Local Analysis of High Dimensional Genetic Data Considering Interaction Effects

Dissertation

by

Tina Müller

Submitted to

Fakultät Statistik,
Technische Universität Dortmund

in Fulfillment of the Requirements for the Degree of
Doktorin der Naturwissenschaften

Berlin, December 2009

Referees:

Prof. Dr. Katja Ickstadt
Prof. Dr. Jörg Rahnenführer

Date of Oral Examination:

January 22^{nd}, 2010

Acknowledgements

I cannot thank my supervisor Katja Ickstadt enough for her support and encouragement throughout my whole time as a PhD student. Without her constant feedback and the fantastic opportunities she gave me (my studies in London and the break from teaching in the last phase of the thesis), I am not sure if I would have finished. Thanks for everything, Katja!!

My special thanks goes to Holger Schwender who engaged in countless discussions and gave me so many valuable ideas, unlimited support as well as constructive criticism. I am also indebted to Gero Szepannek, an extremely creative mind and encouraging friend. Holger and Gero, you guys really rock!

A big 'Thank you' goes to all students who supported my work, in detail to Daniela Breiter, Maria Eveslage, Carolin Pütter, Martin Schäfer and Britta Schulze Waltrup. My colleagues, especially Björn Bornkamp and Arno Fritsch, all helped with their ideas and with proof-reading, for which I am very grateful. For fruitful discussions and feedback on computer scientific aspects, I'd like to thank Ingo Mierswa and Katharina Morik. Big cheers for Ingo Ruczinski who volunteered to read my chapter on his logic regression. A beer well invested...

Last and definitely not least, I thank my parents Volker and Bärbel Müller, especially for their patience and constant support, as well as Irina Czogiel, my best friend throughout all my statistical life.

Financial support of the Deutsche Forschungsgemeinschaft (SFB 475, "Reduction of complexity in multivariate data structures") is gratefully acknowledged.

Contents

List of Figures

List of Tables

List of Symbols

Data

$P(D)$	probability of developing disease D
α	base line odds (genetic model)
θ	effect size (genetic model)

Clustering

$V = \{V_1, \ldots, V_m\}$	set of variables
m	number of variables
n	number of observations
$S(\cdot)$	similarity measure
$D(\cdot)$	distance
$\bar{S}(\cdot)$	similarity measure for clusters
$\mathbf{m}^+$	matching categories
$\mathbf{m}^-$	mismatching categories
m_{ij}	entry of contingency table, $i, j = 0, 1, 2$
n_+	number of matching categories
n_-	number of mismatching categories
$\mathbb{1}_b$	b-dimensional vector of ones
$\mathbf{e}$	unity vector
$SMC(\cdot)$	simple matching coefficient
$\mathbf{w}_F^{+\prime}$	weights for matching categories
$\mathbf{w}_F^{-\prime}$	weights for mismatching categories
$FMC_{\mathbf{w}_F}(\cdot)$	flexible matching coefficients
JC	Jaccard's coefficient
S_p	Pearson's corrected contingency coefficient
$\mathbf{C}_t$	cluster t
$m_{\mathbf{C}_t}$	number of elements of cluster t
$f_1 - f_4$	quality measures
D'	measure of linkage disequilibrium
r^2	measure of linkage disequilibrium

K	number of clusters
K_1	number of clusters with one element
B	number of linkage disequilibrium blocks
$m_{\mathbf{C}_{k},b}$	number of SNPs of block b in cluster k
m_c	number of causative SNPs
$\mathcal{C}_{ca,k_i}$	set of SNPs in cluster k_i for the cases
$\mathcal{C}_{con,k_i}$	set of SNPs in cluster k_i for the controls
U_{k_i}	united set of SNPs from clusters k_i in both partitions
$m_{U_{k_i}}$	number of elements in U_{k_i}
I_{k_i}	intersecting set of SNPs from clusters k_i in both partitions
$m_{I_{k_i}}$	number of elements in I_{k_i}
$m^{nc}_{ca,k'}$	number of non-causative SNPs in cluster k' for the cases
$m^{nc}_{con,k'}$	number of non-causative SNPs in cluster k' for the controls
$d(\cdot)$	desirability function
$q(\cdot)$	desirability index
$q^{Harr}(\cdot)$	Harrington's desirability index
$pur(\cdot)$	purity
V_{opt}	set of optimal variables
m_{opt}	number of optimal variables

Classification

$C(\cdot)$	classifier
y_i	class label of observation i
$\mathbf{y}$	vector of class labels
$\mathbf{x}$	set of covariates
$\mathbf{x_i}$	covariates for observation i
$\mathbf{X}$	matrix of covariates
n_{Tr}	number of observations in training data
n_{Te}	number of observations in test data
n_{miss}	number of misclassified observations
x	variable
MCR	misclassification rate
$\hat{y}_i$	predicted class label of observation i
$m_{\mathcal{I}}$	number of items
I	item
$\mathcal{I}$	set of items

T_i	transaction i
$\mathcal{D}$	data set
$\mathcal{D}^{Tr}$	training data set
$\mathcal{D}^{Te}$	test data set
B	antecedent of an association rule
H	consequent of an association rule
R_r	association rule r
$supp$	support
$supp_{min}$	minimal support treshold
$\mathcal{D}_B$	transactions containing B
con	confidence
con_{min}	minimal confidence treshold
$lift$	lift
$conviction$	conviction
$oddsRatio$	odds Ratio
F_1	set of frequent 1-itemsets
FC_k	candidate set k
FC_T	candidates contained in T
fc	single candidate itemset
F_k	list of frequent itemsets of length k
f^k	element of F_k
n_k	number of elements in F_k
F	ordered list of all frequent itemsets
n_F	number of itemsets in F
T_i^{Tr}	transaction i in training data set
$G_{T_i^{Tr}}$	class label of T_i^{Tr}
G_{misc}	miscellaneous group
G_g	group G_g of transactions
n_{G_g}	number of elements in group G_g
$\mathbf{R}$	set of association rules
$n_{\mathbf{R}}$	number of elements in $\mathbf{R}$
$\mathbf{R}(T_i)$	applicable association rules for transaction T_i
$\delta(\cdot)$	decision rule
γ	voting fraction
w_i	weight i
ν	scaling parameter

y	response variable
L	boolean logic expression
β	regression coefficient
q	number of logic expressions
$\wedge$	*and*-operator
$\vee$	*or*-operator
^{c}	negation
$\mathcal{L}(\cdot)$	likelihood function
$\pi(\mathbf{x}_i)$=π_i	probability of observing y=1 given $\mathbf{x}_i$
$D(\cdot)$	deviance
$W(\cdot)$	weighting function
$d(\cdot)$	distance function
$h(\cdot)$	bandwidth
s_j	scaling parameter
λ	smoothing parameter
s^2	variance
$\tilde{t}$	test statistics to compare misclassification rates
t_5	t distribution with 5 degrees of freedom

CHAPTER 1

Introduction

Starting with Mendel's discovery of hereditary transmission of characteristic traits in pea plants in the 19^{th} century, genetics has ever since been an active field of research. Today, huge interest lies in analysing the association of genetic predisposition with the development of diseases (e.g., cancer), promising a better understanding of the disease mechanisms as well as enabling preventive action and better suited treatment.

One step within this vast framework of ambitious aims is to find relationships between several genetic characteristics and group them into different classes (or clusters). This structural information can be used to form new hypothesis about the interplay between genes in the disease mechanism.

A different task is the classification of potential patients into diseased and non-diseased individuals. The classification method provides information about the given data set and, in addition, can be used to predict the probable disease status of future patients. It can also provide insight into the way the genetic profile influences a specific disease and, in a second step, might help to identify genes that are associated with the disease.

There is a variety of genetic information available. In this thesis, we deal with *Single Nucleotide Polymorphisms* (SNPs). SNPs are single base exchanges within the DNA that are present in at least 1% of a population. They can be found in abundance in the human genome and can be assessed via high throughput methods. SNPs can influence body processes in different ways. For example, they might alter proteins (that are constructed by translating the genetic code that the SNPs belongs to). In the worst case, the protein cannot fulfill its assigned task anymore. E.g, if it is responsible for repairing disrupted DNA, some damages could remain unrepaired and increase the disease risks.

The association between a disease and SNPs can be illuminated with appropriate statistical methods. Together with the rapid development of technology for assessing

SNPs, the need for better suited analysis methods rose with equal speed. In an interactive fashion, mutual improvements in both areas drove and still drive the progress (LaFramboise, 2009). Despite successes and constant improvement, the challenges connected to SNP analyses are still demanding. There is a tremendously huge number of SNPs to consider (about one million for Affymetrix (Affymetrix, 2007) and Illumina (Illumina, 2009) chips, and numbers still increasing), and they are assumed to have only a moderate to small effect on the disease risk, at least if the disease of interest is common ("Common disease, common variant"-hypothesis, Risch and Merikengas (1996)). Additionally, SNPs are assumed to impact disease risk in interaction with each other rather than alone (Garte, 2001). It is also reasonable to assume that there might be different genetic profiles that lead to a similar disease risk.

These aspects of SNP analysis can be summarised by **high dimensionality**, **interaction effects** and **locality** (due to alternative ways of developing a disease). Thus, we adapt and investigate methods that can handle one or more of these challenges.

The analyses are carried out on different data sets: We design a simulation study that reflects the three characteristics of SNP data and incorporates different genetic models of association between SNP interactions and the disease. Different effect sizes of the causative SNPs allow to answer the question "Which effect size is needed to detect differences between diseased and healthy patients?" Furthermore, we analyse data from the GENICA study on sporadic breast cancer (Justenhoven et al., 2004) and a subset of the publicly available HapMap data (The International HapMap Consortium, 2007).

This thesis consists of two types of methodology: Cluster and discrimination analysis. Cluster methods divide all SNPs into several subgroups, and thus enable us to describe their relationship and investigate differences in these relationships between diseased and healthy patients. As clusterings do not rely on known class labels, it is not straight forward how to judge the quality of a partition. To solve this problem, we define sensible goals for a desirable partition and present coefficients that measure how closely these goals are met. To allow for an overall comparison between different partitions, the quality measures are combined into one single desirability index. In a different approach, we use cluster analysis on the observations and try to achieve clusters with a high fraction of either cases or controls.

In the second part, we present suitable classification methods for SNP data that we either developed or adapted from existing methods and show how they incorporate possible interactions and locality. Five methods borrow concepts from the field of data mining. The algorithm we use (called *apriori*, Agrawal et al., 1996) has been developed to handle huge amounts of data, e.g. from internet purchase data bases or spam filters.

It is used to search for *frequent itemsets*, i.e. frequent combination of variable values, and *association rules*, which give information about the likeliness to observe one item if a different itemset is known to be present. We adapt them to suit the genetic environment and translate frequent itemsets as genetic profiles, while association rules are descriptive predictions of the disease status given a certain genetic profile.

The first two methods we present use frequent genetic profiles to either build new interaction variables (*feature construction*) or to divide the data into subgroups and perform profile-specific analysis (*local class*).

The remaining three classification methods originate from associative classification (Liu et al., 1998), i.e. classification and association rules combined into one method. They are based on all interesting association rules that can be found in the data, and classify a new observation based on either the best rule (*naive classification*) or on a voting of all applicable rules (associative classification based on voting and on weighted voting). Additionally, we localise logic regression (Ruczinski et al., 2003), an existing method for analysing SNP data, and investigate the impact of the localisation. All classification methods are compared by their misclassification rates. Logic regression, CART (Breiman et al., 1984) and Random Forests (Breiman, 2001) serve as standards.

This dissertation is organised as follows: It starts with an introduction to SNPs and genetics in Chapter 2, followed by a description in Chapter 3 of the simulation study and the data sets used in the analyses. The methodological section is divided into two different chapters: The cluster analysis and the corresponding quality measures are presented in Chapter 4, while the classification methods are described in Chapter 5. All results of both the cluster and the discrimination analysis are presented in Chapter 6. In the final Chapter 7, the findings will be summarised, and an outlook to future perspective and work will be given.

CHAPTER 2

Single Nucleotide Polymorphisms and Disease

Human genetic information (like all eukaryotic genetic information) is stored in deoxyribonucleic acid (DNA) contained in the nucleus of nearly every body. DNA is a polymer that, for stability reasons, arranges as a helical intertwined double DNA strand. Each strand consists of nucleotides interconnected via phosphodiester bonds. Every nucleotide is built out of a phosphate group, a deoxyribose sugar and of one of four nitrogen bases (adenine (A), thymine (T), cytosine (C) or guanine (G)). Adenine and guanine are pyrimidines, while cytosine and thymine are purines and therefore heterobicyclic. Two kinds of bases (A and T, C and G) are complementary, meaning that they can connected and allow the two strands of DNA to be attached to each other via two (A and T) or three (G and C) hydrogen bonds (Haines and Pericak-Vance, 2006).

Single sections of the DNA build functional units (genes) that contain all necessary information for the production of a certain protein (which consists of an amino acid chain). The DNA provides the required code: three DNA bases together (triplet) form codons that encode for one of the 20 amino acids. The number of codons (4^3=64) exceeds the number of amino acids, which results in a redundancy. Thus, several different codons encode for the same amino acid, while others contain information where to start and stop reading the DNA code. According to the central dogma of molecular biology (Jorde et al., 1995), the relevant genetic code is transcribed from the DNA into mRNA (a single stranded ribonucleic acid) by splitting the double stranded helix open, reading the base order and arranging the respective complementary bases. Note that instead of thymine, mRNA consists of the similar base uracil (U). The flexible mRNA can leave the cell's nucleus and translate its information into proteins with the aid of ribosomes and tRNA molecules (cf. Figure 2.1).

Not every part of the DNA encodes for the production of proteins. The regions that do consist of exons and introns. For translating the genetic information, introns have

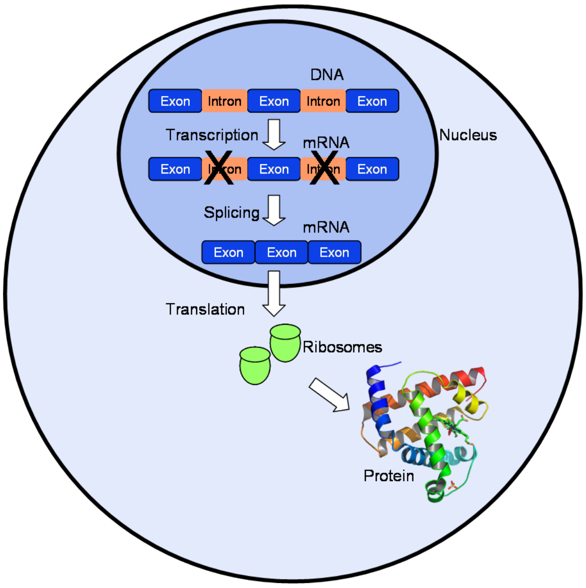

Figure 2.1: Central dogma of molecular biology: The DNA is transcribed into mRNA in the cell's nucleus. After posttranscriptional modifications, the introns are spliced out leaving a strand of exons. The flexible mRNA can leave the nucleus into the cytoplasma and translate its information into proteins with the aid of ribosomes. Source: Haines and Pericak-Vance (2006), figure adapted from Jorde et al. (1995).

to be spliced out from the gene (cf. Figure 2.1). The main proportion of DNA is not transcribed.

DNA is arranged in chromosomes. Humans have 23 pairs of chromosomes, 22 homologous pairs (one maternal and one paternal chromosome) and two gonosomes that differ between men (who inherit one X and one Y chromosome) and women (with two X chromosomes).

In the case of mutation of one base in the coding regions three different situations can occur: Because of the redundancy of the genetic code this mutation can be silent because more than one triplett encode for the specific amino acid. Secondly, the mutation results in a missense mutation which results in an exchange of the specific amino acid the triplett encodes for. The extent of this mutation depends on the substituted amino acid. A modification of an important functional domain is possible which can result a loss of function. At least, this mutation can lead to a nonsense-mutation, which means that the triplett which encodes for the specific amino acid is substituted by the triplett encoding for the stop codon. In this case the translation of this protein is terminated

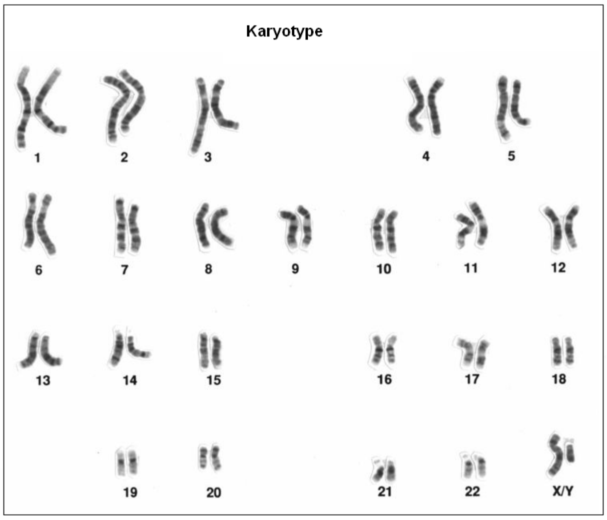

Figure 2.2: Human (male) set of chromosomes, taken from National Human Genome Research Institute (http://www.genome.gov).

misleadingly and important functional domains can be deleted. These changes in the amino acid sequence might play a role in the susceptibility for certain diseases (either malignant or beneficial changes).

If one base in the genome is substituted (in comparison to some reference), and this variation occurs in more than 1% of a population, this is called Single Nucleotide Polymorphism or SNP. These SNPs are the genetic information of interest in this thesis. For the homologous chromosome pairs, each genetic position (called locus, pl.: loci) exists twice, once on each chromosome. Therefore, a SNP can take three possible values (genotypes): Either there is no variant in comparison to some kind of reference coding (homozygous reference) or the variation occurs on one of the two chromosomes (heterozygous), or both chromosomes express the variant base (homozygous variant), cf. Figure 2.3.

SNPs cannot be measure directly, but have to be assessed indirectly and inferred from the results afterwards (called genotyping). Due to a rapid development in technology, genotyping has changed completely recently. The GENICA data (cf. Chapter 3) were measured around the year 2002, measuring one SNP at a time via time-of-flight mass spectrometry (MALDI-TOF, cf. Justenhoven et al. (2004)). Today, SNP chips (e.g., by Affymetrix) allowing to genotype up to one million SNPs simultaneously are available and widely used (LaFramboise, 2009), e.g. for the HapMap data (cf. Chapter 3).

A genetic component plays a certain part in complex human disease (Garte, 2001). SNP association studies can be aimed at finding such genetic risk factors for a binary trait like disease status. For the search, it is important to note that instead of directly causing diseases, SNPs give information about the risk of developing a particular disease. This can be shown by the following example:

The gene *apolipoprotein E* (*apoE*) is associated with the development of Alzheimer's disease (Rocca et al., 1986). *apoE* contains two SNPs which lead to the three different SNP combinations on a chromosome E2, E3, and E4. If at least one chromosome of a person expresses E4, the risk of developing Alzheimer's disease increases. On the other hand, inheriting E2 seems to be protective. However, it is by any means also possible to remain free of Alzheimer's disease even with an inherited copy of E4 while someone with the protective combination can still fall ill.

In addition to the problem that the predisposition can be present without inducing a visible phenotype, there are other challenges that complicate a SNP analysis. As, e.g., stated on the website of the Human Genome Project (2008), even though *apoE* has been successfully associated with the Alzheimer's disease risk, it is highly likely that such a complex disease is influenced by variations in several genes.

This implies problems for methods to analyse SNP data: There might be alternative ways of developing a disease (Clark et al., 2005) and it is unlikely that a single SNP alone influences the risk of developing complex diseases as Alzheimer's or cancer (Garte, 2001, Goldstein and Cavalleri, 2005).

Another challenge is the dimension of the data sets produced by high throughput methods nowadays: SNP association studies usually contain many variables, especially if interactions in addition to main effects are of interest. Multiple testing leads to a high number of false positive test results (Storey and Tibshirani, 2003) plainly because of the vast number of tests. As the data matrix is high dimensional and consists of more columns than rows, it is unsuitable for applying standard procedures, e.g., logistic regression (Hoh and Ott, 2003). Furthermore, a stepwise regression approach can be shown to be suboptimal (Rao and Wu, 2001). Therefore, methods from the field of data mining and machine learning gain a lot of attention in genomic research as they can handle a huge amount of data.

All challenges mentioned above are key words for the main topics of this thesis: **locality** (due to alternative ways of developing a disease), **interaction effects** and **high dimensionality**.

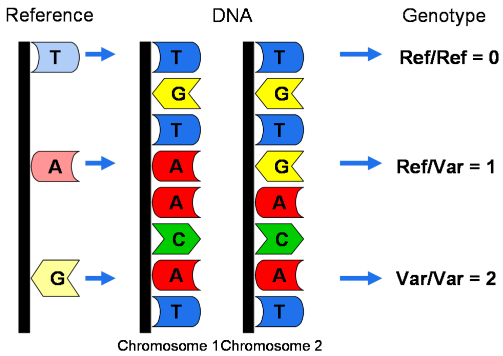

Figure 2.3: Schematic illustration of single nucleotide polymorphisms: On several loci, the allelic state of both chromosomes is compared to the reference bases and results in three possible genotypes.

2.1 Genetic Terms

Throughout this thesis we will use genetic technical terms in order to describe the data and hypothesis underlying the disease risk mechanism.

Definition 2.1. *An allele is the state of a genetic site, either of a single base or of a larger piece of DNA.*

Definition 2.2. *The minor allele frequency is the relative frequency of the less frequent variant of an allele in a population or study.*

Definition 2.3. *The penetrance of an allele is the probability of developing the disease D given the respective allele A, written as $P(D|A)$.*

Definition 2.4. *The haplotype is the allelic state on one of the two chromosomes.*

This means that if SNP values are determined, the result usually gives information about both chromosomes together (genotype). E.g., if two loci A and B from the same chromosome are investigated and both turn out to be heterozygous, this can either mean that on chromosome 1, both loci show the variant, while on chromosome 2, both show the reference base. On the other hand, it is possible that on chromosome 1, locus A shows the variant and B the reference, and vice versa on chromosome 2. If the exact position of the bases on the respective chromosome is known, it is called haplotype.

During the process of meiosis, the parents' DNA is split up to be united in the baby cell as a new genome. This never occurs without disturbance, as the DNA can break into pieces and can be put together in a different fashion (cross over, recombination). Not all loci on the genome are equally likely to break, therefore, some DNA pieces are inherited together more often than others with are frequently separated.(one reason, e.g., is the physical distance of loci). The phenomenon can be measures statistically by linkage disequilibrium.

Definition 2.5. *Linkage disequilibrium is the non-random association of alleles at two or more loci on the genome.*

CHAPTER 3

Data

Throughout the thesis SNP values will be coded with 0 (homozygous reference), 1 (heterozygous) and 2 (homozygous variant).

Our methods will be tested on different kinds of SNP data. Initially, we analyse simulated data with different genetic models underlying the disease risk. The simulation should give information about the performance of the methods on the one hand and help to define minimum detection thresholds under which even a good method is unable to find influential SNPs on the other hand.

