Dimension Reduction and Classification with High-Dimensional Microarray Data

Dissertation an der Fakultät für Mathematik, Informatik und Statistik der Ludwig-Maximilian-Universität München

vorgelegt von

Anne-Laure Isabeau Boulesteix

am 18.11.2004

Dimension Reduction and Classification with High-Dimensional Microarray Data

Dissertation an der Fakultät für Mathematik, Informatik und Statistik der Ludwig-Maximilian-Universität München

vorgelegt von

Anne-Laure Isabeau Boulesteix

am 18.11.2004

1. Gutachter: Prof. Dr. G. Tutz
2. Gutachter: Prof. Dr. L. Fahrmeir
3. Gutachter: Prof. Dr. U. Gather

Rigorosum: 22.02.2005

VORWORT

Diese Arbeit entstand im Laufe der letzten zweieinhalb Jahren während meiner Tätigkeit als wissenschaftliche Mitarbeiterin am Institut für Statistik der Ludwig-Maximilian Universität München. Sie wurde zum Teil durch Mittel des Sonderforschungsbereichs 386 und des Emmy-Noether-Programms der DFG gefördert.

Bedanken möchte ich mich zuallererst bei meinem Doktorvater Gerhard Tutz, der mir durch fruchtbare Gespräche sehr geholfen hat, neue Ideen entstehen zu lassen und diese zu verwirklichen. Er hat mir dabei viele Freiheit und Vertrauen geschenkt. Ein besonderer Dank gilt meinem zweiten Betreuer Korbinian Strimmer, der mir insbesondere am Anfang meiner Promotion sehr hilfsbereit zur Seite stand und als Zimmernachbar für gutes Arbeitsklima gesorgt hat.

Ich bedanke mich auch bei Ludwig Fahrmeir und Ursula Gather, die sich freundlicherweise bereit erklärt haben, diese Arbeit zu begutachten und bei meinem Diplomvater Volkmar Liebscher, der in mir die Lust am wissenschaftlichen Arbeiten geweckt hat. Bedanken möchte ich mich außerdem bei meinen Kollegen des Instituts für Statistik, insbesondere den MitarbeiterInnnen des Seminars für angewandte Stochastik und der Arbeitsgruppe für statistische Genetik und Bioinformatik, die für eine angenehme Arbeitsstimmung gesorgt haben.

Zu guter Letzt möchte ich mich bei meinen Eltern sowie bei meinem Mann für ihre jahrelange Unterstützung und bei meinem Sohn Victor für seine gute Laune und seine aufmunternden Lächeln herzlich bedanken. Ohne den enormen Beitrag meines Mannes bei der Organisation unseres Familienalltags hätte ich bestimmt erst in zwei Jahren promoviert !

ZUSAMMENFASSUNG

Klassische Microarray Datensätze enthalten in der Regel bei Beobachtungszahlen im zweistelligen Bereich Tausende von Prädiktoren. Daher ist es eine große Herausforderung, den hochdimensionalen Prädiktorenraum so zu transformieren, daß damit die Klassifikation wie zum Beispiel die Krebsdiagnose möglich wird. In dieser Arbeit werden verschiedene Ansätze zur Dimensionsreduktion solcher Daten untersucht.

Das Kapitel 2 ist eine Einführung in die Klassifikation mit Microarray Daten und weiterhin enthält es auch einen Überblick einiger spezifischer Probleme (Variablenselektion, Vergleich mehrerer Klassifikationsmethoden). Im Kapitel 3 untersuche ich besondere Interaktionsstrukturen im Kontext der Klassifikation: 'Emerging Patterns'. Ich führe eine neue und allgemeinere Definition, die auf den unterliegenden Wahrscheinlichkeiten beruht, ein und stelle eine neue auf dem CART-Algorithmus basierende einfache Suchmethode, die die entsprechenden empirischen Patterns in konkreten Datensätzen findet, vor. Ich habe den Suchalgorithmus sowie die Klassifikationsmethode in der Programmiersprache `R` implementiert. Einige dieser Programme sind auf meiner Homepage frei verfügbar. Im Kapitel 4 geht es um die klassische lineare Dimensionsreduktion. Im Rahmen der binären Klassifikation mit stetigen Prädiktoren beweise ich die Zusammenhänge zwischen der Partial Least Squares (PLS) Methode, der "between-group" Hauptkomponentenanalyse und der linearen Diskriminanzanalyse. Die PLS Dimensionsreduktion wird im Kapitel 5 im Detail untersucht. Die Klassifikationsmethode der PLS Dimensionsreduktion kombiniert mit der linearen Diskriminanzanalyse wird für neun Microarray Datensätze mit den besten bekannten Methoden verglichen und erweist sich als der beste Ansatz. Außerdem wende ich einen Boosting Algorithmus auf diese Klassifikationsmethode an. Ebenso schlage ich auch einen einfachen Ansatz zur Wahl der Anzahl der PLS Komponenten vor. Zum Schluss untersuche ich den theoretischen Zusammenhang zwischen PLS Dimensionsreduktion und Variablenselektion: ich beweise eine Equivalenzeigenschaft zwischen einem bekannten Kriterium zur Variablenselektion und einem auf der ersten PLS Komponente basierenden Ansatz.

SUMMARY

Usual microarray data sets include only a handful of observations, but several thousands of predictor variables. Transforming the high-dimensional predictor space to make classification (for instance cancer diagnosis) possible is a major challenge. This thesis deals with various dimension reduction approaches which can handle such data.

Chapter 2 gives an introduction into classification with microarray data as well as an overview of a few specific problems such as variable selection and comparison of classification methods. In Chapter 3, I discuss a particular class of interaction structures in the classification framework: "emerging patterns". I propose a new and more general definition referring to underlying probabilities and present a new simple method which is based on the CART algorithm to find the corresponding empirical patterns in concrete data sets. In addition, the detected patterns can be used to define new variables for classification. Thus, I propose a simple scheme to use the patterns to improve the performance of classification procedures. I implemented the search algorithm as well as the classification procedure in the language `R`. Some of these programs are publicly available from my homepage. Chapter 4 deals with classical linear dimension reduction methods. In the context of binary classification with continuous predictors, I prove two properties concerning the connections between Partial Least Squares (PLS) dimension reduction, between-group PCA and between linear discriminant analysis and between-group PCA. PLS dimension reduction for classification is examined thoroughly in Chapter 5. The classification procedure consisting of PLS dimension reduction and linear discriminant analysis on the new components is compared favorably with some of the best state-of-the-art classification methods using nine real microarray cancer data sets. Moreover, I apply a boosting algorithm to this classification method, which is a novel approach. In addition, I suggest a simple procedure to choose the number of PLS components. At last, I examine the connection between PLS dimension reduction and variable selection and prove a property concerning the equivalence between a common univariate selection criterion and a variable selection approach based on the first PLS component.

Contents

1 Introduction **1**

1.1 High-dimensional microarray data 1

1.2 Guideline through the thesis 3

1.3 Notations 4

2 Classification with application to microarray data **7**

2.1 Overview of classification with high-dimensional microarray data 7

2.2 Comparing classification methods 10

2.2.1 Decision theory 10

2.2.2 Comparing two classification methods in practice 11

2.3 Variable selection 13

2.3.1 Univariate ranking methods 14

2.3.2 Optimal subset selection 16

3 Emerging and Interaction Patterns **19**

3.1 Introduction 19

3.2 Definition of interaction patterns 21

3.2.1 Interaction Patterns for two classes 21

3.2.2 Generalization to multicategorical response 24

3.3 Discovering interaction patterns with trees 25

3.3.1 Tree methodology 25

3.3.2 Discovering Algorithm 28

3.3.3 Receiver Operating Characteristic 29
3.3.4 Simulation study 31
3.4 Classification based on interaction patterns 34
3.4.1 Method 34
3.4.2 Study Design 35
3.4.3 Data sets 37
3.4.4 Results 38
3.4.5 An example 43
3.5 Discussion 43

4 Linear dimension reduction for classification **47**
4.1 Introduction 47
4.2 Between-group PCA 49
4.2.1 Definition 49
4.2.2 A special case: $K = 2$ 51
4.3 A connection between PLS dimension reduction and between-group PCA 52
4.3.1 Introduction to PLS dimension reduction 52
4.3.2 A property 52
4.4 A connection between LDA and between-group PCA 53
4.4.1 Linear discriminant analysis 53
4.4.2 A property 54
4.5 Overview of other methods 55
4.6 Discussion 57

5 A study of PLS dimension reduction **59**
5.1 Introduction 59
5.2 Dimension reduction and classification with PLS 63
5.2.1 Outline of the method 63
5.2.2 The SIMPLS algorithm 64
5.2.3 Choosing the number of components 67
5.2.4 Boosting 68

5.3 Data 69
5.3.1 Data sets 69
5.3.2 Data Visualization via PLS dimension reduction 71
5.4 Classification results on real microarray data 73
5.4.1 Study design 73
5.4.2 Classification accuracy of δ_{PLS} 77
5.4.3 Classification accuracy of discrete AdaBoost with $\delta = \delta_{PLS}$ 82
5.5 PLS and gene selection 88
5.6 Discussion 90

6 Conclusion **93**

Chapter 1

Introduction

1.1 High-dimensional microarray data

Microarray technology allows to measure the expression level of thousands of genes simultaneously using gene chips. Typically, a gene is a variable and a chip is an observation from the point of view of statisticians. In the context of cell-cycle experiments, each chip measures the gene expression levels at a different time point during the cell cycle. In cancer studies, each chip measures the gene expression levels of a different cancer patient. In all applications of microarray technology, the number of variables (genes) p is much larger than the number of observations (chips) n: a typical study includes from 1000 to 20000 genes for only 10 to 200 chips. Discovering e.g. interactions between genes, association of gene expression levels with specific clinical outcomes or clusters of functionally related genes in such high-dimensional data is a difficult and challenging task. In the last few years, multivariate statistics for microarray data analysis has been the subject of thousands of publications in statistics, machine learning, bioinformatics and biology. Most of the classical topics of multivariate statistics have been studied in the context of high-dimensional microarray data.

Clustering can be used to find groups of similarly expressed genes, in the hope that they also have a similar function. Different methods have been applied for this task e.g. hierarchical clustering (Eisen et al., 1998), self-organizing maps (Tamayo et al., 1999) or model-based clustering (Yeung et al., 2001). Another application of clustering is the identification of new groups of patients, for instance new tumor subclasses (Golub et al., 1999). Another topic of interest is the prediction of the survival time of a patient using gene expression levels. Nguyen and Rocke (2002b) and Park et al. (2002) answer this question by using a PLS-based method. Other authors (O'Neill and Song, 2003) dichotomize the survival time and transform the problem into a classification problem. Biologists are often interested in finding genes which are related to the survival time. Log-rank statistics can be used to identify these genes (Beer et al., 2002). Methods of times series analysis may also be useful for microarray data analysis, e.g. to identify differentially expressed genes in time-course microarray experiments (Park et al., 2003) or periodically expressed genes in the context of cell-cycle experiments (Wichert et al., 2004). Dimension reduction methods such as principal component analysis (PCA) and related methods have been applied for different purposes. Whereas Yeung and Ruzzo (2001) use it to perform clustering, Alter et al. (2000) propose a PCA-based approach for data processing and elimination of noise. Another PCA-related method is 'gene-shaving' (Hastie et al., 2000), which can be seen as a semi-supervised clustering method with possibly overlapping clusters. Classification methods are widely used for tumor diagnosis, which is one of the most important applications of microarray technology. A related issue is the identification of differentially expressed genes, i.e. genes which have significantly different levels in two or more classes. An overview of classification is given in Section 2.1.

While some authors claim that microarray data require to develop new specific statistical methods, others report good results obtained with well-known standard methods. I think that known approaches, especially classical linear methods, should be emphasized because they are often preferable to new ad-hoc purpose-built procedures. However, the high-dimensionality ($n < p$) should not be overcome by performing a dramatic variable se-

lection. Thus, statisticians should make an effort to develop and adapt methods which can handle a very large number of possibly noisy variables. Such methods could be applied in various fields of natural and engineering sciences, since recent technologies produce data with more and more variables.

1.2 Guideline through the thesis

This thesis deals with one of the major applications of microarray technology: prediction of a categorical variable (such as the tumor type of a patient) using gene expression data. It is divided into four independent chapters. An overview of classification with gene expression data is given in Chapter 2. Chapter 2 discusses some important aspects of classification, such as evaluation of classification methods and variable selection.

Interactions between variables (genes) in classification are an important topic which is often omitted in the context of high-dimensional microarray data. Chapter 3 deals with a special type of interaction structures: interaction patterns. This concept is borrowed from the machine learning community and mapped into a statistical framework. The use of interaction patterns for classification is examined and a new CART-based discovering method is proposed. This approach can be seen as a dimension reduction method, since it extracts specific patterns from high-dimensional data. A version of the methods developed in Chapter 3 has been accepted for publication in "Computational Statistics and Data Analysis" (Boulesteix and Tutz, 2005). This publication is an extension of a more applied paper published in the journal "Bioinformatics" (Boulesteix et al., 2003). I implemented the search algorithms and the classification method in `R`. Some of the programs are publicly available at the URL `www.statistik.lmu.de/~socher/ep.html`.

Another approach for dimension reduction is linear dimension reduction, which is presented and discussed in Chapter 4. The connection between different dimension reduction and classification methods is examined and proved. A brief overview of other existing approaches is given at the end of Chapter 4. A part of Chapter 4 is based on a paper which has been accepted for publication in the journal "International Journal of Pure and

Applied Mathematics" (Boulesteix, 2005).

Partial Least Squares (PLS) is one of the linear dimension reduction methods mentioned in Chapter 4. It is examined more thoroughly in Chapter 5. An extensive comparison study including a PLS-based approach and other top-ranking classification methods is proposed, as well as a study of PLS with boosting, which is a novel approach. The rest of the chapter deals with the use of PLS dimension reduction as a visualization tool and the connection between PLS dimension reduction and variable selection, including the proof that the so-called BSS/WSS ratio is a monotonic transformation of the squared coefficient in the first PLS component. Methods from Chapter 5 are found in a paper published in the journal "Statistical Applications in Genetics and Molecular Biology" (Boulesteix, 2004).

1.3 Notations

$X_1, \ldots, X_p$ denote the gene expression levels. In the whole work, they are continuous predictor variables. $\mathbf{x} = (X_1, \ldots, X_p)^T$ denotes the corresponding random vector. Y denotes the class membership. $(\mathbf{x}_i, Y_i)_{i=1,\ldots,n}$ is the observed stratified data set, with $\mathbf{x}_i^T = (x_{i1}, \ldots, x_{ip})$ denoting measurements of the p predictors and Y_i the class membership for observation i. For $k = 1, \ldots, K$, $\mathbf{x}_{k_1}, \ldots, \mathbf{x}_{k_{n_k}}$ denote the observations from class k, where n_k is the number of observations from class k and $k_1, \ldots, k_{n_k}$ are the indices of the observations from class k in the data set $(\mathbf{x}_i, Y_i)_{i=1,\ldots,n}$. Thus, for $k = 1, \ldots, K$, $i = 1, \ldots, n_k$, one has $Y_{k_i} = k$. $\mathbf{X}$ is the $n \times p$ matrix which contains $\mathbf{x}_i$ in its ith row, for $i = 1, \ldots, n$.

In the following,

$$\boldsymbol{\mu} = \mathrm{E}(\mathbf{x}) = (\mu_1, \ldots, \mu_p)^T$$

denotes the mean vector of $\mathbf{x}$ and $\hat{\boldsymbol{\mu}}$ is the empirical mean vector of $\mathbf{x}$, i.e.

$$\hat{\boldsymbol{\mu}} = \frac{1}{n}\mathbf{X}^T\mathbf{1}_n = (\hat{\mu}_1, \ldots, \hat{\mu}_p)^T,$$

where $\mathbf{1}_n$ is the vector of ones of length n. $\mathbf{\Sigma}$ is the $p \times p$ covariance matrix of $\mathbf{x}$:

$$\mathbf{\Sigma} = \mathrm{COV}(\mathbf{x}) = \mathrm{E}((\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^T).$$

$\mathbf{S}$ denotes the usual unbiased estimator of $\mathbf{\Sigma}$:

$$\mathbf{S} = \frac{1}{n-1} \sum_{i=1}^{n} (\mathbf{x}_i - \hat{\boldsymbol{\mu}})(\mathbf{x}_i - \hat{\boldsymbol{\mu}})^T,$$

whereas $\hat{\mathbf{\Sigma}}$ is the maximum-likelihood estimator of the covariance matrix $\mathbf{\Sigma}$:

$$\hat{\mathbf{\Sigma}} = \frac{n-1}{n}\mathbf{S}.$$

For $k = 1, \ldots, K$, n_k is the number of observations in class k. $\boldsymbol{\mu}_k$ denotes the mean of $\mathbf{x}$ within class k:

$$\boldsymbol{\mu}_k = \mathrm{E}(\mathbf{x}|Y = k) = (\mu_{k1}, \ldots, \mu_{kp})^T,$$

and $\hat{\boldsymbol{\mu}}_k = (\hat{\mu}_{k1}, \ldots, \hat{\mu}_{kp})^T$ is the empirical mean of $\mathbf{x}$ within class k:

$$\hat{\boldsymbol{\mu}}_k = \frac{1}{n_k} \sum_{i=1}^{n_k} \mathbf{x}_{k_i}.$$

$\mathbf{\Sigma}_k$ denotes the within-group covariance matrix of $\mathbf{x}$ for class k:

$$\mathbf{\Sigma}_k = \mathrm{COV}(\mathbf{x}|Y = k).$$

The within-group covariance matrix is defined as

$$\mathbf{\Sigma}_W = \sum_{k=1}^{K} p_k \mathbf{\Sigma}_k,$$

where p_k is the probability of class k: $p_k = p(Y = k)$. In the following,

$$\mathbf{\Sigma}_B = \sum_{k=1}^{K} p_k (\boldsymbol{\mu}_k - \boldsymbol{\mu})(\boldsymbol{\mu}_k - \boldsymbol{\mu})^T,$$

denotes the between-group covariance matrix. The decomposition

$$\mathbf{\Sigma} = \mathbf{\Sigma}_W + \mathbf{\Sigma}_B$$

is a decomposition of the type "total variability = variability within groups + variability between groups" and follows from the formula

$$\mathrm{COV}(\mathbf{x}) = \mathrm{E}(\mathrm{COV}(\mathbf{x}|Y)) + \mathrm{COV}(\mathrm{E}(\mathbf{x}|Y)).$$

$\mathbf{S}_k$ denotes the unbiased estimator of $\mathbf{\Sigma}_k$:

$$\mathbf{S}_k = \frac{1}{n_k - 1} \sum_{i=1}^{n_k} (\mathbf{x}_{k_i} - \hat{\boldsymbol{\mu}}_k)(\mathbf{x}_{k_i} - \hat{\boldsymbol{\mu}}_k)^T$$

Natural unbiased estimators for $\mathbf{\Sigma}_W$ and $\mathbf{\Sigma}_B$ are $\mathbf{W}/(n-K)$ and $\mathbf{B}/(n(K-1))$ respectively, where

$$\mathbf{W} = \sum_{k=1}^{K} (n_k - 1)\mathbf{S}_k.$$

and

$$\mathbf{B} = \sum_{k=1}^{K} n_k (\hat{\boldsymbol{\mu}}_k - \hat{\boldsymbol{\mu}})(\hat{\boldsymbol{\mu}}_k - \hat{\boldsymbol{\mu}})^T$$

$\hat{\mathbf{\Sigma}}$ can be decomposed into

$$\hat{\mathbf{\Sigma}} = \frac{1}{n}\mathbf{W} + \frac{1}{n}\mathbf{B}.$$

Chapter 2

Classification with application to microarray data

2.1 Overview of classification with high-dimensional microarray data

Suppose we have n patients which belong to one of the different classes $1, \ldots, K$, where $K \geq 2$. Let Y denote the categorical random variable 'class membership'. In cancer studies, Y is the tumor class. $X_1, \ldots, X_p$ denote the gene expression levels and $\mathbf{x} = (X_1, \ldots, X_p)^T$ is the corresponding random vector. $X_1, \ldots, X_p$ are continuous variables. $(\mathbf{x}_i, Y_i)_{i=1,\ldots,n}$ is the observed data set, with $\mathbf{x}_i^T = (x_{i1}, \ldots, x_{ip})$ denoting measurements of the p predictors and Y_i the class membership for observation i. $\mathbf{X}$ is the $n \times p$ matrix which contains $\mathbf{x}_i$ in its ith row, for $i = 1, \ldots, n$.

Statisticians know a lot of methods to predict a class membership using continuous predictor variables. Linear and quadratic discriminant analysis, Fisher's linear discriminant, generalized linear models (e.g. logistic or probit regression, ridge logistic regression splines), nearest-neighbor classification, kernel density estimation, classification trees and related methods (bump hunting, multivariate adaptive regression), neural networks and

support vector machines are among the most well-known approaches. An overview of the methods mentioned above can be found in Hastie et al. (2001).

For financial and practical reasons, microarray studies rarely involve more than 200 experiments. Since several thousands of gene expression levels are measured, one faces the problematic situation '$n << p$'. The number of experiments is expected to grow in the next few years, since the price of microarrays tends to decrease and the experimental protocols are getting easier. However it is unrealistic to expect this number to grow exponentially, for both ethical and practical reasons. At the same time, the number of genes included in a microarray study is steadily increasing because of daily progress in the area of sequence analysis. Thus, the number of observations n will not get greater than the number of genes p in the near future, hence the need for statistical methods to handle many variables at the same time or to reduce the dimension p.

There are three main ways to handle high-dimensional data in the classification framework.

- The first approach consists to select a handful of relevant genes and apply a classical classification method on this small subset of genes. In the microarray literature, this approach is often denoted as gene selection, gene screening, variable selection, subset selection or gene filtering. Classical classification methods which have already been used in microarray data analysis are e.g. linear discriminant analysis, quadratic discriminant analysis, Fisher's linear discriminant, nearest-neighborhood classification (Dudoit et al., 2002), artificial neural networks (Kahn et al., 2001), Support Vector Machines (Furey et al., 2000). Some of the classical classification methods, like nearest-neighborhood classification, do not require explicitly $n > p$ but give poor classification accuracy in practice when the number of irrelevant variables is too large, as in microarray data. Other methods can not be applied if $n < p$: in classical discriminant analysis, empirical covariance matrices have to be inverted for the estimation of the discriminant function. This can not be done in the 'small n, large p' framework. A brief overview of current gene subset selection methods is given in section 2.3.

- An alternative approach is dimension reduction: instead of selecting a small subset of genes and eliminating the other, one can create new components which summarize the data as well as possible in some sense. The new components are then used as predictor variables with a classical classification method. This topic is examined in chapters 4 and 5.

- Alternatively, one can use a classification method which performs variable selection or variable weighting intrinsically and does not necessitate any preliminary variable selection. For example, variable selection is intrinsic in classification trees (CARTs). However, using CARTs on the whole microarray data set is not recommended for at least three reasons. First, it is very slow. Second, it lacks robustness: the obtained trees are very sensitive to small changes in the data. Third, trees obtained from microarray data with few observations often have few splittings: thus, many possibly interesting genes are ignored. In the context of classification trees, aggregation methods such as bagging (Breiman, 1996), boosting (Freund, 1995) or random forests (Breiman, 2001) often lead to spectacular improvements of the classification accuracy. They can also be seen as methods which perform variable selection intrinsically. Another method which performs variable selection intrinsically is the nearest centroid classifier which was especially designed for classification with microarray data (Tibshirani et al., 2002). Each observation from the test set is assigned to the nearest shrunken class centroid. Shrunken centroids are determined using only genes with a high d score, where d is a statistic of the type 'signal to noise'. The number of genes included in the analysis depends on the chosen threshold value for the d statistic. Thus, an intrinsic variable selection is performed. Shrinkage methods such as ridge regression and the LASSO (Tibshirani, 1996) and localized logistic classification with variable selection (Binder and Tutz, 2004) can also be considered as classification methods performing variable selection intrinsically.

These approaches can be combined. For example, Dettling and Bühlmann (2003) use a CART-based method after variable selection and Nguyen and Rocke (2002a) perform dimension reduction after variable selection.

The next section discusses four of the most common methods used to compare classification methods in the microarray framework. As an introduction, a few useful concepts from the decision theory are presented.

2.2 Comparing classification methods

An important problem which often occurs in practice is: "Should I rather use method A than method B to make a diagnosis using gene expression data ?". After a short introduction into decision theory, we present and discuss a few common approaches used in the microarray literature to compare classification methods.

2.2.1 Decision theory

In classification, one looks for a decision function d of the type:

$$\begin{array}{rccl} d: & \mathbb{R}^p & \rightarrow & \{1,\ldots,K\} \\ & \mathbf{x} & \mapsto & \hat{Y} = d(\mathbf{x}). \end{array}$$

The classification methods mentioned in section 2.1 (e.g. logistic regression, nearest-neighbor, etc) are methods to define such a decision function d using a learning data set.

