 samples of training data. The methods tested are the standard CART algorithm (CART), the bootstrap aggregating algorithm (Bagging), the CART tree using a linear combination of variables (CART-LC) and linear discriminant analysis for the growth curve model (LDA for GCM). Each of the algorithms is tested for different values of dependence between the repeated measurements and for $10, 8$ and 6 consecutive measurements.

Cov $(t, t+1)$	CART	Bagging	CART-LC	LDA for GCM
10 repeated measurements, $t = 1, 2, ..., 10$				
0.1	0.919	0.978	0.507	0.996
0.2	0.933	0.933	0.507	0.993
0.3	0.631	0.632	0.507	0.989
0.4	0.985	0.986	0.507	0.986
0.5	0.876	0.835	0.507	0.985
0.6	0.993	0.920	0.507	0.981
0.7	0.894	0.894	0.507	0.981
0.8	0.925	0.921	0.507	0.995
0.9	0.960	0.960	0.507	0.995
8 repeated measurements, $t = 1, 2, ..., 8$				
0.1	0.607	0.743	0.499	0.868
0.2	0.923	0.911	0.499	0.855
0.3	0.623	0.623	0.499	0.850
0.4	0.688	0.771	0.499	0.842
0.5	0.772	0.817	0.499	0.842
0.6	0.727	0.834	0.499	0.845
0.7	0.504	0.586	0.499	0.853
0.8	0.783	0.825	0.499	0.878
0.9	0.700	0.813	0.499	0.932
6 repeated measurements, $t = 1, 2, ..., 6$				
0.1	0.567	0.589	0.494	0.640
0.2	0.343	0.490	0.494	0.643
0.3	0.586	0.772	0.494	0.644
0.4	0.546	0.657	0.494	0.653
0.5	0.430	0.699	0.494	0.662
0.6	0.652	0.605	0.494	0.662
0.7	0.745	0.666	0.494	0.678
0.8	0.542	0.561	0.494	0.770
0.9	0.595	0.721	0.494	0.772

Table 3.2: Classification accuracy of different methods using 100 samples of training data. Each of the standard CART algorithm (CART), the bootstrap aggregating algorithm (Bagging) and the CART tree using a linear combination of variables (CART-LC) is tested for different values of dependence between the repeated measurements and for $10, 8$ and 6 consecutive measurements.

Cov $(t, t+1)$	CART	Bagging	CART-LC
10 repeated measurements, $t = 1, 2, ..., 10$			
0.1	0.963	0.990	0.982
0.2	0.976	0.979	0.972
0.3	0.981	0.979	0.993
0.4	0.972	0.976	0.995
0.5	0.979	0.979	0.992
0.6	0.978	0.978	0.992
0.7	0.981	0.981	0.970
0.8	0.978	0.978	0.951
0.9	0.981	0.990	0.991
8 repeated measurements, $t = 1, 2, ..., 8$			
0.1	0.823	0.826	0.891
0.2	0.830	0.812	0.833
0.3	0.815	0.828	0.876
0.4	0.827	0.790	0.584
0.5	0.813	0.830	0.849
0.6	0.826	0.715	0.837
0.7	0.824	0.760	0.771
0.8	0.860	0.856	0.822
0.9	0.860	0.856	0.832
6 repeated measurements, $t = 1, 2, ..., 6$			
0.1	0.586	0.583	0.632
0.2	0.610	0.622	0.788
0.3	0.580	0.580	0.468
0.4	0.574	0.571	0.648
0.5	0.564	0.583	0.731
0.6	0.602	0.621	0.403
0.7	0.589	0.591	0.701
0.8	0.611	0.626	0.735
0.9	0.636	0.676	0.814

ables occured for classifying the data. This was particularly the case when 10 repeated measurements was used to classify the data, in which case the linear combination tree often consists of the final measurement as the variable where a good threshold can be found and no other variable was considered necessary.

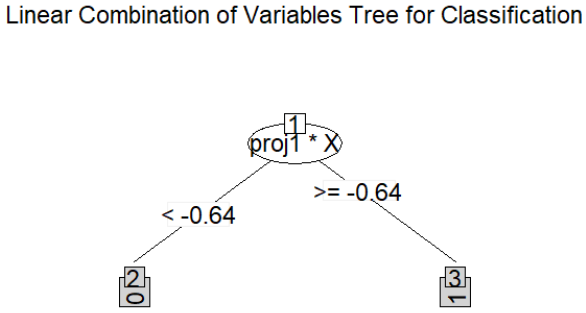

Figure 3.4: A CART-LC tree produced when using 100 samples of training data.

3.3 CART algorithm on a higher number of samples

The linear combination of variables algorithm was tested on more training data than 100 samples and the results are seen in Table 3.3. The results seem to imply that by increasing the number of training samples, the accuracy of the model increases. However, this is not necessarily the case, as in Table 3.3 for the covariance 0.5 and 0.6 the accuracy of the model decreases when increasing the number of training samples from 400 to 600. The reason for not attempting to use more training samples was that the time it took to simulate training data was too long.

Table 3.3: Classification accuracy of CART-LC trees for 200, 400 and 600 samples of training data. The CART-LC algorithm is tested for different values of dependence between the repeated measurements 10 consecutive measurements ($t = 1, 2, ..., 10$).