One simulated data set is inflated to comprise 10 000 SNPs with the same SNP structure as in the smaller simulated data sets. The genetic model is also the same.

The initial real world data set, a subset of the German GENICA study (Justenhoven et al., 2004) on breast cancer, is chosen as this study triggered our research in the area of molecular epidemiology. As a publicly available set we chose the HapMap data (The International HapMap Consortium, 2003). In all real-world data sets, we avoid missing values, either by removing observations or SNPs from the study or by imputing the missing values. This is necessary to guarantee comparability of the results of different classification methods because not all methods can handle missing values.

3.1 Simulation of SNP Data Using the Software SNaP

The simulated data are created using the software SNaP (Nothnagel, 2002), which has been established in different investigations, e.g., Zhao et al. (2005), Wu et al. (2008). Its underlying algorithm is based on the theory that the genome consists of haplotype blocks that are more likely to be inherited in one piece than being split into several

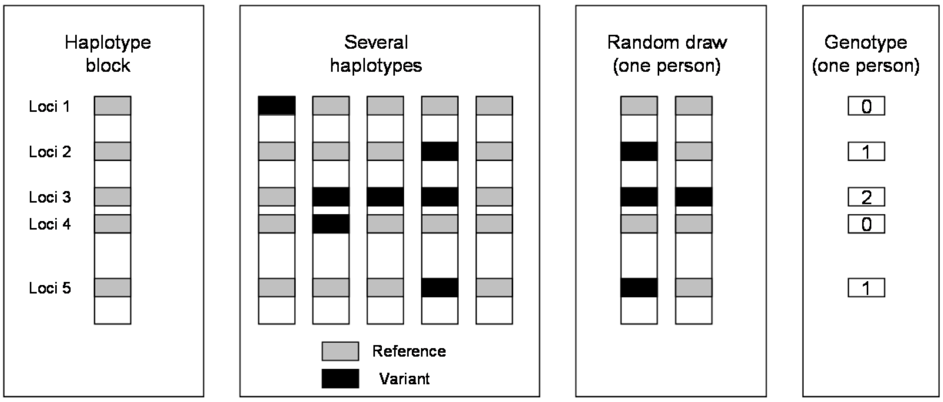

Figure 3.1: Basic idea of SNaP simulation: For a haplotype block with a specified number of loci (five in this example), several possible states are fixed. From the pool of available haplotype blocks, two are randomly drawn (with replacement) for each observation. The genotype is then inferred from the haplotypes.

parts by recombination (Gabriel et al., 2002) and cross-over. That implies a dependency structure between the loci within a block and independence between loci of different blocks.

As can be seen in Figure 3.1, for a haplotype block a given number of loci to investigate is specified. Several states of the different loci of the block are defined. For an individual, two haplotyes from the pool of available haplotype blocks are drawn with given probabilities and then combined to yield the persons's genotype.

For association studies, we simulate cases and controls. The software allows to choose causative SNPs whose values influence the probability to belong to either of the two collectives. By simulating data sets we mimic the characteristics of the GENICA study (cf. Subsection 3.2.1). Thus, all simulations consist of 100 SNPs, 500 cases, 500 controls and 20 blocks (with block sizes varying between three and eight SNPs per block). All parameter choices can be seen in Table 3.1.

The possibilities of specifying genetic interaction models in the software are limited by the maximum number of causative SNPs (not more than six causative SNPs allowed). Therefore, we investigate three different scenarios: They comprise one, two and three two-way interactions, respectively. The most important feature of the simu-

Parameter	**Value(s)**
number of cases and controls	500 and 500
number of SNPs	100
number of blocks	20
number of SNPs per each block (individually)	$\in \{3, 4, \ldots, 8\}$
number of block alleles	$\in \{3, 4, \ldots, 13\}$
frequencies of block alleles	$\in [0.01, 0.62]$
number of causative SNPs	$\in \{2, 4, 6\}$
causative SNPs	SNP4, SNP12, SNP48,
	SNP51, SNP 68, SNP82
frequencies of causative SNPs	$\in [0.22, 0.35]$

Table 3.1: Parameter settings for the simulation software SNaP.

lations are the different penetrances which determine the genetic interaction model and the strength of the disease risk. In the case of two-way interactions, according to Marchini et al. (2005a), a possible genetic interaction model involving two loci (expressed in terms of odds) can take the form described in Table 3.2. The baseline odds α denote the odds of developing the disease given that at least one SNP shows the homozygous reference (note that $P(D)$ is the probability of developing the disease and that SNP variables can take possible values 0, 1 and 2):

$$\alpha = \frac{P(D|\text{SNP A} = 0 \vee \text{SNP B} = 0)}{1 - (P(D|\text{SNP A} = 0 \vee \text{SNP B} = 0))}.$$

This transforms to the penetrance given at least one SNP is homozygous referent:

$$P(D|\text{SNP A} = 0 \vee \text{SNP B} = 0) = \frac{\alpha}{1 + \alpha}.$$

All penetrances can be found in Table 3.3. Here, θ denotes the effect size of the interaction. We assume equal effect sizes for all loci.

In order to obtain sensible parameters for the simulation study, we choose $\alpha = 0.\bar{1}$ and increase the effect size θ by steps of 0.2, starting at $\theta = 0.5$ (small effect, see Marchini et al. (2005b)) to $\theta = 1.9$.

		SNP B		
		0	1	2
SNP A	0	α	α	α
	1	α	$\alpha(1+\theta)$	$\alpha(1+\theta)^2$
	2	α	$\alpha(1+\theta)^2$	$\alpha(1+\theta)^4$

Table 3.2: Odds of developing a disease in the multiplicative genetic interaction model. The baseline odds for an effect is denoted by α, while θ is the effect size determined by the interaction. Source: Marchini et al. (2005a).

		SNP B		
		0	1	2
SNP A	0	$\frac{\alpha}{1+\alpha}$	$\frac{\alpha}{1+\alpha}$	$\frac{\alpha}{1+\alpha}$
	1	$\frac{\alpha}{1+\alpha}$	$\frac{\alpha(1+\theta)}{1+\alpha(1+\theta)}$	$\frac{\alpha(1+\theta)^2}{1+\alpha(1+\theta)^2}$
	2	$\frac{\alpha}{1+\alpha}$	$\frac{\alpha(1+\theta)^2}{1+\alpha(1+\theta)^2}$	$\frac{\alpha(1+\theta)^4}{1+\alpha(1+\theta)^4}$

Table 3.3: Penetrances of developing a disease in the multiplicative genetic interaction model. The baseline odds for an effect is denoted by α, while θ is the effect size determined by the interaction. Adapted from Marchini et al. (2005a).

The simulation process simulates genotypes first and obtains the disease status according to the penetrances chosen afterwards. Due to the setting that cases are generated if variants are present in the causative SNPs and the probability of simulating this genotype is much lower than simulating control genotypes, the study's controls are quickly generated (in our case 500), while we are still lacking many cases needed for the given study size. The simulation process continues with the difference that all additional controls as discarded and only cases are kept in the study until both collectives have reached their pre-specified size. Most of the cases' disease status will be due to a more common genotype with lower penetrance (e.g., heterozygous causative SNPs), rather than a rare genotype with higher penetrance. We will analyse all simulated data sets descriptively and show the respective number of cases belonging to each causative genotype.

Interactions of order 3 or higher might also occur in SNP data sets. Still, the theoretical justification as given for the two-way interactions is not as straight forward anymore. Thus, we set the simulation of higher interactions aside. In ad hoc simulation scenarios it could be shown that the classification methods that will be described in subsequent chapters can handle three-way interactions without problems (Müller et al., 2008).

Due to the process of drawing cases and controls during the simulation, the structure of the simulated data sets differs from the configuration of the penetrances. The empirical genotype distributions of the causative SNPs, separated into cases and controls, can be seen in Figures 3.2, 3.3 and A.1. Note that the frequencies are averaged over all 10 data sets of the same setting.

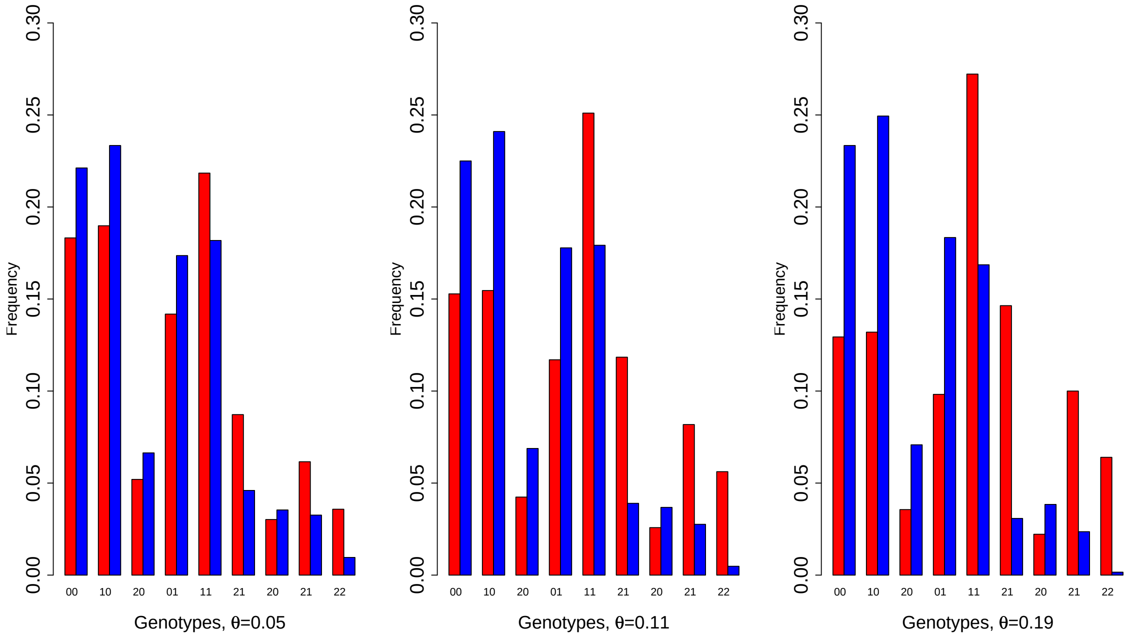

Figure 3.2: Different genotype distributions for causative SNPs between cases (red) and controls (blue) for different effect sizes, given one causative SNP interaction.

For small effect sizes, the differences between cases and controls are very small. They become more pronounced with rising effect size. However, if several interactions are associated with the disease risk, the differences between cases and controls for the single interactions diminish again. Thus, classifying observations into cases and controls will be difficult as in real world SNP data sets.

To investigate the influence of a higher SNP number with similar causation process, one simulated data set is inflated to comprise 10 000 SNPs. This is achieved by retaining the dependence structure of the different SNP blocks. In permutated order, they are added to the individuals to form a genotype of 10 000 SNPs with one causative SNP interaction with an effect size of $\theta = 1.9$. The data set will be referred to as the 10000-SNPs simulation.

3.2 Real-World Data

3.2.1 GENICA

The GENICA study on **G**enetic and **En**vironmental **I**nteractions and Sporadic Breast **Ca**ncer (Justenhoven et al. (2004), Justenhoven et al. (2008)) was designed as an age-matched population-based candidate SNP association study which aims to identify associations between genetic and environmental factors and breast cancer in women. It was carried out in two phases between 2000 and 2004 in the Greater Bonn region, Germany. All women within the study were under 80 years of age, current residents in the study area and of Caucasian ethnicity.

The actual study comprises SNP data as well as epidemiological variables such as reproductive information, occupation, medication etc., but in this thesis we will focus on the genetic information only.

Considering both phases, the GENICA study consists of about 150 polymorphisms and 2298 women. Only data from the first phase is considered in this thesis. We will analyse a subset of 63 SNP variables (belonging to genes from different pathways, the steroid and xenobiotic metabolic pathway, the cell-cycle control mechanisms, and from DNA repair mechanisms) and 1191 observations (561 cases and 630 controls) previously investigated in Ickstadt et al. (2006) and Nunkesser et al. (2007). The preprocessing ensured that all observations show less than six missing values and that all SNP variables included in the subset do not have more than 10% missing values nor have fewer than 30 patients expressing a heterozygous or homozygous variant genotype. Observations and variables that did not meet these criteria were deleted. The few missing values

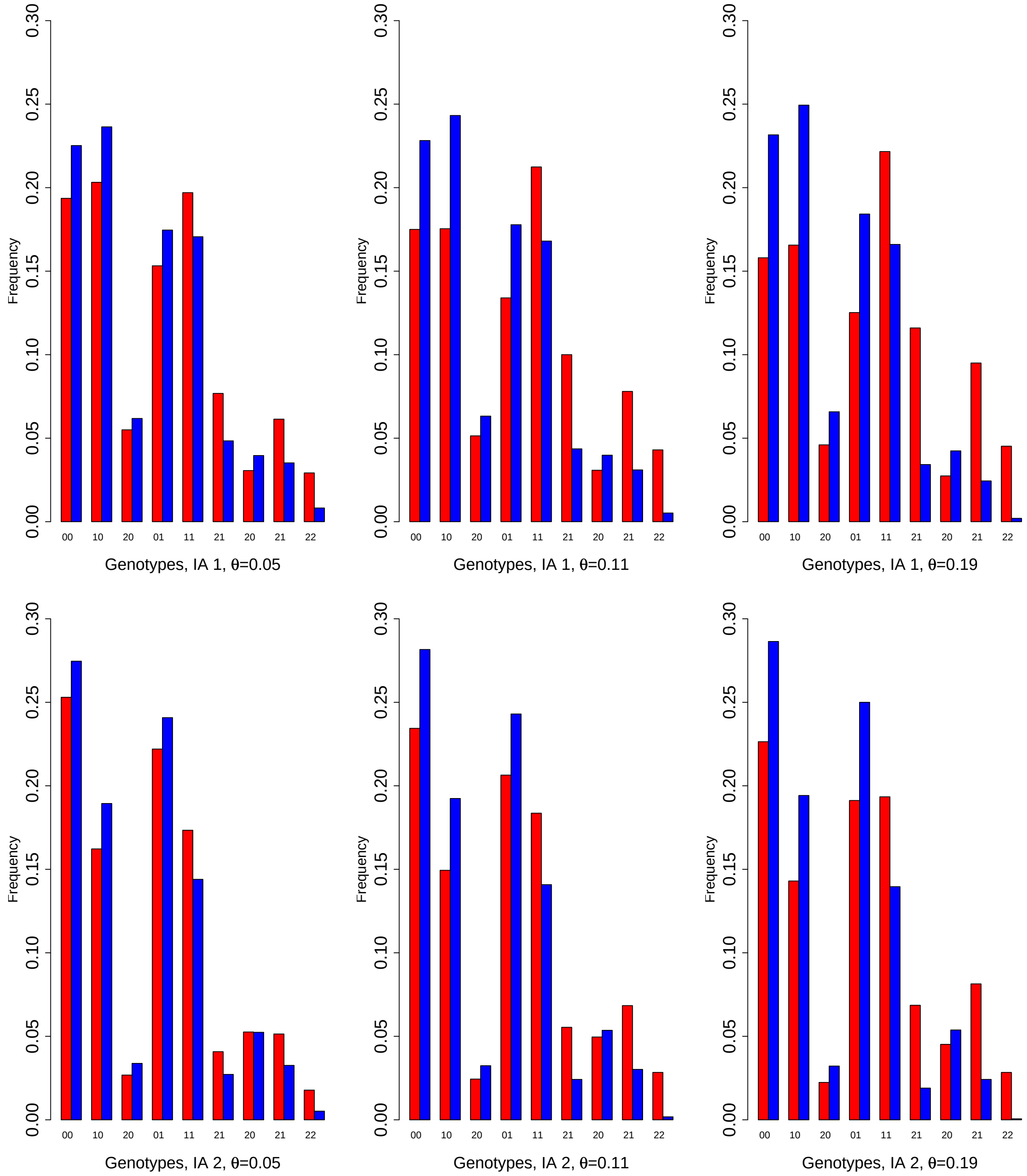

Figure 3.3: Different genotype distributions for causative SNPs between cases (red) and controls (blue) for different effect sizes, given two causative SNP interactions.

remaining in the data set are replaced SNP-wise by random draws from the marginal distribution of the respective SNP.

For evaluating the results, the limits of the study have to be taken into account: It was designed as a candidate study in the beginnings of SNP analysis, therefore, the number of variables within the study is comparatively small. Causative variants might be missing in the data and prevent results from being satisfactory.

3.2.2 HapMap

The HapMap Project (The International HapMap Consortium, 2003) was founded to catalogue SNPs and to identify haplotypes and tagSNPs representing all SNPs of a haplotype. Thus, if a researcher wants to test for an association between a disease and SNPs of a genome-wide scan, the number of tests is reduced if he refers to the tagSNPs only as they contain all information necessary.

Up to 2007, the HapMap Consortium genotyped about 3.1 million SNPs (in two phases) in 270 persons of four different ethnicities (Yoruba in Ibadan (Nigeria), Japanese in Tokyo (Japan), Han Chinese in Beijing (China) and Utah residents with ancestry from northern and western Europe (USA)) without assessing any information about diseases or other traits. Thus, the label for HapMap data is ethnicity. In contrast to the other data sets, the natural relationship between the genetic profile and the outcome is rather deterministic in this example if the respective SNPs are part of the data. However, the analytic interest lies in the data structure which can be compared to that of association studies with cases and controls.

We will analyse a subset of the HapMap data. The SNPs measured with the Affymetrix GeneChip Mapping 500K Array Set (in particular, the SNPs from the Nsp array) and assessed as BRLMM genotypes (Bayesian Robust Linear Model with Mahalanobis distance, Affymetrix (2006)). The data subset is additionally reduced by the following criteria the SNPs have to meet:

1. The SNPs show no missing genotypes for the two ethnicities (omitting (54 400 SNPs).
2. All three possible genotypes are observed (omitting 75 481 SNPs).
3. Their minor allele frequency yields > 0.1 (omitting 10 609 SNPs).

4. The false discovery rate of chosen subset (by Significance Analysis of Microarrays (Tusher et al., 2001) adapted for categorical data (Schwender, 2003)) yields 0.069 (omitting 121 617 SNPs).

The remaining SNP subset comprises 157 SNPs. We chose the two collectives of unrelated Japanese from Tokyo and unrelated Han Chinese from Bejing (45 observations per class) to create a two class problem similar to an association study.
All discrimination methods described in Chapter 5 are then applied to this subset of SNPs, and the misclassification is estimated by 9-fold crossvalidation, where each of the nine subsets is composed of five randomly chosen Han Chinese and five randomly chosen Japanese participants.

CHAPTER 4

Clustering

Cluster analysis is a multipurpose method for unsupervised learning (Hastie et al., 2001). Its main application is to divide a set of variables or observations into subgroups or clusters. As a side effect, the degree of similarity (or distance) between the clustered objects is calculated. It can also be used to check if distinct subgroups in the data sets do exist.

In this thesis, we will use average linkage, a hierarchical agglomerative algorithm described in Subsection 4.1. Average linkage is based on a similarity matrix containing all pairwise similarities between all cluster objects of interest. There are many possibilities to define similarity, but we will focus on matching coefficients, a specific class of similarity measures, and on Pearson's corrected contingency coefficient, all introduced in Section 4.1 . Throughout the first part of this chapter, the cluster objects will be SNP variables, and partitions should give information about genetic structure and relationships between SNPs.

The most relevant part of the cluster section is given in Section 4.3: Validation coefficients for cluster partitions. For the specific situation of genetic SNP data (cf. Section 4.2), certain characteristics for a good clustering are required. We incorporate them into four goodness-of-fit measures which are defined and motivated in this section. Furthermore, all four coefficients are combined to a single score (or *desirability index*) to allow for an unambiguous comparison of clustering partitions.

In the final section, the cluster objects will be the observations instead of the variables. Thus, the cluster analysis now aims at finding distinctive clusters of cases and clusters of controls. We describe a suitable cluster algorithm that clusters observations on an optimal set of SNP variables (due to an integrated variable selection).

4.1 Similarity and Cluster Algorithm

Similarity and Distance

Let the data consist of n objects and m variables, with $V = \{V_1, \dots, V_m\}$ describing the set of variables. For now, we are interested in clustering the variables, not observations. To define a notion of similarity between elements of V, we introduce a function $S : V \times V \to \mathbb{R}$ called similarity measure (Jain and Dubes, 1988). It should yield higher values for cluster objects that are more similar to each other than for objects less similar and meet at least the first three of the following requirements (with $i, j = 1, \dots, m$):

1. $S(V_i, V_j) = S(V_j, V_i)$, symmetry,
2. $S(V_i, V_j) \leq S(V_i, V_i)$, natural order,
3. $S(V_i, V_j) \geq 0$, positivity,
4. $S(V_i, V_i) = 1$, normality.

Assumptions 3 and 4 are often useful, though not necessary for $S(\cdot)$ for being a similarity measure.
In most practical investigations, distances rather than similarities are of interest. For categorical data in general and SNP data in particular, it is common practice to compute the similarity $S(\cdot)$ first and then transform it into a distance $D(\cdot)$ (Cox and Cox, 2001). Large similarities correspond to small distances and vice versa. Therefore, we use the following transformation if $S(\cdot) \in [0, 1]$:

$$D(V_k, V_l) = 1 - S(V_k, V_l), \qquad \forall\, V_k, V_l \in V. \tag{4.1}$$

If $[\min(S(\cdot)), \max(S(\cdot))] \not\subset [0, 1]$, we apply

$$D(V_k, V_l) = \begin{cases} 1 - \frac{S(V_k, V_l)}{|\max S(V_i, V_j)|} & \text{, if } \min S(V_i, V_j) \geq 0 \\ 1 - \frac{S(V_k, V_l) + |\min S(V_i, V_j)|}{|\max S(V_i, V_j)| + |\min S(V_i, V_j)|} & \text{, else,} \end{cases}$$

$\forall\, V_k, V_l \in V$, and $i, j = 1, \dots, m$. The resulting distance satisfies $D(\cdot) \in [0, 1]$ (Fahrmeir et al., 1996).