A 'good' decision function is a function which can predict Y as well as possible, where 'well' can be defined in different ways. For example, one might want to find the decision function d which minimizes the overall error rate ϵ:

$$\epsilon(d) = p(d(\mathbf{x}) \neq Y). \tag{2.1}$$

Let $\epsilon_{kk'}(d)$ be defined as

$$\epsilon_{kk'}(d) = p(d(\mathbf{x}) = k' | Y = k), \text{for } k \neq k'. \tag{2.2}$$

Then one obtains

$$\epsilon(d) = \sum_{k=1}^{K} \sum_{k'=1, k' \neq k}^{K} \epsilon_{kk'}(d) \cdot p(Y = k). \tag{2.3}$$

Now suppose the cost induced by the misclassification of an observation from class k as k' is not equal for all pairs (k, k'). Then it is preferable to look for a decision function d which minimizes the Bayes risk

$$R(d) = \sum_{k=1}^{K} \sum_{k'=1, k' \neq k}^{K} c_{kk'} \cdot \epsilon_{kk'}(d) \cdot p(Y = k), \tag{2.4}$$

where $c_{kk'}$ is the cost occurring when an observation from class k is incorrectly assigned to class k'. As can be seen from equations 2.3 and 2.4, $R(d) = \epsilon(d)$ if $c_{kk'} = 1$ for all $k \neq k'$ and $c_{kk} = 0$ for all k.

In practice, the overall error rate and the Bayes risk of a given classifier d are unknown and have to be estimated from data. Given a decision function d and a data set, a natural unbiased estimator of ϵ is the proportion of observations which are misclassified by d. In concrete studies, one often has to compare classification methods like e.g. CART and linear discriminant analysis, i.e. methods to construct the decision function d. This is a more complex task than comparing two given decision functions d_1 and d_2. Procedures to compare different classification methods using a data set $(\mathbf{x}_i, Y_i)_{i=1,\ldots,n}$ are presented and discussed in the next section. These methods differ in the choice of the observations used to:

- construct the decision function(s),
- estimate ϵ.

2.2.2 Comparing two classification methods in practice

All four approaches output a numerical criterion for each of the considered classification methods. This criterion can be used to evaluate and compare methods. For all four approaches, the classification method A is considered better than the classification method B if the criterion output by A is lower than the criterion output by B.

- **Approach 1**. The whole data set $(\mathbf{x}_i, Y_i)_{i=1,\ldots,n}$ is used to construct the decision function d. The class of the observations from $(\mathbf{x}_i, Y_i)_{i=1,\ldots,n}$ is then predicted us-

ing d. The criterion is the estimated overall error rate of d, i.e. the proportion of misclassified observations. This approach is not recommended, because it underestimates the overall error rate and favors more complex classification methods which overfit the data. Using a complex method, it may be possible to construct a decision function which fits perfectly a particular data set. However, such a decision function might generalize poorly on independent data. Thus, Approach 1 should be avoided.

- **Approach 2**. An alternative approach consists to split the data set $(\mathbf{x}_i, Y_i)_{i=1,\ldots,n}$ into two non-overlapping data sets: a learning data set $\mathcal{L}$ and a test data set $\mathcal{T}$. $\mathcal{L}$ is used to construct the decision function d. $\mathcal{T}$ is then run through d. The criterion is the estimated overall error rate of d, i.e. the proportion of misclassified observations from $\mathcal{T}$. The partition into learning set and test set is sometimes fixed due to 'historical' reasons, for instance because the experiments were performed at two different times or in two different places. In this case, there might be some systematic difference between the learning set and the test set, for instance because of different laboratory assistants. That's why it is generally better to split the original data set at random. The major inconvenience of Approach 2 is that it is very sensitive to changes in the partition into learning and test sets. To circumvent this problem, one might prefer Approach 3.

- **Approach 3**. A preferable option is to repeat N times Approach 2 and compute the global criterion as the empirical mean of the N obtained criteria. Increasing N decreases the variance of the empirical mean. Thus, N should be as large as technically possible. The choice of the ratio between the size of the learning set and the size of the whole data set is important. Decreasing the ratio generally increases the empirical mean error rate, since the decision rules are built using less observations. Increasing the ratio increases the correlation between the estimated error rates obtained with the N partitions. Common values for the ratio are 2/3, 0.7 and 9/10. In a comparative study, the objective is not the estimation of the error rate in itself, but the ranking of different classification methods. Thus, small ratios

(e.g. 2/3 or 0.7) are recommended (Dudoit et al., 2002). Moreover, when comparing several classification methods, one should always use the same partitions.

- **Approach 4**. Cross-validation is a quite popular option. It consists to split the data set $(\mathbf{x}_i, Y_i)_{i=1,\ldots,n}$ into k non-overlapping data subsets of (approximately) equal size $S_1, \ldots, S_i, \ldots, S_k$. For each subset S_i, the following procedure is repeated. S_i is considered as a test data set. The learning data set is formed by the $k-1$ remaining subsets $S_1, \ldots, S_{i-1}, S_{i+1}, \ldots, S_k$ and used to construct a decision function d. The class of the observations from S_i are predicted using d. After this procedure was repeated for $i = 1, \ldots, k$, the criterion is computed as the proportion of misclassified observations. Two common choices for k are $k = 10$ and $k = n$. If $k = n$, the procedure is also called leave-one-out cross-validation and do not necessitate any random splitting. It turns out that approach 3 should be preferred to cross-validation for small sample microarray data (Braga-Neto and Dougherty, 2004).

In the microarray literature, a very common mistake is to perform a preliminary variable selection using the whole data set and follow Approach B, C or D based on the selected genes. Variable selection, if any, should always be considered as a part of the construction of decision functions. As such, variable selection must be performed using only the learning data set. An extensive study of this topic can be found in (Nguyen and Rocke, 2002a).

2.3 Variable selection

Various variable selection schemes have been applied to microarray data with a double purpose:

- Variable selection may be performed as a preliminary step before classification, because the chosen classification method works only with a small subset of variables.
- Variable selection is of crucial interest for biologists who want to identify genes which are associated with specific diseases.

The variable selection methods found in the microarray literature can be divided into two distinct groups: univariate ranking methods and optimal subset selection.

2.3.1 Univariate ranking methods

Each variable is taken individually and a relevance score measuring the discriminating power of the variable is computed. The variables are then ranked according to their score. One can choose to select only the $\tilde{p}$ top-ranking variables (where $\tilde{p} < p$) or the variables whose score exceeds a given threshold. If the distribution of the score is known under the null-hypothesis that the variable is irrelevant, the p-value corresponding to the null-hypothesis can also be used as a variable score. Microarray data analysis is an extreme multiple testing situation, since p hypotheses are tested simultaneously. Thus, p-values should be handled with caution, for example by using a multiple testing procedure to control the false discovery rate, see Dudoit et al. (2003) for an overview of multiple testing in the microarray framework. One of the most common relevance scores is the F-statistic. For variable j, the F-statistic is defined as

$$F_j = \frac{\sum_{k=1}^{K} \sum_{i:Y_i=k} (\hat{\mu}_{kj} - \hat{\mu}_j)^2/(K-1)}{\sum_{k=1}^{K} \sum_{i:Y_i=k} (x_{ij} - \hat{\mu}_{kj})^2/(n-K)}. \tag{2.5}$$

Let us suppose that for all $j = 1, \ldots, p$, $k = 1, \ldots, K$,

- X_j is normally distributed within class k, with mean μ_{kj} and variance σ^2.
- The observations x_{ij}, $i = 1, \ldots, n$ are independent.

Then under the hypothesis

$$H_0 : \mu_{1j} = \cdots = \mu_{Kj} \tag{2.6}$$

versus

$$H_1 : \mu_{1j} \neq \mu_{kj} \text{ for at least one } k, \tag{2.7}$$

F_j is F-distributed with degrees of freedom $K-1$ and $n-K$. The corresponding p-value can be used as a relevance score. The t-statistic, which is also a very commonly used selection criterion in the case of binary responses, turns out to be a special case of the

F-statistic. The so-called BSS/WSS ratio used by Dudoit et al. (2002) equals the F-statistic up to a constant. On the whole, the F-statistic and its variants are by far the most commonly used relevance score in the microarray literature.

Since microarray data contain a lot of outliers and few observations, some authors (e.g. Dettling and Bühlmann (2003)) prefer to use a more robust statistic such as Wilcoxon's rank sum statistic for the case $K = 2$. For variable j, the only assumption to be made is the independence of the observations $x_{1j}, \ldots, x_{nj}$. If $rank(x_{ij})$ denotes the rank of x_{ij} in the sequence $x_{1j}, \ldots, x_{nj}$, the test statistic for variable j is given by

$$W_j = \sum_{i:Y_i=1} rank(x_{ij}). \tag{2.8}$$

Under the hypothesis

$$H_0 : \text{median}(X_j|Y = 1) = \text{median}(X_j|Y = 2) \tag{2.9}$$

versus

$$\text{median}(X_j|Y = 1) \neq \text{median}(X_j|Y = 2), \tag{2.10}$$

W_j has a Wilcoxon distribution with degrees of freedom n_1 and n_2. The corresponding p-value obtained for variable j can be used as a relevance score. For multicategorical responses ($K > 2$), one can use the Kruskal-Wallis test statistic (Hollander and Wolfe, 1973), which can be seen as a generalization of Wilcoxon's rank sum test statistic.

Various ad-hoc relevance scores can be found in the microarray literature, such as the so-called 'TNoM' combinatoric score (Ben-Dor et al., 2000) or a variant of the F-statistic, which is one of the first relevance scores proposed in the literature (Golub et al., 1999). These scores are either of the type 'signal-to-noise' (like the F-statistic) or based on ranks (like Wilcoxon's rank sum statistic). They generally lead to similar orderings of the variables.

If variables are selected according to an individual relevance score, correlations and interactions with other variables are omitted, which is an important drawback. In some cases, the subset of the top-ranking variables is not the best subset in terms of classification accuracy. Two examples are discussed below.

- The top-ranking variables might be strongly correlated, for instance because they correspond to the same DNA sequence or because they are corregulated by a common regulator. Let us consider an extreme case: the two top-ranking genes have identical expression levels across the n patients. It is then better to select genes 1 and 3 than genes 1 and 2, because genes 1 and 2 give redundant information. This topic is examined in detail by Jäger et al. (2003).

- Another reason why the subset of top-ranking genes might not be the optimal subset for classification is the existence of interactions between genes. Two genes which have a low individual relevance score might separate the different classes well when they are considered together. Emerging patterns (see section 3) are an example of such a data structure. In this context, severe univariate variable selection might be inadequate.

That's why many authors try to find optimal subsets based on more complex criteria than individual relevance scores. This approach is briefly presented in the following.

2.3.2 Optimal subset selection

Methods to find optimal subsets are characterized by

- **The relevance score used to evaluate the subsets of variables**. Criteria for subsets of variables can be divided into two groups. The first group consists of scores which can be seen as generalizations of univariate criteria and do not involve the construction of a decision function. For instance, Chilingaryan et al. (2002) compute the empirical Mahalanobis distance between the two classes for each candidate subset of variables. Such methods are usually denoted as filter methods in contrast to wrapper methods. In wrapper methods, the optimality criterion of a each subset of variables is based on the accuracy of decision functions built using only these variables. Wrapper methods are generally computationally intensive and more difficult to set up than filter methods.

- **The search algorithm used to explore the space of the possible subsets**, since it is not possible to examine all the possible subsets. Furey et al. (2000) and Model et al. (2001) use a backward selection selection procedure, whereas Bo and Jonassen (2002) select pairs of variables following a forward selection scheme. Alternative search approaches include genetic algorithms (Ooi and Tan, 2003; Li et al., 2001) or simpler stochastic search algorithms (Chilingaryan et al., 2002).

Methods that look for optimal subsets of variables have two major drawbacks. First, they are often computationally intensive and difficult to set up. Second, they generally suffer from overfitting: even if the found subsets are optimal for the considered learning data set, they might generalize poorly on independent data. In this thesis, we are interested in alternatives to variable selection.

Chapter 3

Emerging and Interaction Patterns

3.1 Introduction

In classification interaction structures among predictors may be used explicitly or implicitly. In linear discriminant analysis or logistic regression a familiar way to exploit interactions is the incorporation of interaction terms into the linear predictor. Nonparametric classifiers like nearest neighborhood classifiers do not specify the interaction structure explicitly but rely on its implicit use. Tree based methods like CARTs (classification and regression trees, registered trademark by Salford Systems) as suggested by Breiman et al. (1984) make interaction structures the central issue. The same holds for early versions of trees, where the detection of interaction structures gave the algorithm its name, i.e. AID for automatic interaction detection (Morgan and Sonquist, 1963). More recently, specific interaction structures called emerging patterns have been introduced by Dong and Li (1999) and applied to high-dimensional gene expression analysis in Li and Wong (2003). An alternative concept which is related to interactions is the search for boxes in the feature space in which the response variable has a particular distribution.

Bump hunting as suggested by Friedman and Fisher (1999) is a method to seek boxes in which the response is as high as possible. A short overview on bump hunting is given in Hastie et al. (2001). In the following we will consider simple interaction structures of the emerging pattern type which have the form

$$\{X_1 > \theta_1\} \cap \{X_2 \leq \theta_2\} \cap \cdots \cap \{X_d > \theta_d\}$$

where $X_1, \ldots, X_d$ are covariates and $\theta_1, \ldots, \theta_d$ are thresholds to be estimated. An interaction structure of this type will be called an interaction pattern. For simplicity, it will be abbreviated by P. Emerging patterns as considered by Dong and Li (1999) are interaction patterns which discriminate between two classes in a specific sense. Let $\mathbf{x}^{\mathrm{T}} = (X_1, \ldots, X_p)$ denote the random vector of covariates and Y the class indicator which can take the values 1 and 2. Let $n_{P,j}$ denote the number of observations from class j in P. According to the definition of Dong and Li (1999), a pattern P is a ρ-emerging pattern from class i to class j if the growth rate from i to j GR_{ij} is larger than ρ, where GR_{ij} is defined as

$$GR_{ij}(P) = \frac{n_{P,j}/n_j}{n_{P,i}/n_i}.$$

The definition is based on a heuristic rather than a statistical criterion. The focus in Dong and Li (1999) is on data mining and therefore on algorithms that find all the ρ-emerging patterns without regard to relevance. The problem of overfitting is neglected. By investigating a large number of possible patterns, it is always possible to find a large growth rate in the training data, but in an independent test data set, growth rates are usually much lower. Another drawback of Dong and Li's patterns is that the definition is restricted to the case $K = 2$.

In this chapter, we suggest a more general definition of interaction patterns which is based on the underlying probability and allows for more than two classes. In addition, a CART-based method is proposed to identify statistically relevant interactions in cases where many variables are potential candidates. In gene expression data where the expression levels of thousands of genes are measured simultaneously the challenge is the number of predictors. The objectives of our approach are identification of interaction patterns as well as their use in classification. In the microarray framework, the detection of interactions

aims at the analysis of gene expression profiles to uncover how combinations of genes are linked to specific diseases. The classification part aims at the improvement of classification rules.

Two main papers address the problem of the discovery of emerging patterns. While Dong and Li (1999) focus on an enumeration based algorithm to find all patterns with large empirical growth rates, Boulesteix et al. (2003) propose a CART-based method. Here, we suggest an improvement of the CART-based method developed in Boulesteix et al. (2003). The method allows to identify candidate patterns and only those which satisfy a statistical criterion are selected as interaction patterns. In addition, a pruning criterion is used to prevent too long and irrelevant *IP*s. A simpler version of the algorithm which is restricted to the case of two classes is given in Boulesteix et al. (2003). The present chapter can be seen as an extension of Boulesteix et al. (2003) with respect to three important issues. First, the concept of interaction patterns is mapped into a theoretical statistical framework. Second, various statistical aspects of interaction patterns are investigated (e.g. receiver operating characteristic, length of the interaction patterns, survival plot). Third, the concept of interaction patterns as well as the discovering algorithm are adapted to handle multicategorical response variables: all the variables involved in the patterns are tested for relevance (not only the variable involved in the last splitting, as in Boulesteix et al. (2003)) .

3.2 Definition of interaction patterns

3.2.1 Interaction Patterns for two classes

In this section, we first consider the binary case. For simplicity, the variables $X_1, \ldots, X_p$ are assumed to be metric, although the method is easily generalized to categorical variables. A pattern may be characterized as a collection of restrictions on a subset of variables $X_{j_1}, \ldots, X_{j_d}$. The restrictions have the simple form $X_j \leq \theta_j$ or $X_j > \theta_j$. Let I_j denote

an interval of this type, then the restrictions are collected in

$$X_{j_1} \in I_1, \ldots, X_{j_d} \in I_d.$$

More formally, the restrictions may be represented as a subset of the observation space $\mathbb{R}^p$ or in terms of the underlying event. As subset of $\mathbb{R}^p$ they are given by

$$\{\mathbf{x}|X_{j_1} \in I_1\} \cap \cdots \cap \{\mathbf{x}|X_{j_d} \in I_d\}.$$

For random variables $X_1, \ldots, X_p$ the underlying event for pattern P is given by

$$P = A_{j_1} \cap \cdots \cap A_{j_d},$$

where $A_s = \{\omega|X_s(\omega) \in I_s\}$. The pattern P may be simply identified by the sequence of variables and corresponding intervals $\{(j_s, I_s), s = 1, \ldots, d\}$ where d is the order of the pattern. In addition, let $P_{\setminus j}$ denote the pattern where the restriction for variable j is omitted, i.e.

$$P_{\setminus j} = \cap_{i \in \{j_1, \ldots, j_d\} \setminus \{j\}} A_i.$$

The original pattern is easily obtained by $P = P_{\setminus j} \cap A_j$.

Definition 3.1. *Interaction pattern for two classes*
For $\eta > 1$, P is called a η- Interaction Pattern (IP) for class k_0 if

$$\frac{p(P|Y = k_0)}{p(P|Y \neq k_0)} > \eta, \tag{3.1}$$

and for all $j \in \{j_1, \ldots, j_d\}$ the condition

$$\frac{p(P_{\setminus j}|Y = k_0)}{p(P_{\setminus j}|Y \neq k_0)} < \frac{p(P|Y = k_0)}{p(P|Y \neq k_0)} \tag{3.2}$$

holds.

In simple words, an interaction pattern is a condition on a collection of covariates for which the probability of occurrence is larger in one of the classes (equation (3.1)) and such that every involved covariate actually contributes to the ratio between the probabilities of occurrence within classes (equation (3.2)). The probabilities involved in the definition are unknown. Therefore, given a candidate pattern P, the data are used to decide if it

is an interaction pattern fulfilling equations (3.1) and (3.2). One option is to base the decision on a statistical test. For fixed k_0, condition (3.1) may be investigated by testing the hypothesis

$$H_0^{(1)} : p(P|Y = k_0) \ \leq \ p(P|Y \neq k_0).$$

For simplicity $\eta = 1$ is used. Then testing of $H_0^{(1)}$ is equivalent to one-sided independence testing in the following 2×2 contingency table with rows given by presence or non-presence of pattern P and columns defined by the classes.

	$Y = k_0$	$Y \neq k_0$	
P	n_{P,k_0}	$n_{P,\overline{k}_0}$	n_P
$\overline{P}$	$n_{\overline{P},k_0}$	$n_{\overline{P},\overline{k}_0}$	$n_{\overline{P}}$

In the contingency table P stands for presence of a specific pattern P and $\overline{P} = \mathbb{R}^p \setminus P$ denotes the non-presence of P. One can use for instance Fisher's exact test, which allows one-sided testing and is also valid for small numbers of observations. An overview on independence testing in contingency tables is given in Agresti (2002). The hypothesis $H_0^{(1)}$ is rejected by the chosen independence test (for instance Fisher's test) to the significance level α_1 if $p^{(1)} < \alpha_1$, where $p^{(1)}$ denotes the p-value obtained by testing of $H_0^{(1)}$. P is selected as an interaction pattern only if $p^{(1)} < \alpha_1$ holds. For the investigation of condition (3.2) it is useful to reformulate the condition. Since

$$\frac{p(P_{\setminus j}|Y = k_0)}{p(P_{\setminus j}|Y \neq k_0)} < \frac{p(P|Y = k_0)}{p(P|Y \neq k_0)}$$

is equivalent to

$$\frac{p(P_{\setminus j} \cap \overline{A}_j|Y = k_0)}{p(P_{\setminus j} \cap \overline{A}_j|Y \neq k_0)} < \frac{p(P|Y = k_0)}{p(P|Y \neq k_0)} \tag{3.3}$$

condition (3.2) may be investigated by one-sided independence testing in the following contingency table:

	$Y = k_0$	$Y \neq k_0$	
$P = P_{\setminus j} \cap A_j$	$n^{(j)}_{A,k_0}$	$n^{(j)}_{A,\overline{k}_0}$	n_P
$P_{\setminus j} \cap \overline{A}_j$	$n^{(j)}_{\overline{A},k_0}$	$n^{(j)}_{\overline{A},\overline{k}_0}$	$n_{P_{\setminus j}} - n_P$

Let $\gamma^{(j)}$ denote the associated odds ratio

$$\gamma^{(j)} = \frac{p(P \cap \{Y = k_0\})/p(P \cap \{Y \neq k_0\})}{p(P_{\setminus j} \cap \overline{A}_j \cap \{Y = k_0\})/p(P_{\setminus j} \cap \overline{A}_j \cap \{Y \neq k_0\})}.$$

Then, condition (3.3) can be reformulated as $\gamma^{(j)} > 1$. To investigate condition (3.3), one has to test for all j the hypothesis

$$H_0^{(2,j)} : \gamma^{(j)} = 1 \text{ vs. } H_1^{(2,j)} : \gamma^{(j)} > 1.$$

An option is to use Fisher's one-sided independence test again. The hypothesis $H_0^{(2,j)}$ is rejected by the chosen independence test to the significance level α_2 if $p^{(2,j)} < \alpha_2$, where $p^{(2,j)}$ denotes the p-value obtained by testing of $H_0^{(2,j)}$. P is selected as an interaction pattern only if $\max_j p^{(2,j)} < \alpha_2$ holds, i.e. for all $j \in \{j_1, \ldots, j_d\}$, $H_0^{(2,j)}$ has to be rejected. The number of involved variables represents the order of the interaction pattern and is denoted by d. Patterns of order 1 are explicitly allowed. In the following, empirical interaction patterns are simply denoted as *IP*s. The connection to emerging patterns is easily derived. In the emerging pattern literature which uses terminology from data mining the support is defined by $supp_k(P) = n_{P,k}/n_k$. This is an unbiased estimate of the probability $p(P|Y = k)$. The crucial difference between the present approach and the emerging pattern approach in data mining is that in the latter approach growth rates are simple descriptive tools and only condition (3.1) is investigated.

3.2.2 Generalization to multicategorical response

In practice, categorical variables often have more than two possible classes. In this section, we address the problem of multicategorical responses ($K > 2$) and propose a generalization of the definition of *IP*s.

Definition 3.2. *Interaction pattern for more than two classes*
For $\eta > 1$, P is called a η-Interaction Pattern (IP) for the class k_0 if

$$\frac{p(P|Y = k_0)}{p(P|Y = k)} > \eta \tag{3.4}$$

holds for all k and for all j from $\{j_1, \ldots, j_d\}$ one has

$$\frac{p(P_{\setminus j}|Y = k_0)}{p(P_{\setminus j}|Y \neq k_0)} < \frac{p(P|Y = k_0)}{p(P|Y \neq k_0)}. \tag{3.5}$$

For fixed k_0, condition (3.4) may be investigated by testing the hypotheses

$$H_0^{(1,k)} : p(P|Y = k_0) \leq p(P|Y = k)$$

for all $k \neq k_0$. The hypothesis $H_0^{(1,k)}$ is rejected by the chosen independence test (for instance Fisher's test) to the significance level α_1 if $p^{(1,k)} < \alpha_1$, where $p^{(1,k)}$ denotes the p-value obtained by testing of $H_0^{(1,k)}$. For fixed α_1, P is selected as an interaction pattern if $\max_{k \neq k_0} p^{(1,k)} < \alpha_1$ holds.
Condition (3.5) can be investigated using the same procedure as for *IP*s for two classes.

3.3 Discovering interaction patterns with trees

Interaction patterns and single leaves of classification trees have similar structures and properties. Thus, we propose to use the well-known and fast CART-algorithm proposed in Breiman et al. (1984) to discover interaction patterns.

3.3.1 Tree methodology

Classification trees are an efficient exploratory tool to detect structures in data (Breiman et al., 1984). They are based on recursive partitioning whereby the measurement space $\mathbb{R}^p$ is successively split into subsets. Let $\mathbf{x}^{\mathrm{T}} = (X_1, \ldots, X_p) \in \mathbb{R}^p$ denote the vector of covariates. If C is a subset of $\mathbb{R}^p$ (corresponding to the partitioning of $\mathbb{R}^p$ into C and $\overline{C} = \mathbb{R}^p \setminus C$), the split of C based on variable X_j divides C into

$$C_1(j, \theta) = \{\mathbf{x} \in C | X_j \leq \theta\},$$

$$C_2(j, \theta) = \{\mathbf{x} \in C | X_j > \theta\}.$$

Thus the subset C is split by use of one variable, X_j, with the split simply depending on a threshold θ from the range of X_j. By starting with $C = \mathbb{R}^p$ and performing successive splittings one obtains a tree. After d splittings, one obtains subsets of $\mathbb{R}^p$ of the form

$$\{\mathbf{x}|X_{i_1} \leq \theta_1\} \cap \{\mathbf{x}|X_{i_2} > \theta_2\} \cap \cdots \cap \{\mathbf{x}|X_{i_d} \leq \theta_d\}.$$

A subset is identical to a pattern P given by the sequence $\{(j_s, I_s), s = 1 \ldots, d\}$ where j_s identifies the variable and I_s specifies the interval which in the simple case of binary splits has the form $I_s = (-\infty, \theta_s]$ or $I_s = (\theta_s, +\infty)$. The relationship between decision trees and patterns is simple: a pattern is equivalent to a leaf.