Cov $(t, t+1)$	200	400	600
0.1	0.977	0.994	0.995
0.2	0.984	0.991	0.992
0.3	0.969	0.992	0.982
0.4	0.986	0.952	0.988
0.5	0.953	0.988	0.980
0.6	0.955	0.978	0.970
0.7	0.983	0.978	0.982
0.8	0.981	0.948	0.983
0.9	0.982	0.995	0.995

3.4 CART algorithm using more features

As described in the Chapter 2, another method for attempting to account for the dependence in between the variables is to add new features to the data set which would account for this dependence. Some of the features that were added were the coefficients of an estimation of a linear fit for the repeated measurements and the coefficients of an estimation of a quadratic fit, cubic fit, and quartic fit in addition to the predicted values for each of these estimates. The estimated accuracy for the models constructed using these features are seen in Table 3.4. The main idea was to use the repeated measurements to find derived features that would better describe the difference between the two groups.

The results in Table 3.4 show that the accuracy of the trees constructed with training data consisting of more variables is not clearly better than when no additional information was included. Using 10 repeated measurements, the results were, in fact, lower than when only the original measurements were used. This could be due to the additional variables being good at distinguishing the data samples within the training data, but not the data set as a whole. For 8 and 6 measurements, the accuracy of the decision trees in Table 3.4 were overall higher than those in Table 3.2, which seems to imply that some of the additional information was useful, but not as good as using the last measurement.

Table 3.4: Classification accuracy of trees using 100 samples as training data. The standard CART algorithm (CART) and the bootstrap aggregating algorithm (Bagging) is tested for different values of dependence between the repeated measurements and for $10, 8$ and 6 consecutive measurements. The training data includes variables such as the coefficients for the estimation of a polynomial fit.

$\mathrm{Cov}(t, t+1)$	CART	Bagging
10 repeated measurements		
0.1	0.988	0.937
0.2	0.978	0.982
0.3	0.500	0.501
0.4	0.970	0.970
0.5	0.977	0.977
0.6	0.978	0.978
0.7	0.979	0.979
0.8	0.978	0.978
0.9	0.980	0.980
8 repeated measurements		
0.1	0.841	0.848
0.2	0.829	0.806
0.3	0.820	0.834
0.4	0.832	0.779
0.5	0.809	0.819
0.6	0.831	0.823
0.7	0.830	0.756
0.8	0.871	0.899
0.9	0.871	0.899
6 repeated measurements		
0.1	0.604	0.556
0.2	0.537	0.544
0.3	0.572	0.537
0.4	0.500	0.542
0.5	0.569	0.587
0.6	0.607	0.621
0.7	0.689	0.506
0.8	0.596	0.613
0.9	0.648	0.701

Chapter 4

Discussion

When comparing the results of the standard CART algorithm, boostrap aggregating and the CART-LC algorithm with those of the linear discrimination analysis for classification of repeated measurements, it becomes evident that the linear discrimination analysis works better than decision trees for classifying using a smaller training data set. Even if cases did occur where the decision tree algorithm performed better than the linear discrimination analysis, the results showed that a decision tree is more likely to perform worse. However, there might be decision tree algorithms which have not been explored in this thesis which would work well for small sample sizes as well.

The results of this thesis show that for the given true models, with 10 simulated measurements, it is possible to, by using the standard CART algorithm, construct a function that correctly classifies samples into one of two classes up to 98.1% of the time. By extending the CART algorithm and using bootstrap aggregating, the accuracy of the final function can be at least 99% for certain data sets. It is, by using the CART-LC algorithm, possible to increase the accuracy of the final model to 99.5%.

As increasing the size of the training data tends to increase the accuracy of the decision tree, it is likely possible to further improve upon the accuracy in all cases. Similarly, an increased number of repeated measurements would probably improve the performance of the model. From the results in this thesis, it is not possible to estimate to what extent the accuracy can be improved using any of these suggestions.

An unexpected result is that the accuracy of the decision trees using linear combination did not necessarily increase when using more samples. A possible explanation for this result is that though the number of samples is few, the training data provided the information needed for the classifier to be as good

as it can be. This argument would also mean that the misclassified samples are outliers, and it would not be possible for the decision tree to classify the samples correctly regardless of the tuning performed on the splits.

The benefit of decision trees is that they are, given that bagging is not used, easy to interpret. In some instances, this benefit may outweigh the fact that they are not the classifiers with the highest accuracy. This means that decision trees may still be a good choice as a classifier, particularly when there is a larger amount of data.

A limitation of this thesis is that only one true data model was used for testing the classifiers. The model was such that the difference between the classes is large when $t = 10$. To be able to generalize the results of this thesis, it would be needed to test the results for a variety of data sets.

There are expansions to the basic CART algorithm that may produce better results. These are for example using the random forest algorithm. In addition, there may exist properties of the data which were not explored in this thesis. These explorations are, however, outside of the scope of this thesis.

Bibliography

[1] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Taylor & Francis Group, 1984.

[2] A. Lindholm, N. Wahlström, F. Lindsten, and T. B. Schön. *Machine Learning A First Course for Engineers and Scientists*. Cambridge University Press, 2022.

[3] S. Marsland. *Machine Learning - An Algoritmic Perspective*. Taylor & Francis Group, 2009.

[4] E. Ngailo, D. von Rosen, and M. Singull. Approximation of misclassification probabilities in linear discriminant analysis with repeated measurements. Technical report, Linköping University, 2020.

[5] D. von Rosen. *Bilinear Regression Analysis: An Introduction*. Springer International Publishing, 2018.

[6] X. Wu, V. Kumar, Q. J. Ross, J. Ghosh, Q. Yang, McLachlan G. J. Motoda, H., A. Ng, B. Liu, P. S. Yu, and Z.H. Zhou. Top 10 algorithms in data mining. *KAIS*, 14(1):1–37, 2006.