		V_l			
		0	1	2	
	0	m_{00}	m_{01}	m_{02}	
V_k	1	m_{10}	m_{11}	m_{12}	
	2	m_{20}	m_{21}	m_{22}	
					n

Table 4.1: 3×3 - contingency table for matching coefficients in the case of SNP data

The choice of a particular similarity measure S depends on the data structure. A suitable option for SNP data are matching coefficients (Anderberg (1973) and Cox and Cox (2001)). Based on the contingency table of two variables V_k and V_l with three levels each, resulting into nine categories (Table 4.1), matching coefficients relate the number of matching objects (given in the *matching categories*) to the remaining objects and thus evaluate the similarity between V_k and V_l. Classically, the n_+ categories on the diagonal of the contingency table are considered as matching categories $\mathbf{m}^+ := (m_{00}, m_{11}, m_{22})'$, while all other classes build up the n_- *mismatching categories* $\mathbf{m}^- := (m_{01}, m_{02}, m_{10}, m_{12}, m_{20}, m_{21})'$. With $\mathbb{1}_b$ a b-dimensional vector of ones, the ordinary simple matching coefficient SMC, giving the fraction of matches compared to the total number of observations, can be written as:

$$SMC(V_k, V_l) = \frac{\mathbb{1}'_{n_+} \cdot \mathbf{m}^+}{\mathbb{1}'_{n_+} \cdot \mathbf{m}^+ + \mathbb{1}'_{n_-} \cdot \mathbf{m}^-}.$$

The SMC can be generalised in two ways: Initially, the division into matching and mismatching categories can be relaxed. All entries of the contingency table can be labeled user-specified as either matching or mismatching, regardless of their position within the table. Furthermore, the user can assign individual weights $\mathbf{w}_F^+$ (for matches) or $\mathbf{w}_F^-$ (for mismatches), respectively, to all categories. The resulting flexible matching coefficient reflectcs the characteristics of SNP data described in Section 4.2.

Definition 4.1. *Let $V = \{V_1, \dots, V_m\}$ be the set of variables, let $\mathbf{m}_F^+$ ($\mathbf{m}_F^-$) be the vector of the numbers of elements in the n_+ matching (the n_- mismatching) categories and let $\mathbf{w}_F^+$ ($\mathbf{w}_F^-$) be respective weights that reflect the matches (mismatches) importance. Then the flexible matching* *coefficient $FMC_{\mathbf{w}_F}$: $V \times V \to \mathbb{R}$ is given by*

$$FMC_{\mathbf{w}_F}(V_k, V_l) := \frac{\mathbf{w}_F^{+\prime} \cdot \mathbf{m}_F^{+}}{\mathbf{w}_F^{+\prime} \cdot \mathbf{m}_F^{+} + \mathbf{w}_F^{-\prime} \cdot \mathbf{m}_F^{-}}.$$

The following restrictions apply:

1. $\mathbb{1}'_{n_+} \cdot \mathbf{w}_F^{+} > 0$ and $\mathbb{1}'_{n_-} \cdot \mathbf{w}_F^{-} > 0$.

2. $\mathbf{e}'_i \cdot \mathbf{w}_F^{+} \geq 0 \, \forall \, i$, $\mathbf{e}'_j \cdot \mathbf{w}_F^{-} \geq 0 \, \forall \, j$, with $\mathbf{e}_i$, $\mathbf{e}_j$ being the unity vectors of respective order with value 1 at position i and j, respectively, with $i \in \{1, \ldots, n_+\}$ and $j \in \{1, \ldots, n_-\}$.

3. $\mathbf{w}_F^{+\prime} \cdot \mathbf{m}_F^{+} + \mathbf{w}_F^{-\prime} \cdot \mathbf{m}_F^{-} > 0$.

The distinction into matches and mismatches as well as the specification of the weights can be chosen to fit best by the user. The properties of the FMC are given in Selinski and Ickstadt (2005). Müller (2004), Ickstadt et al. (2006) and Selinski and Ickstadt (2008) show that the clustering performance on SNP data can be improved by employing FMCs. Additionally, conventional matching coefficients can be formulated as special cases of FMCs, e.g. Jaccard's coefficient (Anderberg, 1973). It is defined as all matching objects except if they share a null category (m_{00}) divided by all objects excluding the objects of the null category. In our notation, this yields an FMC with $\mathbf{w}_F^{+\prime} = (0, 1, 1)$, $\mathbf{m}_F^{+} = (m_{00}, m_{11}, m_{22})$, $\mathbf{w}_F^{-\prime} = (1, 1, 1, 1, 1, 1)$ and $\mathbf{m}_F^{-} = (m_{01}, m_{02}, m_{10}, m_{12}, m_{20}, m_{21})$.
A different idea of similarity is introduced by using the χ^2-coefficient and yielding Pearson's corrected contingency coefficient

$$PCC = \sqrt{\frac{3}{2}\frac{\chi^2}{m + \chi^2}}. \tag{4.2}$$

Here, variables are considered similar if they are dependent.

Cluster Algorithm

Based on the similarities of cluster objects, a cluster algorithm divides the data into different clusters. Agglomerative hierarchical clustering methods build a data partition for every number of classes $m, \ldots, 1$, starting with m clusters (each object forming its own cluster) and finishing with all objects fused into a single cluster. All fusions are based on the similarity matrix obtained by calculating pairwise similarities. Following the notation $S(\cdot)$, we call $\bar{S}(\cdot)$ the similarity measure for clusters. The average linkage cluster method computes the similarity between two clusters $\mathbf{C}_t$ and $\mathbf{C}_r$ with $m_{\mathbf{C}_t}$ and $m_{\mathbf{C}_r}$ elements, respectively, as follows:

$$\bar{S}(\mathbf{C}_t, \mathbf{C}_r) = \frac{1}{m_{\mathbf{C}_t} m_{\mathbf{C}_r}} \sum_{V_i \in \mathbf{C}_t} \sum_{V_j \in \mathbf{C}_r} S(V_i, V_j), \qquad \begin{array}{l} i = 1, \ldots, m_{\mathbf{C}_t}, \\ \\ j = 1, \ldots, m_{\mathbf{C}_r}. \end{array}$$

Given the initial situation that every variable is regarded as a cluster with only one element, the algorithm works as follows:

1. Fuse the two clusters with the highest similarity.
2. Recompute the similarity for the newly established group to all remaining clusters.
3. Iterate steps 1 and 2 until all variables form a single cluster.

Number of Clusters

As there are specific data partitions for m to 1 clusters, a certain cluster number choice is not necessary in advance to start the algorithm. Nevertheless, to receive interpretable results or compare and evaluate different partitions, a best number of clusters has to be determined. A choice could be made graphically by regarding the resulting dendrogram and assessing the most relevant increase in overall distance between clusters. However, that quickly becomes too time-consuming when analysing many partitions. Alternatively, background knowledge can determine a number of clusters or a range of sensible numbers of clusters.

For the simulated data, we want to find the best partition according to the desirability index (cf. Subsection 4.3). We use one data set of the same configuration as training set for obtaining the optimal number of clusters and employ the results on a respective test data set to calculate the desirability index. For the real world data, we will consider reasonable assumptions and the results from the simulated data.

4.2 Application to Genetic SNP Data

The flexible matching coefficients described above meet the data characteristics of SNPs as they are suited for categorical data and they give possibilities to down-weight the less informative (but generally more abundant) category of homozygous references, while stressing the importance of matching variants (for details, see Technical Report Müller et al. (2005)). Pearson's corrected contingency coefficient can also be applied to categorical data.

We use cluster analysis for a first insight into genetic SNP data. All SNP variables should be clustered separately for cases and controls to allow for differences between the genetic profiles to be visible. Ideally, similar SNPs cluster in the same group, while diverse variables are assigned to different clusters. Several assumptions concerning the emerging partitions can be made:

1. SNPs in high linkage disequilibrium (cf. Subsection 2.1) might be grouped in one cluster.
2. SNPs with similar or linked biological functions might be grouped in one cluster.
3. SNPs responsible for disease risk might be clustered differently in the case and in the control collective.

For the simulated SNP data, the notion of linkage disequilibrium (Goal 1) is expressed in the blocking structure of the SNP variables (cf. Chapter 3). Several blocks with different numbers of SNPs are used to generate genotypes in a way that SNPs within a block are correlated, while SNPs of different blocks are assumed to be independent. Therefore, the blocks are used to investigate Goal 1 in the following description of cluster validation.

4.3 Cluster Validation

As there are numerous possibilities to choose a cluster method and a similarity measure or weights for a particular similarity measure, a means of evaluating and comparing different clusterings is desirable. Existing validation methods can be grouped into three different types (Halkidi et al., 2001): External (comparing the resulting partition with

a standard partition given beforehand, internal (using the data the partition was built of itself) and relative criterion (evaluating partitions from the same algorithm, but with different parameters). The introduced measures f_1, f_2, f_3 and f_4 will all be of the third type, but f_1, f_3 and f_4 also rely on external information.
According to the aspired assumptions of the last section, a good partition should fulfill the following goals:

1. **Detecting linkage disequilibrium (or the underlying blocking structure) of the variables** (cf. assumption 1. in Section 4.2).

2. **Avoiding clusters with only one element**. Uninformative partitions tend to add each variable separately to one big cluster, leaving most other clusters with just one element. It is more desirable to obtain a clustering with several distinctive clusters consisting of a couple of SNPs each.

3. **Help to distinguish between causative and non-causative SNPs in the case, but not in the control collective**. If a causative SNP combination influences the risk of developing a disease, it is likely that all involved SNPs express similar genotypes in the case collective. On the other hand, they should not show any pattern in the controls as they do not have an effect in these observations. SNPs clustering in both collectives seem to be in linkage disequilibrium rather than to be responsible for the disease under investigation.

We have designed quality measures for each of these goals (Selinski and Ickstadt (2008), Müller (2004)). For ease of understanding, note that the letter m with some meaningful index corresponds to numbers of variables and capital calligraphy letters refer to sets of variables.
The first goal corresponds to measuring the linkage disequilibrium between two loci. Usually, a common measure like D′ or r^2 (Lewontin (1964), Devlin and Risch (1995), Hill and Robertson (1968)) would be used to assess linkage disequilibrium. However, as our goal is not to compute linkage disequilibrium, but to measure properties of the similarity coefficients used, we calculate f_1 as an indicator for an existing blocking structure (as used in the simulated data):

Definition 4.2. *For a data partition with K clusters* $\mathbf{C}_k$ *and B linkage disequilibrium blocks underlying the SNP structure (cf. Chapter 3), the number of SNPs of block b in cluster k is given by* $m_{\mathbf{C}_k,b}$, $b = 1, \ldots, B$ *and* $k = 1, \ldots, K$, *while the number of elements in cluster k is* *given by $m_{\mathbf{C}_k}$. Then the average proportion of SNPs of the same block in a common cluster is calculated by*

$$f_1 := \frac{1}{K}\sum_{k=1}^{K}\frac{m^*_{\mathbf{C}_k}}{m_{\mathbf{C}_k}}, \quad m^*_{\mathbf{C}_k} := \begin{cases} \max_{b=1,\ldots,B} m_{\mathbf{C}_k,b} & \text{, if } m_{\mathbf{C}_k} > 1, \\ 0 & \text{, if } m_{\mathbf{C}_k} = 1, \end{cases}$$

with $b = 1, \ldots, B, k = 1, \ldots, K$.

Some properties of f_1 are that $f_1 \in [0, 1]$ with $f_1 = 0$ if no SNPs of the same block share a cluster and $f_1 = 1$ if all linkage disequilibrium blocks form their own clusters. This means that high values of f_1 correspond to a good clustering according to goal 1.

The second goal, avoiding clusters with just one element, is examined by f_2.

Definition 4.3. *Given a data partition with K clusters $\mathbf{C}_k$, the proportion of the K_1 clusters with just one element is calculated by*

$$f_2 := 1 - \frac{K_1}{K}.$$

Again, it is desirable to yield high values of $f_2 \in [0, 1]$.

Both f_1 and f_2 should be calculated for cases and controls separately. Thus, in order to obtain a single measure for the clustering of a data set, the two values have to be combined. This can be done, e.g., by building the arithmetic mean, or by taking two values per measure into the desirability index (see following subsection).

To test the performance of a clustering concerning the third goal, we need to split its different aspects into two measures. They should investigate if the clusterings of the two collectives differ with respect to the causative SNPs (f_3) and if they resemble each other in clustering the non-causative SNPs (f_4).

Definition 4.4. *Two separate data partitions are given for the case and for the control collective. Let m_c be the total number of causative SNPs, which are arbitrarily labeled from 1 to m_c. They are distributed over different clusters in the case and in the control collective, with k_i indicating the index of the cluster containing causative SNP i, $i = 1, \ldots, m_c$. $\mathcal{C}_{ca,k_i}$ denotes the set of*

SNPs in cluster k_i for the cases and $\mathcal{C}_{con,k_i}$ for the controls, respectively. The united set of SNPs from clusters k_i in both partitions is called $U_{k_i} = \mathcal{C}_{ca,k_i} \cup \mathcal{C}_{con,k_i}$ and contains $m_{U_{k_i}}$ elements. Equivalently, the intersecting set of SNPs belonging to cluster k_i in both partitions is called $I_{k_i} = \mathcal{C}_{ca,k_i} \cap \mathcal{C}_{con,k_i}$ with $m_{I_{k_i}}$ being the number of SNPs within I_{k_i}. The measure

$$f_3 := \frac{1}{m_c} \sum_{i=1}^{m_c} \frac{m_{U_{k_i}} - m_{I_{k_i}}}{m_{U_{k_i}}}$$

gives the mean fraction of SNPs (over all clusters $k_i, i = 1, \ldots, m_c$) present in only one of the two corresponding clusters of both partitions containing a causative SNP.

In contrast to showing different clusterings for the causative SNPs, we need to ensure that the two partitions do not differ considerably in the clustering of the non-causative SNPs. It is not detectable straight forward, as clusters are labeled arbitrarily and not consistently over different partitions. Therefore, we search for the clusters with a maximum agreement of shared non-causative SNPs between the two clusterings.

Definition 4.5. *Two separate data partitions are given for the case and for the control collective. Let $m^{nc}_{ca,k'}$ be the number of non-causative SNPs in cluster k' in the case collective $\mathcal{C}_{ca,k'}$ and $m^{nc}_{con,k}$ the number of SNPs in cluster k of the control collective $\mathcal{C}_{con,k}$, respectively. With $m^{nc}_{U_{k,k'}}$ as the number of non-causative SNPs in the union $U_{k,k'} = \mathcal{C}_{ca,k'} \cup \mathcal{C}_{con,k}$, the maximum number of common variables in clusters k and k' is given by*

$$ma^{nc}_k := \begin{cases} \max_{k'} \frac{m^{nc}_{U_{k,k'}}}{\min\{m^{nc}_{ca,k'}, m^{nc}_{con,k}\}} & \text{, if } \min\{m^{nc}_{ca,k'}, m^{nc}_{con,k}\} > 1 \\ 0, & \text{else.} \end{cases}$$

With ma^{nc}_k,

$$f_4 := \frac{1}{K_+} \sum_{k=1}^{K_+} ma^{nc}_k,$$

gives the mean number of agreeing SNPs between maximally similar clusters of the two clusterings (with K_+ being the number of cluster pairs where both clusters contain more than one non-causative SNP).

For both measures, $f_3 \in [0,1]$, $f_4 \in [0,1]$, with high values corresponding to a better partition.

There might be partitions which are good in terms of one of these measures, but yield inferior values for another. For a valid overall comparison, we can employ a desirability index (Harrington, 1965) which is built up by the values of all four validation measures, but gives only one score for the total performance of a clustering (Neumann, 2007). A short introduction to the theory of desirabilities and desirability indices will be given in the next section.

Desirability Index

If different quality aspects have to be assessed simultaneously, it is effective to condense the quality values from different measures into a single score value. Useful methods for this purpose are desirability functions and desirability indices (Steuer (2005), Harrington (1965), Trautmann and Weihs (2006)) and come from the field of product quality management.

Definition 4.6. *Let Y_i be a quality measure with real-valued realisations y_i, $i = 1, \dots, z$. A function*

$$\begin{aligned} d : \mathbb{R} &\rightarrow [0,1] \\ y_i &\mapsto d(y_i) \end{aligned} \tag{4.3}$$

*is called **desirability function**. Higher values of d indicate better performance in terms of the desired quality. If $d(y_i) = 0$, the desirability is unacceptable, while a value of 1 indicates that further improvement will be unnecessary.*

Harrington (1965) proposes a certain family of functions for d which ensure that the desirabilities are scale-free. In our current situation, f_1, f_2, f_3 and f_4 take values within the interval $[0, 1]$ already. Even though they are not necessarily scale-free, we treat them as desirabilities from now on (cf. Neumann (2007)).
They can be condensed into a desirability index (Steuer, 2005).

Definition 4.7. *Let $d_1, \dots, d_z$ be z desirability functions belonging to the quality components $y = (y_1, \dots, y_z)$ of $Y_i, \dots Y_z$, respectively. The function*

$$\begin{aligned} q : [0,1]^z &\rightarrow [0,1] \\ (d_1(y_1), \dots, d_z(y_z))' &\mapsto q(y) \end{aligned} \tag{4.4}$$

is called ***strong desirability index****, if the following monotonicity characteristic holds for two different measurements $y_{1,i}$ and $y_{2,i}$, $i = 1, \dots, z$:*

$$d_i(y_{1,i}) \geq d_i(y_{2,i}) \forall i \in \{1, \dots, z\} \text{ and } \exists\, i \in \{1, \dots, z\} \text{ with } d_i(y_{1,i}) > d_i(y_{2,i}) \Rightarrow q(y_1) > q(y_2).$$

If $>$ is replaced by $\geq$ the former inequality, then the index is be called ***weak desirability index****.*

Harrington (1965) suggests to use the geometric mean of the desirability functions as the desirability index. As a reason for this choice, he states that this index turns zero if at least one desirability function takes the unacceptable value zero. Thus, no desirability function is neglected.

Definition 4.8. *Let $d_1, \dots, d_z$ be z desirability functions belonging to the quality components $y = (y_1, \dots, y_z)$. The function*

$$q^{Harr} := \left(\prod_{i=1}^{z} d_i(y_i) \right)^{1/z} \tag{4.5}$$

is called ***Harrington's desirability index****.*

In the following we base our analyses on this choice as our desirability index as all quality measures (desirability functions) should yield values sufficiently different from zero. Further possibilities for desirability functions and indices are given in, e.g., Steuer (2005) and Trautmann and Weihs (2006).

4.4 Clustering Objects - Variable Selection and Classification

Instead of clustering variables, a different approach is to cluster the observations into distinct clusters according to their genetic profile. Note that the number of objects to be clustered is now n instead of m.

In an idealised setting, the resulting clusters would be pure and contain only observations of the same class. Leaving the field of unsupervised learning, we could use the class information, label the clusters accordingly and classify future observations by assigning them to a cluster and let them inherit the class label.

Pure clusters rarely evolve in real data, therefore we label a cluster $\mathbf{C}_k$ "case" if the majority of all observations within the cluster are cases. A possible validation of this kind of classification clustering is the average over the *purity* (Xinhua et al., 2006) of the clusters $\mathbf{C}_k$, $k = 1, \ldots, K$. Given that $n_{\mathbf{C}_k}$ is the number of observations in cluster $\mathbf{C}_k$, with $n_{\mathbf{C}_k,ca}$ and $n_{\mathbf{C}_k,con}$ being the respective number of cases and controls in $\mathbf{C}_k$, the purity takes the form

$$pur(\mathbf{C}_k) := \frac{\max\{n_{\mathbf{C}_k,ca},\, n_{\mathbf{C}_k,con}\}}{n_{\mathbf{C}_k}}.$$

Clusters with just one element do not yield a sensible statement and are therefore neglected. That leaves a total of K^{-1} clusters. Purity needs the information about the class labels and is therefore an external quality measure. Other possible measures are, e.g., entropy (measuring the uncertainty) and the overall F-measure that quantifies the relationship between recall and precision of objects assigned to clusters (Steinbach et al. (2000), Larsen and Aone (1999)). We choose purity because if can be seen as a natural estimator of the misclassification rate (cf. Chapter 5) and is therefore consistent with the evaluation of the classification methods in the next chapter.

As irrelevant variables can blur actual effects, we improve the purity of clusters by choosing a subset V_{opt} of m_{opt} variables that best explains cases and controls by means of algorithm 1 (backward selection). Note that $\mathbf{C}_k|V/\{V_i\}$ refers to cluster $\mathbf{C}_k$ formed by a clustering based on all variables except variable V_i.

Algorithm 1 Backward Variable Selection by Means of Clustering

```
$V^1 = argmax_{\{V_i \in V\}} \sum_{k=1}^{K-1} \frac{n_{\mathbf{C}_k}}{n} pur(\mathbf{C}_k|V/\{V_i\})$
while $j \leq (m-1)$ do
  $V^j = argmax_{\{V_i \in V\}} \sum_{k=1}^{K-1} \frac{n_{\mathbf{C}_k}}{n} pur(\mathbf{C}_k|V/\{V_i, V^1, \ldots V^{j-1}\})$
  if $\sum_{k=1}^{K-1} \frac{n_{\mathbf{C}_k}}{n} pur(\mathbf{C}_k|V/\{V^1, \ldots V^j\}) < \sum_{k=1}^{K-1} \frac{n_{\mathbf{C}_k}}{n} pur(\mathbf{C}_k|V/\{V^1, \ldots V^{j-1}\})$ then
    Stop; set $m_{opt}^{-} = j-1$ and $V_{opt} = V/\{V^1, \ldots V^{m_{opt}^{-}}\}$
  else
    Iterate until $j = (m-1)$
  end if
  Set $m_{opt}^{-} = m$ and $V_{opt} = V\{V^1, \ldots V^{m_{opt}^{-}}\}$
end while
```

CHAPTER 5

Classification

The last part of Chapter 4 has already built a bridge from clustering to classification because the achieved clustering is used for both labelling observations with known class labels and selecting variables that are important for a good classification.
In this chapter, we focus on different classification approaches and their success at classifying patients into cases and controls according to their genetic profile. We will pay special attention to the existence of influential SNP interactions and of several local mechanisms that alter the disease risk for different subgroups of patients.
The first kind of classifiers will be based on the theory of frequent itemsets and association rules. After giving all necessary background on the topic, we introduce two classification methods based on frequent itemsets, namely *feature construction* and *local class*. More sophisticated methods from the field of *associative classification* (in which the classification is based on association rules) are described in Subsection 5.3. Finally, in Section 5.4 we present a localised version of *logic regression* (Ruczinski et al., 2003), a tree-based method that was especially designed for SNP data, to classify new patients.

The main task in any classification scenario consists of finding a classifier $C(\cdot)$ that assigns the correct class label y_i to a given observation based on one or more predictors $\mathbf{x}$. Ideally, the classifier (or model) is built on one data set, while its quality is evaluated on a separate data set. In real life, circumstances usually leave us with only one data set to be used for both the model fitting and the model evaluation. Depending on the sample size which determines the feasibility, we will either employ cross validation (for smaller data sets) or a partition into training and test data (for larger data sets) to prevent overfitting. In the following description, we will concentrate on the test and training data approach. Note that if an optimisation procedure is part of building a classifier, the optimisation has to be carried out on a subset of the training data that is treated as test data within the training data.

Throughout the analyses, logic regression (Ruczinski et al. (2003), cf. Chapter 5.4) as a regression method specifically designed for SNP analysis as well as CART (Breiman et al., 1984) and Random Forests (Breiman, 2001) as standard classification procedures will be used as benchmark classification methods.