Splitting criterion

Given a pattern P of order d, an additional split in variable j at θ yields a $(d+1)$-dimensional pattern. Let

$$P \cap A = P \cap \{\omega|X_j(\omega) \in I_j\}$$

denote the new pattern where $I_j = (-\infty, \theta_j]$ or $I_j = (\theta_j, +\infty)$. Thus starting from P one obtains for the transition from P to $P \cap A$ the transition contingency table

	$Y=1$	$\ldots$	$Y=K$
$P \cap A$	$n_{PA,1}$	$\ldots$	$n_{PA,K}$
$P \cap \overline{A}$	$n_{P\overline{A},1}$	$\ldots$	$n_{P\overline{A},K}$
	$n_{P,1}$	$\ldots$	$n_{P,k}$

The margins $n_{P,k}$ for k from $\{1, \ldots, K\}$ represent the number of observations from class k in pattern P.

The new split is chosen to minimize a splitting criterion. One of the most common criteria is the deviance, also called cross-entropy, see Hastie et al. (2001). The deviance of a pattern P corresponds to the fit of the model

$$p(P|Y = 1) = \cdots = p(P|Y = K).$$

Let n denote the total number of observations and n_k the number of observations from class k. The deviance has the form

$$\begin{aligned} D(P) &= 2\sum_{k=1}^{K}\{n_{P,k}\log\frac{n_{P,k}/n_k}{n_P/n} + n_{\overline{P},k}\log\frac{n_{\overline{P},k}/n_k}{n_{\overline{P}}/n}\} \\ &= 2\sum_{k=1}^{K}\{n_{P,k}\log\frac{\hat{p}(P|k)}{\hat{p}(P)} + n_{\overline{P},k}\log\frac{\hat{p}(\overline{P}|k)}{\hat{p}(\overline{P})}\} \\ &= 2\sum_{k=1}^{K} n_k KL(\hat{p}(P|k),\hat{p}(P)) \end{aligned}$$

where $n_P = \sum_{k=1}^{K} n_{P,k}$, $\hat{p}(P|k) = \frac{n_{P,k}}{n_k}$, $\hat{p}(P) = \frac{n_P}{n}$, and KL stands for the Kullback-Leibler distance

$$KL(p,q) = p\log\frac{p}{q} + (q-p)\log\frac{1-p}{1-q}.$$

The new split which characterizes A is chosen to minimize the conditional deviance $D(P\cap A|P)$ given by

$$D(P\cap A|P) = D(P\cap A) - D(P)$$

and tests the hypothesis

$$p(P\cap A|Y=1) = \cdots = p(P\cap A|Y=K)$$

given $p(P|Y=1) = \cdots = p(P|Y=K)$. The conditional deviance can also be written as

$$D(P\cap A|P) = 2\sum_{k=1}^{K} n_{P,k} KL(\hat{p}(P\cap A|k),\hat{p}(P\cap A)).$$

Various other splitting criteria have been used to grow trees, for instance the Gini-Index or the misclassification error, see Hastie et al. (2001).

Stopping Criterion

The splitting criterion characterizes the way the tree is grown. In addition a stopping-criterion has to be chosen. In the tree literature, various stopping criteria have been proposed, for instance by Breiman et al. (1984). Let us consider a leaf P. One can decide not to split this leaf if its order exceeds a fixed number, if it contains less than a fixed number of observations or if the best split would yield at least one leaf with less than a fixed number of observations. Many other more sophisticated methods to limit the depth of trees such as cost-complexity pruning described in Hastie et al. (2001) have been investigated.

3.3.2 Discovering Algorithm

When using trees for the detection of interaction patterns the main problem is that trees are constructed by recursive partitioning. What is an advantage in terms of computation time and structuring turns into a disadvantage since the leaves share splits in the same variables. In particular, all leaves share the same root splitting. Patterns that do not involve the root splitting variable will never be found by a single tree. Therefore the proposed algorithm is based on the growing of several trees which use different sets of variables from which the splitting starts.

The first stage is designed to find candidate patterns. Here candidate patterns are generated which are investigated in the following steps. The selection is directly based on classification trees. The iterative algorithm grows a tree on the available set of variables and then removes the variable that generates the first split from the available set of variables. Thus patterns result which include different sets of variables. In applications we use the CART-algorithm `tree` (Ripley, 1996) implemented in the `tree` library in `R` (R-Development-Core-Team, 2004) with the deviance as splitting criterion. As stopping criterion, we fix `mincut` (minimal number of observations to include in either child node) at 5, `minsize` (minimal allowed node size) at 10 and `mindev` (minimal ratio between within-node deviance and the root node deviance for the node to be split) at 0.01. These settings are the default values of the `R` program.

In a second stage, conditions (3.1) and (3.2) resp. (3.4) and (3.5) are tested for the selected candidates patterns. The significance levels for the tests (α_1 and α_2) as well as the test T to be used (e.g. Fisher's exact test) have to be specified as input. The whole procedure can be summarized by the following algorithm.

Stage 1: Candidate patterns

Grow a classification tree. Store the obtained leaves and eliminate the variable defining the first splitting of the tree from the set of input variables . Repeat this procedure until there is no more variable in the input set. Define S as the set of all obtained leaves.

Stage 2: Relevance of candidate patterns

For each leaf from S, define k_0 as the class that maximizes $\hat{p}(P|k)$.

1. For each leaf, for all $k \neq k_0$, test $H_0^{(1,k)}$ with test T to the significance level α_1. Eliminate from S all the patterns for which $\max_{k \neq k_0} p^{(1,k)} > \alpha_1$. This step corresponds to the testing of condition (3.1) resp. (3.4).

2. For all the remaining leaves from S, test $H_0^{(2,j)}$ for all j in $\{j_1, \ldots, j_d\}$ with test T to the significance level α_2. If $\max_j p^{(2,j)} > \alpha_2$, eliminate the variable for which $p^{(2,j)}$ is maximal from the interaction pattern. Repeat this procedure as long as variables are eliminated. This step corresponds to the testing of condition (3.2) resp. (3.5).

3. Repeat step 1 for all the leaves that have be shortened in step 2. This step is necessary to ascertain that the shortened patterns still fulfill condition (3.1) resp. (3.4).

4. Eliminate from S all the duplicated patterns.

The algorithm yields empirical interaction patterns which are based on tests with significance levels α_1 and α_2. Since many tests are performed the question of the overall significance level arises. This might be controlled for the given set of candidate patterns. It is however hard to control for the total procedure. Approaches to control the level for trees by maximally selected rank statistics are found in Lausen and Schumacher (1992). Instead of performing multiple testing, which would be very difficult in this framework, we follow an alternative approach by defining receiver operating curves which capture the performance of the algorithms for varying significance levels. This topic is addressed in the following section, where it is shown that the algorithm can detect 'ideal' theoretical interaction patterns with quite good accuracy.

3.3.3 Receiver Operating Characteristic

A popular method for summarizing the accuracy of a classification rule are receiver operating characteristic (ROC) curves. A ROC curve is a plot of the true-positive rates against the false-positive rates. In classification, curves result from the consideration of varying thresholds on the diagnostic scale. Let a disease be diagnosed if the diag-

nostic scale is larger than threshold γ. Then the true-positive and false-positive rates are functions of the threshold. The resulting ROC curve is convex under quite natural assumptions. A large body of literature deals with the concept and estimation of ROC curves. An early reference is Swets and Pickett (1982), more recent approaches to estimation are proposed in Lloyd (2000) and Venkatraman (2000). A version of the ROC curve is suggested here to illustrate the power of the method for detecting relevant interactions. The empirical ROC curve shows the hit rate HR (or sensitivity) against the false alarm rate FAR (or specificity), where HR and FAR depend on the parameters α_1 and α_2 and on the order of the interaction patterns. Let, for example, the order of the interaction patterns be fixed at $d = 2$, i.e. only pairs of variables are investigated. If p is the total number of variables, the total number of possible pairs of variables is $p(p-1)/2$. For each possible pair of variables, two binary variables are defined: r, which equals 1 if the pair forms a real *IP* of order 2 and 0 else, and d, which equals 1 if the pair is detected as an *IP* of order 2 by our method and 0 else. For each parameter setting (α_1 and α_2), we are interested in the following contingency table.

	d	$\overline{d}$	Σ
r	$n_{r,d}$	$n_{r,\overline{d}}$	n_{IP}
$\overline{r}$	$n_{\overline{r},d}$	$n_{\overline{r},\overline{d}}$	$p(p-1)/2 - n_{IP}$

The hit rate (HR) is defined as the proportion of discovered *IP*s among the n_{IP} real *IP*s, i.e.

$$HR = \frac{n_{r,d}}{n_{IP}}.$$

Similarly, the false alarm rate (FAR) is defined as the proportion of patterns which were detected as *IP*s among the non-*IP* patterns of the same order, i.e.

$$FAR = \frac{n_{\overline{r},d}}{p(p-1)/2 - n_{IP}}.$$

3.3.4 Simulation study

Study Design

In a simulation study it is investigated if the algorithm is able to detect simulated patterns. To make the problem more simple and reduce the number of parameters in the study, we consider only the case of two classes. Simulated data are obtained by following procedure. The number of variables contained in the data set is fixed at $p = 50$ and the number of observations is varied ($n = 50, 80$). These sample sizes correspond roughly to the typical values found in real gene expression data sets. From the 50 variables, 20 variables form pairwise interaction patterns (variable 1 forms an interaction pattern with variable 2, variable 3 with variable 4, and so on). The two threshold values defining each pattern are drawn randomly from the uniform distribution in $[0.25, 0.75]$. The type of inequality defining the pattern ($\leq$ or $>$) are also chosen randomly. Thus, various data configurations are obtained. In the subsets defined by the pattern and in its complement, the distribution is uniform. The rest of the 50 variables are generated randomly and independently of the class, following the uniform distribution in $[0, 1]$.
The simulation study is designed as follows. We generate 100 random data matrices following the procedure described above. Then the discovering algorithm is run on each data matrix with different values of α_1 and α_2. HR and FAR are estimated for each parameter setting from the contingency tables obtained for the 100 random data matrices. If an IP for class 1 is detected as an IP for class 2 or vice-versa, the IP is considered as false alarm. Finally the means across simulations are built.

Simulation results

Figure 3.1 displays the estimated ROCs for two values of n ($n = 50$ and $n = 80$): the hit rate is represented against the false alarm rate for different values of α_1 (ranging from 10^{-20} to 10^{-2}) and α_2 (ranging from 10^{-14} to 10^{-4}).It is seen that for decreasing significance levels the ROCs rather soon become horizontal, signaling a stable level of detection rate with the level depending on sample size. Within this stable level the

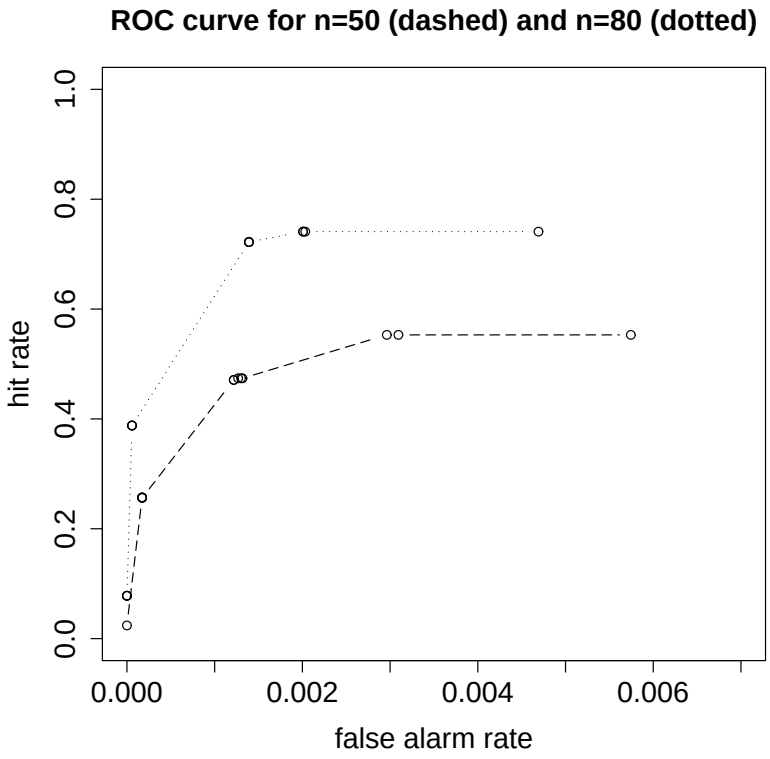

Figure 3.1: ROC curve for $n = 50$ and $n = 80$ (simulated data)

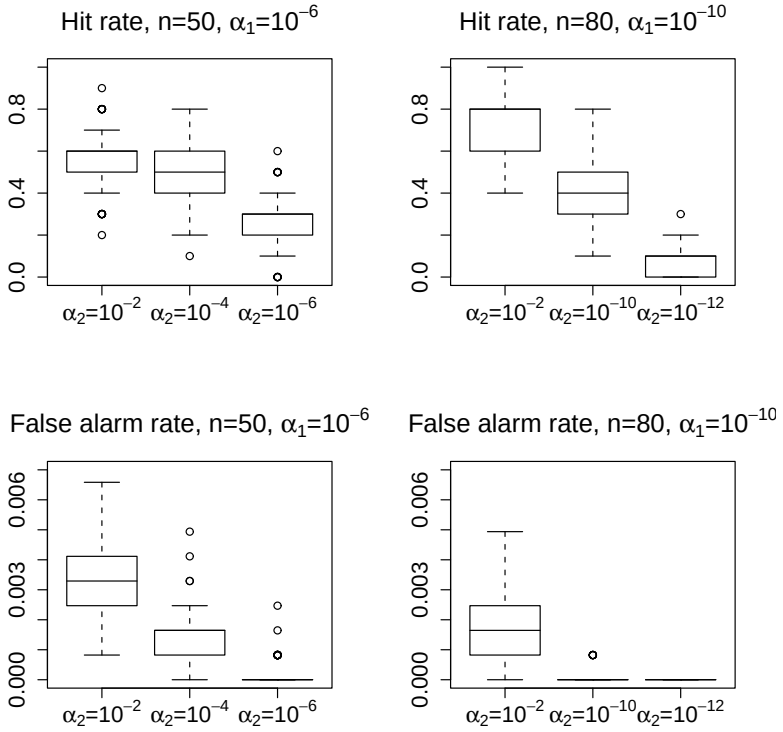

Figure 3.2: Boxplot of the hit rate (top) and false alarm rate (bottom) for $n = 50$ (left) and $n = 80$ (right), for different values of α_1 and α_2.

increase of significance levels only increases the false alarm rate. Figure 3.2 displays the boxplots of the hit rate and false alarm rate for $n = 50$ and $n = 80$, for three parameter settings which correspond to different zones of each ROC curve. As can be seen from Figure 3.2, the variance of the hit rate and false alarm rate across the 100 simulations is quite low, although not negligible.

3.4 Classification based on interaction patterns

3.4.1 Method

As can be seen from their definition, *IP*s might be useful to define predictors for classification. An inconvenience of the CART approach for data sets with many variables and few observations is that the tree often consists of few splittings. If one stops growing the tree too late, then some splittings might be statistically irrelevant. And if the growing is stopped too early, the decision rule depends on very few variables, and does not use most of the potentially interesting variables from the data set. By using *IP*s instead of tree leaves as a basis for the decision rule, one avoids a major problem: the decision rule uses much more information from the data set than a single tree does. In the following, a simple method to use *IP*s for classification is proposed. It is particularly suited for data sets with many (metric or categorical) variables and few observations. It can also be used for data sets with fewer variables, however without spectacular gain in accuracy.
From now on, we suppose that we have a learning data set $\mathcal{L}$ and a test data set $\mathcal{T}$. To predict the class of the observations from $\mathcal{T}$, we proceed as follows: First, *IP*s are found by applying the discovering algorithm on the training set $\mathcal{L}$. Second, m new binary covariates $Z_1, \ldots, Z_m$ are defined, where m denotes the number of found *IP*s. The variables

$$Z_j = \begin{cases} 1 & \text{for the j-th IP} \\ 0 & \text{otherwise} \end{cases}$$

indicate if the considered observation fulfills the conditions defining the considered *IP*. One obtains a transformed learning data set and a transformed test data set. Then virtually any supervised learning method can be applied to these data matrices, for instance linear discriminant analysis (with Bayes or Maximum-Likelihood rule), nearest neighborhood, logistic regression (if m is not too large), etc.

3.4.2 Study Design

Fifty random partitions into a learning data set $\mathcal{L}$ (containing $n-10$ observations) and a test data set $\mathcal{T}$ (containing 10 observations) are generated. For each partition, we proceed as follows. If the number of variables is high, a prescreening step is necessary. It is done by selecting the $\tilde{p}$ variables with lowest p-value for Wilcoxon's test testing the equality of the median in two classes, using only $\mathcal{L}$, as described in Dettling and Bühlmann (2003). If the number of classes K is greater than 2, the procedure is repeated K times: for the K classes successively, one tests the equality of the medians in the considered class and in all the other classes together. Then K groups of variables are selected. An alternative, which might seem more appropriated for multiclass problems, is to use the Kruskal-Wallis statistic. One applies the Kruskal-Wallis test to all genes and selects the $\tilde{p}$ genes with lowest p-values. However, the results obtained with this method are worse than with our procedure. One possible explanation is that the variables selected by the Kruskal-Wallis statistic do not necessarily separate well all K classes.

A prescreening is performed for three of the four investigated data sets: the leukemia, the colon and the SRBCT data sets, which are described in the following subsection. For each data set, the number of selected variables is fixed successively at $\tilde{p} = 50$, $\tilde{p} = 100$, $\tilde{p} = 200$ and $\tilde{p} = 300$. These values have been chosen, because for greater values of $\tilde{p}$, the discovering algorithm is computationally very intensive and for lower values of $\tilde{p}$, the number of found *IP*s is too low (or even zero for some of the partitions).
We run the discovering algorithm to find *IP*s, with different values for the parameter α_1 and $\tilde{p}$. To reduce the number of parameters, α_2 was fixed at 10^{-4}. α_1 is chosen on a heuristic basis. It is chosen so that the number of found *IP*s is non zero and smaller than, say 200 for all the partitions. For the tree topology parameters, the default values of the `R` program as described in section 3 are used.
Once the *IP*s are found, the new covariates are determined for all observations from $\mathcal{L}$ and $\mathcal{T}$. Then classification is carried out, either with nearest neighborhood classification based on 5 nearest neighbors (5-NN) or with linear discriminant analysis. Since the results were slightly better with 5-NN, the results with linear discriminant analysis are not shown.

For the nearest neighborhood classification, the Euclidean distance was used.

Mean error rates over the 50 partitions: For each parameter combination, the mean error rate over the 50 random partitions (i.e. the mean proportion of observations from the test set that were misclassified) is computed. The results are summarized in a table. For comparison, we also show the mean error rate obtained with classical CART, using the same `R` program as in the discovering algorithm, and with 5-NN applied directly on the $\tilde{p}$ genes. The latter is known to be one of the best performing discrimination methods for microarray data (Dudoit et al., 2002).

Observation-wise error rate: For each parameter combination and for each single observation, the proportion of times it was misclassified (out of the runs in which it was in the test set) is recorded. We summarize the results by means of survival plots as described in Dudoit et al. (2000): the proportion of observations classified correctly in at least $V\%$ of the runs is represented against V. The results are shown only for the best parameter combination for each data set.

Variables involved in *IP*s: An interesting issue is whether the variables involved in the *IP*s also perform good individually. To answer this question, we first rank the variables according to the Wilcoxon-statistic using all the observations. Then we represent the proportion of runs in which the variables were selected against their rank. We show the results for the colon data and the leukemia data with $\tilde{p} = 300$ and $p_G = 10^{-6}$ (for colon) and $p_G = 10^{-10}$ (for leukemia).

Number of *IP*s: The number of found *IP*s depends highly on the parameters. Typically, it increases with $\tilde{p}$ and α_1. The number of found *IP*s of each order is stored each time the discovering algorithm is run. The results are summarized by plotting the mean number of found *IP*s of each order over the 50 random partitions, for each data set and for different values of α_1. For the 3 gene expression data sets (leukemia, colon, SRBCT), we show only the results for $\tilde{p} = 300$. For smaller values of $\tilde{p}$ the plots show similar patterns, but the absolute numbers of *IP*s are lower.

3.4.3 Data sets

Leukemia Data: This data set was introduced in Golub et al. (1999) and contains the expression levels of 7129 genes for 47 ALL-leukemia patients and 25 AML-leukemia patients. It is included in the `R` library `golubEsets`. After data preprocessing following the procedure described in Dudoit et al. (2002), only 3571 variables remain. It is easy to achieve excellent classification accuracy on this data set, even with quite trivial methods as described in the original paper (Golub et al., 1999). Indeed, we found out that it is possible to find many *IP*s even if α_1 is very low. Thus, we set α_1 to $\alpha_1 = 10^{-10}$, $\alpha_1 = 10^{-12}$ and $\alpha_1 = 10^{-14}$ successively in our study.

Colon microarray data: The colon data set is a publicly available 'benchmark' gene expression data set which is extensively described in Alon et al. (1999). The data set contains the expression levels of $p = 2000$ genes for $n = 62$ patients from two classes. 22 patients are healthy patients and 40 have colon cancer. This data set is not as 'easy' as the leukemia data set. The classification accuracy is usually much lower, for instance using Support Vector Machines as described in Furey et al. (2000). It is also more difficult to find good *IP*s: α_1 was set heuristically to $\alpha_1 = 10^{-6}$, $\alpha_1 = 10^{-8}$ and $\alpha_1 = 10^{-10}$. Note that it is also possible to run the algorithm with $\alpha_1 = 10^{-12}$ and $\alpha_1 = 10^{-14}$ as for the leukemia data set, but with such values for α_1, no *IP* would be found.

SRBCT microarray data: This gene expression data set is presented in Kahn et al. (2001). It contains the expression levels of 2308 genes for 83 Small Round Blue Cells Tumor (SRBCT) patients belonging to one of the 4 tumor classes: Ewing family of tumors (EWS), non-Hodgkin lymphoma (BL), neuroblastoma (NB) and rhabdomyosarcoma (RMS). For this data set, α_1 was set to $\alpha_1 = 10^{-3}$, $\alpha_1 = 10^{-4}$, $\alpha_1 = 10^{-5}$ and $\alpha_1 = 10^{-6}$. These values are considerably higher than for the leukemia and colon data sets. One the possible explanations is that to be selected as an *IP* of type k ($k \in \{1, 2, 3, 4\}$), a pattern must have higher frequency in class k than in all three other classes, which is a stronger requirement than for the two-classes case.

Iris data: The famous (Fisher's and Anderson's) iris data set is included in the `R` library `MASS`. It gives 4 different measurements (sepal length and width, petal length and width)

for 150 flowers from each of the three species (class labels) setosa, versicolor, virginica. α_1 was set successively to $\alpha_1 = 10^{-4}$, $\alpha_1 = 10^{-8}$ and $\alpha_1 = 10^{-12}$.

3.4.4 Results

Mean error rate: The mean error rates for different values of the parameters are shown in Table 1 for the four data sets. For all four data sets, the new method performs much better than CART and is comparable to nearest neighborhood classification. Thus it is a competitor to one of the best classification procedures in microarray data with the advantage of providing information on the relevance of variables and interaction patterns. Surprisingly, the number of variables as well as the significance level α_1 do not seem to have strong influence on the results, provided *IP*s are found. For the case of two classes the method may be compared to the method suggested in Boulesteix et al. (2003). It turns out that the classification results with the new method are as good as with the former method for the colon data and better for the leukemia data.

Observation-wise error rate: As can be seen from the survival plot depicted in Figure 3.3, a large part of the error rate is due to observations that are misclassified each time they are included in the test data set. Indeed, even for small V, the proportion of observations classified correctly in at least $V\%$ of the runs is not 1, and it decreases slowly for large V. We found out that most of the 'problematic' observations are also misclassified by other classification methods (data not shown).

Number of *IP*s: As can be seen from Figure 3.4, the most frequent *IP*s are *IP*s of order 2. We did not found any *IP* of order 4, and few *IP*s of order 3. If the data sets contained more observations, it would certainly be possible to find more *IP*s of order 3 and 4 (or more). *IP*s of order 1 are quite frequent and correspond to variables that can separate the classes well. Unsurprisingly, the number of found *IP*s increases with α_1. An important fact which can not be seen in the figure is the high variability of the numbers of *IP*s over the random partitions: like CART, our learning method is not very robust, which can be seen as a drawback from the statistical point of view.

Variables involved in *IP*s: As can be seen from Figure 3.5 (for the colon and the

Colon data	$\alpha_1 = 10^{-6}$	$\alpha_1 = 10^{-8}$	$\alpha_1 = 10^{-10}$	$tree$	$5-NN$
50variables	0.16	0.17	0.19	0.30	0.16
100variables	0.14	0.14	0.16	0.30	0.14
200variables	0.15	0.15	0.15	0.29	0.15
300variables	0.15	0.15	0.15	0.29	0.15
Leukemia data	$\alpha_1 = 10^{-10}$	$\alpha_1 = 10^{-12}$	$\alpha_1 = 10^{-14}$	$tree$	$5-NN$
50variables	0.042	0.042	0.042	0.15	0.042
100variables	0.025	0.025	0.025	0.15	0.025
200variables	0.016	0.016	0.016	0.15	0.016
300variables	0.016	0.016	0.016	0.15	0.016
SRBCT data	$\alpha_1 = 10^{-4}$	$\alpha_1 = 10^{-5}$	$\alpha_1 = 10^{-6}$	$tree$	$5-NN$
20variables	0.0077	0.0077	0.0080	0.25	0.0077
50variables	0.0046	0.0046	0.0048	0.25	0.0046
Iris data	$\alpha_1 = 10^{-4}$	$\alpha_1 = 10^{-8}$	$\alpha_1 = 10^{-12}$	$tree$	$5-NN$
	0.035	0.035	0.035	0.059	0.035

Table 3.1: Mean error rate over 50 random partitions

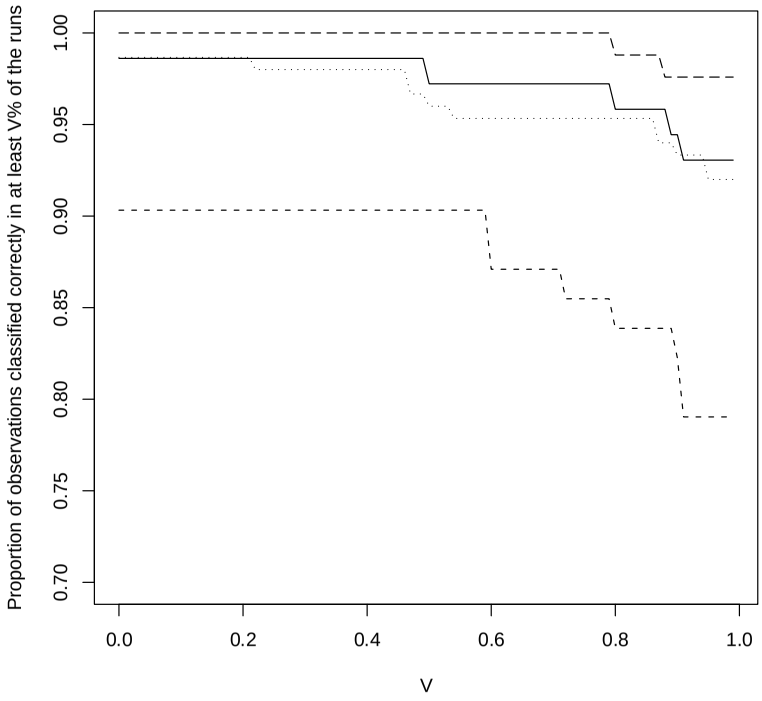

Figure 3.3: Survival plot for leukemia (solid), colon (dashed), SRBCT (longdash) and iris (dotted).

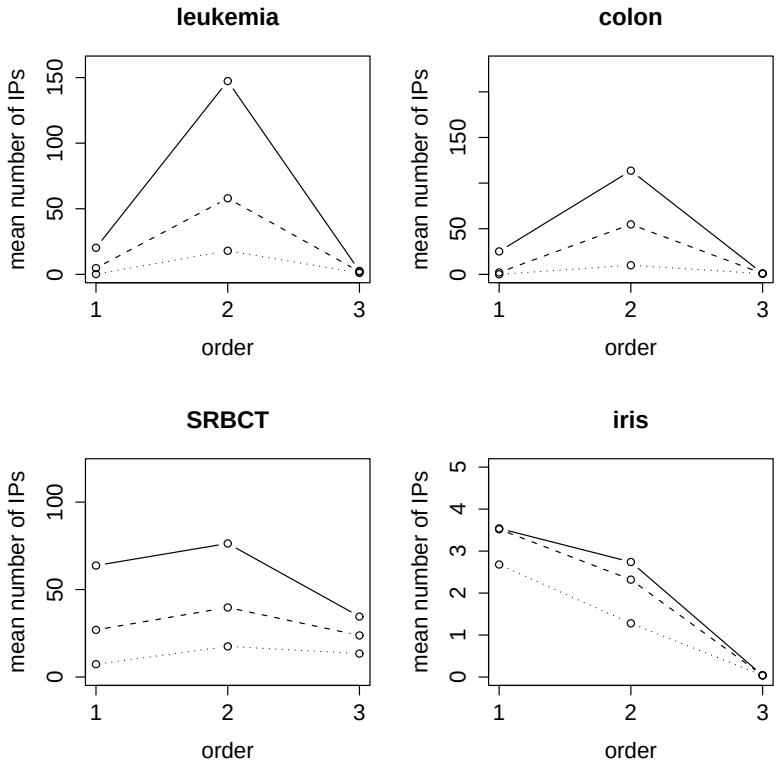

Figure 3.4: Number of *IP*s of each order. **Leukemia**: $\tilde{p} = 300$ and $\alpha_1 = 10^{-10}$ (solid), $\alpha_1 = 10^{-12}$ (dashed),$\alpha_1 = 10^{-14}$ (dotted). **Colon**: $\tilde{p} = 300$ and $\alpha_1 = 10^{-6}$ (solid), $\alpha_1 = 10^{-8}$ (dashed),$\alpha_1 = 10^{-10}$ (dotted). **SRBCT**: $\tilde{p} = 50$ and $\alpha_1 = 10^{-4}$ (solid), $\alpha_1 = 10^{-5}$ (dashed),$\alpha_1 = 10^{-6}$ (dotted). **Iris**: $\alpha_1 = 10^{-4}$ (solid), $\alpha_1 = 10^{-8}$ (dashed),$\alpha_1 = 10^{-12}$ (dotted).

leukemia data sets), most of the 'best' variables appear in at least one *IP* in most runs. But some 'less relevant' variables are involved in *IP*s in many runs as well, thus showing that variables that perform poorly individually might be interesting in association with others. On the whole, there seems to be a weak linear dependence between the variable rank and the frequency of selection. Separate analysis for *IP*s of order 1,2,3 would probably show stronger dependence for *IP*s of order 1 than for *IP*s of order 2 and 3. In the next section, we give an example on how interactions patterns can be interpreted in practice.

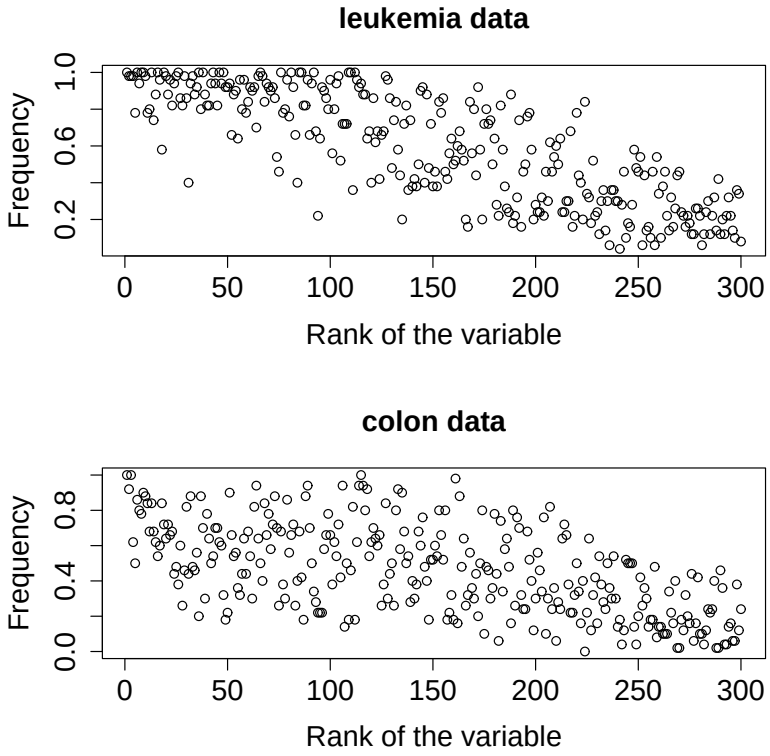

Figure 3.5: Proportion of runs in which the variable is involved in at least one *IP*

3.4.5 An example

In this section, we illustrate the concept of interaction patterns using a concrete example from the colon data. Since the goal is not the evaluation of the classification performance but the identification of relevant patterns, the discovering algorithm is run on the whole colon data set with $\alpha_1 = 10^{-10}$ and $\alpha_2 = 10^{-6}$. The discovering algorithm outputs a list of 9 IPs. For example, the genes R55310 and H72234 are found to form an IP for class 1 (normal tissue) which is defined by the restrictions

$$R55310 \ > \ 0.40$$

and

$$H72234 \ < \ -0.1,$$

as depicted in Figure 3.6. The corresponding biological hypothesis can be formulated as "in normal tissues, gene R55310 has a high expression level and gene H72234 has a low expression level". Hypotheses of this type might be used as a basis for the design of biological experiments.

3.5 Discussion

CART is one of the most popular classification methods in many application fields of statistics, for instance medicine. The main advantages that make it so popular are its simplicity and its interpretability. Moreover, scientists are often interested in the interaction structures implied by the CART decision rules. However, when the number of variables is high and the number of observations small, like in microarray data, CART usually performs poorly, because it uses only a very small part of the available information. Among the huge number of variables, it is often possible to find a few that separate the classes very good or even perfectly in the learning set. Thus, the obtained trees have very short branches and often perform poorly on new data sets. Modern methods based on aggregation of trees do improve the results a little as argued in Dudoit et al. (2002), but do not seem to overcome the problem completely. Instead of partitioning the input

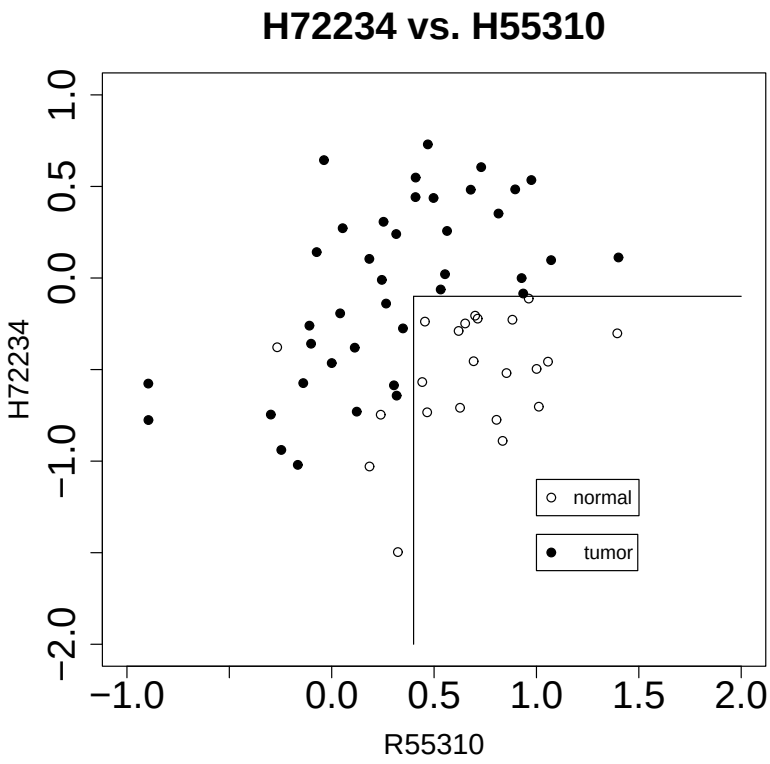

Figure 3.6: An interaction pattern from the colon data

space like in CART, our method defines a wide collection of leaves with non-empty intersection, thus allowing more robust classification.

Another advantage of our classification method is its interpretability in terms of interaction structures. This is a very important issue for applied scientists, especially those working on gene expression data. Indeed, although it is almost certain that genes somehow interact, the challenging question of modeling these interactions remains partly unanswered. The proposed method can detect quite successfully interaction patterns in simulated 'perfect' data.

The proposed approach differs significantly from Dong and Li's approach in several aspects. First, we use a statistical criterion to define the patterns instead of the heuristic growth rate. Second, while Dong and Li find patterns of high order, we argue that short pattern involving only relevant variables are preferable, in order to avoid overfitting of the learning data. Therefore condition (3.2) was added in the definition. Third, the method to detect the patterns is completely different: while Dong and Li perform a dramatic variable selection and enumerate all the possible patterns built with the selected variables, we use a CART-based algorithm which accelerates the search considerably and do not necessitate such a strong variable selection. The approach described in Boulesteix et al. (2003) may be seen as a simplification of the method for binary responses. The search algorithm is similar, but the testing of condition (3.2) is replaced by a pruning step while building the trees. Thus, only the variables involved in the subsequent splittings can be eliminated from a pattern. This approach is often appropriate for binary responses, since the successive splittings of the trees are chosen to minimize the deviance. However, it is too restrictive for multicategorical responses or for highly correlated predictors. The proposed definition and search algorithm overcome this inconvenience and generalize the framework developed in Boulesteix et al. (2003).

Chapter 4

Linear dimension reduction for classification

4.1 Introduction

Variable selection is very popular in the field of microarray data analysis, since conceptually simple. However, it presents two major drawbacks. First, a large part of the information contained in the data set gets lost, since most genes are eliminated by the procedure. Second, interactions and correlations between variables are almost always ignored. A few sophisticated procedures try to overcome this problem by selecting optimal subsets with respect to a given criterion instead of filtering out the apparently uninteresting variables. However, these methods generally suffer from overfitting: the obtained variable subsets might be optimal for the learning data set, but do not perform well on independent test data. Moreover, they are based on computationally intensive iterative algorithms and thus very difficult to implement and interpret.

Dimension reduction is a wise alternative to variable selection to handle high-dimensional data. In the literature, it is also denoted as 'feature extraction' or 'projection onto a low-dimensional subspace'. Dimension reduction methods present several advantages over

variable selection:

- They may allow data visualization in a low-dimensional space.
- They incorporate interactions and correlations between variables.
- Although information on thousands of genes are used, statistical methods which can handle only few variables may be employed. In contrast, if one wants to apply e.g. logistic regression to microarray data without dimension reduction, a drastic variable selection is required, which results in a loss of information.
- In the ideal case, the new components may be interpreted by applied scientists.

Although dimension reduction can serve different purposes, e.g. clustering, regression, classification, we will focus on dimension reduction methods for classification. They can be categorized into:

- Linear and non-linear methods. Linear methods are usually faster, more robust and more interpretable than non-linear methods. In turn, non-linear methods can sometimes discover complicated structures (e.g. embedments) that linear methods fail to detect.
- Supervised and unsupervised methods. Supervised methods use the class information Y for constructing the new components, contrary to unsupervised methods. It is well-known and quite intuitive that supervised methods are recommended when dealing with a supervised problem such as classification (Nguyen and Rocke, 2002a).

Since our interest is in dimension reduction for high-dimensional microarray data, we focus on methods which can handle the case $n < p$. Non-linear methods for dimension reduction (e.g. Isomap or Sammon's non-linear mapping) are computationally very intensive for high-dimensional data. Moreover, they are known to perform poorly when the number of observations is low, as in microarray data. Unsupervised dimension reduction methods are effective tools for graphical representation or to discover structures in data. However, they are generally inappropriate in the context of classification, because the obtained new components are not necessarily linked to the response variable Y. Thus, we restrict

ourselves to supervised linear methods.

In Principal component analysis (PCA), the goal is to find uncorrelated linear transformations of the random vector $\mathbf{x}$ which have high variance. The same analysis can be performed on $\mathrm{E}(\mathbf{x}|Y)$ instead of $\mathbf{x}$. In this chapter, this approach is denoted as between-group PCA and examined in Section 4.2. An alternative approach for linear dimension reduction is Partial Least Squares (PLS): the goal is to find linear transformations which have high covariance with the response Y. In Section 4.3, the PLS approach is briefly presented and a connection between between-group PCA and the first PLS component is shown for the case $K = 2$.

If one assumes that $\mathbf{x}$ has a multivariate normal distribution within each class and that the within-group covariance matrix is the same for all the classes, decision theory tells us that the optimal decision function is a linear transformation of $\mathbf{x}$. This approach is called linear discriminant analysis (Hastie et al., 2001). For $K = 2$, we show in Section 4.4 that under a stronger assumption, linear discriminant analysis is based on the same linear transformation of $\mathbf{x}$ as between-group PCA.

In Section 4.5, we give a brief overview of other linear dimension reduction methods.

4.2 Between-group PCA

4.2.1 Definition

Linear dimension reduction consists to define new random variables $Z_1, \ldots, Z_m$ as linear combinations of $X_1, \ldots, X_p$, where m is the number of new variables. For $j = 1, \ldots, m$, Z_j has the form

$$Z_j = \mathbf{a}_j^T \mathbf{x},$$

where $\mathbf{a}_j$ is a $p \times 1$ vector. In Principal Component Analysis (PCA), $\mathbf{a}_1, \ldots, \mathbf{a}_m \in \mathbb{R}^p$ are defined successively as follows.

Definition 4.1. . **Principal Components.**
$\mathbf{a}_1$ *is the* $p \times 1$ *vector maximizing* $VAR(\mathbf{a}^T\mathbf{x}) = \mathbf{a}^T\mathbf{\Sigma}\mathbf{a}$ *under the constraint* $\mathbf{a}_1^T\mathbf{a}_1 = 1$. *For* $j = 2, \ldots, m$, $\mathbf{a}_j$ *is the* $p \times 1$ *vector maximizing* $VAR(\mathbf{a}^T\mathbf{x})$ *under the constraints* $\mathbf{a}_j^T\mathbf{a}_j = 1$ *and* $\mathbf{a}_j^T\mathbf{a}_i = 0$ *for* $i = 1, \ldots, j-1$.

The vectors $\mathbf{a}_1, \ldots, \mathbf{a}_m$ defined in definition 4.1 are the (normalized) eigenvectors of the matrix $\mathbf{\Sigma}$. The number of eigenvectors with strictly positive eigenvalues equals $rank(\mathbf{\Sigma})$, which is $p-1$ if $X_1, \ldots, X_p$ are linearly independent. $\mathbf{a}_1$ is the eigenvector of $\mathbf{\Sigma}$ with the greatest eigenvalue, $\mathbf{a}_2$ is the eigenvector of $\mathbf{\Sigma}$ with the second greatest eigenvalue, and so on. For an extensive overview of PCA, see e.g. Jolliffe (1986).

In PCA, the new variables $Z_1, \ldots, Z_m$ are built independently of Y and the number of new variables m is at most $p-1$. If one wants to build new variables which contain information on the categorical response variable Y, an alternative to PCA is to look for linear combinations of $\mathbf{x}$ which maximize $\mathrm{VAR}(\mathrm{E}(\mathbf{a}^T\mathbf{x}|Y))$ instead of $\mathrm{VAR}(\mathbf{a}^T\mathbf{x})$. In the following, this approach is denoted as between-group PCA. $\mathbf{\Sigma}_B$ denotes the between-group covariance matrix:

$$\mathbf{\Sigma}_B = \mathrm{COV}(\mathrm{E}(\mathbf{x}|Y)). \tag{4.1}$$

In between-group PCA, $\mathbf{a}_1, \ldots, \mathbf{a}_m$ are defined as follows.
Definition 4.2. . **Between-group Principal Components**.
$\mathbf{a}_1$ *is the* $p \times 1$ *vector maximizing* $VAR(E(\mathbf{a}^T\mathbf{x}|Y)) = \mathbf{a}^T\mathbf{\Sigma}_B\mathbf{a}$ *under the constraint* $\mathbf{a}_1^T\mathbf{a}_1 = 1$. *For* $j = 2, \ldots, m$, $\mathbf{a}_j$ *is the* $p \times 1$ *vector maximizing* $VAR(\mathbf{a}^T\mathbf{x}|Y)$ *under the constraints* $\mathbf{a}_j^T\mathbf{a}_j = 1$ *and* $\mathbf{a}_j^T\mathbf{a}_i = 0$ *for* $i = 1, \ldots, j-1$.

The vectors $\mathbf{a}_1, \ldots, \mathbf{a}_m$ defined in definition 4.2 are the eigenvectors of the matrix $\mathbf{\Sigma}_B$. Since $\mathbf{\Sigma}_B$ is of rank at most $K-1$, there are at most $K-1$ eigenvectors with strictly positive eigenvalues. Since $\mathrm{E}(\mathbf{a}^T\mathbf{x}|Y) = \mathbf{a}^T\mathrm{E}(\mathbf{x}|Y)$, between-group PCA can be seen as PCA performed on the random vector $\mathrm{E}(\mathbf{x}|Y)$ instead of $\mathbf{x}$. In the next section, the special case $K = 2$ is examined.

4.2.2 A special case: $K = 2$

If $K = 2$, $\mathbf{\Sigma}_B$ has only one eigenvector with strictly positive eigenvalue. This eigenvector is denoted as $\mathbf{a}_B$. $\mathbf{a}_B$ can be derived from simple computations on $\mathbf{\Sigma}_B$.

$$\begin{aligned}
\mathbf{\Sigma}_B &= p_1(\boldsymbol{\mu}_1 - \boldsymbol{\mu})(\boldsymbol{\mu}_1 - \boldsymbol{\mu})^T + p_2(\boldsymbol{\mu}_2 - \boldsymbol{\mu})(\boldsymbol{\mu}_2 - \boldsymbol{\mu})^T \\
&= p_1(\boldsymbol{\mu}_1 - p_1\boldsymbol{\mu}_1 - p_2\boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - p_1\boldsymbol{\mu}_1 - p_2\boldsymbol{\mu}_2)^T \\
&\quad + p_2(\boldsymbol{\mu}_2 - p_1\boldsymbol{\mu}_1 - p_2\boldsymbol{\mu}_2)(\boldsymbol{\mu}_2 - p_1\boldsymbol{\mu}_1 - p_2\boldsymbol{\mu}_2)^T \\
&= p_1p_2^2(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T + p_2p_1^2(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T \\
&= p_1p_2(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T \\
\mathbf{\Sigma}_B(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) &= p_1p_2(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2).
\end{aligned}$$

Since

$$p_1p_2(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) > 0, \tag{4.2}$$

$(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$ is an eigenvector of $\mathbf{\Sigma}_B$ with strictly positive eigenvalue. Since $\mathbf{a}_B$ has to satisfy $\mathbf{a}_B^T\mathbf{a}_B = 1$, we obtain

$$\mathbf{a}_B = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)/||\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2||. \tag{4.3}$$

In practice, $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ are often unknown and must be estimated from the available data set $(\mathbf{x}_i, Y_i)_{i=1,\ldots,n}$. $\mathbf{a}_B$ may be estimated by replacing $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ by $\hat{\boldsymbol{\mu}}_1$ and $\hat{\boldsymbol{\mu}}_2$ in equation (4.3):

$$\hat{\mathbf{a}}_B = (\hat{\boldsymbol{\mu}}_1 - \hat{\boldsymbol{\mu}}_2)/||\hat{\boldsymbol{\mu}}_1 - \hat{\boldsymbol{\mu}}_2||. \tag{4.4}$$

Between-group PCA is applied by Culhane et al. (2002) in the context of high-dimensional microarray data. However, Culhane et al. (2002) formulate the method as a data matrix decomposition (singular value decomposition) and do not define the between-group principal components theoretically. In the following section, we examine the connection between-group PCA and Partial Least Squares.

4.3 A connection between PLS dimension reduction and between-group PCA

4.3.1 Introduction to PLS dimension reduction

Partial Least Squares (PLS) dimension reduction is another linear dimension reduction method. It is especially appropriate to construct new components which are linked to the response variable Y. Studies of the PLS approach from the point of view of statisticians can be found in e.g. Stone and Brooks (1990); Frank and Friedman (1993); Garthwaite (1994). In the PLS framework, $Z_1, \ldots, Z_m$ are not random variables which are theoretically defined and then estimated from a data set: their definition is based on a specific sample. Here, we focus on the binary case ($Y = 1, 2$), although the PLS approach can be generalized to multicategorical response variables (de Jong, 1993). For the data set $(\mathbf{x}_i, Y_i)_{i=1,\ldots,n}$, the vectors $\mathbf{a}_1, \ldots, \mathbf{a}_m$ are defined as follows (Stone and Brooks, 1990).

Definition 4.3. . **PLS components**

Let $\hat{COV}$ denote the sample covariance computed from $(\mathbf{x}_i, Y_i)_{i=1,\ldots,n}$. $\mathbf{a}_1$ is the $p\times 1$ vector maximizing $\hat{COV}(\mathbf{a}_1^T\mathbf{x}, Y)$ under the constraint $\mathbf{a}_1^T\mathbf{a}_1 = 1$. For $j = 2, \ldots, m$, $\mathbf{a}_j$ is the $p\times 1$ vector maximizing $\hat{COV}(\mathbf{a}^T\mathbf{x}, Y)$ under the constraints $\mathbf{a}_j^T\mathbf{a}_j = 1$ and $\hat{COV}(\mathbf{a}_j^T\mathbf{x}, \mathbf{a}_i^T\mathbf{x}) = 0$ for $i = 1, \ldots, j-1$.