The following notation will be used throughout this chapter until stated otherwise. The response for a certain observation $i, i = 1, \ldots, n$, is given by y_i, with corresponding covariates $\mathbf{x}_i = (x_{i1}, \ldots, x_{im})'$. A variable considered for all observations is denoted by x_j, $(x_{1j}, \ldots, x_{n,j})'$ $j = 1, \ldots, m$. Even though it is a vector, it is written in normal script to distinguish it from $\mathbf{x}_i$. The complete data can be written as

$$\mathbf{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_i \\ \vdots \\ y_n \end{pmatrix}, \mathbf{X} = \begin{pmatrix} x_{11} & \ldots & x_{1m} \\ \vdots & \ddots & \vdots \\ x_{i1} & \ldots & x_{im} \\ \vdots & \ddots & \vdots \\ x_{n1} & \ldots & x_{nm} \end{pmatrix}.$$

For SNP data analysis, the response variable is binary with outcomes 1 (= case) and 0 (= control), while the covariates are categorical with three possible values 0, 1 and 2.

If a classifier yields good results, only few observations should be assigned to the wrong class which can be assessed by the misclassification rate MCR. Assume that we have built a classifier on the training data $(\mathbf{y}, \mathbf{X})$ with m predictors and n_{Tr} observations. Furthermore, we call $\hat{y}_i$, $i = 1, \ldots, n_{Te}$ the respective predictions for the test observations with $\hat{y}_i \in \{0, 1\}$. The proportion of misclassified observations is given by

$$MCR = \frac{\sum_i^{n_{Te}} |\hat{y}_i - y_i|}{n_{Te}}.$$

For convenience, we write n_{miss} instead of $\sum_i^{n_{Te}} |\hat{y}_i - y_i|$.

5.1 Frequent Itemsets and Association Rules

To introduce the theory of frequent itemsets and association rules, we start with an example from their field of origin: market basket analysis.
Consider a supermarket that provides several thousand (total number = $m_{\mathcal{I}}$) items I_j, $j = 1, \ldots, m_{\mathcal{I}}$, all items forming the set of items $\mathcal{I}$. When shopping, a customer buys a specific set of items. The corresponding *transaction* T_i is a binary vector of length $m_{\mathcal{I}}$ with entry 1 at all positions corresponding to the items the customer bought and 0 otherwise. All transactions $T_i \subseteq \mathcal{I}$, $i = 1, \cdots, n$, build the data set $\mathcal{D}$ (cf. Table 5.1).

	Item I_1	Item I_2	Item I_3	Item I_4
transaction T_1	0	1	1	0
transaction T_2	0	0	1	0
transaction T_3	1	1	1	1

Table 5.1: Example of a data set $\mathcal{D}$ consisting of four items and three transactions. The customer corresponding to transaction T_1 bought items I_2 and I_3.

Given $\mathcal{D}$, the supermarket manager seeks knowledge about the behaviour of his customers to react accordingly and maximise profit. If he knew which combinations of items are frequently bought together (*frequent itemsets*), he could, e.g., place the shelves containing these items adjacent to each other to increase the number of customers buying both items instead of just one (Borgelt and Kruse, 2002).
We extend the concept of frequent itemsets to *association rules*. They are of the form $B \rightarrow H$, with B and H being itemsets ($B, H \subset \mathcal{I}$, $B \cap H = \emptyset$). B is called *antecedent* or *body*, while H is the *consequent* or *head*. An association rule gives information about the occurrence of H if B is known to be part of the transaction. A common example taken from the supermarket environment says that "people who buy bread and milk are likely to buy butter as well". Here, bread and milk build the antecedent and butter is the consequent. Only rules that occur with a given frequency and that are true for a given percentage are of interest. These two concepts and a search algorithm will be

introduced in the following subsection.

Beforehand, we explain the relationship of SNP data $(\mathbf{y}, \mathbf{X})$ and transactions and items. All covariates $x_j, j = 1, \dots m$ are categorical with range $\{0, 1, 2\}$. For frequent itemsets, we need a binary data base with a 0-1-variable for each possible genotype at each possible locus. This is achieved by coding each variable x_j into three binary dummy variables, now called items:

$$x_j \in \{0,1,2\} \Rightarrow \begin{array}{l} x_{j.0} \in \{0,1\} = I_{3 \cdot j-2} \\ x_{j.1} \in \{0,1\} = I_{3 \cdot j-1} \\ x_{j.2} \in \{0,1\} = I_{3 \cdot j}, \, j = 1, \dots, m. \end{array}$$

Note that different dummy codings, especially if only two dummy variables are involved, cannot be used as we clearly need one dummy variable/item for each level of the original SNP.

Each of the three items belonging to a SNP are responsible for one genotype. If their genotype is present in an observation i they take the value 1 at position i while the other two items are 0, resulting into $3 \cdot m = m_{\mathcal{I}}$ binary predictors that, together with two binary variables for disease status, form the data base $\mathcal{D}$. All observations given the binary data are now called transactions T_i, $i = 1, \dots, n$. They correspond to the $\mathbf{x}_i$ of the original data.

We show the relationship of variables and items as well as observations and transactions in the following example: For two cases ($\mathbf{x}_1$ and $\mathbf{x}_2$), two controls ($\mathbf{x}_3$ and $\mathbf{x}_4$) and two SNPs (x_1 and x_2), an example data set may look like the right table in Table 5.2. We then translate the data set into transaction data given in the left table.

A frequent itemset corresponds to a certain genetic profile shared by a sufficiently large percentage of the subjects. If the consequent of an association rule is restricted to consist of one of the class labels only, it indicates that a certain fraction of the subjects that inherit the genetic profile in the antecedent belong to the class given in the consequent. A quantitative definition of these qualities is given in the following section.

	SNP1	SNP2	status
	x_1	x_2	y
$\mathbf{x}_1$	0	1	1
$\mathbf{x}_2$	0	1	1
$\mathbf{x}_3$	0	1	0
$\mathbf{x}_4$	2	0	0

	SNP1			SNP2			status	
	I_1	I_2	I_3	I_4	I_5	I_6	I_{status1}	I_{status0}
T_1	1	0	0	0	1	0	1	0
T_2	1	0	0	0	1	0	1	0
T_3	1	0	0	0	1	0	0	1
T_4	0	0	1	1	0	0	0	1

Table 5.2: This table is a toy example SNP data set with $n = 4$ observations and $m = 2$ variables (lefthand side). On the righthand side, the data has been transformed into transactional data with $n = 4$ transactions and $m_{\mathcal{I}} = 6$ items.

5.1.1 Quality Measures and Statistical Equivalence

An exhaustive search over all possible itemsets and association rules fails in terms of computational feasibility if the data set gets large. Therefore, Agrawal et al. (1996) introduced the famous apriori algorithm (cf. Section 5.1.2) which is able to discover itemsets and association rules that satisfy the quality measures *support* and *confidence*.

Definition 5.1. *Given a set of items $\mathcal{I} = \{I_1, I_2, ..., I_{m_{\mathcal{I}}}\}$ and a data set $\mathcal{D}$ consisting of transactions $T_i \subseteq \mathcal{I}$, $i = 1, \ldots, n$, the* **support** *of an itemset $B \subset \mathcal{I}$ is defined by:*

$$supp(B) = \frac{|\mathcal{D}_B|}{n} \cdot 100\%,$$

where $\mathcal{D}_B = \{T_i \in \mathcal{D} | B \subset T_i\}$ and $| \cdot |$ denotes the magnitude of a set (Borgelt and Kruse (2002)).

A minimum support $supp_{min}$ is set as a threshold. All itemsets exceeding this threshold are considered frequent.

Definition 5.2. *Given a set of items $\mathcal{I} = \{I_1, I_2, ..., I_{m_{\mathcal{I}}}\}$ and a data set $\mathcal{D}$ consisting of transactions $T_i \subseteq \mathcal{I}$, $i = 1, \ldots, n$, the* **confidence** *of an association rule $B \rightarrow H$, $B, H \subset \mathcal{I}$, $B \cap H = \emptyset$ is given by:*

$$con(B \rightarrow H) = \frac{supp(B \cup H)}{supp(B)} \cdot 100\%,$$

where $supp(B \cup H) = \frac{|\mathcal{D}_{B \cup H}|}{n} \cdot 100\%$, $\mathcal{D}_{B \cup H} = \{T_i \in \mathcal{D} | B \cup H \subset T_i\}$ *and* $| \cdot |$ *denotes the magnitude of a set (Borgelt and Kruse, 2002).*

Again, for the algorithm a minimum confidence threshold con_{min} has to be chosen.
An association rule's support can be measured analogously. The original support of an association rule introduced by Agrawal et al. (1993) is defined as the support of the combined set $suppOrig(B \rightarrow H) = supp(B \cup H)$, but Borgelt and Kruse (2002) proposed $supp(B \rightarrow H) = supp(B)$ instead to achieve flexibility in case another quality measure than confidence is employed. We will use the latter, more flexible definition.
For the toy example in Table 5.2, choosing a support of 0.5 and a confidence of 0.6, we find the association rules given in Table 5.3 if we restrict the consequent to consist of the disease status.

Besides support and confidence, there exist numerous proposals of additional quality or interest measures for association rules, see, e.g., Tan et al. (2002) or Piatetsky-Shapiro (1991). They can either be used to prune the mined set of association rules further or to order the rules. This is important for the classification approaches described in Section 5.3. Different choices of interest measures are meaningful for different kinds of data; there is no overall best coefficient (Tan et al., 2002).
A selection of commonly used measures are *lift*, *conviction* and *oddsRatio*.

rule	support	confidence	translation
$I_1 \rightarrow$ status = 1	$\frac{3}{4}$	$\frac{2}{3}$	SNP1 = 0 → $\frac{2}{3}$ of observations with this genotype are cases
$I_5 \rightarrow$ status = 1	$\frac{3}{4}$	$\frac{2}{3}$	SNP2 = 1→ $\frac{2}{3}$ of observations with this genotype are cases
$\{I_1, I_5\} \rightarrow$ status = 1	$\frac{3}{4}$	$\frac{2}{3}$	SNP1 = 0 and SNP2 = 1 → $\frac{2}{3}$ of observations with this genotype are cases

Table 5.3: For the data set given in Table 5.2, we find the following association rules (with the disease status in the consequent) meeting a minimum support of 0.5 and a minimum confidence of 0.6.

Definition 5.3. *Given a set of items $\mathcal{I} = \{I_1, I_2, ..., I_{m_\mathcal{I}}\}$, a data set $\mathcal{D}$ consisting of transactions $T_i \subseteq \mathcal{I}$, $i = 1, \ldots, n$ and $supp(B) = P(B)$,*

*(a) the **lift** of an association rule $B \to H$, $B, H \subset \mathcal{I}$, $B \cap H = \emptyset$ is given by:*

$$lift(B \to H) = \frac{con(B \to H)}{P(H)}$$

*(b) the **conviction** of an association rule $B \to H$, $B, H \subset \mathcal{I}$, $B \cap H = \emptyset$ and $\bar{H} = \mathcal{I} \setminus H$ is given by:*

$$conviction(B \to H) = \frac{P(B)P(\bar{H})}{P(B \cup \bar{H})}$$

*(c) the **oddsRatio** of an association rule $B \to H$, $B, H \subset \mathcal{I}$, $B \cap H = \emptyset$ and $\bar{H} = \mathcal{I} \setminus H$, and $\bar{B} = \mathcal{I} \setminus B$ is given by:*

$$oddsRatio(B \to H) = \frac{P(B \cup H)P(\bar{B} \cup \bar{H})}{P(B \cup \bar{H})P(\bar{B} \cup H)}.$$

Statistical Equivalents

The support of an itemset B equals an estimate of the probability of its occurrence in a random transaction T_i, denoted by $P(\{B\} \subset T_i) =: P(\{B\})$. The confidence of a rule can be seen as an estimate of the conditional probability $P(\{H\}|\{B\})$ while the probability that H and B are both present in a transaction is written as $P(\{B \cup H\})$.
In the data mining literature, the brackets are left out, resulting, e.g., in $P(\{B \cup H\}) = P(B \cup H)$. The concept is clarified by the following relationship:

$$\begin{aligned} P(\{B \cup H\}) \text{ estimated by } \quad & \frac{|\mathcal{D}_{B \cup H}|}{n} \\ = \quad & \frac{|\{T_i \in \mathcal{D} | \{B \cup H\} \subset T_i\}|}{n} \\ = \quad & \frac{|\{T_i \in \mathcal{D} | B \subset T_i\} \cap \{T_i \in \mathcal{D} | H \subset T_i\}|}{n} \\ = \quad & \frac{|\mathcal{D}_B \cap \mathcal{D}_H|}{n}. \end{aligned} \tag{5.1}$$

We go back to the example in Table 5.2 with n=4, $B = \{I_1, I_5\}$ and $H = I_{\text{status}}$. The relationship in formula 5.1 translates into the following tables.

		SNP1			SNP2		status	
$\mathcal{D}$	I_1	I_2	I_3	I_4	I_5	I_6	I_{status1}	I_{status0}
T_1	1	0	0	0	1	0	1	0
T_2	1	0	0	0	1	0	1	0
T_3	1	0	0	0	1	0	0	1
T_4	0	0	1	1	0	0	0	1

$\mathcal{D}_B$	I_1	I_5	I_{status1}
T_1	1	1	1
T_2	1	1	1
T_3	1	1	0

$$\mathcal{D}_B = \{T_1, T_2, T_3\}$$

$\mathcal{D}_H$	I_1	I_5	I_{status1}
T_1	1	1	1
T_2	1	1	1

$$\mathcal{D}_H = \{T_1, T_2\}$$

$\mathcal{D}_{\{B\cup H\}}$	I_1	I_5	I_{status1}
T_1	1	1	1
T_2	1	1	1

$$\mathcal{D}_{B\cup H} = \mathcal{D}_B \cap \mathcal{D}_H = \{T_1, T_2\}$$

5.1.2 The apriori Algorithm and Its Implementation

The apriori algorithm (Agrawal et al., 1996) mines the frequent itemsets first and builds association rules in a second step. The pseudo-code of the algorithm can be seen in Algorithm 2. The fundamental idea which is exploited at several steps within the algorithm is called *downward-closure*, meaning that a superset of an itemset can only be frequent if all contained items and subsets of items themselves are already frequent.

The apriori algorithm starts with itemsets of length 1 (*1-itemsets*). Each item whose support exceeds the minimum support is stored in F_1, the set of frequent 1-itemsets. In

Algorithm 2 apriori Algorithm, adapted from Agrawal et al. (1996)

```
Frequent itemsets
  $F_1 = \{\text{frequent 1-itemsets}\}$
  for ($k \geq 2$, $F_{k-1} \neq \emptyset$) do
    $FC_k$ = apriori-gen($F_{k-1}$); # new candidates with function apriori-gen
    for all transactions $T \in \mathcal{D}$ do
      $FC_T$ =subset($FC_k$, $T$); # candidates contained in transaction $T$
      for all candidates $fc \in FC_T$ do
        $fc$.count = $fc$.count +1
      end for
    end for
    $F_k = \{fc \in FC_k \mid fc.\text{count} \geq supp_{min}\}$;
    $k = k + 1$;
  end for
  set of frequent itemsets = $\bigcup_k F_k$
```

the $(k-1)^{th}$ pass over the data set, a seed set of frequent $(k-1)$-itemsets is used to generate a set of potentially frequent k-itemsets (the *candidate set* FC_k) using the function *apriori-gen*. It consists of two steps: the *join step* and the *prune step*. In the join-step, apriori-gen generates k-itemsets by merging $(k-1)$-itemsets that share the first $k-2$ elements. Afterwards, during the prune-step, all new itemsets that contain infrequent subsets are deleted. Each transaction is now examined to see if it contains candidate itemsets. The itemsets' support is updated during every pass. Among the candidate itemsets, the actually frequent itemsets build the seed set for the next pass. Finally, all sets of frequent itemsets F_k are joint to build the output.
Subsequently, the association rules are generated from the set of frequent itemsets. In the general case, different consequents out of the frequent itemsets have to be tested, but as we will restrict the rules' consequent to consist of one of two specific single items only, this process is abbreviated. Only the itemsets containing these chosen consequent items are investigated, and if their confidence exceeds the minimum confidence, then the association rule and its quality measure values are returned.

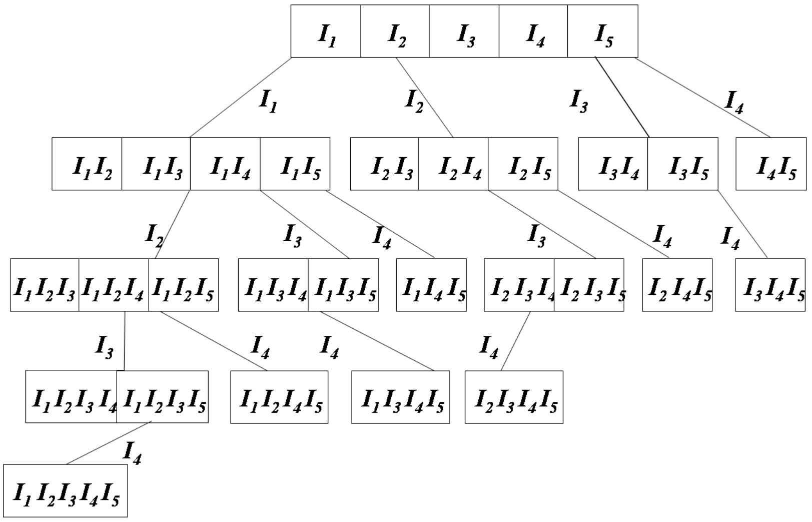

Figure 5.1: Prefix tree of five items. All possible itemsets made out of I_1, I_2, I_3, I_4, I_5, with no regard to the ordering within a set. If an item or an itemset is not frequent, its branch further down the tree is not searched.

Implementation

A very popular version of apriori was implemented by Borgelt and Kruse (2002). One of their achievements is the arrangement of items and possible itemsets in a *prefix tree* (cf. Figure 5.1). The top boxes I_1, I_2, I_3, I_4, I_5 display the single items. The boxes in the second layer give all possible combinations of two items. Further down the tree, in each layer the number of items in a set is increased. The letters next to the lines show which items are added to the sets in the next layer, until all five items are covered by one set at the bottom of the tree. Note that the tree is unbalanced as the order of the items in a set does not matter. If an item or itemset cannot be found in a large number of transactions, it is impossible that any of its combinations with other items occurs frequently. Therefore, beginning with the single items at the top, the search space over the tree is pruned and only itemsets with frequent subsets are investigated.

Furthermore, Borgelt and Kruse (2002) improved the node organisation, the item coding and used recursive counting for assessing support values. Hahsler et al. (2005) implemented this software in the package `arules` into the R environment (R Development Core Team, 2008).

Other Algorithms

Despite its age (initially published in 1996), the apriori algorithm and extensions to it are still widely used to mine association rules. Nevertheless, there exist alternative approaches. Two of them, Eclat (Zaki et al., 1997) and D-Miner (Besson et al., 2004), were investigated in comparison to apriori with respect to their performance on genetic data in Breiter (2008). In contrast to apriori which mines frequent itemsets, Eclat searches for maximal frequent itemsets (frequent itemsets that do not have a frequent superset), while D-Miner finds closed frequent itemsets (frequent itemsets with no subset of the same support). Breiter (2008) showed that D-Miner is unsuitable for the given problem as the minimum support could not be chosen low enough to produce reasonable results and still ensure feasibility. Eclat mined itemsets on which the formed association rules gave good classification results (cf. Section 5.3). The same is true for the results for apriori. Eclat was faster at generating itemsets as the tighter restriction on the characteristics of the itemsets can be exploited during the run.
On the other hand, the association rule generation is not part of the Eclat algorithm, while they are automatically produced with apriori. Therefore, the time saved during the frequent itemset generation is lost again with Eclat while generating rules. All necessary customisation of the association rule search as well as the calculation of quality measures is implemented in the apriori algorithm. To allow for the best possible handling of the data, we therefore choose the apriori algorithm to mine association rules in this thesis.

5.2 Classification Approaches Using Frequent Itemsets

5.2.1 Local Class

The approach *local class* borrows the idea of clustering objects described in Section 4.4. The difference lies in the construction of clusters, or groups in this case, as with local class, patients that share a frequent itemsets are grouped together. We will restrict the analysis to itemsets with a maximum number of three items.
In particular, the algorithm allocates groups according to Algorithm 4 (in the appendix): Choose three different support thresholds $supp_{min_k}$ for $k = 1, 2, 3$ and find all frequent k-itemsets on the training data that exceed their threshold. Order the frequent

k-itemsets according to their support (descending) within the different itemset groups, then combine the three ordered sets, starting with the 3-itemsets. The result is a list of itemsets F, starting with the 3-itemset $f_{3,1}$ with the highest support among all 3-itemsets, followed by all other 3-itemsets with descending support. Then the 2-itemset with the highest support within its group of itemsets and so forth. Each itemset corresponds to a future group which is labeled with the rule's rank number (e.g. group G_1 for the first itemset of F).

Allocate the transactions from the test and from the training data separately to the first group whose corresponding itemset is present in the transaction. All patients that did not fit into an existing group or that are allocated to a group with less than 20 members are fused into the *miscellaneous group* G_{misc}. If one or several groups are only present for training or for test data, dissolve the respective group and allocate its transactions to G_{misc}. Now call the number of still existing groups besides the miscellaneous group n_G. The number of transactions within each group G_g for the test data will be denoted by n_{G_g}.

After establishing the subclasses, build an individual model (using CART) on the training observations in each subgroup G_g, $g = 1, \ldots, n_G$ as well as in the miscellaneous group and assess the misclassification rate by applying the model to the corresponding test data by $MCR_{G_g} = \frac{n^{Gg}_{miss}}{n_{G_g}}$ and $MCR_{G_{misc}} = \frac{n^{Gmisc}_{miss}}{n_{G_{misc}}}$. Averaging over all groups yields the overall misclassification rate $MCR = \frac{1}{n_G+1}(\sum_{g=1}^{n_G} MCR_{G_g} + MCR_{G_{misc}})$.

5.2.2 Feature Construction

Feature construction is a well known tool in data mining (Flach and Lavrac, 2000). Instead of building a classifier using the original variables, new variables or features are constructed first and, in a second step, a classifier based on these constructed features is learned and evaluated. Flach and Lavrac (2000) distinguish between features that describe an interesting subgroup and features that describe a frequent itemset. As our analyses are based on the apriori algorithm and frequent itemsets, we choose the latter version of feature construction.

In particular, we use all frequent itemsets found in local class as input variables for building a classifier. The resulting new data set is binary, as it contains the information if the itemset forming the variable is present (=1) or not (=0) in a patient/transaction.

CART will be employed on the constructed new features.