In the following, the vector $\mathbf{a}_1$ defined in definition 4.3 is denoted as $\mathbf{a}_{PLS}$. An exact algorithm to compute the PLS components can be found in Martens and Naes (1989). In Section 4.3.2, we study the connection between the first PLS component and the first between-group principal component.

4.3.2 A property

Proposition 4.1. .

For a given data set $(\mathbf{x}_i, Y_i)_{i=1,\ldots,n}$, the first PLS component equals the first between-group principal component:

$$\mathbf{a}_{PLS} = \hat{\mathbf{a}}_B.$$

Proof. For all $\mathbf{a} \in \mathbb{R}^p$,

$$\begin{aligned}
\hat{\mathrm{COV}}(\mathbf{a}^T\mathbf{x}, Y) &= \mathbf{a}^T \hat{\mathrm{COV}}(\mathbf{x}, Y) \\
\hat{\mathrm{COV}}(\mathbf{x}, Y) &= \tfrac{1}{n}\textstyle\sum_{i=1}^n \mathbf{x}_i Y_i - \tfrac{1}{n^2}(\sum_{i=1}^n \mathbf{x}_i)(\sum_{i=1}^n Y_i) \\
&= \tfrac{1}{n}(n_1\hat{\boldsymbol{\mu}}_1 + 2n_2\hat{\boldsymbol{\mu}}_2) - \tfrac{1}{n^2}(n_1\hat{\boldsymbol{\mu}}_1 + n_2\hat{\boldsymbol{\mu}}_2)(n_1 + 2n_2) \\
&= \tfrac{1}{n^2}(nn_1\hat{\boldsymbol{\mu}}_1 + 2nn_2\hat{\boldsymbol{\mu}}_2 - n_1^2\hat{\boldsymbol{\mu}}_1 - 2n_1n_2\hat{\boldsymbol{\mu}}_1 - n_1n_2\hat{\boldsymbol{\mu}}_2 - 2n_2^2\hat{\boldsymbol{\mu}}_2) \\
&= n_1n_2(\hat{\boldsymbol{\mu}}_2 - \hat{\boldsymbol{\mu}}_1)/n^2
\end{aligned}$$

The only unit vector maximizing $n_1n_2\mathbf{a}^T(\hat{\boldsymbol{\mu}}_2 - \hat{\boldsymbol{\mu}}_1)/n^2$ is

$$\begin{aligned}
\mathbf{a}_{PLS} &= (\hat{\boldsymbol{\mu}}_2 - \hat{\boldsymbol{\mu}}_1)/||\hat{\boldsymbol{\mu}}_2 - \hat{\boldsymbol{\mu}}_1|| \\
&= \hat{\mathbf{a}}_B
\end{aligned}$$

□

Thus, the first component obtained by PLS dimension reduction is the same as the first component obtained by between-group PCA. This is an argument to support the (controversial) use of PLS dimension reduction in the context of binary classification. The connection between between-group PCA and linear discriminant analysis is examined in the next section.

4.4 A connection between LDA and between-group PCA

4.4.1 Linear discriminant analysis

In this section, linear discriminant analysis is briefly introduced. The connection to between-group PCA is examined in section 4.4.2.

If $\mathbf{x}$ is assumed to have a multivariate normal distribution with mean $\boldsymbol{\mu}_k$ and covariance matrix $\boldsymbol{\Sigma}_k$ within class k,

$$\begin{aligned}
P(Y = k|\mathbf{x}) &= p_k \cdot f(\mathbf{x}|Y = k)/f(\mathbf{x}) \\
\ln P(Y = k|\mathbf{x}) &= \ln p_k - \ln f(\mathbf{x}) - \ln(\sqrt{2\pi}|\boldsymbol{\Sigma}_k|^{1/2}) - \tfrac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k^{-1/2}(\mathbf{x} - \boldsymbol{\mu}_k),
\end{aligned}$$

where f represents the density function. The Bayes classification rule predicts the class of an observation $\mathbf{x}_0$ as

$$\begin{aligned} C(\mathbf{x}_0) &= \arg\max_k P(Y=k|\mathbf{x}) \\ &= \arg\max_k (\ln p_k - \ln(\sqrt{2\pi}|\mathbf{\Sigma}_k|^{1/2}) - \tfrac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^T\mathbf{\Sigma}_k^{-1/2}(\mathbf{x}-\boldsymbol{\mu}_k)). \end{aligned}$$

For $K=2$, the discriminant function d_{12} is

$$\begin{aligned} d_{12}(\mathbf{x}) &= \ln P(Y=1|\mathbf{x}) - \ln P(Y=2|\mathbf{x}) \\ &= -\tfrac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_1)^T\mathbf{\Sigma}_1^{-1/2}(\mathbf{x}-\boldsymbol{\mu}_1) + \tfrac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_2)^T\mathbf{\Sigma}_2^{-1/2}(\mathbf{x}-\boldsymbol{\mu}_2) \\ &\quad + \ln p_1 - \ln p_2 - \ln(\sqrt{2\pi}|\mathbf{\Sigma}_1|^{1/2}) + \ln(\sqrt{2\pi}|\mathbf{\Sigma}_2|^{1/2}) \end{aligned}$$

If one assumes $\mathbf{\Sigma}_1 = \mathbf{\Sigma}_2 = \mathbf{\Sigma}$, d_{12} is a linear function of $\mathbf{x}$ (hence the term linear discriminant analysis):

$$\begin{aligned} d_{12}(\mathbf{x}) &= (\mathbf{x} - \tfrac{\boldsymbol{\mu}_1+\boldsymbol{\mu}_2}{2})^T\mathbf{\Sigma}^{-1/2}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2) + \ln p_1 - \ln p_2 \\ &= \mathbf{a}_{LDA}^T\mathbf{x} + b, \end{aligned}$$

where

$$\mathbf{a}_{LDA} = \mathbf{\Sigma}^{-1/2}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) \tag{4.5}$$

and

$$b = -\frac{1}{2}(\boldsymbol{\mu}_1+\boldsymbol{\mu}_2)^T\mathbf{\Sigma}^{-1/2}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2) + \ln p_1 - \ln p_2. \tag{4.6}$$

4.4.2 A property

Proposition 4.2. .
If $\mathbf{\Sigma}$ is assumed to be of the form $\mathbf{\Sigma} = \sigma^2\mathbf{I}_p$, where $\mathbf{I}_p$ is the identity matrix of dimensions $p \times p$ and σ is a scalar, $\mathbf{a}_{LDA}$ and $\mathbf{a}_B$ are collinear.

Proof. The proof follows from equations (4.3) and (4.5). □

Thus, we showed the strong connection between linear discriminant analysis and between-group PCA in the case $K=2$. In practice, $\mathbf{a}_B$ is estimated by $\hat{\mathbf{a}}_B$ and $\mathbf{a}_{LDA}$ is estimated

by $\hat{\mathbf{a}}_{LDA} = (\hat{\boldsymbol{\mu}}_1 - \hat{\boldsymbol{\mu}}_2)/\hat{\sigma}$, where $\hat{\sigma}$ is an estimator of σ. Thus, $\hat{a}_B$ and $\hat{a}_{LDA}$ are also collinear.

The assumption about the structure of $\boldsymbol{\Sigma}$ is quite strong. However, such an assumption can be wise in practice when the available data set contains a large number of variables p and a small number of observations n. If $p > n$, which often occurs in practice (for instance in microarray data analysis), $\hat{\boldsymbol{\Sigma}}$ can not be inverted, since it has rank at most $n-1$ and dimensions $p \times p$. In this case, it is sensible to make strong assumptions on $\boldsymbol{\Sigma}$. Proposition 4.2 tells us that between-group PCA takes only between-group correlations into account, not within-group correlations.

4.5 Overview of other methods

Many other methods for dimension reduction in the context of classification have been proposed in the statistical literature. Here, we give a short overview of these approaches and discuss their utility for microarray data analysis.

The most common projection method is probably the so-called discriminant coordinates approach (Gnanadesikan, 1977), which can be seen as a generalization of Fisher's linear discriminant for more than two classes (Rao, 1952). An adaptation of this method for data with asymmetric classes (i.e. classes with strongly different within-group covariance matrices) can be found in Hennig (2004). The idea behind discriminant coordinates is to find linear transformations of $\mathbf{x}$ "such that the between-group variance is maximized relative to the within-group variance" (Hastie et al., 2001). For a given data set $(\mathbf{x}_i, Y_i)_{i=1,\ldots,n}$, the vectors $\mathbf{a}_1, \ldots, \mathbf{a}_m$ are defined as follows.

Definition 4.4. . **Discriminant coordinates**
$\mathbf{a}_1$ *is the* $p \times 1$ *vector maximizing* $\mathbf{a}^T\mathbf{B}\mathbf{a}/\mathbf{a}^T\mathbf{W}\mathbf{a}$ *under the constraint* $\mathbf{a}_1^T\mathbf{W}\mathbf{a}_1 = 1$. *For* $j = 2, \ldots, m$, $\mathbf{a}_j$ *is the* $p \times 1$ *vector maximizing* $\mathbf{a}^T\mathbf{B}\mathbf{a}/\mathbf{a}^T\mathbf{W}\mathbf{a}$ *under the constraints* $\mathbf{a}_j^T\mathbf{W}\mathbf{a}_j = 1$ *and* $\mathbf{a}_j^T\mathbf{W}\mathbf{a}_i = 0$ *for* $i = 1, \ldots, j-1$.

It is easy to show that the vectors $\mathbf{a}_1, \ldots, \mathbf{a}_m$ are the normalized eigenvectors of the matrix

$\mathbf{W}^{-1}\mathbf{B}$ corresponding to the maximal eigenvalues. The number of such eigenvectors with non-zero eigenvalues equals the rank of the matrix $\mathbf{W}^{-1}\mathbf{B}$, which is at most $K-1$. This method requires the inversion of the sample within-group covariance matrix, which is not possible if $n < p$. That's why it is not applicable to microarray data.

In the case $K = 2$, another approach consists to look for linear transformations of $\mathbf{x}$ which maximize the so-called Bhattacharyya distance D (Fukunaga, 1990) measuring the dissimilarity between the two classes. D is defined by Fukunaga (1990) as

$$D(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2) = \frac{1}{8}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T (\frac{\boldsymbol{\Sigma}_2 + \boldsymbol{\Sigma}_1}{2})^{-1} (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1) + \frac{1}{2} \log \frac{det(\frac{\boldsymbol{\Sigma}_2 + \boldsymbol{\Sigma}_1}{2})}{\sqrt{det(\boldsymbol{\Sigma}_1) det(\boldsymbol{\Sigma}_2)}}. \quad (4.7)$$

In Hennig (2004), the so-called mean-dominated Bhattacharyya coordinates are defined as follows for a given data set $(\mathbf{x}_i, Y_i)_{i=1,\ldots,n}$.

Definition 4.5. Mean-dominated Bhattacharyya coordinates
Let $\frac{1}{2}(\mathbf{S}_1 + \mathbf{S}_2)$ *be denoted as* $\mathbf{W}_D$. $\mathbf{a}_1$ *is the* $p \times 1$ *vector maximizing*

$$D(\mathbf{a}^T \hat{\boldsymbol{\mu}}_1, \mathbf{a}^T \hat{\boldsymbol{\mu}}_2, \mathbf{a}^T \mathbf{W}_D \mathbf{a}, \mathbf{a}^T \mathbf{W}_D \mathbf{a})$$

under the constraint $\mathbf{a}_1^T \mathbf{W}_D \mathbf{a}_1 = 1$. *For* $j = 2, \ldots, m$, $\mathbf{a}_j$ *is the* $p \times 1$ *vector maximizing* $D(\mathbf{a}^T \hat{\boldsymbol{\mu}}_1, \mathbf{a}^T \hat{\boldsymbol{\mu}}_2, \mathbf{a}^T \mathbf{W}_D \mathbf{a}, \mathbf{a}^T \mathbf{W}_D \mathbf{a})$ *under the constraints* $\mathbf{a}_j^T \mathbf{W}_D \mathbf{a}_j = 1$ *and* $\mathbf{a}_j^T \mathbf{S}_1 \mathbf{a}_i = 0$ *for* $i = 1, \ldots, j-1$.

In practice, the maximization of $D(\mathbf{a}^T \hat{\boldsymbol{\mu}}_1, \mathbf{a}^T \hat{\boldsymbol{\mu}}_2, \mathbf{a}^T \mathbf{W}_D \mathbf{a}, \mathbf{a}^T \mathbf{W}_D \mathbf{a})$ requires the inversion of $\mathbf{S}_1 + \mathbf{S}_2$, which is impossible if $n < p$.

Young et al. (1987) propose another approach denoted as mean/variance difference coordinates. This method is applicable for multicategorical responses and captures the difference between the within-group covariance matrices. Hennig (2004) interprets this method as the maximization of the "sum of the projected squared between-group differences in mean and variance". In practice, the procedure can be applied even if $n < p$. However, it requires the eigendecomposition of a $p \times (p+1) \cdot (K-1)$ matrix. Thus, it is not recommended when p is very large.

Sufficient dimension reduction represents a large family of dimension reduction methods for regression. Some of these methods can be applied in the context of binary classification.

An overview of sufficient dimension reduction for binary classification can be found in Cook and Lee (1999). The general idea of sufficient dimension reduction is to find a $p \times m$ matrix $\mathbf{A}$ (with $m \leq p$) such that the distribution of $Y|\mathbf{x}$ is the same as the distribution of $Y|\mathbf{A}^T\mathbf{x}$ for all $\mathbf{x}$. This condition can also be formulated as

$$Y \perp\!\!\!\perp \mathbf{x}|\mathbf{A}^T\mathbf{x}, \tag{4.8}$$

where $\perp\!\!\!\perp$ denotes stochastic independence.

It implies that the random vector $\mathbf{x}$ of length p can be replaced by the random vector $\mathbf{A}^T\mathbf{x}$ of length m without loss of information on Y (Cook and Lee, 1999). Such a matrix $\mathbf{A}$ always exists, since if one sets $m = p$, the identity matrix $\mathbf{I}_p$ satisfies (4.8). This matrix is not unique, since any matrix whose columns form a base of the same subspace as the subspace spanned by $\mathbf{A}$'s columns satisfies (4.8) as well. m must be as small as possible. $\mathbf{A}$ and m can be estimated in several ways. The most common methods to estimate $\mathbf{A}$ are sliced inverse regression (SIR) proposed by Li (1991), slice average variance estimation (SAVE) proposed by Cook and Weisberg (1991) and principal Hessian directions (PHD) proposed by Li (1992). In practice, these methods require the inversion of the sample covariance matrix, which is not possible if $n < p$.

4.6 Discussion

We showed the strong connection between PLS dimension reduction for classification, between-group PCA and linear discriminant analysis for the case $K = 2$. PCA and PLS are useful techniques in practice, especially when the number of observations n is smaller than the number of variables p, for instance in the context of microarray data analysis (Nguyen and Rocke, 2002a). The connection between PLS and between-group PCA can also support the use of PLS dimension reduction in the classification framework. The conclusion of this theoretical study is that PLS and between-group PCA, which are the two main dimension reduction methods allowing $n < p$ are tightly connected to a special case of linear discriminant analysis with strong assumptions. In future work, one

could examine the connection between the three approaches for multicategorical response variables. The connection between linear dimension reduction methods and alternative methods such as shrinkage methods could also be investigated in future.

Chapter 5

A study of PLS dimension reduction

5.1 Introduction

The output of n microarray experiments can be summarized as a $n \times p$ data matrix, where p is the number of analyzed genes. p is always much larger than the number of experiments n. An important application of microarray technology is tumor diagnosis, i.e. class prediction. High-dimensionality makes the application of most classification methods difficult, if not impossible. To overcome this problem, one can either extract a small subset of interesting variables (gene selection) or construct m new components which summarize the original data as well as possible, with $m < p$ (dimension reduction).

Gene selection has been studied extensively in the last few years. The most commonly used gene selection procedures are based on a score which is calculated for all genes individually. Then the genes with the best scores are selected. These methods are often denoted as univariate gene selection. Several selection criteria have been used in the literature, e.g. the t statistic (Hedenfalk et al., 2001), Wilcoxon's rank sum statistic (Dettling and Bühlmann, 2003) or Ben Dor's combinatoric 'TNoM' score (Ben-Dor et al.,

2000). When using a test statistic as criterion, it is useful to adjust the p-values with a multiple testing procedure (Dudoit et al., 2003). The main advantages of gene selection are its simplicity and interpretability. Gene selection procedures output a list of relevant genes which can be experimentally analyzed by biologists. Moreover, univariate gene selection is generally quite fast.

The scores mentioned in the previous paragraph are all based on the association of individual genes with the classes. Interactions and correlations between genes are omitted, although they are of great interest in system biology. For illustration, let us consider three genes A, B and C. A relevance score like the t statistic might tell us: gene A is more relevant than gene B and gene B is more relevant than gene C for classification. Now suppose we want to select two of these three genes to perform classification. The t statistic does not tell us if it is better to select A and B, A and C or B and C. A few sophisticated procedures intend to overcome this problem by selecting optimal subsets with respect to a given criterion instead of ranking the genes. Bo and Jonassen (2002) look for relevant pairs of genes, whereas Li et al. (2001) want to find optimal gene subsets via genetic algorithms. However, these methods generally suffer from overfitting: the obtained gene subsets might be optimal for the training data, but they do not perform as well on independent test data. Moreover, they are based on computationally intensive iterative algorithms and thus very difficult to interpret and implement.

Dimension reduction is a wise alternative to variable selection in order to overcome this dimensionality problem. It is also denoted as feature extraction. Unlike gene selection, such methods use all the genes included in the data set. The whole data are projected onto a low-dimensional space, thus allowing a graphical representation. The new components often give information or hints about the data's intrinsic structure, although there is no standard concept and procedure to do this. Dimension reduction is sometimes criticized for its lack of interpretability, especially for applied scientists who often need more concrete answers about individual genes. In this chapter, we show that PLS dimension reduction is tightly connected to gene selection.

Dimension reduction methods for classification can be categorized into linear and nonlin-

ear, supervised and unsupervised methods. Intuitively, supervised methods, i.e. methods which use the class information of the observations to construct new components, should be preferred to unsupervised methods, which work only 'by chance' in 'good' data sets (Nguyen and Rocke, 2002a). Since nonlinear methods are generally computationally intensive and lack robustness, they are not recommended for microarray data analysis. To our knowledge, the only well-established supervised linear dimension reduction method working even if $n < p$ is the Partial Least Squares method (PLS). PLS is a linear method in the sense that the new components are linear combinations of the original variables. However, the coefficients defining the new components are not linear. Another approach denoted as between-group analysis has been proposed by Culhane et al. (2002), but it turns out that it is strongly related to PLS. Principal component analysis (Ghosh, 2002; Kahn et al., 2001) is an unsupervised method: its goal is to find uncorrelated linear transformations of the original variables which have high variance. As an unsupervised method, it is inappropriate for classification. Sufficient dimension reduction for classification is reviewed in Cook and Lee (1999) and applied to microarray data in Chiaromonte and Martinelli (2001). It is a supervised approach: it looks for components which summarize the predictor variables such that the class and the predictor variables are independent given the new components. This method can not be applied if $p > n$. A few other dimension reduction methods for classification are reviewed in Hennig (2004). Some of them, such as discriminant coordinates or the Bhattacharyya distance approach can not be applied if $p > n$. The mean/variance difference coordinates approach is introduced in Young et al. (1987) and discussed in Hennig (2004). It can theoretically be applied if $p > n$, but it requires the eigendecomposition of a $p \times p$ empirical covariance matrix, which is not recommended when $p >> n$. To our knowledge, PLS is the only fast supervised dimension reduction method which can handle a huge number of predictor variables.

It is known that PLS dimension reduction can be used for classification problems in the context of microarray data analysis (Nguyen and Rocke, 2002a; Huang and Pan, 2003). However, these papers do not include any extensive comparative study of classification methods. Moreover, they treat the PLS technique as a 'black box' which is only meant

to improve classification accuracy, without concern for the components themselves. In this chapter, two aspects of PLS dimension reduction are examined. First, the classification performance is compared with the classification performance of top-ranking methods which have already been studied in the literature. Second, the connection between PLS dimension reduction and gene selection is examined.

In recent years, aggregation methods such as bagging (Breiman, 1996) and boosting (Freund, 1995) have been extensively analyzed. They lead to spectacular improvements of prediction accuracy when they are applied to classification problems. In microarray data analysis, accuracy improvement is also observed (Dettling and Bühlmann, 2003; Dudoit et al., 2002). So far, aggregating methods have been applied with weak and unstable classifiers such as stumps or classification trees. To our knowledge, boosting has never been used with dimension reduction techniques. In this chapter, we apply a classical boosting algorithm (AdaBoost) in the framework of PLS dimension reduction.

The chapter is organized as follows. PLS dimension reduction and boosting are introduced in section 2. In Section 3, the data are introduced and a few examples of data visualization using PLS dimension reduction are given. Classification results using PLS, PLS with boosting and various other methods are presented in section 4. In section 5, the connection between PLS and gene selection is studied and an interesting property of the first PLS component is proved in the case of binary responses.

In the following, $X_1, \ldots, X_p$ denote the continuous predictors (genes) and $\mathbf{x} = (X_1, \ldots, X_p)^T$ the corresponding random vector. $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})^T$ for $i = 1, \ldots, n$ denote independent identically distributed realizations of the random vector $\mathbf{x}$. Each row of the $n \times p$ data matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$ contains a realization of $\mathbf{x}$.

5.2 Dimension reduction and classification with PLS

5.2.1 Outline of the method

Suppose we have a learning set $\mathcal{L}$ consisting of observations whose class is known and a test set $\mathcal{T}$ consisting of observations whose class has to be predicted. The data matrices corresponding to $\mathcal{L}$ respectively $\mathcal{T}$ are denoted as $\mathbf{X}_L$ respectively $\mathbf{X}_T$. The vector containing the classes of the observations from $\mathcal{L}$ is denoted as $\mathbf{Y}_L$. A classification method can be formalized as a function δ of $\mathbf{X}_L$, $\mathbf{Y}_L$ and the vector of predictors $\mathbf{x}_{new,i}$ corresponding to the ith observation from the test set:

$$\begin{aligned} \delta(., \mathbf{X}_L, \mathbf{Y}_L): \quad \mathbb{R}^p \quad &\rightarrow \quad \{1, \ldots, K\} \\ \mathbf{x}_{new,i} \quad &\rightarrow \quad \delta(\mathbf{x}_{new,i}, \mathbf{X}_L, \mathbf{Y}_L). \end{aligned}$$

In this section, we describe briefly the function δ which is discussed in this chapter. From now on, it is denoted as δ_{PLS}. δ_{PLS} consists of two steps.

The first step is dimension reduction. The idea is to look for m appropriate linear transformations $Z_1, \ldots, Z_m$ of the vector of predictors $\mathbf{x}$, where m has to be chosen by the user (this topic is discussed in Section 5.2.3). In the whole chapter, $\mathbf{a}_1, \ldots, \mathbf{a}_m$ denote the $p \times 1$ vectors which are used to construct the linear transformations $Z_1, \ldots, Z_m$:

$$\begin{aligned} Z_1 &= \mathbf{a}_1^T \mathbf{x}, \\ \ldots &= \ldots, \\ Z_m &= \mathbf{a}_m^T \mathbf{x}. \end{aligned}$$

In this chapter, the vectors $\mathbf{a}_1, \ldots, \mathbf{a}_m$ are determined using the SIMPLS algorithm (de Jong, 1993), which is one of the variants of PLS dimension reduction. The SIMPLS algorithm is introduced in Section 5.2.2. The linear transformations $Z_1, \ldots, Z_m$ are denoted as new components, for consistency with the PLS literature.

The second step is linear discriminant analysis using the new components $Z_1, \ldots, Z_m$ as predictor variables. Linear discriminant analysis is described in Section 5.4. One could use another classification method such as logistic regression. However, logistic regression is known to give worse results for some specific data configurations. For example,

logistic regression does not perform well when the different classes are completely or quasi-completely separated by the predictor variables, as claimed by Nguyen and Rocke (2002a). Since this configuration is quite common in microarray data, logistic regression is not a good choice. Linear discriminant analysis, which is not recommended when the number of predictor variables is large (see Section 5.4), performs well when applied to a small number of approximately normally distributed PLS components.

The procedure to predict the class of the observations from $\mathcal{T}$ using $\mathcal{L}$ can be summarized as follows.

1. Determine the vectors $\mathbf{a}_1, \ldots, \mathbf{a}_m$ using the SIMPLS algorithm (see Section 5.2.2) on the learning set $\mathcal{L}$. If $\mathbf{A}$ denotes the $p \times m$ matrix containing the vectors $\mathbf{a}_1, \ldots, \mathbf{a}_m$ in its columns, the matrix $\mathbf{Z}_L$ of new components for the learning set is obtained as

$$\mathbf{Z}_L = \mathbf{X}_L \mathbf{A}. \tag{5.1}$$

2. Compute the matrix $\mathbf{Z}_T$ of new components for the test data set as

$$\mathbf{Z}_T = \mathbf{X}_T \mathbf{A}. \tag{5.2}$$

3. Predict the class of the observations from $\mathcal{T}$ by linear discriminant analysis, using $Z_1, \ldots, Z_m$ as predictor variables. The classifier is built using only $\mathbf{Z}_L$.

This two-step approach is applied to microarray data by Nguyen and Rocke (2002a). In this chapter, we use the SIMPLS algorithm by de Jong (1993), which can be seen as a generalization for multicategorical response variables of the algorithm used by Nguyen and Rocke (2002a). The SIMPLS algorithm is presented in the next section.

5.2.2 The SIMPLS algorithm

Partial Least Squares (PLS) is a wide family of methods which are originally developed as a multivariate regression tool in the context of chemometrics (Martens and Naes, 1989). Later on, PLS regression has been studied by statisticians (Stone and Brooks, 1990; Garthwaite, 1994; Frank and Friedman, 1993). An overview of the history of PLS regression

is given in Martens (2001). PLS regression is especially appropriate to predict a univariate or multivariate continuous response using a large number of continuous predictors. The underlying idea of PLS regression is to find uncorrelated linear transformations of the original predictor variables which have high covariance with the response variables. These linear transformations can then be used as predictors in classical linear regression models to predict the response variables. Since the p original variables are summarized into a small number of relevant new components, linear regression can be performed even if the number of original variables p is much larger than the number of available observations. The different PLS algorithms differ in the definition of the linear transformations. Here, the focus is on the SIMPLS algorithm, because it can handle efficiently both univariate and multivariate variables in the same framework.

If Y is a binary response, it can be treated as a continuous response variable, since PLS regression does not require any distributional assumption. However, if Y is a multicategorical variable, it can not be treated as a continuous response variable. The problem can be circumvented by dummy-coding. The multicategorical random variable Y is transformed into a K-dimensional random vector $\mathbf{y} \in \{0,1\}^K$ as follows:

$$\begin{aligned} y_{i1} &= 1 \quad \text{if} \;\; Y_i = k, \\ y_{ik} &= 0 \quad \text{else}, \end{aligned}$$

where $\mathbf{y}_i = (y_{i1}, \ldots, y_{iK})^T$ denotes the ith realization of $\mathbf{y}$. In the following, $\mathbf{y}$ denotes the (one-dimensional) random variable Y if Y is binary ($K = 2$) or the K-dimensional random vector as defined above if Y is multicategorical ($K > 2$).

The SIMPLS algorithm proposed by de Jong (1993) computes the vectors $\mathbf{a}_1, \ldots, \mathbf{a}_m$ defined as follows.

Definition 5.1. *Let $\hat{COV}$ denote the empirical covariance computed from the available data set.* $\mathbf{a}_1$ *and* $\mathbf{b}_1$ *are the unit vectors maximizing* $\hat{COV}(\mathbf{a}_1^T\mathbf{x}, \mathbf{b}_1^T\mathbf{y})$. *For all* $j = 2, \ldots, m$, $\mathbf{a}_j$ *and* $\mathbf{b}_j$ *are the unit vectors maximizing* $\hat{COV}(\mathbf{a}_j^T\mathbf{x}, \mathbf{b}_j^T\mathbf{y})$ *subject to the constraint* $\hat{COV}(\mathbf{a}_j^T\mathbf{x}, \mathbf{a}_i^T\mathbf{x}) = 0$ *for all* $i = 1, \ldots, j-1$.

In words, the SIMPLS algorithm computes linear transformations of $\mathbf{x}$ and linear trans-

formations of $\mathbf{y}$ which have maximal covariance, under the constraint that the linear transformations of $\mathbf{x}$ are mutually uncorrelated. This formulation might seem familiar to those working with canonical correlation analysis. However, in contrast to canonical correlation analysis, the PLS approach defined above is based on the empirical covariances, not on the correlations. In PLS regression, a multivariate regression model is then built using $\mathbf{y}$ as multivariate response variable and $\mathbf{a}_1^T\mathbf{x}, \ldots, \mathbf{a}_m^T\mathbf{x}$ as predictors, hence the name PLS regression. The regression coefficients for each response variable and each predictor variable are also output by the SIMPLS algorithm. However, they are not used here, since we use the SIMPLS algorithm for dimension reduction only: our focus is on the new components $Z_1, \ldots, Z_m$, which are then used for linear discriminant analysis.

The predictor variables as well as the response variables have to be centered to have zero mean before running the SIMPLS algorithm. The `R` library `pls.pcr` includes an implementation of the SIMPLS algorithm, which is used in this chapter. To illustrate PLS dimension reduction, let us consider the following data matrix $\mathbf{X}$:

X_1	X_2	X_3	X_4	X_5
1	5	4	4	3
2	9	3	2	6
5	6	7	2	7
3	1	2	4	3

and the vector of classes

$$\mathbf{Y}^T = (1 \;\; 1 \;\; 2 \;\; 2).$$

$\mathbf{Y}$ and the columns of $\mathbf{X}$ are first centered to zero mean by substracting the empirical mean. The SIMPLS algorithm is then applied with e.g. $m = 2$. One obtains:

$$\begin{aligned} \mathbf{a}_1^T &= (1.77 \quad -4.86 \quad 0.53 \quad 0.76 \quad -0.82) \\ \mathbf{a}_2^T &= (2.31 \quad 3.01 \quad 3.02 \quad -1.79 \quad 3.45) \end{aligned}$$

The matrix of new components is obtained as

$$\mathbf{Z} = \mathbf{X}\mathbf{A},$$

where $\mathbf{A}$ is the 5×2 matrix containing $\mathbf{a}_1$ and $\mathbf{a}_2$ in its columns:

$$\begin{array}{rr} Z_1 & Z_2 \\ -0.20 & -0.46 \\ -0.71 & 0.13 \\ 0.33 & 0.76 \\ 0.58 & -0.43 \end{array}.$$

As can be seen from the matrix $\mathbf{Z}$, Z_1 seems to separate the two classes very well. Z_2, which is uncorrelated with Z_1, seems to be less relevant. It indicates that $m = 1$ might be a sensible choice in this trivial case. However, it is generally difficult to choose the right number m of PLS components to use for classification. In the following section, we address the problem of the choice of m.

5.2.3 Choosing the number of components

There is no widely accepted procedure to determine the right number of PLS components. Here, we propose to use a simple method based on cross-validation. Suppose we have a learning set $\mathcal{L}$ and a test set $\mathcal{T}$. Only the learning set $\mathcal{L}$ is used to choose m. The following procedure is repeated N_{run} times: the classifier δ_{PLS} is built using only $\alpha\%$ of the observations from $\mathcal{L}$ and applied to the remaining observations, with m taking successively different values. For each of the N_{run} runs, the error rate is computed using only the remaining observations from $\mathcal{L}$. After N_{run} runs, the mean error rate over the N_{run} runs is computed for each value of m. For a more precise description of the mean error rate, see Section 5.4.1. The value of m minimizing the mean error rate is then used to predict the class of the observations from $\mathcal{T}$. In the following, it is denoted as m_{opt}. In our analysis, we set α to 0.7 for consistency with Section 5.4 and $N_{run} = 50$, which seems to be a good compromise between computation time and estimation accuracy. It seems that m_{opt} does not depend highly on the parameters α and N_{run}.

When the procedure described above is used to choose the number of PLS components, the classification method consisting of PLS dimension reduction and linear discriminant

analysis does not involve any parameter. Since boosting is known to improve classification accuracy in many situations, we suggest to apply a boosting strategy to this classification method. Boosting is briefly introduced in the following section.

5.2.4 Boosting

Bagging and boosting consist of building a simple classifier using successively different bootstrap samples. In bagging, the bootstrap samples are based on the unweighted bootstrap and the predictions are made by majority voting. In boosting, the bootstrap samples are built iteratively using weights that depend on the predictions made in the last iteration. An early study focusing on statistical aspects of boosting is Schapire et al. (1998). A classifier based on a learning set $\mathcal{L}$ containing n_L observations is represented as in section 5.2.1 as a function of the p-dimensional vector of predictors $\mathbf{x}_{new,i}$:

$$\begin{aligned} \delta(., \mathbf{X}_L, \mathbf{Y}_L): \quad \mathbb{R}^p &\rightarrow \{1, \ldots, K\} \\ \mathbf{x}_{new,i} &\rightarrow \delta(\mathbf{x}_{new,i}, \mathbf{X}_L, \mathbf{Y}_L). \end{aligned}$$

In boosting, perturbed learning sets $\mathcal{L}_1, \ldots, \mathcal{L}_B$ are formed adaptively by drawing from the learning set $\mathcal{L}$ at random, where the probability of an observation to be selected in $\mathcal{L}_j$ depends on the prediction made by $\delta(., \mathbf{X}_{L_{b-1}}, \mathbf{Y}_{L_{b-1}})$. Observations which are incorrectly classified by $\delta(., \mathbf{X}_{L_{b-1}}, \mathbf{Y}_{L_{b-1}})$ have greater probability to be selected in $\mathcal{L}_b$.

The discrete AdaBoost procedure was proposed by Freund (1995). In the first iteration, the weights are initialized to $w_1 = \cdots = w_{n_L} = 1/n_L$. In the following we show the b-th step of the algorithm as described by Tutz and Hechenbichler (2004).

Discrete AdaBoost algorithm

1.
   - Based on the resampling probabilities $w_1, \ldots, w_{n_L}$, the learning set $\mathcal{L}_b$ is sampled from $\mathcal{L}$ with replacement.
   - The classifier $\delta(., \mathbf{X}_{L_b}, \mathbf{Y}_{L_b})$ is built.

2. The learning set $\mathcal{L}$ is run through the classifier $\delta(., \mathbf{X}_{L_b}, \mathbf{Y}_{L_b})$ yielding an error indicator $\epsilon_i = 1$ if the i-th observation is classified incorrectly and $\epsilon_i = 0$ otherwise.

3. With $e_b = \sum_{i=1}^{n_L} w_i \epsilon_i$, $b_b = (1 - e_b)/e_b$ and $c_b = \log(b_b)$ the resampling probabilities are updated for the next step by

$$w_{i,new} = \frac{w_i b_b^{\epsilon_i}}{\sum_{j=1}^{n_L} w_j b_b^{\epsilon_j}} = \frac{w_i \exp(c_b \epsilon_i)}{\sum_{j=1}^{n_L} w_j \exp(c_b \epsilon_j)}$$

After B iterations the aggregated voting for observation $\mathbf{x}_{new}$ is obtained by

$$\arg\max_j (\sum_{k=1}^{B} c_b I(\delta(x, \mathbf{X}_{L_b}, \mathbf{Y}_{L_b}) = j))$$

We propose to apply the AdaBoost algorithm with $\delta = \delta_{PLS}$ with different numbers of components. To our knowledge, boosting has never be used in the context of dimension reduction. In the whole study, we use 9 real microarray cancer data sets which are introduced in the following section.

5.3 Data

5.3.1 Data sets

Colon: The colon data set is a publicly available 'benchmark' gene expression data set which is extensively described in Alon et al. (1999). The data set contains the expression levels of 2000 genes for 62 patients from two classes. 22 patients are healthy patients and 40 patients have colon cancer.

Leukemia: This data set is introduced by Golub et al. (1999) and contains the expression levels of 7129 genes for 47 ALL-leukemia patients and 25 AML-leukemia patients. It is included in the `R` library `golubEsets`. After data preprocessing following the procedure described in Dudoit et al. (2002), only 3571 variables remain. It is easy to achieve excellent classification accuracy on this data set, even with quite trivial methods as described in the original paper by Golub et al. (1999).

Prostate: This data set gives the expression levels of 12600 genes for 50 normal tissues and 52 prostate cancer tissues. We threshold the data and filter genes as described in Singh et al. (2002). The filtering step leaves us with 5908 genes.

Breast cancer (ER+/ER-): This data set gives the expression levels of 7129 genes for 46 breast cancer patients from which 23 have status ER+ and 23 have status ER-. It is presented in West et al. (2002).

Carcinoma: This data set comprises the expression levels of 7463 genes for 18 normal tissues and 18 carcinomas. We standardize each array to have zero mean and unit variance. For an extensive description of the data set, see Notterman et al. (2001).

Lymphoma: The data set presented by Alizadeh et al. (2000) comprises the expression levels of 4026 genes for 62 patients from 3 different classes (B-CLL, FL and DLBCL). The missing values are inputed as described in Dudoit et al. (2002) using the function `pamr.inpute` from the `R` library `pamr` (Tibshirani et al., 2002).

SRBCT: This gene expression data set is presented in Kahn et al. (2001). It contains the expression levels of 2308 genes for 83 Small Round Blue Cells Tumor (SRBCT) patients belonging to one of the 4 tumor classes: Ewing family of tumors (EWS), non-Hodgkin lymphoma (BL), neuroblastoma (NB) and rhabdomyosarcoma (RMS).

Breast cancer (BRCA): This breast cancer data set contains the expression levels of 3227 genes for breast cancer patients with one of the three tumor types: sporadic, BRCA1 and BRCA2. It is described in Hedenfalk et al. (2001). The data are preprocessed as described in Simon et al. (2004).

NCI: This dataset comprises the expression levels of 5244 genes for 61 patients with 8 different tumor types: 7 breast, 5 central nervous system, 7 colon, 6 leukemia, 8 melanoma, 9 non-small-cell-lung-carcinoma, 6 ovarian, 9 renal Ross et al. (2000). The data are preprocessed as described in Dudoit et al. (2002).

In this next section, some of these data sets are visualized graphically using PLS dimension reduction.

5.3.2 Data Visualization via PLS dimension reduction

An advantage of PLS dimension reduction is the possibility to visualize the data by graphical representation. For instance, one can plot the second PLS component against the first PLS component using different colors for each class. As a visualization method, PLS might be useful for applied researchers who need simple graphical tools. In the following, we give a few concrete examples and show briefly and qualitatively that PLS dimension reduction can outline relevant cluster structures.

Suppose we have to analyze a data set with a binary response. One of the classes, e.g. class 2, consists of 2 subclasses: 2a and 2b. In the following, we try to interpret the PLS components in terms of clusters. For example, the first PLS component may discriminate between class 1 and class 2a and the second PLS component between class 1 and class 2b. In order to illustrate this point, we perform PLS dimension reduction on the whole prostate data set. We also cluster the observations from class 2 into two subclasses 2a and 2b using the k-means algorithm on the original variables $X_1, \ldots, X_p$. For the k-means clustering, we set the maximal number of iterations to 10. As can be seen from Figure 5.1, the first PLS component separates almost perfectly class 1 and class 2a, whereas the second PLS component separates almost perfectly class 1 and class 2b. Thus, the two PLS components can be interpreted in terms of clusters. A similar result can be obtained with the breast cancer data. We perform PLS dimension reduction on the whole breast cancer data set and cluster the observations from class 2 into 2a and 2b using the k-means algorithm on $X_1, \ldots, X_p$. The first and the second PLS components are represented as a scatterplot in Figure 5.2. We observe that the first PLS component can separate class 1 from class 2 perfectly. The second PLS component separates only 1 and 2a from 2b. Similar results are observed for the carcinoma and the leukemia data. Thus, for 4 of 5 data sets with binary class, the PLS components can be easily interpreted in terms of clusters.

However, in our examples, we do not know whether the subclasses 2a and 2b are biologically interpretable: they are only the output of the k-means clustering algorithm. Thus,

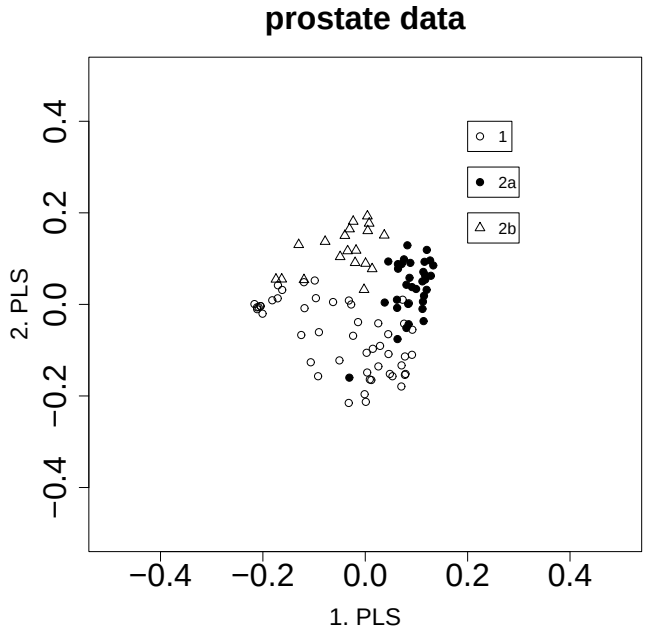

Figure 5.1: First and second PLS components for the prostate data

we also perform the same analysis on the lymphoma data set, for which three biologically interpretable classes are known. Patients with tumor type DLBCL are assigned to class 1, B-CLL to class 2a and FL to class 2b. PLS dimension reduction is performed as if the class were binary. As can be seen from Figure 5.3, the first PLS discriminates between class 1 and class 2, whereas the second PLS discriminates between class 2a and classes 1 and 2b.

As a conclusion, we recommend the PLS technique as a visualization tool, because it can outline relevant cluster structures. As can be seen from the figures presented in this section, the PLS components can be used to predict the class of new observations. The next section is dedicated to the classification method δ_{PLS} consisting of PLS dimension reduction and linear discriminant analysis.

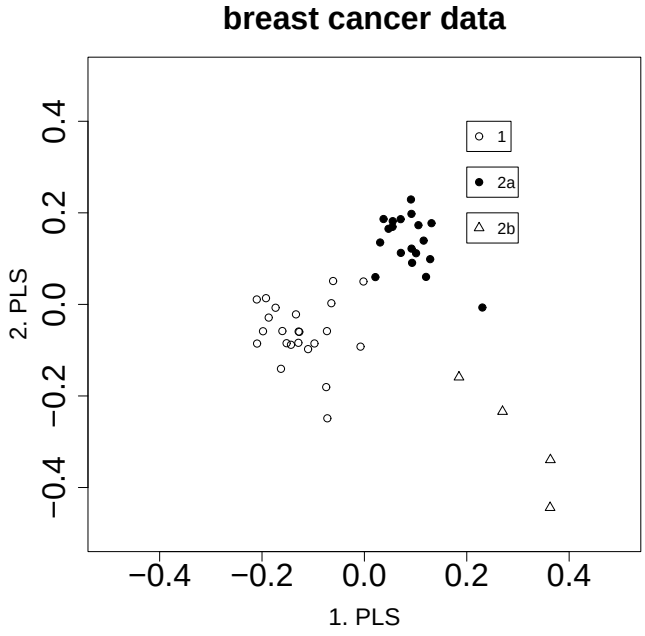

Figure 5.2: First and second PLS components for the breast cancer data

5.4 Classification results on real microarray data

5.4.1 Study design

For each data set, 200 random partitions into a learning data set $\mathcal{L}$ containing n_L observations and a test data set $\mathcal{T}$ containing the $n - n_L$ remaining observations are generated. This approach for evaluating classification methods was used in one of the most extensive comparative studies of classification methods for microarray data (Dudoit et al., 2002). It is believed to be more reliable than leave-one-out cross-validation (Braga-Neto and Dougherty, 2004). We fix the ratio n_L/n at 0.7, which is a usual choice. For each partition $\{\mathcal{L}, \mathcal{T}\}$, we predict the class of the observations from $\mathcal{T}$ using δ_{PLS} with successively 1,2,3,4,5 PLS components for the data sets with a binary response. We also use the discrete AdaBoost boosting algorithm based on the classifier $\delta = \delta_{PLS}$ with 1,2,3 PLS components. For data sets with multicategorical responses, we use 1,2,3,4,5,6 PLS

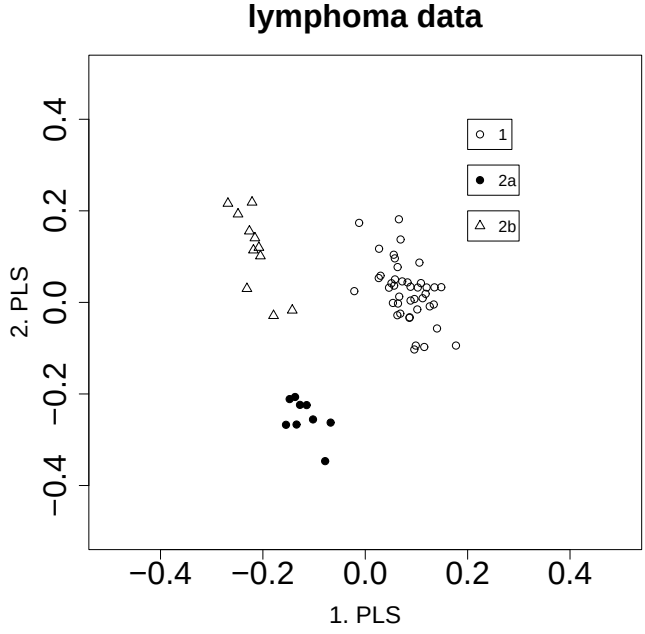

Figure 5.3: First and second PLS components for the lymphoma data with 2 classes

components for the lymphoma and BRCA data, 1,2,3,4,5,6,8,10 for the SRBCT data and 1,5,10,15,20 components for the NCI data.

For each approach and for each number of components, the mean error rate over the 200 partitions is computed using only the test set. Let $n_{\mathcal{T}_j}$ $(j = 1, \ldots, 200)$ denote the number of observations in the test set $\mathcal{T}_j$, $\mathcal{L}_1, \ldots, \mathcal{L}_{200}$ denote the 200 learning sets and $\mathcal{T}_1, \ldots, \mathcal{T}_{200}$ the 200 corresponding test sets. For a given approach, a given number of components and a given partition, $\hat{Y}_i$ denotes the predicted class of the ith observation of the test set. The mean error rate MER over the 200 partitions is given by

$$MER = \frac{1}{200} \sum_{k=1}^{200} \frac{1}{n_{\mathcal{T}_j}} \sum_{i=1}^{n_{\mathcal{T}_j}} I(\hat{Y}_i \neq Y_i), \tag{5.3}$$

where I is the standard indicator function ($I(A) = 1$ if A is true, $I(A) = 0$ otherwise).

The results are summarized in Tables 5.1 and 5.2.

For each partition $\{\mathcal{L}_j, \mathcal{T}_j\}$, the optimal number of PLS components m_{opt} is estimated

following the procedure described in section 2.3 and the error rate of δ_{PLS} with m_{opt} PLS components is computed. The corresponding mean error rate over the 200 random partitions is given in Table 5.1 (last column). The candidate numbers of components used to determine m_{opt} by cross-validation are also given in the table for each data set. For the data sets with a binary response, m_{opt} is chosen from $1, 2, 3, 4, 5$. For data sets with a multicategorical response (except the NCI data), m_{opt} is chosen from $1, 2, 3, 4, 5, 6, 7, 8, 9, 10$. For the NCI data set, which has much more classes, m_{opt} is chosen from $1, 5, 10, 15, 20$.

For comparison, the mean error rate obtained with some of the best classification methods for microarray data is also computed. The first one is nearest-neighbor classification based on 5 neighbors (5NN). This method can be summarized as follows. For each observation from the test set, the 5 closest observations ('neighbors') in the learning set are found and the observation is assigned to the class which is most common among those k neighbors. Closeness is measured using a specified distance metric. The most common distance metric, which we use here, is the euclidean distance metric. Nearest-neighbor classification is implemented in the `R` library `class`. This method is known to achieve good classification accuracy with microarray data (Dudoit et al., 2002).

The second method is linear discriminant analysis (LDA), which is also known to give good classification accuracy (Dudoit et al., 2002). A short description of linear discriminant is given in the following. Suppose we have p predictor variables. The random vector $\mathbf{x} = (X_1, \ldots, X_p)^T$ is assumed to a multivariate normal distribution within class k ($k = 1, \ldots, K$) with mean $\boldsymbol{\mu}_k$ and covariance matrix $\boldsymbol{\Sigma}_k$. In linear discriminant analysis, $\boldsymbol{\Sigma}_k$ is assumed to be the same for all classes: for all k, $\boldsymbol{\Sigma}_k = \boldsymbol{\Sigma}$. Using estimates $\hat{\boldsymbol{\mu}}_k$ and $\hat{\boldsymbol{\Sigma}}$ in place of $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}$, the maximum-likelihood discriminant rule assigns the ith new observation $\mathbf{x}_{new,i}$ to the class

$$\delta(\mathbf{x}_{new,i}) = \arg\min_k (\mathbf{x}_{new,i} - \hat{\boldsymbol{\mu}}_k)\hat{\boldsymbol{\Sigma}}^{-1}(\mathbf{x}_{new,i} - \hat{\boldsymbol{\mu}}_k)^T. \tag{5.4}$$

This approach is usually denoted as linear discriminant analysis, because $\delta(\mathbf{x}_{new,i})$ is a linear function of the vector $\mathbf{x}_{new,i}$. In our study, it does not perform as well as 5NN, SVM and PAM, probably because the estimation of the inverse of $\hat{\boldsymbol{\Sigma}}$ is not robust when the

number of variables is too large. Thus, the classification results using linear discriminant analysis are not shown.