5.3 Associative Classification

Association rules can also be employed in a classification framework. The concept of *associative classification* is not new (introduced by Ali et al. (1997) and Liu et al. (1998)), but so far it is not heavily used. A review on most of the current algorithms can be found in Thabtah (2005).
The underlying idea of associative classification is to combine the advantages of association rule mining (exhaustive search for interesting patterns instead of heuristic) and classification (extending the description to prediction) to improve the accuracy of assigning objects to certain classes. Several studies (e.g., Liu et al. (1998), Dong et al. (1999)) show that associative classification performs well in comparison with other classification approaches such as C4.5 (Quinlan, 1992).
In contrast to association rule discovery, the search for association rules in associative classification is restricted to rules with a class label as consequent, in our case this translates to a consequent H consisting either of "status = case" or "status = control". For convenience, we adopt the notation $H = 1$ for "status = case" and $H = 0$ for "status = control".
There are numerous associative classification algorithms, but the general procedure of all methods can be divided into three separate steps:

1. Discover association rules, yielding a rule set $\mathbf{R}$ consisting of $n_{\mathbf{R}}$ elements.
2. Select the association rules used for the classification.
3. Evaluate the classifier on the test data.

We will concentrate on methods close to Classification Based Associations (CBA) by Liu et al. (1998). It generates association rules $R_r \in \mathbf{R}$, $r = 1, ..., n_{\mathbf{R}}$, which satisfy the given thresholds for support and confidence using the apriori algorithm. Then, all R_r are ordered (with $R_a \succ R_b$ meaning that R_a has a higher rank than R_b):

1. $R_a \succ R_b$ if $con(R_a) > con(R_b)$.
2. If $con(R_a) = con(R_b)$: $R_a \succ R_b$ if $supp(R_a) > supp(R_b)$.

3. If $con(R_a) = con(R_b)$ and $supp(R_a) = supp(R_b)$: $R_a \succ R_b$ if R_a was generated earlier than R_b.

Thus, the classifier consists of an ordered list of association rules. A new observation is classified according to the label of the first rule of that list which is applicable, meaning whose antecedent is contained in the new observation's transaction. If no association rule of the list is applicable, the new observation is assigned to a default class.

We choose CBA because it will be most comparable to our different approaches for using frequent itemsets and association rules as analysis tools for genetic association studies. Furthermore, adaptions (e.g. in weighting rules) can be related directly to themselves instead of diffusing processes within the classification.

5.3.1 Naive Associative Classification

The associative classifier that we call *naive associative classification* resembles actually the original CBA if the ranking of association rules is conducted according to the instruction given above. We use our own implementation which also allows the ordering of the rule set to be conducted according to any interest measure, leading to several different rankings and different classification results.

An advantage of employing the confidence is that it was computed during the algorithm, whereas the values of the other interest measures would have to be computed separately. Additionally, it can be shown that two of the interest measures we introduced in Subsection 5.1.1, lift and conviction, result in the same order of rules than ordering them by their confidence value if the consequent is the same item for the rules (cf. Appendix). This is due to the proportion of cases $P(H = 1)$ being a constant in this case. With this constraint, all measures mentioned above form monotonous transformations of the confidence on the interval [0, 1] (for similar considerations see Bayardo and Agrawal (1999)). The proofs can be found in the Section C.1 in the Appendix.

As we are dealing with several rules from now on, the antecedent and consequent of a rule have to be labeled with an index, yielding $B_r \rightarrow H_r$ for association rule R_r . We set H_r to equal 1 if the rule is a case rule and 0 if it is a control rule. Let $\mathbf{R}(T_i)$ be the ordered set of $n_{\mathbf{R}(T_i)}$ applicable rules for transaction T_i, applicable meaning that $B_r \subset T_i$. The appropriate decision rule is

$$\delta_1(\mathbf{R}(T_i)) := \begin{cases} 1, \text{ if } & H_1 = 1 \text{ with } H_1 \in R_1, R_1 \subset \mathbf{R}(T_i) \\ 0, & \text{else} \end{cases} \tag{5.2}$$

and it implies that transactions with an empty set of applicable rules are classified as controls by default.

5.3.2 Voting

Even though the naive method proves itself intuitive and sensible, one of its weaknesses lies in the lack of robustness. If by chance the first applicable rule in the ordered set is random instead of meaningful, the respective observation might be misclassified. Also, alternative interactions influencing the disease risk can only be captured by allowing for different association rules.

To solve this problem, we use the whole set of applicable rules for a new test observation instead of just the best applicable rule. The respective rules vote for the new observations class label. An intuitive choice would be a majority vote (Baralis and Garza, 2003). However, in the case of SNP data, several mechanisms typical for healthy persons might also be present in a diseased person. If there are many association rules describing this control-specific genetic profile and only one rule representing the risk factors, the control rules might lead to a wrong prediction of the diseased observation. Therefore, the information carried by a case rule seems to be more valuable than the information given by an applicable control rule, which will be expressed in terms of weights the different kinds of rules will be given in the voting process. For our analysis, the decision rule

$$\delta_2(\mathbf{R}(T_i)) := \begin{cases} 1, \text{ if } & \frac{1}{n_{\mathbf{R}(T_i)}} \sum_{r=1}^{n_{\mathbf{R}(T_i)}} H_r \geq \gamma \\ 0, & \text{else} \end{cases} \tag{5.3}$$

with $H_r = 1$ for case rules and $H_r = 0$ for control rules will be investigated in two different settings.

1. We apply the classical majority vote ($\gamma = 0.5$).
2. We perform a grid search over possible proportions $\gamma \in [0,1]$ to find the best proportion between case and control rules in the vote.

For an even more specific approach, each rule gets an individual weight:

$$\delta_3(\mathbf{R}(T_i)) := \begin{cases} 1, \text{ if} & \sum_{r=1}^{n_{\mathbf{R}(T_i)}} w_r I_{\{H_r=1\}} \geq \sum_{R_r \subset \mathbf{R}(T_i)} w_r I_{\{H_r=0\}} \\ & \Leftrightarrow \ \sum_{r=1}^{n_{\mathbf{R}(T_i)}} w_r H_r \geq 0.5 \\ 0, & \text{else} \end{cases} \tag{5.4}$$

The first part of the equivalence on the right-hand side says that the sum of weighted case rules has to exceed the sum of the weighted control rules. As the weights sum up to 1, so do the two combined sums. Thus, comparing the weighted sum of case rules to 0.5 gives and equivalent decision .
We construct weights w_r^* by combining support and confidence values by their geometric mean, $w_r^* = \sqrt{supp(R_r) \cdot con(R_r)}$). As the applicable rule set differs for every test observation to be classified, weights w_r^* would not always sum up to one if they were initially chosen for the whole set of rules. Therefore, we scale them if necessary with scaling parameter ν to ensure $\sum_{r=1}^{n_{\mathbf{R}(T_i)}} \frac{w_r^*}{\nu} = 1$, and $w_r = \frac{w_r^*}{\nu}$.

5.3.3 Locality and Interaction

The methods based on association rules can be considered as local and can handle interaction effects. The interactions are directly present in itemsets (antecedents) of length greater than one. A resulting effect on the disease risk is attributable to the described genotype. Another huge advantage of frequent itemsets and association rules is the ready interpretability which allows to present the rules directly to biologists or toxicologists interested in the results of the analysis.
The proof of locality of the methods is more complicated as the term "local" does not have a clean cut definition. Although, e.g., in computer science, the terms *local* and *global* are used as well, their meaning differs from the one used in this thesis. Thabtah (2005) considers associative classification global, as the complete (training) data set was used for every rule generated. Other search strategies, such as divide-and-conquer

(Fürnkranz, 2005) remove all observations covered by a mined rule and discover the next rule based only on the remaining observations. Fürnkranz (2005) calls this approach local. Following this definition, associative classification and feature construction would not be considered local.

But from our point of view, associative classification definitely is local: It employs a local subset of all mined rules that is applicable to each specific observation, in the voting approach even with individual observation-wise weights. Alternative ways of influencing the disease risk can therefore be modeled by using different association rules for classifying observations from different subgroups.

The statement that associative classification is a local method is also supported by Mielikäinen (2005) who says that pattern detection in general is a local model, and finding association rules is a method of pattern detection.

In the following section, local logic regression is introduced. Local regression has a fixed definition; every observation is classified by an individual regression model that ensures that observations close to the observation to be classified contribute to the model with higher weights.

5.4 Localised Logic Regression

Logic regression (Ruczinski et al., 2003) is a regression and classification technique which was especially developed for the analysis of SNP data. It is based on logic expressions formed by boolean variables. Compared to other classification approaches, it performed considerably well on different genetic data sets (Kooperberg et al. (2001), Ruczinski et al. (2004), Schwender (2007)). In this thesis, we combine the proposed idea of localisation from previous chapters with logic regression using the well established theoretical background of *local regression* and *local likelihood* methodology (Loader, 1999).

The first part of this section introduces logic regression, followed by a short insight into local regression theory in Subsection 5.4.2. Subsection 5.4.3 combines the local aspects with logic regression and compares the result to a *boosting* approach (which might seem similar to localisation).

5.4.1 Logic Regression

This section is primarily based on Ruczinski (2000) and Ruczinski et al. (2003). In addition to the notation given in the introduction to this chapter, in the following the response variable will be denoted as y, while the m predictor variables will be $\mathbf{x} = (x_1, \ldots, x_m)'$.
The main idea of logic regression in contrast to ordinary regression, where mostly main effects or at most interactions of low order are considered, is to allow for the integration of high-order interaction effects in the form of a generalized additive model based on boolean logic expressions $L_i(\mathbf{x}), i = 1, \ldots, q$:

$$f(y) = \beta_0 + \sum_{i=1}^{q} \beta_i \cdot I_{L_i(\mathbf{x})\text{is true}}. \tag{5.5}$$

The logic expressions of the explanatory variables are not given beforehand, but have to be constructed during the fitting process.
Suppose you have a set of p binary predictors $\mathbf{x}$ and a response variable y. In our case, y is binary as well, but this is not mandatory for logic regression. These binary predictors $\mathbf{x}$ can form logic expressions L of the form, e.g.,

$$L = (x_1 \wedge x_2^C) \vee x_3 \tag{5.6}$$

which are either true or false for a specific observation. Possible operators to connect the predictors are *and* ($\wedge$) and *or* ($\vee$), while *not* (^{C}) is needed to incorporate both values of a predictor.
Each logic expression can be constructed iteratively as a combination of two predictors (e.g., x_1 and x_2^C in the expression above), of a predictor and a logic expression or of two logic expressions. A visualisation of Equation 5.6 in form of a *logic tree* following the iterative construction scheme can be seen in Figure 5.2. Logic trees (we will use this term equivalently to logic expression) can either form a *logic model* consisting of a single tree (e.g. as in classification with two possible class labels), or be incorporated in a bigger logic model in which several logic trees are combined appropriately (e.g., in multiple logistic regression). A good logic model should describe the outcome y as precisely as possible, as well as predict the outcome of a new observation reliably. For the different contexts that logic regression can be applied to, different measures of quality (*score functions*) have to be chosen. For logistic regression, e.g., the score function is the deviance (cf. Equation 5.8).

Due to computational infeasibility, we cannot carry out an exhaustive search over all models to find the best one as there are just too many possibilities. Therefore, we have to employ search algorithms. Two different types are considered in logic regression: a greedy search and a simulated annealing approach. As the greedy search is computational less complex, we will concentrate on this search method only even though it might get stuck in local optimal solutions. Simulated annealing, combined with a localised approach, will be computational infeasible.
Let's start with a single tree scenario: Initially, a greedy search algorithm builds a tree by selecting the single variable that yields the best quality with regard to its predictive power on y, using the appropriate scoring function. Afterwards, all neighbouring trees are generated using the following move set (Ruczinski, 2000):

- Alternating a leave,
- changing an operator,
- growing a new branch,
- pruning a branch as well as
- splitting or deleting a leave.

If the new tree scores better than the original tree and all other neighbours, it replaces the initial tree, and the algorithm starts again. It stops as soon as no improvement can be made in a single step.
Equipped with the means to find the best single tree, the methodology can be extended to bigger logic models $y = \beta_0 + \sum_{i=1}^{q} \beta_i \cdot I_{L_i(\mathbf{x})\text{is true}}$. If we allow for multiple trees in the model, the move set to find neighbour logic expressions has to be enlarged by the move *Start a new tree*.
The search algorithm starts with an empty model. Then, for all possible single predictor, a model is build by maximising the likelihood $\mathcal{L}(\boldsymbol{\mu}, \mathbf{y}) = \sum_{i=1}^{n} ln\, f(y_i, \mu)$, with

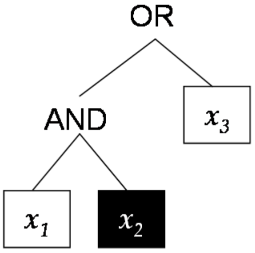

Figure 5.2: Logic Tree

$f(y_i, \mu)$ being the density of y and μ being the parameter of interest. The model which scores best with respect to the score function is chosen as basis for the next iteration of the search algorithm, the enlarged move set is employed, and the best of the resulting models is chosen for the next iteration step.

Logistic Regression

The data we want to analyse contain a binary outcome y_i, $i = 1, \ldots, n$. The respective probabilities are given by $P(y_i = 1|L(\mathbf{x}_i)) = \pi(\mathbf{x}_i)$ and $P(y_i = 0|L(\mathbf{x}_i)) = 1 - \pi(\mathbf{x}_i)$ with $\mathbf{x}_i = (x_{i,1}, \ldots, x_{i,m})$. For convenience, we will write π_i instead of $\pi(\mathbf{x}_i)$, and $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_n)$. The suitable regression technique is logistic regression. Instead of modelling the plain probabilities by a linear combination of the logic expressions, $ln\left(\frac{\pi}{1-\pi}\right)$ = the log odds of belonging to class 1 are modeled (cf. Section 5.4.2). Therefore, given q independent logic expressions L_j, $j = 1, \ldots, q$ that can be either true or false, the following regression equation evolves:

$$ln\left(\frac{\boldsymbol{\pi}}{1-\boldsymbol{\pi}}\right) = \beta_0 + \beta_1 I_{\{L_1 \text{ is true}\}} + \cdots + \beta_q I_{\{L_q \text{ is true}\}} \tag{5.7}$$

$$\Leftrightarrow \quad \boldsymbol{\pi} = \frac{e^{(\beta_0 + \beta_1 I_{\{L_1 \text{ is true}\}} + \cdots + \beta_q I_{\{L_q \text{ is true}\}})}}{1 + e^{(\beta_0 + \beta_1 I_{\{L_1 \text{ is true}\}} + \cdots + \beta_q I_{\{L_q \text{ is true}\}})}}.$$

The coefficients β_j, $j = 1, \ldots, q$ determine the influence of each indicator variable $I_{\{L_j \text{ is true}\}}$. They depend on the set of predictors $\mathbf{x}$ and could be rewritten as $\beta_j(x_j)$ to stress the issue.

An estimate $\hat{\boldsymbol{\pi}}$ for $\boldsymbol{\pi}$ can be found by maximising the log likelihood

$$\mathcal{L}(\boldsymbol{\pi}, \mathbf{y}) = \sum_{i=1}^{n} y_i \cdot ln(\frac{\pi_i}{1-\pi_i}) + ln(1-\pi_i)$$

numerically via iterative Fisher-Scoring (cf. e.g. Cramer (2003)).

Returning to the algorithm, the competing models of an algorithmic step are evaluated by their score function, the deviance (McCullagh and Nelder, 1989):

$$D(\mathbf{y}, \hat{\boldsymbol{\pi}}) = 2\sum_{i=1}^{n}(y_i \cdot ln(\frac{y_i}{\hat{\pi}_i}) + (1-y_i)ln(\frac{1-y_i}{1-\hat{\pi}_i}))$$

$$\Leftrightarrow D(\mathbf{y}, \hat{\boldsymbol{\pi}}) = -2\sum_{i=1}^{n}(y_i \cdot ln(\hat{\pi}_i) + (1-y_i)ln(1-\hat{\pi}_i)). \tag{5.8}$$

Classification

In a classification scenario with two classes, we restrict the number of logic trees in the model to a single tree L, which we use to construct the best classification rule C:

$$C = I_{\{L \text{ is true}\}}. \tag{5.9}$$

If L is true for an observation, it is labeled class 1, otherwise it is assigned to class 0. An appropriate score function is the misclassification rate

$$MCR = \frac{n_{miss}}{n}, \tag{5.10}$$

where n_{miss} is the number of misclassified observations. MCR gives the percentage of misclassified observations.
A more detailed description of the methodology, theoretic background and applications can be found in Ruczinski (2000).

5.4.2 Local Regression and Local Likelihood

All methodology is taken from Loader (1999), Hastie et al. (2001) and Bornkamp (2006). There are numerous applications where a local regression approach is more advisable than ordinary regression. One of them is the analysis of SNP data, where we assume local relationships which are true in a close neighbourhood of $\mathbf{x}_i$, but may change for a different $\mathbf{x}_j$, corresponding to alternative ways of developing a disease.
The basic idea of local likelihood regression says: Instead of building one global explanatory model for y_i where all observations $\mathbf{x}_i$, $i = 1, \ldots, n$ influence the fit in equal parts, different models for different fitting points $\mathbf{x}_0$ are computed for assessing the corresponding $\hat{y}_0$. During the estimation of the regression parameters, observations in a close neighbourhood of $\mathbf{x}_0$ contribute with a higher weight $w_i(\mathbf{x}_0)$ to the estimate $\hat{y}_0$ than distant observations. The necessary measure for the amount of closeness between the fitting point and the data points within the window $\{\mathbf{x}_i | d(\mathbf{x}_i, \mathbf{x}_0) \leq h(\mathbf{x}_0)\}$ is a weighting function

$$w_i(\mathbf{x}_0) = W\left(\frac{d(\mathbf{x}_i, \mathbf{x}_0)}{h(\mathbf{x}_0)}\right). \tag{5.11}$$

It assigns a weight to each observation $\mathbf{x}_i$ according to its distance d to $\mathbf{x}_0$ and depending on the bandwidth $h(\mathbf{x}_0)$. All sensible weighting functions increases as $d(\mathbf{x}_i, \mathbf{x}_0)$ decreases (cf. Subsection 5.4.2.1).

We will now show the underlying theory for local logistic regression as a special case of likelihood regression. It was already investigated in the case of high dimensions and continuous regression variables by Tutz and Binder (2005) and applied to SNP data by Schiffner et al. (2009).

The initial situation is similar to Section 5.4.1: We have a binary outcome y_i, $i = 1, \ldots, n$ and a set of binary predictors $\mathbf{x}_i$. The probabilities of belonging to class one or zero, respectively, look like (cf. Section 5.4.1):

$$P(y_i = 1|\mathbf{x}_i) = \pi_i, \quad P(y_i = 0|\mathbf{x}_i) = 1 - \pi_i,$$

with $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_n)$. The corresponding log likelihood l of π_i gives:

$$\begin{aligned} l(\pi_i) &= y_i ln(\pi_i) + (1 - y_i) ln(1 - \pi_i) \\ &= y_i ln(\frac{\pi_i}{1 - \pi_i}) + ln(1 - \pi_i). \end{aligned} \tag{5.12}$$

As already mentioned in Section 5.4.1, π_i should not be modeled directly, therefore, we use the logit function as a link function $\theta(\cdot)$ to map the range of the estimation from $[0, 1]$ to $(-\infty, \infty)$:

$$\begin{aligned} \theta(\mathbf{x}) &= ln(\frac{\boldsymbol{\pi}}{1 - \boldsymbol{\pi}}) \qquad (5.13) \\ \Leftrightarrow \quad \boldsymbol{\pi} &= \frac{e^{\theta(\mathbf{x})}}{1 + e^{\theta(\mathbf{x})}}. \end{aligned}$$

The function $\theta(\mathbf{x}) = \mathbf{x}'\boldsymbol{\beta}$, with $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)'$ being the vector of regression coefficients.

Now, using Equations 5.12 and 5.13, a global log likelihood for logistic regression is

$$\mathcal{L}(\boldsymbol{\theta}, \mathbf{y}) = \sum_{i=1}^{n} (y_i \cdot \theta(\mathbf{x}_i) + ln(1 - \frac{e^{\theta(\mathbf{x}_i)}}{1 + e^{\theta(\mathbf{x}_i)}})).$$

In a general logistic regression scenario, the local approximation of $\theta(\mathbf{x})$ following Taylor's theorem would be achieved in fitting a local polynomial of a user specified degree around $\mathbf{x}_0$ to $\theta(\mathbf{x})$ and incorporate it together with the weights into the likelihood regression. As in our case all input variables are binary, the approximation via a polynomial is only sensible for a constant polynomial of degree 0, which means that the

original $\theta(\mathbf{x})$ remains in the likelihood equation.
This translates to the local log likelihood for $\mathbf{x}_0$ including weights $w_i(\mathbf{x}_0)$:

$$\mathcal{L}_{\mathbf{x}_0}(\boldsymbol{\theta}, \mathbf{y}) \quad = \quad \sum_{i=1}^{n} w_i(\mathbf{x}_0)(y_i \cdot \theta(\mathbf{x}_i) + ln(1 - \frac{e^{\theta(\mathbf{x}_i)}}{1 + e^{\theta(\mathbf{x}_i)}})).$$

The estimate $\hat{\theta}(\mathbf{x}_0)$ is given by optimising $\mathcal{L}_{\mathbf{x}_0}(\boldsymbol{\theta}, \mathbf{y})$. As we are rather interested in the local $\boldsymbol{\pi}$, we invert the link function and get

$$\hat{\boldsymbol{\pi}} = \frac{e^{\hat{\theta}(\mathbf{x}_0)}}{1 + e^{\hat{\theta}(\mathbf{x}_0)}}.$$

Tutz and Binder (2005) choose a notation which directly relates to $\boldsymbol{\pi}$ instead of $\boldsymbol{\theta}$, yielding

$$\mathcal{L}_{\mathbf{x}_0}(\boldsymbol{\pi}, \mathbf{y}) = \sum_{i=1}^{n} w_i(\mathbf{x}_0)(y_i \cdot ln(\pi_i) + (1 - y_i) \cdot ln(1 - \pi_i)), \tag{5.14}$$

which is the form we want to use in later contexts.
A major problem, discussed, e.g., in Hastie et al. (2001) and Tutz and Binder (2005), is the curse of dimensionality (Bellman, 2000). It implies that for high dimensions, a local likelihood approach is hardly local as a neighbourhood with a sufficient number of observations has to be chosen too large to be considered local anymore. Therefore, Tutz and Binder (2005) suggest a variable selection procedure prior to the fitting of local regression or likelihood methods to reduce the dimension. We will both adapt their suggestion of preselecting relevant variables as well as carry out a local analysis of the complete data.

5.4.2.1 Weighting Function and Bandwidth

As all local models relevant to our analysis will be multivariate, we will describe distances and weights directly for vectors $\mathbf{x}_i$.
The weighting function $W(x)$ determines the influence of an observation $\mathbf{x}_i$ on the estimate at the local fitting point $\mathbf{x}_0$. Its general structure was already given in Equation 5.11. Useful features (defined for kernels but also useful for weighting functions) of $W(x)$ are (Fahrmeir et al., 1996):

- Symmetry around 0,
- maximum value at 0 and
- non-negativity.

There are numerous possibilities to define such a function; we will use the structure of the classical tricube weighting function $W_{twf}(x) = (1 - |x|^3)^3 \cdot I_{\{x \leq 1\}}$ described in Loader (1999). $W_{twf}(x)$ fulfills all desired features as it is only applied to observations within the window $d(\mathbf{x}_i, \mathbf{x}_0) \leq h(\mathbf{x}_0)$. With this restraint, in contrast to, e.g., a Gaussian kernel function for which all observations add to the estimation, $W_{twf}(x)$ saves computation time and effort.

The distance function d used in most multivariate applications is based on the length of a vector $\mathbf{x} \in R^m$ (Loader, 1999):

$$||\mathbf{x}||^2 = \sum_{j=1}^{m} \left(\frac{x_j}{s_j} \right)$$

with $s_j > 0$ being a scaling parameter in dimension j. We will use this standard approach for comparison, but focus on an adaption of our Flexible Matching Coefficient FMC from Section 4.1 for calculating the distances d. In contrast to the standard approach, they are suitable for categorical data. They also fulfill all required useful features if the similarity is transformed into a distance according to Equation 4.1. This yields

$$W_{\text{FMC}} \left(\frac{1 - \text{FMC}(\mathbf{x}_i, \mathbf{x}_0)}{h(\mathbf{x}_0)} \right) \quad \text{with} \quad W_{\text{FMC}}(x) = (1 - |x|^3)^3 \cdot I_{\{x \leq 1\}}.$$

The parameters chosen for FMC will reflect quality aspects derived in Chapter 4 and promise a coherent analysis according to data structure.

The appropriate bandwidth lies in between too small values leading to an estimate that might be based on very few up to no observations and a noisy fit with a high variance. If $h(\mathbf{x}_0)$, $\mathbf{x}_0$ being the fitting point, is very large, the locality of the model ceases, resulting in fits with a low variance, but a high bias.

In spite of the high dimension of the data, we can still ensure to have sufficient observations in a close neighbourhood around $\mathbf{x}_0$ by choosing a bandwidth reflecting a nearest neighbourhood quality: We compute all distances $d(\mathbf{x}_i, \mathbf{x}_0)$, $i = 1, \ldots, n$ (based on the distance used in the weighting function $W(\cdot)$). Then we choose a smoothing parameter $\lambda \in [0, 1]$ and define $h(\mathbf{x}_0)$ to be the kth smallest distance with $k = \lambda n$.

5.4.3 Local Logic Regression

Now we want to localise logic regression, which means that we need a separate model for every observation $\mathbf{x}_i$, $i = 1, \ldots, n$. For the case of using logic regression in the regression environment with binary outcome y and multiple trees, a straight-forward approach is to incorporate weights in the log likelihood during the fitting process (cf. equation 5.14). The resulting likelihood equation is

$$\begin{aligned}
\mathcal{L}_{\mathbf{x}_0}(\boldsymbol{\beta}, \mathbf{y}) &= \sum_{i=1}^{n} w_i(\mathbf{x}_0)(y_i \cdot ln(\frac{e^{(\beta_0+\beta_1 I_{\{L_1 \text{ is true}\}}+\cdots+\beta_q I_{\{L_q \text{ is true}\}})}}{1+e^{(\beta_0+\beta_1 I_{\{L_1 \text{ is true}\}}+\cdots+\beta_q I_{\{L_q \text{ is true}\}})}}) \\
&+ (1-y_i) \cdot ln(1 - \frac{e^{(\beta_0+\beta_1 I_{\{L_1 \text{ is true}\}}+\cdots+\beta_q I_{\{L_q \text{ is true}\}})}}{1+e^{(\beta_0+\beta_1 I_{\{L_1 \text{ is true}\}}+\cdots+\beta_q I_{\{L_q \text{ is true}\}})}})) \\
\Leftrightarrow \mathcal{L}_{\mathbf{x}_0}(\boldsymbol{\pi}, \mathbf{y}) &= \sum_{i=1}^{n} w_i(\mathbf{x}_0)(y_i \cdot ln(\pi_i) + (1-y_i) \cdot ln(1-\pi_i)) \qquad (5.15)
\end{aligned}$$

and it reflects all characteristics shown in the previous section. However, logic regression consists of two steps: the model fitting and the model selection step. The latter employs the deviance as the scoring function for logistic regression. It is directly related to the log likelihood via

$$D(\mathbf{y}, \hat{\boldsymbol{\pi}}) = 2(\mathcal{L}(\mathbf{y}, \mathbf{y}) - \mathcal{L}(\hat{\boldsymbol{\pi}}, \mathbf{y})),$$

and as for all comparisons between different fitted values $\hat{\pi}$ the first part of the deviance stays constant, it reduces to $-2\sum_{i=1}^{n}(y_i \cdot ln(\hat{\pi}_i) + (1-y_i)ln(1-\hat{\pi}_i))$. Now, assuming a local likelihood as in Equation 5.15, this translates to

$$D_{\mathbf{x}_0}(\mathbf{y}, \hat{\boldsymbol{\pi}}) = 2(\mathcal{L}_{\mathbf{x}_0}(\mathbf{y}, \mathbf{y}) - \mathcal{L}_{\mathbf{x}_0}(\hat{\boldsymbol{\pi}}, \mathbf{y}))$$

and the scoring can be assessed by calculating

$$-2(\mathcal{L}_{\mathbf{x}_0}(\hat{\boldsymbol{\pi}}, \mathbf{y})) = -2\sum_{i} w_i(\mathbf{x}_0)[y_i \cdot ln(\hat{\pi}_i) + (1-y_i)ln(1-\hat{\pi}_i)].$$

However, the application of logic regression is much broader than the special case of logistic regression only; it comprises more regression scenarios as well as a methodology for classification. The locality can be inserted analogously to the proceeding described above.

Primarily, in our analysis we will be dealing with the classification approach. Although logistic regression can without any difficulty be adapted to serve as a classifier as well, the computational effort of a localised logic regression with logistic regression application is merely infeasible. On the other hand, most of the other classifiers we describe in different sections resemble the ad hoc kind of classification as the one we will stick to for logic regression better, and therefore, a comparison is more meaningful.
The effect of localisation in classification is expressed explicitly in the scoring function, the misclassification rate, which is weighted by the respective values w_i:

$$MCR_{\mathbf{x}_0} = \sum_i w_i(\mathbf{x_0}) \cdot (\hat{y}_i - y_i)^2. \tag{5.16}$$

The formula looks slightly different than the local variant in Equation 5.10, but for $w_i = \frac{1}{n}$, it yields the same structure as the squared difference is only 1 if prediction $\hat{y}_i$ and original value y_i differ:

$$\begin{aligned} MCR_{\mathbf{x}_0} &= \sum_i w_i(\mathbf{x_0}) \cdot (\hat{y}_i - y_i)^2 \\ MCR_{\mathbf{x}_0} &= \sum_i \frac{1}{n} \cdot (\hat{y}_i - y_i)^2 \\ MCR_{\mathbf{x}_0} &= \frac{1}{n} \cdot \sum_i (\hat{y}_i - y_i)^2 \\ MCR_{\mathbf{x}_0} &= \frac{n_{miss}}{n}. \end{aligned}$$

Locality for the general regression model in 5.5 for a constant polynomial fit and a parameter vector of interest $\boldsymbol{\mu}$ results in a likelihood of the form $\mathcal{L}_{\mathbf{x}_0}(\boldsymbol{\mu}, \mathbf{y}) = \sum_{i=1}^{n} w_i(ln\, f(y_i, \mu))$. The respective score functions inherit the local weight in a similar manner to the deviance and the misclassification rate.

5.4.4 Separation from Boosting Logic Regression

In contrast to the previous section, we will concentrate on the classification scenario only when describing the boosting approach. All classifiers will be build on training data, while the final misclassification error will be assessed using the training model on the test data.
At first glance, boosting logic regression, as boosting does include multiplicative weights

as well, might seem similar to the local approach described in the previous section. But taking a closer look, we find that boosting works from a completely different perspective: The underlying idea of boosting is to construct a series of weak classifiers L_j, logic trees in our case, and predict the outcome of a new observation according to a weighted vote given by these classifiers (Hastie et al., 2001). Weak means that the prediction of the classifier itself is quite close to random guessing. Still, during the iterative construction of new classifiers, they perform better and better for difficult observations as more emphasis is laid upon observations that have been misclassified before by adjusting the individual observation weights $w_{B\,i}$. Thus, the weights have a completely different interpretation as in local regression; instead of reflecting a closeness to and therefore an influence on a fitting point $\mathbf{x}_0$, the $w_{B\,i}$ are increased or decreased due to how well the corresponding outcome can be predicted by the classifiers.

The popular algorithm *AdaBoost.M1.* (Freund and Schapire, 1997) is described in Algo-

Algorithm 3 AdaBoost.M1.

```
Set all initial observation weights to $w^1_{B\,i} = \frac{1}{n}, i = 1, \ldots, n$.
for $j = 1$ to $J$ do
    Fit a classifier/logic expression $L_j(x)$ to the training data using weights $w^j_{B\,i}$.
    Compute the weighted misclassification rate
        $MCR^{w_B}_j = \frac{\sum_{i=1}^n w^j_{B\,i} I(y_i \neq L_j(x)))}{\sum_{i=1}^n w^j_{B\,i}}.$
    Compute the voting weight of $L_j(x)$: $\alpha_j = ln(\frac{1-MCR^{w_B}_j}{MCR^{w_B}_j})$
    Update weights: $w^{j+1}_{B\,i}$ <- $w^j_{B\,i} \cdot exp(\alpha_j \cdot I(y_i \neq L_j(x))), i = 1, \ldots, n.$
    Recode class labels:
        $c_j(x) = \begin{cases} 1 & \text{, if } L_j(x) = 1 \\ -1 & \text{, if } L_j(x) = 0 \end{cases}$
end for
Set class to
    $C(\mathbf{x}_0) = sign\left(\sum_{j=1}^{J} \alpha_j \cdot c_j(\mathbf{x}_0)\right)$
```

rithm 3 (structure adapted from Hastie et al. (2001)) with slight changes as AdaBoost.M. expects class labels to be -1 and 1 instead of 0 and 1.

5.5 Comparing Classification Results

To avoid an arbitrary comparison of the performance of two classifiers C_A and C_B, we employ a test to evaluate differences between two estimated misclassification rates. Dietterich (1998) compared five different tests designed for this purpose and suggested the use of 5×2*cv paired t-test*. It works as follows:

Initially, the data set $\mathcal{D}$ is randomly divided into two equal parts $\mathcal{D}_1$ and $\mathcal{D}_2$. Both parts serve as training and test set once for both classifiers, yielding four different estimated misclassification rates $MCR^1_{C_A}$, $MCR^1_{C_B}$, $MCR^2_{C_A}$ and $MCR^2_{C_B}$ with estimated differences $MCR^1 = MCR^1_{C_A} - MCR^1_{C_B}$ and $MCR^2 = MCR^2_{C_A} - MCR^2_{C_B}$. The superscript indicates which data subset was used for training. A suitable estimated variance is $s^2 = (MCR^1 - M\bar{C}R)^2 + (MCR^2 - M\bar{C}R)^2$, with $M\bar{C}R = (MCR^1 + MCR^2)/2$.

This procedure is replicated five times, yielding one estimate s_i^2, $i = 1, \ldots, 5$ for each run. All five values are used to provide a stabilized estimated variance of the difference of the misclassification rates. The null hypothesis of "both classifiers yielding the same misclassification rates" results into

$$H_0 : MCR_{C_A} = MCR_{C_B} \quad \text{vs.} \quad H_1 : MCR_{C_A} \neq MCR_{C_B},$$

with MCR_{C_A} and MCR_{C_B} being the true misclassification rates of the two classifiers. We test the hypothesis by means of a test statistics whose structure is similar to a paired t-test (cf. Equation 5.17). Even though we perform five classifications for each classifier, only the difference MCR^1_1 from the first of the five runs serves as numerator and the stabilized variance estimate as denominator:

$$\tilde{t} = \frac{MCR^1_1}{\sqrt{\frac{1}{5}\sum_{i=1}^{5} s_i^2}} \overset{approx}{\sim} t_5. \tag{5.17}$$

Under the assumptions that the binomial distribution of the proportions (= misclassification rates) approximates the normal distribution, that MCR^1_i and MCR^2_i are independent of each other, that all s_i^2 are independent and that the numerator and denominator of $\tilde{t}$ are independent, $\tilde{t}$ approximately follows a t-distribution with five degrees of freedom. Dietterich (1998) shows that minor violations of these assumptions do not worsen the result of the testing procedure substantially.

In the case of the simulation study, we do not need the cross-validation as we have results of ten different training and test sets already. In contrast to the test procedure described above, we use the ten different test and training settings for assessing the variance estimate and again choose the first difference in misclassification rates for the test statistic's numerator. The resulting test statistics approximately follows a *t*-distribution with 10 degrees of freedom.

CHAPTER 6

Results

All analysis was carried out using R version 2.8.1 (R Development Core Team, 2008). We used the packages *arules* (Hahsler et al., 2009), *LogicReg* (Kooperberg and Ruczinski, 2008), *rpart* (Therneau and Atkinson., 2008) and *scrime* (Schwender and Fritsch, 2009).

6.1 Clustering

By assessing the desirability index, we compare the performance of the clusterings obtained by employing five different similarity coefficients: simple matching coefficient (SMC), Jaccard's coefficient (JC), the flexible matching coefficients FMC1 and FMC2 with

- $\mathbf{w}_F^{+\prime} = (1, 2, 4)$,
- $\mathbf{m}_F^{+} = (m_{00}, m_{11}, m_{22})$,
- $\mathbf{w}_F^{-\prime} = (1, 1, 1, 1, 1, 1)$ and
- $\mathbf{m}_F^{-} = (m_{01}, m_{02}, m_{10}, m_{12}, m_{20}, m_{21})$

for FMC1 (stressing mutual variants), and for FMC2 with

- $\mathbf{w}_F^{+\prime} = (1, 2, 0.5, 0.5, 4)$,

- $\mathbf{m}_F^+ = (m_{00}, m_{11}, m_{12}, m_{21}, m_{22})$,
- $\mathbf{w}_F^{-\prime} = (1, 1, 1, 1)$ and
- $\mathbf{m}_F^- = (m_{01}, m_{02}, m_{10}, m_{20})$

(allowing pairs with at least one common variant to be considered similar), as well as Pearson's corrected contingency coefficient (PCC).

6.1.1 Simulation

The best number of clusters for the different simulation scenarios is unknown. For a good choice, we divide the ten data sets into two groups of five data sets each. Within each group, we optimise the desirability index by generating a candidate best number of clusters from each data set, taking the average desirability index for this number over all five data sets and choosing the number which yields the highes average value. The optimal numbers of clusters are used for clustering the data sets from the other group (cf. Table 6.1). The minimum number of clusters is 5 (5% of all variables), while 50 (50 % of the variables) is the maximal number of clusters. If more clusters were allowed, we would encourage the existence of clusters with one element only, a characteristic that should be avoided.

number of clusters computed on	data sets 1 - 5	data sets 6 - 10
number of clusters used for	data sets 6 - 10	data sets 1 - 5

Table 6.1: The number of clusters is determined in two different parts of the data sets. The result is used for the clustering of data sets in the other group.

The maximum desirability index is obtained by the number of clusters shown in Tables B.1 - B.6 in the Appendix. PCC requires a smaller number of clusters (mostly between 12 and 16 clusters) than the other similarity measures (SMC mostly between 30 and 49, JC mostly between 29 and 50, FMC1 mostly between 39 and 50 and FMC2 mostly between 33 and 44).

All final results were achieved by employing these optimal numbers of clusters in each case.

The ranking of similarity measures according to their desirability index is the same for all scenarios and all θ. PCC yields the highest desirability values, followed by JC. The two flexible matching coefficients perform slightly worse than JC, with FMC1 giving better desirability index values than FMC2. SMC gives the worst results. The respective results for the ten data sets (with $\theta = 1.1$ as an exemplary value as all plots look similar) of a scenario are displayed in Figure 6.1.

Desirability indices increase with increasing θ. This is not strictly true from each level of θ to the next as the risk change in the data for small steps (0.2) in θ is not always reflected in the desirability index, but the tendency is visible.

The phenomenon of equal ranking over all levels of θ and all scenarios is also true for the values of the single measures f_1 and f_2. Recall that they give information about the existence of a blocking structure and about the fraction of clusters with just one element, respectively. Their values do not increase or decrease with changing θ as they evaluate the partition in general and do not take the characteristics of the causative SNPs into account. For the exemplary data sets in all scenarios with $\theta = 1.1$, results are given in the top left and top right part of Figure 6.2 (and Figures B.1 and B.2 in the Appendix). The variance across data sets in one scenario is very small, thus, the boxes in the plots are narrow. The gap between the desirability index of PCC and the other measures for f_2 is remarkable. While values of PCC are close to the optimum 1, no other measure reaches values above 0.5. Thus, PCC is especially useful if clusters with one element should be avoided by all means. It can be concluded that the overall best performance of PCC regarding the desirability index is mainly due to its excellent performance for f_1 and f_2.

Measure f_3 reflecting the difference of clusters containing the causative SNPs between the case and in the control collective increases with θ as expected. PCC is not the best for f_3: JC yields higher values (i.e., more differences between cases and controls). For the scenario with one two-way interaction, FMC1 comes second best (cf. bottom left part of Figure 6.2). For two two-way interactions, the ranking is JC, PCC, FMC1, FMC2 and SMC (cf. bottom left part of Figure B.1), while FMC2 performs better than FMC1 with respect to f_3 for three two-way interactions (cf. bottom left part of Figure B.2). In some cases, the variances are high yielding broad boxes or long whiskers. The inconsistency shows that the allocation of causative SNPs into clusters is unstable across data sets, and that small changes in the data structure seem to be responsible for completely different values of f_3. Thus, the measure is not robust.

For f_4 (giving information about the similarity of clusters without causative SNPs in

cluster number	SMC	JC	FMC1	FMC2	PCC
10	0.1203	0.1461	0.1203	0.1203	0.5879
27	0.0660	0.0671	0.0000	0.0434	0.5006

Table 6.2: Desirability index achieved by clusterings based on the different similarity measures for two different cluster numbers.

cases and controls), all similarity measures give satisfactory results. The variance is small (except for three two-way interactions and the flexible matching coefficients with moderate variances), and, again, SMC gives the lowest desirability index values.

FMC1 and FMC2 are not the best similarity measures according to all quality measures, but they still perform better than SMC. Thus, the generalisation of SMC improves the cluster results.

6.1.2 GENICA

The GENICA data have been analysed previously by means of cluster analysis (Selinski and Ickstadt (2008), Justenhoven et al. (2008), Ickstadt et al. (2006)), but without the aid of objective functions to assess the quality of the resulting clustering. Thus, we analyse the GENICA data again in this thesis. Even though the causative SNPs are not known, we assume from previous results (Justenhoven et al., 2004) that SNPs ERCC2.6540 and ERCC2.18880 are associated with the disease risk and serve as causative SNPs for measures f_3 and f_4. We do not have reliable information about an underlying blocking structure either, but we assume that SNPs from the same gene form a block. Thus, we have 41 blocks, most consisting of one or two SNPs only.

Due to the small size of the data set, we omit the search for optimal cluster numbers and use the information from the simulated data. Thus, we choose 10 and 27 clusters. The first number is close to a good fraction of clusters compared to the number of variables for PCC clustering, while the higher number matches a good cluster number proportion for all other coefficients. The results are displayed in Table 6.2.

Interestingly, for the GENICA data, small cluster numbers give better results for SMC,

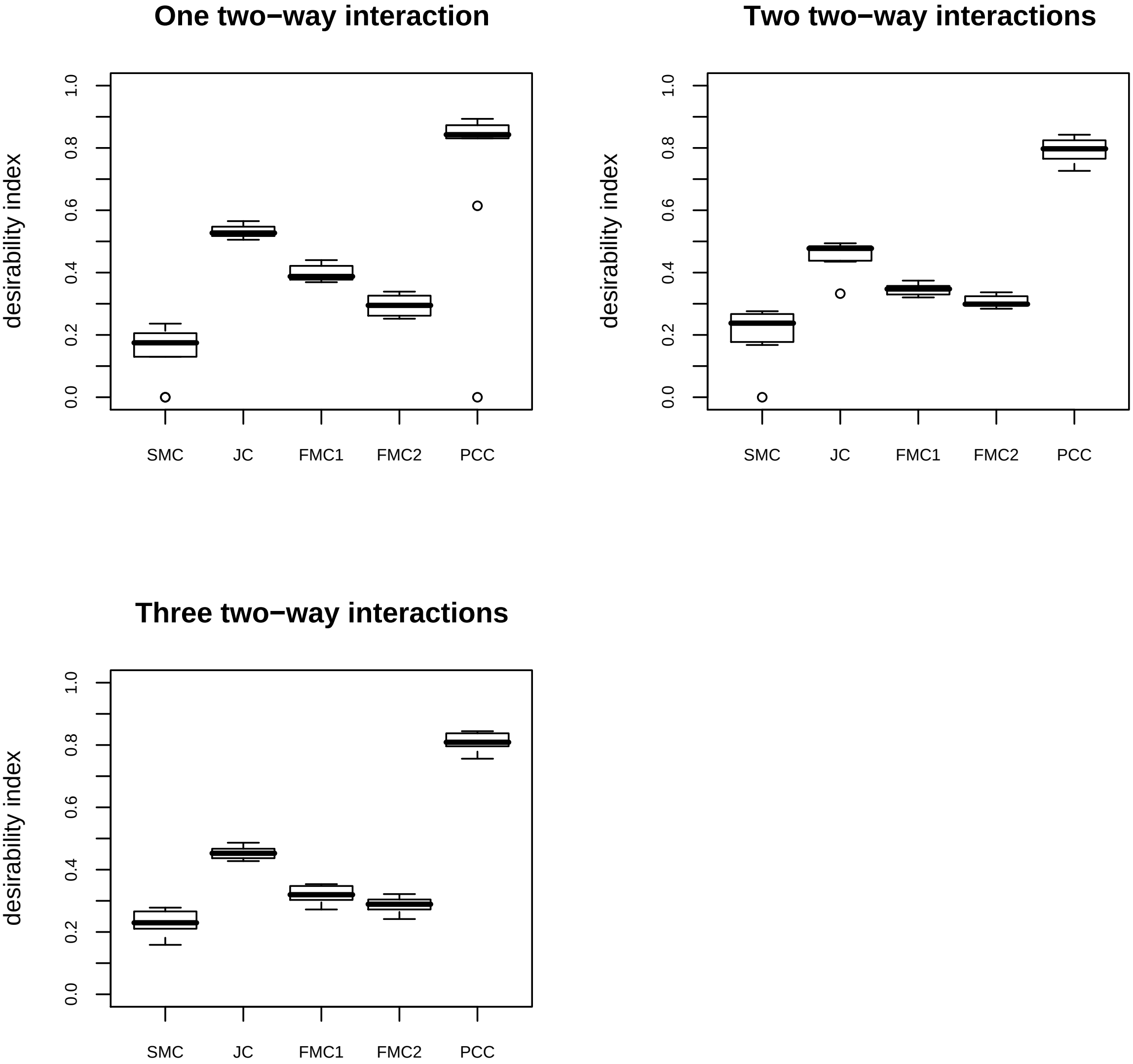

Figure 6.1: Desirability index for clusterings based on the different similarity measures for ten data sets and $\theta = 1.1$; top left: one two-way interaction, top right: two two-way interactions, bottom: three two-way interactions.