The third method is Support Vector Machines (SVM). This method is used by Furey et al. (2000) and seems to perform well on microarray data. The idea is to find a separating hyperplane which separates the classes as well as possible in an enlarged predictor space. This leads to a complex optimization problem in high dimension. In our study, the optimal hyperplan is determined using the function `svm` from the `R` library `e1071`.

A short overview of NN, LDA and SVM is given in Hastie et al. (2001). These three methods require preliminary gene selection, either because they can not be applied if $n < p$ (LDA) or because they perform much better in practice if the number of noisy variables is not too large (NN and SVM). The gene selection is performed by ranking genes according to the BSS/WSS-statistic, where BSS denotes the between-group sum of squares and WSS the within-group sum of squares. For gene j the BSS/WSS-statistic is calculated as

$$BSS_j/WSS_j \;=\; \frac{\sum_{k=1}^{K}\sum_{i:Y_i=k}(\hat{\mu}_{jk}-\hat{\mu}_j)^2}{\sum_{k=1}^{K}\sum_{i:Y_i=k}(x_{ij}-\hat{\mu}_{jk})^2},$$

where $\hat{\mu}_j$ is the sample mean of X_j and $\hat{\mu}_{jk}$ is the sample mean of X_j within class k, for $k = 1, \ldots, K$. The genes with the highest BSS/WSS-statistic are selected. There is no well-established rule to choose the number of genes to select, which is a major drawback of classification methods requiring gene selection. In this study, we decide to use 20 or 50 genes for data sets with a binary response and 100 and 200 genes for data sets with a multicategorical response. The results obtained using other numbers of genes turn out to be similar or worse. Moreover, these numbers are in agreement with similar studies found in the literature (Dudoit et al., 2002).

At last, we apply a recent method called 'Prediction Analysis of Microarray'(PAM) which was especially designed for high-dimensional microarray data (Tibshirani et al., 2002). To our knowledge, it is the only fast classification method beside PLS which can be applied to high-dimensional data without gene selection. PAM is based on shrunken centroids. The user has to choose the shrinkage parameter Δ. The number of genes used to compute

the shrunken centroids depends on Δ. A possible choice is $\Delta = 0$: all genes are used to compute the centroids. Tibshirani et al. (2002) propose to select the best value of Δ by cross-validation: the classification accuracy is evaluated by leave-one-out cross-validation for a set of 30 values of Δ. The value of Δ minimizing the number of misclassifications is chosen. In our study, we try successively both approaches: $\Delta = 0$ (denoted as PAM) and $\Delta = \Delta_{opt}$ (denoted as PAM-opt), where Δ_{opt} is determined by leave-one-out cross-validation as described in Tibshirani et al. (2002). The PAM method as well the choice of Δ by cross-validation are implemented in the `R` library `pamr` (Tibshirani et al., 2002).

The table of results contains only the error rates obtained with 5NN, SVM, PAM and PAM-opt, because the classification accuracy with LDA was found to be comparatively bad for all data sets. The number of selected genes is specified for each method: for example, 'SVM-20' stands for Support Vector Machines with 20 selected genes.

The classification results obtained with δ_{PLS}, 5NN, SVM and PAM are presented in the next section, whereas the results obtained with boosting are discussed in Section 5.4.3.

5.4.2 Classification accuracy of δ_{PLS}

The classification results using the PLS-based approach δ_{PLS} are summarized in Table 5.1. The data sets with a binary response can be divided in two groups. For the leukemia and carcinoma data, the classification accuracy does not depend highly on the number of PLS components. It seems that subsequent components are only noise. On the contrary, the error rate is considerably reduced by using more than one component for the colon, prostate and breast cancer data. The improvement is rather dramatic for the prostate data. Thus, it seems that for data sets with low error rates (leukemia, carcinoma), the classes are optimally separated by one component, whereas subsequent components are useful for data sets with high error rates (prostate, colon, breast cancer).

PLS dimension reduction is very fast because it is based on linear operations with small matrices. The proposed procedure is much faster than the standard approach consisting of selecting a gene subset and building a classifier on this subset. For the lymphoma

Colon	1	2	3	4	5		m_{opt}
$(K = 2)$	0.136	0.114	0.119	0.143	0.147		0.124
Leukemia	1	2	3	4	5		m_{opt}
$(K = 2)$	0.020	0.028	0.03	0.030	0.028		0.024
Prostate	1	2	3	4	5		m_{opt}
$(K = 2)$	0.366	0.140	0.076	0.081	0.077		0.078
Breast cancer	1	2	3	4	5		m_{opt}
$(K = 2)$	0.14	0.110	0.104	0.106	0.103		0.110
Carcinoma	1	2	3	4	5		m_{opt}
$(K = 2)$	0.025	0.021	0.022	0.024	0.023		0.024
Lymphoma	1	2	3	4	5	6	m_{opt}
$(K = 3)$	0.037	0.0003	0.002	0.001	0.004	0.003	0.004
SRBCT	1	2	3	4	6	10	m_{opt}
$(K = 4)$	0.343	0.200	0.056	0.027	0.009	0.003	0.003
BRCA	1	2	3	4	5	6	m_{opt}
$(K = 3)$	0.468	0.348	0.310	0.268	0.285	0.303	0.304
NCI	1	5	10	15	20		m_{opt}
$(K = 8)$	0.715	0.338	0.293	0.318	0.325		0.329

Table 5.1: Mean error rate over 200 random partitions obtained with δ_{PLS} with different number of PLS components and with m_{opt} components

data and the SRBCT data, $K - 1$ seems to be the minimum number of PLS components required to obtain good classification accuracy. As can be seen from Table 5.1, δ_{PLS} can also perform very well on data sets with many classes ($K = 8$ for the NCI data).

As can be seen from Table 5.1, the number of components giving the best classification accuracy is not the same for all data sets. When our procedure to determine the number of useful PLS components is used for each partition $(\mathcal{L}, \mathcal{T})$, the classification accuracy turns out to be quite good. In Figure 5.4, histograms of m_{opt} over the 200 random partitions are represented for each data set. These histograms agree with Table 5.1. For instance,

Colon	5NN-20	5NN-50	$SVM-20$	$SVM-50$	PAM	PAM-opt
$(K=2)$	0.182	0.19	0.134	0.139	0.143	0.130
Leukemia	5NN-20	5NN-50	$SVM-20$	$SVM-50$	PAM	PAM-opt
$(K=2)$	0.034	0.039	0.038	0.05	0.022	0.046
Prostate	5NN-20	5NN-50	$SVM-20$	$SVM-50$	PAM	PAM-opt
$(K=2)$	0.119	0.124	0.086	0.085	0.370	0.099
Breast cancer	5NN-20	5NN-50	$SVM-20$	$SVM-50$	PAM	PAM-opt
$(K=2)$	0.117	0.123	0.100	0.093	0.120	0.147
Carcinoma	5NN-20	5NN-50	$SVM-20$	$SVM-50$	PAM	PAM-opt
$(K=2)$	0.020	0.021	0.024	0.029	0.036	0.096
Lymphoma	5NN-100	5NN-200	$SVM-100$	$SVM-200$	PAM	PAM-opt
$(K=3)$	0.014	0.003	0.038	0.019	0.013	0.042
SRBCT	5NN-100	5NN-200	$SVM-100$	$SVM-200$	PAM	PAM-opt
$(K=4)$	0.012	0.0052	0.010	0.014	0.046	0.069
BRCA	5NN-100	5NN-200	$SVM-100$	$SVM-200$	PAM	PAM-opt
$(K=3)$	0.378	0.318	0.588	0.581	0.331	0.396
NCI	5NN-100	5NN-200	$SVM-100$	$SVM-200$	PAM	PAM-opt
$(K=8)$	0.394	0.366	0.466	0.452	0.316	0.296

Table 5.2: Mean error rate over 200 random partitions with classical methods

the most frequent value of m_{opt} for the colon data is 2. It can be seen in Table 5.1 that the best classification accuracy is obtained with 2 PLS components for the colon data.

Some of the tested classical methods also perform well, especially SVM and PAM. SVM performs slightly better than PAM for most data sets. However, a pitfall of SVM is that it necessitates gene selection in practice, although not in theory. On the whole, the PLS-based method performs at least as good as the other methods for most data sets. More specifically, PLS performs better than the other methods for the colon, the prostate data, the SRBCT and the BRCA data. It is (approximately) as good as PAM and better than SVM and 5NN for the leukemia data, as good as SVM and better than PAM and 5NN for the breast cancer data, as good as 5NN and better than PAM and 5NN for the carcinoma data and the lymphoma data, and a bit worse than PAM-opt but much better than 5NN and PAM for the NCI data. Each of the three tested methods (5NN,SVM,PAM) performs much worse than PLS for at least two data sets. PLS is the only method which ranges among the two best methods for all data sets. This accuracy is not reached at the expense of computational time, except if one performs many cross-validation runs for the choice of the number of components. The problem of the choice of the number of components is one of the major drawbacks of the PLS approach. This problem is partly solved by the procedure based on cross-validation, but this procedure is computationally intensive and not optimal. Another inconvenience of the PLS approach which is often mentioned in the statistical literature is that it is based on an algorithm rather than on a theoretical probabilistic model, like LDA or PAM. However, PLS is a fast and efficient method which never fails to give a good to excellent classification accuracy for all the studied data sets. Since the best number of components can be estimated by cross-validation, the method does not involve any parameter like the number of selected genes for SVM or 5NN. Boosting does not improve the classification obtained with δ_{PLS} in most cases. However, the results are interesting because they indicate a qualitative similarity between boosting and PLS. This topic is discussed in the next section.

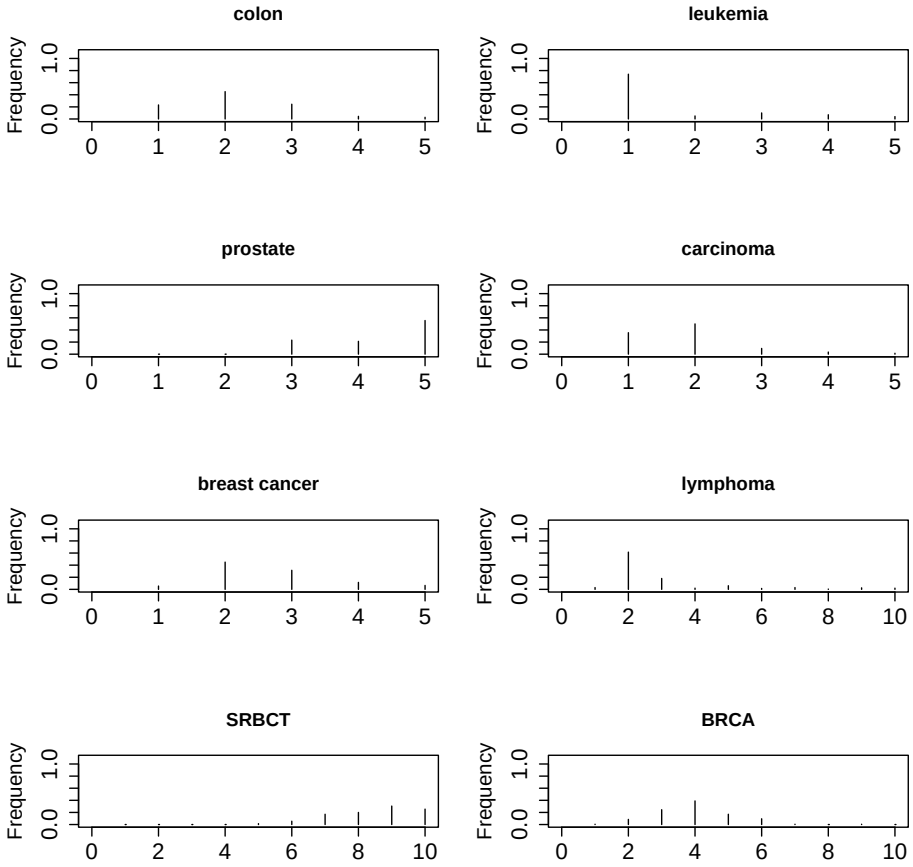

Figure 5.4: Histogram of the estimated optimal number of components for 200 partitions with different data sets (relative frequencies).

5.4.3 Classification accuracy of discrete AdaBoost with $\delta = \delta_{PLS}$

Real Data

In this section, we compute the mean classification error rate over 50 random partitions using the AdaBoost algorithm with $\delta = \delta_{PLS}$ and $B = 30$. $B = 30$ turns out to be a sensible choice for all data sets, because the classification accuracy remains constant after approximately 20 iterations. The results are represented in Figure 5.5 (top) for the prostate data. Boosting can reduce the error rate when one or two PLS components are used. However, the classification accuracy of δ_{PLS} with three PLS components is not improved by boosting. It can be seen from Table 5.1 that the best classification accuracy for δ_{PLS} is reached with three PLS components: the fourth and fifth PLS components do not improve the classification accuracy. Thus, with a fixed number m of PLS components, boosting improves the classification accuracy if and only the $(m+1)$th PLS component also does.

In order to examine the connection between boosting and PLS, we perform PLS dimension reduction on the whole prostate data set. We also run the AdaBoost algorithm with $\delta = \delta_{PLS}$ (with 1 component) and compute the empirical correlations between the four first PLS components and the first PLS component obtained at each boosting iteration (the correlations are computed based on the n observations). The results are shown for 5 boosting iterations in Table 5.3. The first component at each boosting iteration is strongly correlated with the first and the second PLS component, but not with the subsequent components. This statement agrees with the classification accuracy results: it can be seen from Figure 5.5 (top) that the classification accuracy obtained by boosting with one component equals approximately the classification accuracy of δ_{PLS} with two components.

Thus, both the classification results and the study of the correlations suggest a similarity between the PLS components obtained in subsequent boosting iterations and the subsequent PLS components obtained when δ_{PLS} is used without boosting. The same can be observed with the multicategorical responses. Here we focus on the SRBCT data, but the

	$B = 1$	$B = 2$	$B = 3$	$B = 4$	$B = 5$
PLS 1	0.80	−0.74	0.79	−0.74	0.60
PLS 2	−0.48	0.63	−0.35	0.58	−0.30
PLS 3	0.03	0.00	−0.00	0.00	0.14
PLS 4	−0.06	−0.01	−0.03	−0.02	−0.19

Table 5.3: Correlations between 4 PLS components and the 5 first PLS components with boosting (prostate data)

study of other data sets yields similar results. The mean error rate of δ_{PLS} with boosting is depicted in Figure 5.5 (bottom) for different numbers of PLS components. As for the prostate data, boosting reduces the error rate when one or two PLS components are used, but not when three PLS components are used. As can be seen from Table 5.1, three is the minimal number of components required to obtain good classification accuracy. Thus, with a fixed number m of PLS components, boosting improves the classification accuracy if and only the $(m + 1)$th PLS component also does.

The similarity between PLS and boosting can be intuitively and qualitatively explained as follows. In this paragraph, 'boosting' stands for 'boosting of δ_{PLS} with one component'. At iteration b in boosting, an observation is either in or out of the learning set, and the probability depends on how the observation was classified at iteration $b - 1$. The observations which are misclassified at iteration $b-1$ have higher probability to be selected in the learning set at iteration b. At each iteration, the error rate in the learning set is expected to decrease, since the algorithm focuses on 'problematic' observations. In practice, the PLS components computed at subsequent iterations have low correlations with the PLS component computed at the first iteration. The PLS component computed at the first iteration has high covariance with the class in the whole learning set, whereas the PLS components computed at subsequent iterations have high covariance with the class in particular learning sets where observations which are incorrectly predicted by the first PLS component are over-represented.

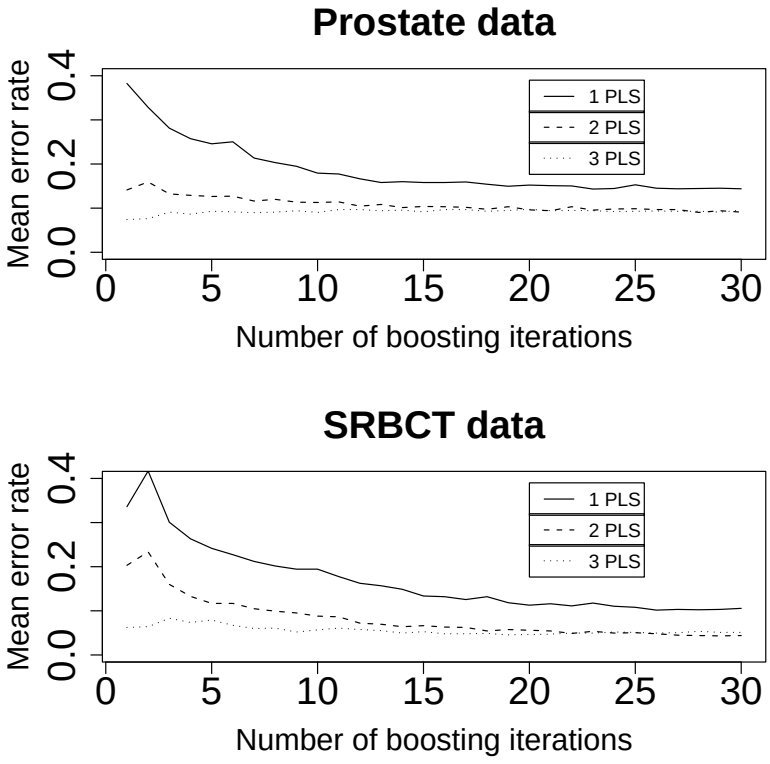

Figure 5.5: Mean error rate over 50 random partitions with AdaBoost and δ_{PLS} with different numbers of PLS components for the prostate data (top) and the SRBCT data (bottom)

Let us consider δ_{PLS} without boosting, but with several PLS components. For the computation of each PLS component, all the observations remain in the learning set, but the mth PLS component is uncorrelated with the $m-1$ first PLS components. Thus, observations which are correctly predicted by the $m-1$ first PLS components do not participate as much in the construction of the mth PLS component as the observations which are incorrectly predicted. In conclusion, both algorithms (boosting and PLS with several components) focus on observations or directions which have been neglected in the previous runs (for boosting) or components (for PLS). The theoretical connection between boosting and PLS could be examined in future work in a probabilistic framework.

Simulated Data

In simulations, we examine the effect of boosting on the classification accuracy for multicategorical data. For the generation of simulated data, the number of classes K is set successively to $K = 3$ and $K = 4$ and the number of observations in each class is set to 30 for the learning sets. The test sets contain 100 observations for each class, in order to improve the accuracy of the estimation of the error rate. To limit the computation time, the number of predictor variables p is set to $p = 200$. Similar results can be obtained with different values of n and p. Each class k is separated from the other classes by a group of 10 genes. The K groups of relevant genes are distinct, which is a simplifying but realistic hypothesis. For each class k, the 10 relevant genes are assumed to have the following conditional distributions:

$$\begin{aligned} X|Y = k &\sim \mathcal{N}(\mu = 0, \sigma = 1) \\ X|Y \neq k &\sim \mathcal{N}(\mu = 1, \sigma = 1), \end{aligned}$$

where $\mathcal{N}(\mu, \sigma)$ denotes the normal distribution with mean μ and standard deviation σ, and to be mutually uncorrelated within each class. Furthermore, the n observations are assumed to be identically and independently distributed given the class.

For $K = 3$ and $K = 4$ successively, we generate 50 learning data sets $\{\mathcal{L}_1, \ldots, \mathcal{L}_{50}\}$ and 50 test data sets $\{\mathcal{T}_1, \ldots, \mathcal{T}_{50}\}$ as follows. First, the K groups of 10 relevant genes are drawn

	1	2	3
K=3	0.328	0.077	0.113
K=4	0.504	0.283	0.104

Table 5.4: Mean error rate over 50 simulated learning sets and test sets with δ_{PLS} for different numbers of PLS components.

within each class from the conditional distributions given above. The remaining genes are drawn from the standard normal distribution for all classes. For each pair $\{\mathcal{L}_j, \mathcal{T}_j\}$ $(j = 1, \ldots, 50)$, δ_{PLS} with boosting $(B = 30)$ for 1,2,3 components is used to predict the classes of the observations from $\mathcal{T}_j$. The mean error rate over the 50 runs is then computed at each boosting iteration. The results are depicted in Figure 5.6 for $K = 3$ (top) and $K = 4$ (bottom). As can be seen from Figure 5.6, boosting improves the classification accuracy of δ_{PLS} if and only if less than $K - 1$ components are used. It seems that using boosting with a larger number of components can even decrease the classification accuracy. For comparison, the classification accuracy of δ_{PLS} without boosting is given in Table 5.4 for different numbers of PLS components. The best classification accuracy is achieved with $K - 1$ PLS components for both $K = 3$ and $K = 4$. Thus, the similarity between boosting and PLS which is observed for real data can also be observed for simulated data: for a given number m of PLS components, boosting improves the classification accuracy if and only if the $(m + 1)$th PLS component also does.

In the following section, we show a connection between the first PLS component and gene selection: the squared coefficient in the first PLS component can be seen as a score of relevance for single genes (see section 4 for more details). 'Boosted gene selection' might be an interesting application of boosting with PLS: we suggest that selecting the top-ranking genes at each boosting iteration might improve the classification accuracy of classifiers based on small gene subsets, although the study of this topic would be beyond the scope of this thesis.

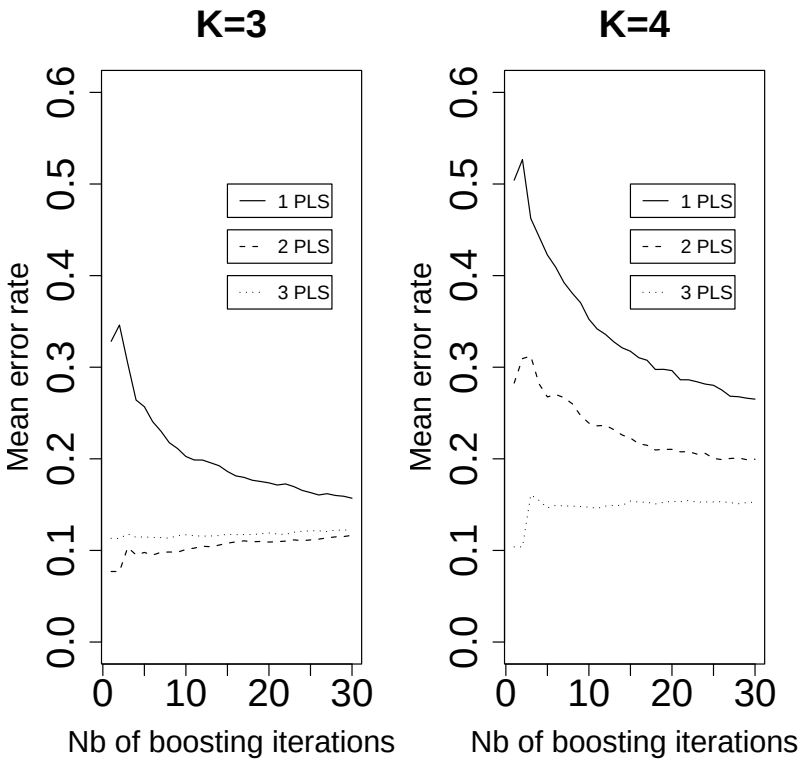

Figure 5.6: Mean error rate over 50 simulated learning sets and test sets with AdaBoost and δ_{PLS} with different numbers of PLS components for simulated data for $K = 3$ (left) and $K = 4$ (right)

5.5 PLS and gene selection

Biologists often want statisticians to answer questions such as 'which genes can be used for tumor diagnosis ?'. Thus, gene selection remains an important issue and should not be neglected. Dimension reduction is sometimes wrongly described as a black box which looses the information about single genes. In the following, we will see that PLS is strongly connected to gene selection.

In this section, only binary responses are considered: Y can take values 1 and 2. We denote as $\mathbf{Y}_C = (Y_{C1}, \ldots, Y_{Cn})^T$ the vector obtained by centering $\mathbf{Y} = (Y_1, \ldots, Y_n)^T$ to have zero mean:

$$\begin{aligned} Y_{Ci} &= -n_2/n \quad \text{if } Y_i = 1, \\ &= \ \ n_1/n \quad \text{if } Y_i = 2, \end{aligned}$$

where n_1 respectively n_2 are the numbers of observations in class 1 respectively 2.

To perform PLS dimension reduction, it is not necessary to scale each column of the data matrix $\mathbf{X}$ to unit variance. However, the first PLS component satisfies an interesting property with respect to gene selection if $\mathbf{X}$ is scaled. In this section, the columns of the data matrix $\mathbf{X}$ are supposed to be have been scaled to unit variance and, as usual in the PLS framework, centered to zero mean. $\mathbf{a} = (a_1, \ldots, a_p)^T$ denotes the $p \times 1$ vector defining the first PLS component as calculated by the SIMPLS algorithm.

A classical gene selection scheme consists of ordering the p genes according to BSS_j/WSS_j and selecting the top-ranking genes. For data sets with a binary response, we argue that a_j^2 can also be seen as a scoring criterion for gene j and we prove that the ordering of the genes obtained using BSS_j/WSS_j is the same as the ordering obtained using a_j^2.

Theorem 5.1. *If $K = 2$, there exists a strictly monotonic function f such that*

$$BSS_j/WSS_j = f(a_j^2),$$

for $j = 1, \ldots, p$.