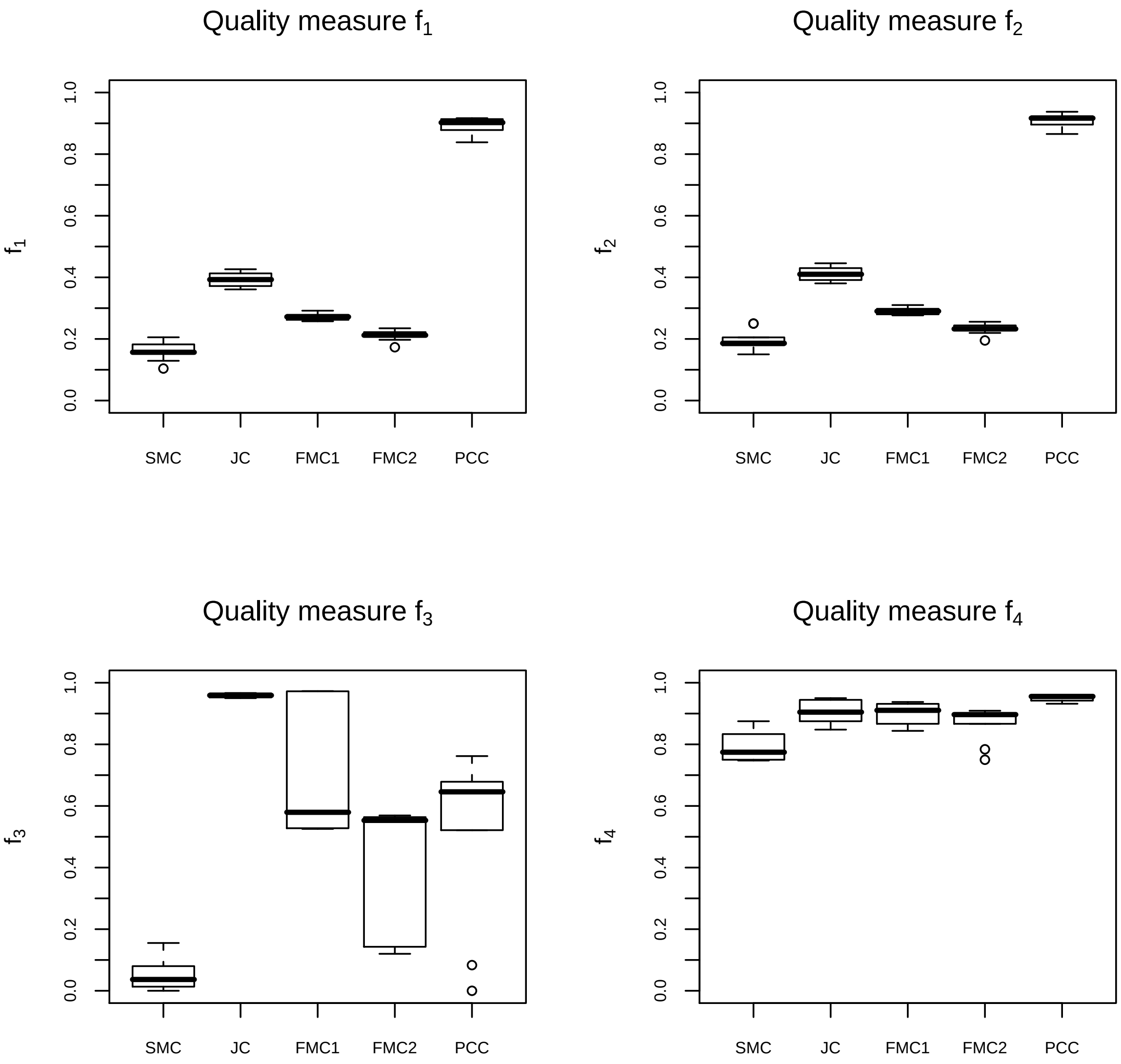

Figure 6.2: Quality measures for the different similarity measures for data sets with one two-way interaction and $\theta = 1.1$.

JC and the FMCs, while PCC gives stable results for both numbers of clusters. All similarity measures except PCC yield unacceptable results. In order to check how the different measures react to the different number of clusters, Figure 6.3 shows the values of f_1- f_4 on cluster numbers between 2 (3% of all variables) and 31 ($\sim$ 50% of all variables). The respective desirability index can be found in Figure 6.4.

As we assess f_1 and f_2 separately for cases and controls, the two plots in the top row of Figure 6.3 contain twice as many points than the plots in the bottom row. Some points are not displayed as they lie on top of each other. It can be seen that PCC gives higher results for f_1, f_2 and f_3 measures over the whole range of number of clusters. The other measures (with JC giving slightly higher values than SMC and the FMCs) yield decreasing values for an increasing number of clusters. However, there seem to be break points where a monotonous decrease is interrupted and picked up at a higher level of the measure again, e.g. in the top left plot of Figure 6.3 at 28 clusters: JC showed decreasing values before, but at 28 clusters, the values rise spontaneously to start another decrease. A similar behaviour can be seen for the desirability index (Figure 6.4). It seems that some changes in the clustering cause abrupt changes in quality (e.g., if two cluster with one element are fused).

For the GENICA data, only PCC makes an adequate choice to achieve a good clustering according to f_1- f_3. However, FMC2 and JC outperform it for f_4. The statements on clustering quality concerning blocking structure and causative SNPs are doubtful.

For the HapMap data, we do not carry out a cluster analysis as quality assessment of clusterings as we do not have information about causative SNPs.

6.1.3 Supervised Clustering

As PCC gave the most satisfying results for clustering SNPs, it is also employed for clustering patients. The number of clusters is set to 18. It allows for different clusters for cases belonging to the different interactions.

Variable selection for the simulation did not improve the purity essentially. For most data sets, only between two and four variables were deleted before the average purity could not be improved. The effect size did not influence the purity within the different scenarios, thus, we display mean results in Table 6.3. More interactions slightly increased purity values. Still, on average, all clusters contain both cases and controls, and a discrimination on bases of the different clusters is not sensible. This is due to the

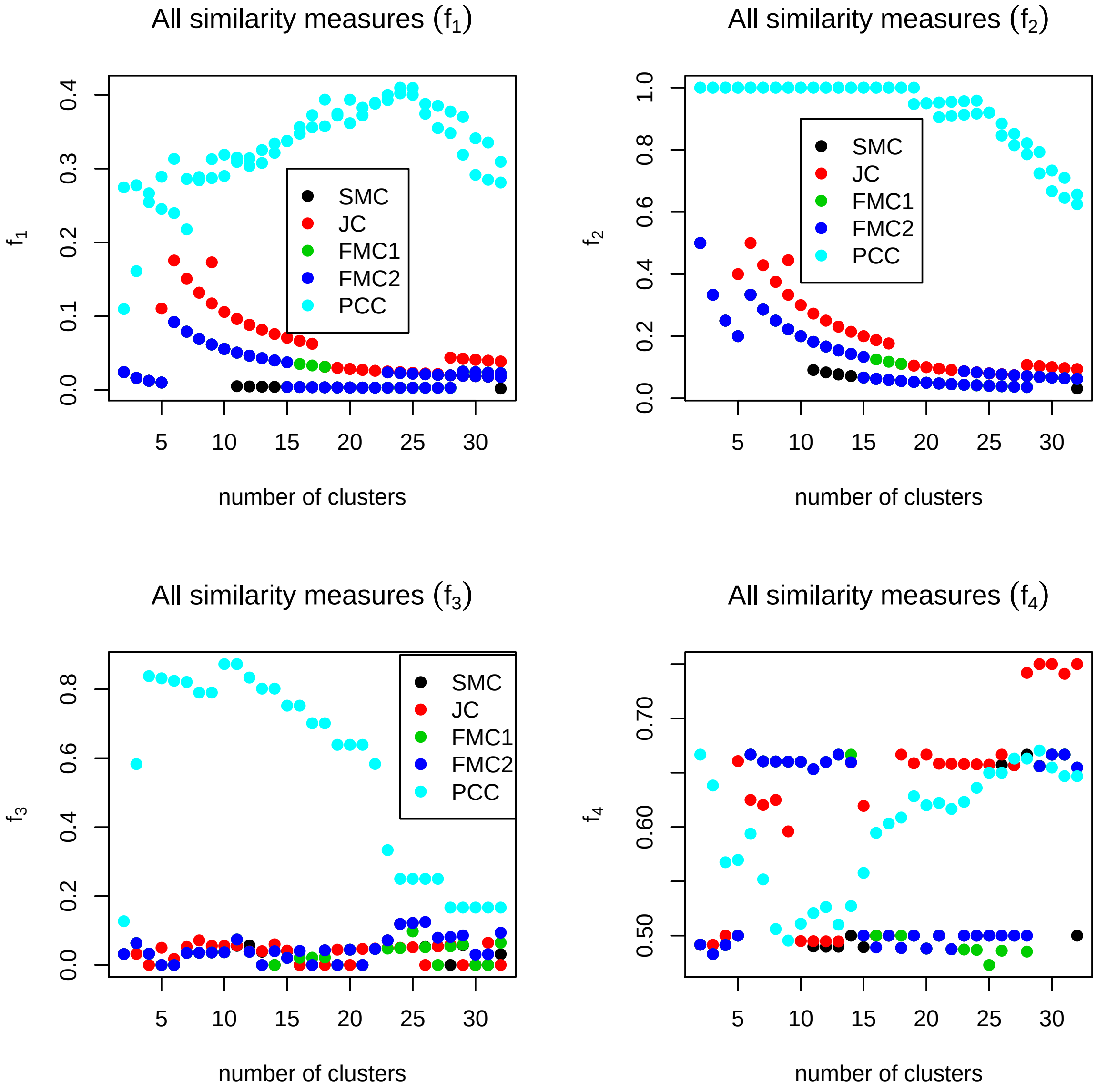

Figure 6.3: Comparison of clusterings based on different similarity measures for f_1 (top left), f_2 (top right) , f_3 (bottom left) and f_4 (bottom right) for different numbers of clusters. Note the different limits of the y axis for the different plots.

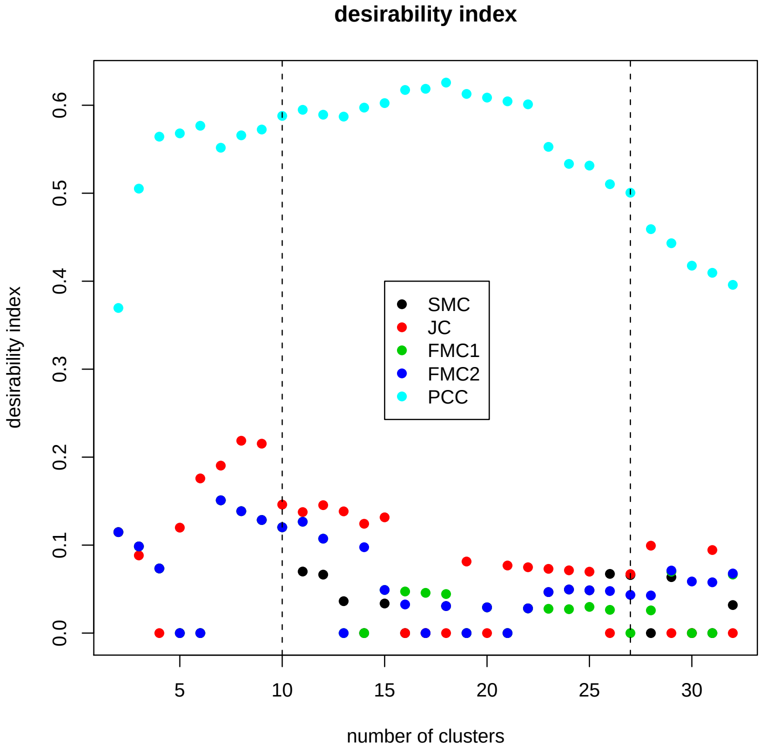

Figure 6.4: Comparison of clusterings based on different similarity measures the desirability index for different numbers of clusters.

scenario	mean purity	sd.dev purity	# variables removed	c-SNPs removed
one two-way	0.5455	0.0091	2-4	yes (1,1)
two two-way	0.5648	0.0093	2-4	yes (1,1,0,0)
three two-way	0.5660	0.0085	2-4	yes (0, 2, 3, 2, 5, 1)

Table 6.3: Characteristics of supervised clusterings. The abbreviation sd.dev denotes standard deviation.

weighting regime of the overall purity: There are several smaller clusters with a higher purity value, but the bigger clusters with more weight contain mostly even proportions of cases and controls. This reflects the idea that the association of the disease and the genetic profile contains a local component.
We set the cluster number to 18 due to background considerations. Subsequent analysis of different cluster numbers showed that the results do not change substantially over different cluster numbers.
The GENICA data set is not well separable. With 61 variables, an average purity of 0.5699 can be achieved.

The Hapmap data can easily be divided into pure clusters with an overall purity of 1, a result constant for several numbers of clusters (tested for 10 - 20 clusters). The only two variables that are deleted in every run are SNP_A-1859383 and SNP_A-1854291.

6.2 Classification

The simulated data sets are used to find suitable threshold values for all methods that require a parameter specification. To guarantee that the results will be generalisable, we split each data set of a simulation setting into training and test data (ratio 2:1). Via grid search, different combinations of parameters are chosen to fit models on the training data. Afterwards, the results are evaluated on the test data. As this is performed on ten data sets per setting, the results have to be averages over all data sets. Detailed results can be found in Subsection C.2.2 in the Appendix. With the chosen parameters, models are now build on the complete data set and evaluated on a different data set of the same setting, so that each complete data set serves as training and as test data once. The results gained from the simulated data will be used to choose thresholds and parameters for the real world data sets.
In the remaining chapters, we will use the following abbreviations:

- FC = feature construction
- LC = local class
- NC = naive associative classification
- ACV = AC vote (constant weights)

- ACSC = AC vote (weights = $supp \cdot conf$)
- LR = logic regression
- LLR = local logic regression
- CART = CART
- RF = Random Forest
- MCR = misclassification rate

6.2.1 Simulation

Each of the ten equally designed studies ($(\mathbf{y}, \mathbf{X})_1, \ldots, (\mathbf{y}, \mathbf{X})_{10}$) per simulation scenario reflects the same properties. Thus, we can use them for parameter specification as well as for prediction if we follow certain principles: We do not test a model on the same data that it is built on and we search for optimal parameters on data sets we do not use the parameters for. An overview over the structure is given in the following tables.

parameter specification computed on	$(\mathbf{y}, \mathbf{X})_1 - (\mathbf{y}, \mathbf{X})_5$	$(\mathbf{y}, \mathbf{X})_6 - (\mathbf{y}, \mathbf{X})_{10}$
parameter specification used for	$(\mathbf{y}, \mathbf{X})_6 - (\mathbf{y}, \mathbf{X})_{10}$	$(\mathbf{y}, \mathbf{X})_1 - (\mathbf{y}, \mathbf{X})_5$

With the optimal parameter combination (if necessary) obtained on one half of the data sets, its setting is used on the other half. The division into test and training sets can be found here:

training	$(\mathbf{y}, \mathbf{X})_1$	$(\mathbf{y}, \mathbf{X})_2$	$(\mathbf{y}, \mathbf{X})_3$	$(\mathbf{y}, \mathbf{X})_4$	$(\mathbf{y}, \mathbf{X})_5$
test	$(\mathbf{y}, \mathbf{X})_5$	$(\mathbf{y}, \mathbf{X})_1$	$(\mathbf{y}, \mathbf{X})_2$	$(\mathbf{y}, \mathbf{X})_3$	$(\mathbf{y}, \mathbf{X})_4$
training	$(\mathbf{y}, \mathbf{X})_6$	$(\mathbf{y}, \mathbf{X})_7$	$(\mathbf{y}, \mathbf{X})_8$	$(\mathbf{y}, \mathbf{X})_9$	$(\mathbf{y}, \mathbf{X})_{10}$
test	$(\mathbf{y}, \mathbf{X})_{10}$	$(\mathbf{y}, \mathbf{X})_6$	$(\mathbf{y}, \mathbf{X})_7$	$(\mathbf{y}, \mathbf{X})_8$	$(\mathbf{y}, \mathbf{X})_9$

The analysis of the simulated data has two main focuses: We want to compare different classification approaches according to their MCR and, implicity, give information about how large an effect size has to be in order to leave detectable differences in the simulated data sets.

With one causative two-way interaction and an effect size of 0.5, a meaningful classification into cases and controls is not possible with the employed classification methods. As can be seen in Figure 6.5, all boxplots either cross or touch the dashed line indicating an MCR of 0.5 (equal to random guessing in the balanced case control design).
FC yields unacceptable results on the simulated data. Even with increasing effect size, the classification results do not improve, and MCRs remain constantly around 0.5 with very small variance. LC gives median MCRs below 0.5 for all effect sizes, but the decrease in MCR with increasing θ is not pronounced. In seven out of eight scenarios, at least one MCR for one data set lies above 0.5. In contrast to this result, CART, RF and LR yield satisfactory results for the given circumstances. The achieved median MCRs decrease monotonously with increasing θ, while the variance of MCRs of different data sets is small. LR gives the best performance as expected. The LLR as implemented is not suitable for the analysis of the simulated data as it achieves MCRs around 0.5.
The NC approach works surprisingly well, especially compared to ACV. Its MCRs do decrease with increasing θ, and the overall performance over the 10 data sets is stable. The only drawback is that for three data sets, MCRs are very high (0.601, 0.739, 0.691), even worse than random guessing. The ACV approach shows the most unstable results of all methods. The achieved MCRs are higher than for NC and ACSC, with maximal MCRs close to 0.5 in all settings. Still, the minimal MCR is similar to CART and LR.
The analysis of the simulated data shows that even with a substantial effect size in the population, in an association study the effect is hard to detect, a finding which is consistent with results from real-world SNP studies. With an increasing number of causative interactions (cf. Figure 6.6 for two and Figure C.1 for three two-way interactions), the MCRs rise for all methods. The minimum mean MCR (on highest effect sizes) is 0.3342 for two causative two-way interactions and 0.3752 for three causative two-way interactions (both achieved by LR). FC, LLR and ACV yield stable uninformative classification results for both interaction scenarios. LC gives median results around 0.5, too, but with considerable variation into both directions, a characteristic which gets more pronounced in the case of three two-way interactions. It results into LC yielding the lowest MCRs for the small effect sizes for specific data sets.
NC shows good performance for higher effect sizes, and ACSC gives good results with decreasing MCRs with increasing effect sizes with only one exception in each scenario.

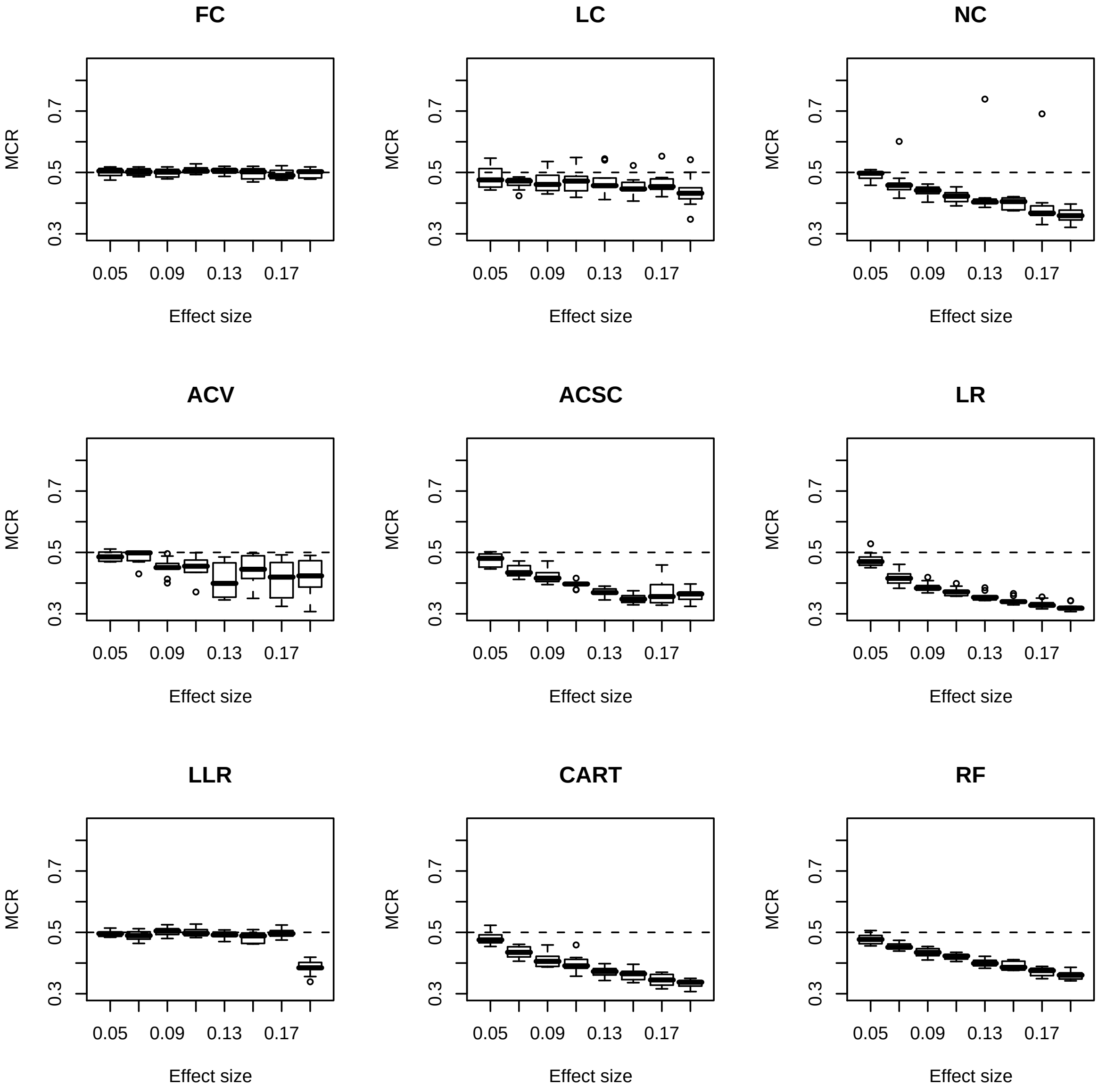

Figure 6.5: Misclassification rates for the simulated data achieved by the different classification methods for one causative two-way interaction.