Proof. From the SIMPLS algorithm, we get

$$\mathbf{a} = c_1 \cdot \mathbf{X}^T \mathbf{Y}_C,$$

where c_1 is a scalar. For $j = 1, \ldots, p$,

$$a_j = c_1 \cdot \sum_{i=1}^{n} x_{ij} Y_{Ci}.$$

It leads to

$$\begin{aligned} a_j &= c_1 \cdot (-(n_2/n) \textstyle\sum_{i:Y_i=1} x_{ij} + (n_1/n) \sum_{i:Y_i=2} x_{ij}) \\ a_j^2 &= c_1^2 \cdot (n_1 n_2/n)^2 (\hat{\mu}_{j2} - \hat{\mu}_{j1})^2 \end{aligned}$$

For $K = 2$,

$$\begin{aligned} BSS_j &= n_1(\hat{\mu}_{j1} - \hat{\mu}_j)^2 + n_2(\hat{\mu}_{j2} - \hat{\mu}_j)^2 \\ &= n_1((n\hat{\mu}_{j1} - n_1\hat{\mu}_{j1} - n_2\hat{\mu}_{j2})/n)^2 + n_2((n\hat{\mu}_{j2} - n_2\hat{\mu}_{j2} - n_1\hat{\mu}_{j1})/n)^2 \\ &= (n_1 n_2^2/n^2 + n_2 n_1^2/n^2)(\hat{\mu}_{j2} - \hat{\mu_{j1}})^2 \\ &= c_2 a_j^2, \end{aligned}$$

where c_2 is a positive constant which does not depend on j. $BSS_j + WSS_j$ is proportional to the sample variance of X_j. Since the variables $X_1, \ldots, X_p$ all have equal sample variance, there exists a constant c_3 which is independent of j such that

$$\begin{aligned} BSS_j/WSS_j &= \tfrac{BSS_j}{c_3 - BSS_j} \\ &= \tfrac{c_2 a_j^2}{c_3 - c_2 a_j^2}. \end{aligned}$$

□

As a consequence, the first PLS component calculated by the SIMPLS algorithm can be used to order and select genes and the ordering is the same as the ordering produced by one of the most widely accepted selection criteria. As an illustration, the BSS/WSS ratio can be computed for the 2000 genes of the colon data set. For the 5 first genes, one obtains:

$$1.069 \cdot 10^{-2}, \quad 3.979 \cdot 10^{-5}, \quad 6.439 \cdot 10^{-3}, \quad 2.431 \cdot 10^{-3}, \quad 9.492 \cdot 10^{-4}.$$

The coefficients of these 5 genes for the first PLS component are

$$9.280 \cdot 10^{-5}, \quad -5.691 \cdot 10^{-6}, \quad 7.217 \cdot 10^{-5}, \quad -4.444 \cdot 10^{-5}, \quad 2.779 \cdot 10^{-5}.$$

As can be seen from these partial results, the ordering of the genes produced by the BSS/WSS ratio is the same as the ordering produced by the absolute value of the coefficient for the first PLS component. For the colon data, the 5 top-ranking genes are gene 493 (Hsa.37937), gene 377 (Hsa.36689), gene 249 (Hsa.8147), gene 1635 (Hsa.2097) and gene 1423 (Hsa.1832).

Up to a constant, the BSS/WSS-statistic equals the F-statistic which is used to test the equality of the means within different groups. Thus, we have proved that the SIMPLS algorithm can be used as a gene selection procedure which is exactly equivalent to the procedure based on the BSS/WSS ratio or on the F-statistic. This method tends to be sensitive to outliers, which are common in microarray data. Moreover, it does not incorporate interactions and correlations between genes, as all univariate criteria. However, it is one of the most widely used criteria for gene selection and seems to perform well in most cases (Dudoit et al., 2002). We claim that one should rather use the first PLS component than the BSS/WSS ratio because it is faster to compute.

5.6 Discussion

In this chapter, several aspects of PLS dimension reduction for classification are examined. First, PLS is compared to several other classification methods which are known to give excellent classification accuracy. To our knowledge, this work is the first extensive comparison study including PLS. The classifier δ_{PLS} turns out to be the best one in terms of classification accuracy for most of the data sets. Another advantage is its computational efficiency. Even if PLS dimension reduction is originally designed for continuous regression, it can be successfully applied to classification problems. To determine the optimal number of PLS components, a simple cross-validation procedure is proposed. The reliability of this procedure is quite good, although not perfect. An aggregation strategy (AdaBoost) is used in the hopes of improving the classification accuracy, because aggregation methods are known to be very effective in reducing the error rate on independent test data. The conclusion is that boosting does not improve the classification accuracy

of PLS, except in some special cases. The second topic of this chapter is gene selection. We show that the first PLS component can be used for gene selection and prove that the proposed procedure is equivalent to a well-known gene selection procedure found in the literature. Thus, the information on single genes does not get lost through PLS dimension reduction. Moreover, we claim that PLS dimension reduction can be used as a visualization tool. Contrary to principal component analysis, PLS is a supervised procedure which uses the information about the class of the observations to construct the new components. Unlike sufficient dimension reduction and related methods, PLS can handle all the genes simultaneously and performs gene selection intrinsically. In a word, PLS is a very fast and competitive tool for classification problems with high-dimensional microarray data as regards to prediction accuracy, feature selection and visualization. In future work, one could examine the theoretic connection between PLS and boosting, as well as the use of boosting in gene selection. Since the best classification accuracy is often reached with more than one PLS component, the subsequent PLS components could also be used to perform a refined gene selection. One could also try to improve the procedure to choose the number of components. It seems that cross-validation is appropriate, but a more sophisticated cross-validation scheme could maybe improve the classification performance of our PLS-based approach.

Chapter 6

Conclusion

In this thesis, different aspects of dimension reduction for high-dimensional microarray data have been studied. In this chapter, a summary of the achieved objectives and the possible directions for future methodological research are given for each topic, as well as a brief outlook on the evolution of biological research and its consequences for the statistical analysis.

- In the introductory Chapter 2, I reviewed briefly a few usual variable selection criteria, as well as a few common methods used to compare different classification methods. Variable selection methods can be divided in two categories: methods based on univariate selection criteria and methods looking for optimal variable subsets. While univariate selection criteria miss potentially useful information, optimal subsets tend to overfit the learning data set. Dimension reduction is an interesting alternative to variable selection. I also reviewed four common approaches used to compare classification methods. In this thesis, I always use Approach 3 (successive random splittings of the available data set into learning data set and test data set) to compare classification methods, because it is the most reliable design according to several studies.

- In Chapter 3, I mapped the data mining tool "emerging pattern" into a statistical framework and proposed a new and more general probabilistic definition. I proposed a new fast search algorithm to identify such patterns in high-dimensional data sets and showed its efficiency in simulations. I also proposed a new approach to use emerging patterns for classification. The programs are implemented in the language `R`. The concept of emerging patterns which is proposed in this thesis could be generalized to categorical or ordinal variables in order to integrate other types of data. In the context of low-dimensional data, emerging patterns could also be examined in the multiple testing framework.

- In Chapter 4, the strong connection between three linear dimension reduction and classification approaches is proved for $K = 2$. I showed that the first PLS component is the same as the first between-group principal component and that between-group principal component analysis is strongly related to classical linear discriminant analysis for binary response variables ($K = 2$). In future work, one could examine the case $K > 2$, which is common in practical microarray studies. The connection to the wide family of sufficient dimension reduction methods could also be investigated, as well as an adaptation of these methods to very high-dimensional data with $n < p$.

- Chapter 5 deals with PLS dimension reduction with application to classification problems. PLS regression is a powerful tool for the analysis of various types of high-dimensional data, ranging from chemometric data to microarray data. In an extensive comparison study based on nine real microarray data sets, I showed that the classification method consisting of PLS dimension reduction and linear discriminant analysis using the PLS components ranges among the best classification methods and should be recommended because of its computational efficiency and conceptual simplicity. Moreover, PLS dimension reduction can be used to visualize microarray data in two or three dimensions in the context of classification. I showed that the obtained directions can often be interpreted in terms of subclasses in practice. I also examined the connection between PLS dimension reduction and boosting and pointed out a qualitative similarity between these two approaches. Lastly, I proved

a property concerning the connection between the so-called BSS/WSS ratio, which is one of the most common selection criteria in microarray data analysis, and the coefficient in the first PLS component. The theoretical connection between boosting and PLS dimension reduction, as well as the application of boosting to variable selection could be examined in further work. In the context of class prediction, the performance of PLS dimension reduction could be improved by refining the choice of the number of components. Another direction for future research is the application of PLS regression or dimension reduction to other types of high-dimensional data and other biological issues.

Indeed, the multiplication of experimental technologies generating high-dimensional data in the field of life sciences constitutes a major challenge for biostatisticians who now have to develop methods integrating different types of data. For instance, protein-protein interaction data, which give information on single pairs of proteins, are becoming common. In the near future, gene expression data will be only a small piece of a giant puzzle. Detecting interaction structures using several high-dimensional data sets will probably be one of the most important tasks for biostatisticians.

Bibliography

Agresti, A., 2002. Categorical Data Analysis. Wiley, New York.

Alizadeh, A. A., Eisen, M. B., Davis, R. E., Ma, C., Lossos, I. S., Rosenwald, A., Boldrick, J. C., Sabet, H., Tran, T., Yu, X., Powell, J. I., Yang, L., Marti, G. E., Moore, T., Hudson, J., Lu, L., Lewis, D. B., Tibshirani, R., Sherlock, G., Chan, W. C., Greiner, T. C., Weisenburger, D. D., Armitage, J. O., Warnke, R., Levy, R., Wilson, W., Grever, M. R., Byrd, J. C., Botstein, D., Brown, P. O., Staudt, L. M., 2000. Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. Nature 403, 503–511.

Alon, U., Barkai, N., Notterman, D. A., Gish, K., Ybarra, S., Mack, D., Levine, A., 1999. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. Proceedings of the National Academy of Sciences 96, 6745–6750.

Alter, O., Brown, P. O., Botstein, D., 2000. Singular value decomposition for genome-wide expression data processing and modelling. Proceedings of the National Academy of Sciences 97, 10101–10106.

Beer, D. G., Kardia, S. L. R., Huang, C.-C., Giordano, T. J., Levin, A. M., Misek, D. E., Lin, L., Chen, G., Gharib, T. G., Thomas, D. G., Lizyness, M. L., Kuick, R., Hayasaka, S., Taylor, J. M. G., Iannettoni, M. D., Orringer, M. B., Hanash, S., 2002. Gene-expression profiles predict survival of patients with lung adenocarcinoma. Nature Medicine 4, 816–824.

Ben-Dor, A., Bruhn, L., Friedman, N., Nachman, I., Schummer, M., Yakhini, Z., 2000. Tissue classifiction with gene expression profiles. Journal of Computational Biology 7, 559–584.

Binder, H., Tutz, G., 2004. Localized logistic classification with variable selection. In: J. Antoch (Ed.) COMPSTAT 2004, Physica Verlag.

Bo, T. H., Jonassen, I., 2002. New feature subset selection procedures for classification of expression profiles. Genome Biology 3, R17.

Boulesteix, A. L., 2004. PLS dimension reduction for classification with microarray data. Statistical Applications in Genetics and Molecular Biology 3, Issue 1, Article 33.

Boulesteix, A. L., 2005. A note on between-group PCA. International Journal of Pure and Applied Mathematics (to appear).

Boulesteix, A. L., Tutz, G., 2005. Identification of interaction patterns and classification with applications to microarray data. Computational Statistics and Data Analysis (to appear).

Boulesteix, A. L., Tutz, G., Strimmer, K., 2003. A cart-based approach to discover emerging patterns in microarray data. Bioinformatics 19, 2465–2472.

Braga-Neto, U., Dougherty, E. R., 2004. Is cross-validation valid for small-sample microarray classification ? Bioinformatics 20, 374–380.

Breiman, L., 1996. Bagging predictors. Machine Learning 24, 123–140.

Breiman, L., 2001. Random forests. Machine Learning 45, 5–32.

Breiman, L., Friedman, J. H., Olshen, R. A., Stone, J. C., 1984. Classification and Regression Trees. Wadsworth, Monterey, CA.

Chiaromonte, F., Martinelli, J., 2001. Dimension reduction strategies for analyzing global gene expression data with a response. Mathematical Biosciences 176, 123–144.

Chilingaryan, A., Gevorgyan, N., Vardanyan, A., Jones, D., Szabo, A., 2002. Multivariate

approach for selecting sets of differentially expressed genes. Mathematical Biosciences 176, 59–72.

Cook, R. D., Lee, H., 1999. Dimension reduction in binary response regression. Journal of the American Statistical Association 94, 1187–1200.

Cook, R. D., Weisberg, S., 1991. Discussion of "sliced inverse regression" by k. c. li. Journal of the American Statistical Association 86, 328–332.

Culhane, A. C., Perriere, G., Considine, E., Gotter, T., Higgins, D., 2002. Between-group analysis of microarray data. Bioinformatics 18, 1600–1608.

de Jong, S., 1993. SIMPLS. an alternative approach to partial least squares regression. Chemometrics and Intelligent Laboratory Systems 18, 251–253.

Dettling, M., Bühlmann, P., 2003. Boosting for tumor classification with gene expression data. Bioinformatics 19, 1061–1069.

Dong, G., Li, J., 1999. Efficient mining of emerging patterns: Discovering trends and differences. In: Proceedings of the SIGKDD (5th ACM International Conference on Knowledge Discovery and Data Mining), San Diego, CA. pp. 43–52.

Dudoit, S., Fridlyand, J., Speed, T. P., 2000. Comparison of discrimination methods for the classification of tumors using gene expression data. Technical Report 576: http://www.stat.berkeley.edu/tech-reports/index.html.

Dudoit, S., Fridlyand, J., Speed, T. P., 2002. Comparison of discrimination methods for the classification of tumors using gene expression data. Journal of the American Statistical Association 97, 77–87.

Dudoit, S., Shaffer, J. P., Boldrick, J. C., 2003. Multiple hypothesis testing in microarray experiments. Statistical Science 18, 71–103.

Eisen, M. B., Spellman, P. T., Brown, P. O., Botstein, D., 1998. Cluster analysis and display of genome-wide expression patterns. Proceedings of the National Academy of Sciences 95, 14863–14868.

Frank, I. E., Friedman, J. H., 1993. A statistical view of some chemometrics regression tools. Technometrics 35, 109–135.

Freund, Y., 1995. Boosting a weak learning algorithm by majority. Information and Computation 121, 256–285.

Friedman, J. H., Fisher, N., 1999. Bump-hunting in high-dimensional data. Statistics and Computing 9, 123–143.

Fukunaga, K., 1990. Introduction to Statistical Pattern Recognition. Academic Press, Boston, MA.

Furey, T. S., Cristianini, N., Duffy, N., Bednarski, D. W., Schummer, M., Haussler, D., 2000. Support vector machine classification and validation of cancer tissue samples using microarray expression data. Bioinformatics 16, 906–914.

Garthwaite, P. H., 1994. An interpretation of partial least squares. Journal of the American Statistical Association 89, 122–127.

Ghosh, D., 2002. Singular value decomposition regression modelling for classification of tumors from microarray experiments. Proceedings of the Pacific Symposium on Biocomputing 98, 11462–11467.

Gnanadesikan, R., 1977. Methods for Statistical Data Analysis of Multivariate Observations. Wiley, New York.

Golub, T., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M. L., Downing, J., Caligiuri, M. A., Bloomfield, C. D., Lander, E. S., 1999. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science 286, 531–537.

Hastie, T., Tibshirani, R., Eisen, M. B., Alizadeh, A., Levy, R., Staudt, L., Chan, W. C., Botstein, D., Brown, P., 2000. 'Gene shaving' as a method for identifying distinct sets of genes with similar expression patterns. Genome Biology 1, R3.

Hastie, T., Tibshirani, R., Friedman, J. H., 2001. The elements of statistical learning. Springer-Verlag, New York.

Hedenfalk, I., Duggan, D., Chen, Y., Radmacher, M., Bittner, M., Simon, R., Meltzer, P., Gusterson, B., Esteller, M., Raffeld, M., Yakhini, Z., Ben-Dor, A., Dougherty, E., Kononen, J., Bubendorf, L., Fehrle, W., Pittaluga, S., Gruvberger, S., Loman, N., Johannsson, O., Olsson, H., Wilfond, B., Sauter, G., Kallioniemi, O.-P., Borg, A., Trent, J., 2001. Gene-expression profiles in hereditary breast cancer. New England Journal of Medicine 344, 539–548.

Hennig, C., 2004. Asymmetric linear dimension reduction for classification. Journal of Computational and Graphical Statistics 13, 930–945.

Hollander, M., Wolfe, D. A., 1973. Nonparametric statistical inference. Wiley, New York.

Huang, X., Pan, W., 2003. Linear regression and two-class classification with gene expression data. Bioinformatics 19, 2072–2078.

Jäger, J., Sengupta, R., Ruzzo, W. L., 2003. Improved gene selection for classification of microarray. Proceedings of the 2003 Pacific Symposium on Biocomputing, 53–64.

Jolliffe, I. T., 1986. Principal Component Analysis. Springer-Verlag, New York.

Kahn, J., Wei, J. S., Ringner, M., Saal, L. H., Ladanyi, M., Westermann, F., Berthold, F., Schwab, M., Antonescu, C. R., Peterson, C., Meltzer, P. S., 2001. Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. Nature Medicine 7, 673–679.

Lausen, B., Schumacher, M., 1992. Maximally selected rank statistics. Biometrics 48, 73–85.

Li, J., Wong, L., 2003. Simple rules underlying gene expression profiles of more than six subtypes of acute lymphoblastic leukemia (ALL) patients. Bioinformatics 19, 71–78.

Li, K. C., 1991. Sliced inverse regression for dimension reduction. Journal of the American Statistical Association 86, 316–342.

Li, K. C., 1992. On principal hessian directions for data visualization and dimension reduction: Another application of Stein's lemma. Journal of the American Statistical Association 87, 1025–1039.

Li, L., Weinberg, C. R., Darden, T. A., Pedersen, L. G., 2001. Gene selection for sample classification based on gene expression: study of sensitivity to choice of parameters of the GA/KNN method. Bioinformatics 17, 1131–1142.

Lloyd, C. J., 2000. Regression models for convex ROC curves. Biometrics 56, 562–567.

Martens, H., 2001. Reliable and relevant modelling of real world data: a personal account of the development of PLS regression. Chemometrics and Intelligent Laboratory Systems 58, 85–95.

Martens, H., Naes, T., 1989. Multivariate Calibration. Wiley, New York.

Model, F., Adorjan, P., Olek, A., Piepenbrock, C., 2001. Feature selection for DNA methylation based cancer classification. Bioinformatics 17, 157–164.

Morgan, J. N., Sonquist, J. A., 1963. Problems in the analysis of survey data, and a proposal. Journal of the American Statistical Association 58, 415–435.

Nguyen, D., Rocke, D. M., 2002a. Tumor classification by partial least squares using microarray gene expression data. Bioinformatics 18, 39–50.

Nguyen, D. V., Rocke, D. M., 2002b. Partial least squares proportional hazard regression for application to DNA microarray survival data. Bioinformatics 18, 1625–1632.

Notterman, D. A., Alon, U., Sierk, A. J., Levine, A. J., 2001. Transcriptional gene expression profiles of colorectal adenoma, adenocarcinoma, and normal tissue examined by oligonucleotide arrays. Cancer Research 61, 3124–3130.

O'Neill, M. C., Song, L., 2003. Neural network analysis of lymphoma microarray data: prognosis and diagnosis near-perfect. BMC Bioinformatics 4, 13.

Ooi, C. H., Tan, P., 2003. Genetic algorithms applied to multi-class prediction for the analysis of gene expression data. Bioinformatics 19, 37–44.

Park, P. J., Tian, L., Kohane, I. S., 2002. Linking gene expression data with patient survival times using partial least squares. Bioinformatics 18, 120–127.

Park, T., Yi, S.-G., Lee, S., Lee, S. Y., Yoo, D.-H., Ahn, J.-I., Lee, Y.-S., 2003. Statistical tests for identifying differentially expressed genes in time-course microarray experiments. Bioinformatics 19, 694–703.

R-Development-Core-Team, 2004. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria, ISBN 3-900051-00-3.

Rao, C. R., 1952. Advanced Statistical Methods in Biometric Research. Wiley, New York.

Ripley, B. D., 1996. Pattern Recognition and Neural Networks. Cambridge University Press, Cambridge.

Ross, D. T., Scherf, U., Eisen, M. B., Perou, C. M., Spellman, P., Iyer, V., Jeffrey, S. S., de Rijn, M. V., Waltham, M., Pergamenschikov, A., Lee, J. C. F., Lashkari, D., Shalon, D., Myers, T. G., Weinstein, J. N., Botstein, D., Brown, P. O., 2000. Systematic variation in gene expression patterns in human cancer cell lines. Nature Genetics 24, 227–234.

Schapire, R., Freund, Y., Bartlett, P., Lee, W., 1998. Boosting the margin: A new explanation for the effectiveness of voting methods. Annals of Statistics 26, 1651–1686.

Simon, R. M., Korn, E. L., McShane, L. M., Radmacher, M. D., Wright, G. W., Zhao, Y., 2004. Design and Analysis of DNA Microarray Investigations. Springer-Verlag, New York.

Singh, D., Febbo, P. G., Ross, K., Jackson, D. G., Manola, J., Ladd, C., Tamayo, P., Renshaw, A. A., D'Amico, A. V., Richie, J. P., Lander, E. S., Loda, M., Kantoff, P. W., Golub, T. R., Sellers, W. R., 2002. Gene expression correlates of clinical prostate cancer behavior. Cancer Cell 1, 203–209.

Stone, M., Brooks, R. J., 1990. Continuum regression: cross-validated sequentially constructed prediction embracing ordinary least squares, partial least squares and principal component regression. Journal of the Royal Statistical Society B 52, 237–269.

Swets, J., Pickett, R., 1982. Evaluation of Diagnostic Systems; Methods from Signal Detection Theory. Academic Press, New York.

Tamayo, P., Slonim, D., Mesirov, J., Zhu, Q., Kitareewan, S., Dmitrovsky, E., Lander, E. S., Golub, T. R., 1999. Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation. Proceedings of the National Academy of Sciences 96, 2907–2912.

Tibshirani, R., 1996. Regression shrinkage and selection via the LASSO. Journal of the Royal Statistical Society B 58, 267–288.

Tibshirani, R., Hastie, T., Narasimhan, B., Chu, G., 2002. Diagnosis of multiple cancer types by shrunken centroids of gene expression. Proceedings of the National Academy of Sciences 99, 6567–6572.

Tutz, G., Hechenbichler, K., 2004. Aggregating classifiers with ordinal response structure. Journal of Statistical Computation and Simulation (to appear).

Venkatraman, E. S., 2000. A permutative test to compare receiver operating characteristic curves. Biometrics 56, 1134–1138.

West, M., Blanchette, C., Dressman, H., Huang, E., Ishida, S., Spang, R., Zuzan, H., Olson, J., Marks, J., Nevins, J., 2002. Predicting the clinical status of human breast cancer using gene expression profiles. Proceedings of the National Academy of Sciences 98, 11462–11467.

Wichert, S., Fokianos, K., Strimmer, K., 2004. Identifying periodically expressed transcripts in microarray time series data. Bioinformatics 20, 5–20.

Yeung, K. Y., Fraley, C., Murua, A., Raftery, A. E., Ruzzo, W. L., 2001. Model-based clustering and data transformation for gene expression data. Bioinformatics 17, 977–987.

Yeung, K. Y., Ruzzo, W. L., 2001. Principal component analysis for clustering gene expression data. Bioinformatics 17, 763–774.

Young, D. M., Marco, V. R., Odell, P. L., 1987. Quadratic discrimination: some results on optimal low-dimensional representation. Journal of Statistical Planning and Inference 17, 307–319.

Lebenslauf

Persönliche Daten

Name	Anne-Laure Isabeau Boulesteix
Familienstand	verheiratet mit Günther Socher, 1 Kind (geb. 2003)
Staatsangehörigkeit	französisch
Geburtsdatum/-ort	11.03.1979 in Paris XIV (Frankreich)

Ausbildung

09.1984-06.1989	Grundschule in Rueil-Malmaison (Frankreich).
09.1989-06.1996	Gymnasium in Rueil-Malmaison (Frankreich).
09.1996-12.2001	Studium der Mathematik und Ingenieurwissenschaften im Rahmen eines Doppeldiplomprogramms.
Studienabschlüsse	2001: Diplom Mathematikerin der Universität Stuttgart. 2002: Diplom Ingenieurin der Ecole Centrale Paris (Frankreich).
Seit 04.2002	Promotion in Statistik bei Prof. Dr. Gerhard Tutz (LMU München).

Berufliche Erfahrung

Sommer 1999	Werkstudentin bei der Bayer AG (Monheim) im Biolabor.
Sommer 2000	Fachpraktikum in Verfahrenstechnik bei der Bayer AG (Leverkusen).
10.2000-01.2001	Hilfswissenschaftlerin am Institut für Mathematik A, Universität Stuttgart.
Seit April 2002	Wissenschaftliche Mitarbeiterin am Institut für Statistik, LMU München. 2 Stunden Lehrverpflichtungen seit Oktober 2002 Frauenbeauftragte des Instituts für Statistik seit Oktober 2004 Sokrates/Erasmus Programmbeauftragte