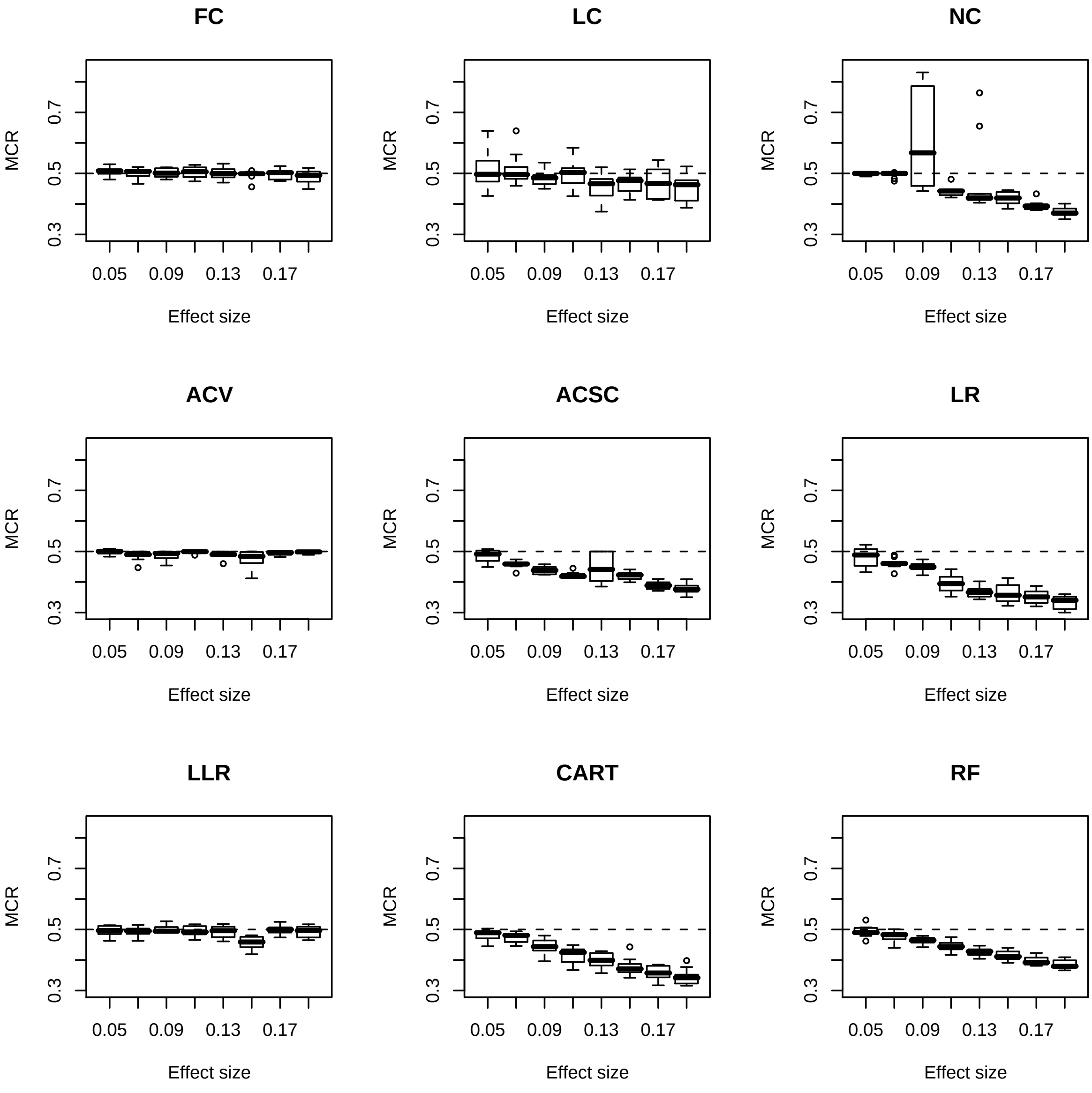

Figure 6.6: Misclassification rates for the simulated data achieved by the different classification methods for two causative two-way interactions.

LR and CART give the lowest MCRs, while the MCRs achieved by RF are slightly higher, but less variable.

6.2.2 HapMap

The choice of parameters for the methods for the HapMap data is not resulting from an optimisation process as the data set consists only of 90 observations. Instead, we choose parameters based on the results from the simulation with two causative two-way interactions, on the size of the set of itemsets or rules and on the fact that the HapMap data can be divided into the two classes more easily than in the case of association study data. Parameter choices can be found in (cf. Table 6.4). For LC, the simulation suggests 0.6 for the frequent itemsets of all lengths. For FC, we lower the support as we assume that more features based on informative frequent itemsets can improve prediction. For all association rules, the high thresholds for confidence account for the relevant information contained in the data. However, as the number of observations is smaller than in the simulation, minimum support values are kept as low as possible to still yield a sufficiently small set of association rules (in this case, ~ 500 rules). The threshold specifications for ACSC are less strict because the method is the one amongst the three that can build best on more rules.
For logic regression, we choose 8 leaves (more than in the simulation) to account for the separability of the data without assuming a structure that is too simple. In LLR, λ is very small because the observations are supposed to show high similarities. Thus, with a small λ we ensure the locality of the local approach.

Figure 6.7 shows the misclassification rates, obtained by nine-fold cross validation. ACSC and RF classify only one (two, respectively) observations to the wrong class, thus, their performance is excellent. LR is second best and superior to LLR. CART and LC give a similar performance, both reveal problems to separate the two classes. FC gives MCRs around 0.2 on average with the most symmetric variation across the different parts of the data due to cross validation. The NC and ACV do not give satisfying results as they fail to distinguish between the two classes.

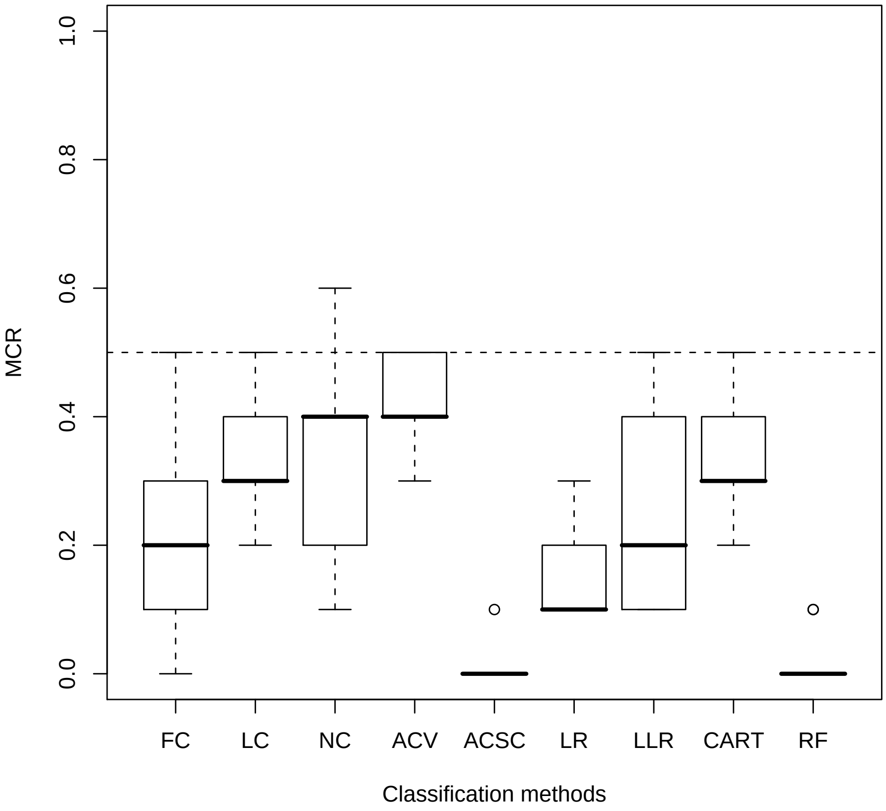

Figure 6.7: Misclassification rates for the subset of HapMap data achieved by the different classification methods.

Method	parameter(s)		
FC	$supp(f_i^k)$= 0.55,	$i = 1, 2, 3$	
LC	$supp(f_i^k)$= 0.6,	$i = 1, 2, 3$	
NC	$supp$ = 0.3,	$conf$=0.9 (both)	
ACV	$supp$ = 0.3,	$conf$=0.9 (both), $\gamma = 0.5$	
ACSC	$supp$ = 0.1,	$conf$=0.7 (both)	
LR	leaves = 8		
LLR	leaves = 10,	λ = 0.1	
RF	trees = 500,	sample size = n,	variables at split = $\sqrt{157}$

Table 6.4: Parameter specification of the different classification methods for the analysis of the HapMap data.

6.2.3 GENICA

The GENICA data have been analysed before (Justenhoven et al. (2008), Ickstadt et al. (2006), Müller et al. (2008)), thus, the parameter chosen are based on this experience. They can be found in Table 6.5. It was established before that the observations in this data set are hard to classify (Justenhoven et al. (2004), Ickstadt et al. (2006), Schiffner et al. (2009), Nunkesser et al. (2007)). It can be seen in Figure 6.8 that the methods presented in this thesis reveal similar results.

NC gives completely unacceptable MCRs (over 0.8). The specificity is 0 for all cross validation runs, i.e., no control is correctly classified. Thus, the best rule for controls is always a case rule, while the only times when the best rule is a control rule occurs for case observations, leading to an even worse classification result. LLR, with $\lambda = 0.5$ gives results close to random guessing and is thus not sensible for the data. LC and CART give similar results and seem to be best suited for the data set, even though their MCRs are still above 0.4. Logic regression as the standard classification method for SNP data gives slightly higher MCRs. It is interesting to know if the difference is significant, therefore, we apply the test to compare MCRs (Dietterich, 1998). The null

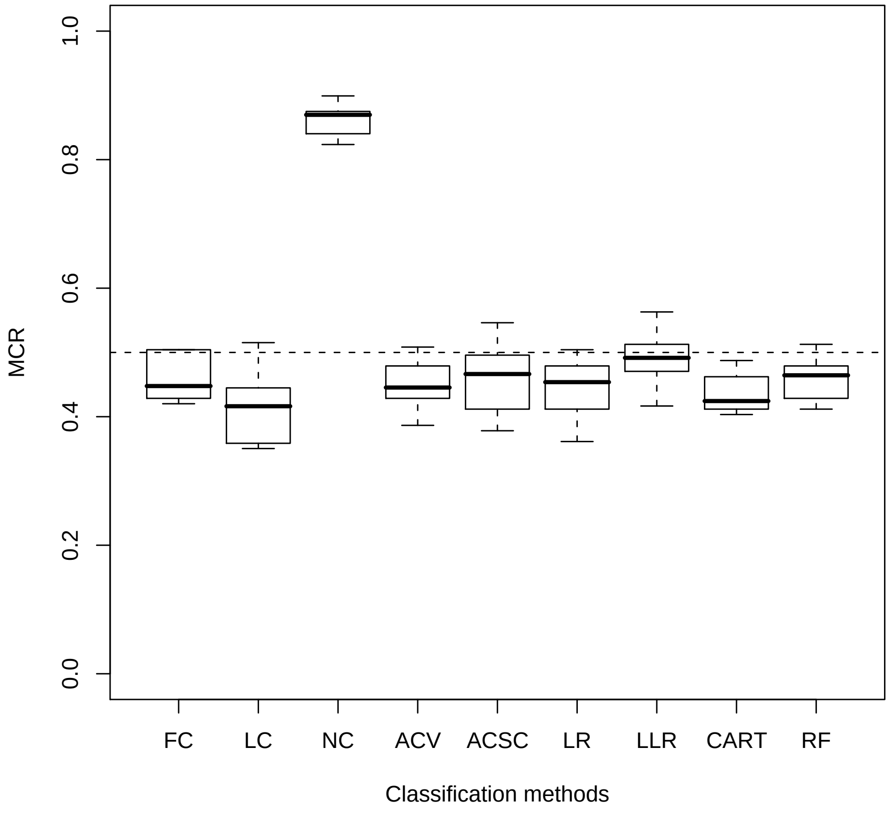

Figure 6.8: Misclassification rates for the subset of GENICA data achieved by the different classification methods.

Method	parameter(s)			
FC	$supp(f_1^k)$= 0.6,	$supp(f_i^k)$=0.85	$i = 2, 3$	
LC	$supp(f_i^k)$= 0.6,	$i = 1, 2, 3$		
NC	$supp$ = 0.07,	$conf$=0.6 (both)		
ACV	$supp_{ca}$ = 0.04,	$conf_{ca}$=0.65 ,	$supp_{co}$ = 0.15,	$conf_{ca}$=0.6, $\gamma = 0.2$
ACSC	$supp_{ca}$ = 0.04,	$conf_{ca}$=0.65 ,	$supp_{co}$ = 0.15,	$conf_{co}$=0.6
LR	leaves = 8			
LLR	leaves= 10,	λ = 0.5		
RF	trees = 500,	sample size = n,	variables at split = $\sqrt{63}$	

Table 6.5: Parameter specification of the different classification methods for the analysis of the GENICA data.

and alternative hypotheses are

$$H_0 : MCR_{LC} = MCR_{LR} \quad \text{vs.} \quad H_1 : MCR_{LC} \neq MCR_{LR},$$

saying that both MCRs are equal versus there is a difference between them. The division of the GENICA data used for cross validation is now neglected, and new subsets are formed for the test. The test statistics yields a value of 1.5801 and is smaller than the respective 0.05/2 - quantile of a t distribution with 5 degrees of freedom (2.5706). The p-value of 0.9125 additionally indicates that the classification results do not differ significantly between LR and LC.

6.3 10000-SNPs Simulation

The largest data set we analyse in this thesis is the 10000-SNPs data set. Logic regression and local logic regression are not applicable anymore. We omit the methods that rely on frequent itemsets as they require many parameter settings and did not achieve good results on the smaller simulation. Thus, we concentrate on the methods based on association rules. The implementation of association rule search in R is not able to cope

Method	MCR	Sensitivity	Specificity
NC	0.3343	0.6325	0.6990
ACV ($\gamma = 0.2$)	0.4910	1.0000	0.0181
ACSC	0.1837	0.9217	0.7108
CART	0.4548	0.5542	0.5361

Table 6.6: Based on 392 rules that satisfied support = 0.4 and confidence = 0.6.

with bigger data sets, thus, we implemented a manually programmed version of association rules search that is able to cope with the amount of data. Once the mined rule set reached a reasonable size, the subsequent analysis steps run quickly. According to the analysis from the smaller data sets and feasibility, we chose a minimum support of 0.4, a minimum confidence of 0.6 and, for the ACV approach, a voting fraction of γ=0.2. The data were divided into a test and a training data set. In total, 392 rules of one or two items in the antecedent were mined from the training data. The classification results on the test data can be found in Table 6.6.

ACSC gives the best misclassification rate which is even below the MCR achieved in the same scenario with less SNPs.

CHAPTER 7

Summary and Discussion

The risk to develop a disease is not only due to environmental circumstances, but supposedly to a high degree related to one's genetic predisposition. Based on the genetic information, an improved understanding of the disease mechanisms can lead to timely preventive action and better suited treatment.

Single nucleotide polymorphisms (SNPs) that are a special type of genetic data can be found in abundance in the human genome. They are part of the genetically triggered process and are comparatively easy to assess. Thus, they have been the target for association studies for a couple of years by now.

However, the analysis of SNP data is still a hard task, e.g., due to their high numbers and their likely small influence when regarded in univariate analysis. In this thesis, we have introduced methods that can cope with the high dimensionality of SNP data, with interactions that affect disease risk, and with the possibility that there are several genetic ways that alter the disease risk. These procedures are applied to real world data sets and to a simulation study that we designed to contain all characteristics that are assumed for SNP data.

Two types of the analysis are employed: Cluster and discrimination analysis. We have performed cluster analysis to find suitable partitions of all SNPs in a study. As a first step, we have defined "suitability": A good partition should allow for the possible detection of linkage disequilibrium (or the underlying blocking structure) of the variables, it should avoid clusters with only one element, and help to distinguish between causative and non-causative SNPs in the case, but not in the control collective. We developed one measure for each of the first two goals and two for the third goal. All measures are combined in a desirability index to allow for an overall comparison between clusterings. As we need information about possible linkage disequilibrium and causative SNPs for the desirability index, only the simulated data and the GENICA data set have been taken into account.

We showed that clusterings based on Pearson's corrected contingency coefficient (PCC) yields better results according to the desirability index than partitions by matching coefficient for all settings of the simulated data as well as for the GENICA data. Still, the result is mainly due to the good performance of PCC for the first two goals (detecting blocking structure and avoiding clusters with one element). For the different grouping of causative SNPs between cases and controls, the partitions based on Jaccard's coefficient and the Flexible Matching coefficient showed equal or better results than PCC.

A different cluster approach has been applied that utilises the clustering of observations for a classification into cases and controls. It has shown no advantages over the classification methods.

A classical classification method for SNP data is logic regression that specifically searches for combinations of SNP interactions as classifier. We developed a local version of logic regression (LLR) that takes the similarity of new observations to given observations within the training data into account and builds individual classification models for each new observation. This structure allows to include different causative genotypes that represent alternative ways of affecting disease risk in one model. In addition, we developed and adapted several classification methods based on frequent itemsets and association rules, two related concepts from the field of data mining (originally designed to handle huge amounts of data). Frequent itemsets, translated into the genetic data world, can be interpreted as genetic profiles. Association rules as we use them can be considered as descriptive predictions of the disease status given a certain genetic profile. If classification and association rules are combined into one method, this is called associative classification.

The most promising of our approaches, ACSC (associative classification with association rules weighted according to their quality), classifies a new observation as case or control according to a weighted voting of all applicable association rules found in a training data set. Other methods derived from association rules that we adapted to the usage for SNP data are naive classification (NC, classifying a new observation according to the best association rule only, also known as CBA) as well as a newly developed voting of rules (ACV) that is based on the optimal proportion of case and control rules. Local class (LC) and feature construction (FC) are both based on frequent itemsets: LC divides a data set into subgroups of certain genetic profiles first and builds a separate classifier in each subgroup afterwards. FC searches frequent itemsets in an initial step and uses these interactions as new input variables in a new data set to built the final classifier. Subsequently, the classifier is built on this newly created data set. For comparisons with well established methods, we also employ CART, Random Forests based on CART trees (RF), and logic regression in all classification scenarios.

All methods for classification of the patients were first applied to the simulated data. Even for the largest effect size of 1.9 and the easiest interaction scenario of one two-way interaction, the best result was a misclassification rate of 0.307 (achieved by ACV, logic regression and CART). This exemplifies that population-based association studies already limit the ability to detect the differences between cases and controls. In reality, this effect is still harder to detect as external factors as lifestyle and environment might affect the disease risk and act different for different genetic susceptibilities. For the GENICA data, another problem is that the choice of SNPs reflects only a fraction of all possible SNPs, thus, important genetic factors might be missing as well. Both reasons give explanations for the classification results than were worse than for the simulated data. A satisfying discrimination according to disease status was not possible on basis of the given data.

In contrast to these data sets, the HapMap data can be classified almost perfectly. Note that due to the change of outcome (ethnicity instead of disease status), the circumstances are quite different than for the other data sets, and only 90 observations and 157 variables are analysed. Nonetheless, the data structure is still the same. ACSC performs best and yields a misclassification rate of 0.01.

The simulation study revealed that some of the newly developed methods can still be improved. FC and LC did not perform better than random guessing and yield misclassification rates around 0.5. Both make use of frequent itemsets corresponding to frequent genetic profiles or SNP interactions. For FC, this results in a number of interactions (used as new input variables for a subsequent classification) that are not specific enough to yield a good classification. LC which formed subclasses in each of which CART was used as a classifier never performed substantially better than CART itself. Apparently, CART already allows for a sufficient amount of locality, and several splits can be interpreted as reflecting interactions. Still, for the HapMap data, the performance of both LC and FC was similar to the overall performance of all methods.

The local logic regression (LLR) is not completely comparable to logic regression. More variability is introduced due to optimising a further parameter which leads to less stable results. We assume that with an additional concept like boosting, the classifier could be stablised. As LLR is much more computer intensive, we used a greedy approach for the model building, while logic regression was itself performed using simulated annealing. The influence of the mechanism on the classification result is supposed to be substantial and should be kept in mind when reporting classification results. Logic regression, nevertheless, proved to be the best suited method over all data sets for analysing SNPs. Its only drawback is that it is not applicable to genome-wide data.

As a step towards the analysis of genome-wide data, we also analysed a simulated data set with 10 000 SNPs. The methods based on association rules were still applicable, and ACSC gave even better classification results than on the data sets with the same genetic model with less SNPs. This might be due to the fact that we retained the dependency structure of the SNP blocks, which results in a higher number of association rules of similar prediction power. It can be concluded that ACSC can not only handle correlated variables (e.g., SNPs in linkage disequilibrium), but also uses the information to achieve better classification results. The other two methods based on frequent itemsets, NC and ACV, did not yield better results than before. Logic regression and LLR were no longer applicable, and CART does not yield satisfactory results.

An often quoted phrase is thus also true for the application of classification methods to genotype data: There is no free lunch (Wolpert, 1996). Still, we find that especially the methods based on association rules contain potential for the future analysis of high-dimensional genetic data. As they are sensitive to the choice of parameters, though, either prior knowledge about the SNP data at hand or a thorough optimisation of parameters is necessary to yield reliable results. ACSC, the best associative classification method in most settings, could be refined by changing the weighting scheme of the individual rules. Moreover, other interest measures than support and confidence could be used.

Our implementation of the algorithm for finding association rules in the 10 000 SNPs is rudimentary. Sophisticated programming could be used to improve its speed and storing capacities, thus, make it suitable for genome-wide SNP data which is the ultimate goal of SNP analysis.

An important idea has to be kept in mind: The analysis of genetic predisposition gains immensely by additional environmental information, e.g. about one's lifestyle (a susceptibility towards a certain substance does not come into play as long as one is not exposed to the substance). If clinical, environmental or epidemiological variables can be split into meaningful categories, the introduced methods based on frequent itemsets and association rules are flexible enough to incorporate the new kind of information as well, adding to their importance for future SNP research.

Appendix A

Simulation

A.1 Settings

The software package SNaP allows for different simulations, e.g. family data, but we concentrate on association data only. In the following, we show an exemplary jobfile containing all settings and an exemplified penetrance matrix. The blocks and their frequencies were generated at random.

```
# ========================== #
# Job and parameter settings #
# ========================== #


# General job specifications #
[General]
DataFilename              = ’TestCarolin.out’
SettingsFilename          = ’TestCarolin.set’
OutputType                = ’genotypes’
OutputFormat              = ’Compact’


# Study and sampling design specifications #
[Design]
```

```
TypeOfPhenoExpression      = 'qualitative'
StudyDesign                = 'individuals'
SamplingDesign             = 'separate'
NumberOfCases              = 500
NumberOfControls           = 500


# Genotype-phenotype model specifications #
[Model]
NumberOfLoci               = 2
NumberOfStates             = 9
Penetrances                =
    0.36000     0.20000     0.10000
    0.20000     0.14286     0.10000
    0.10000     0.10000     0.10000
BiallelicCheck             = 'n'
RemoveCausalSNPs           = 'n'
GenotypingError            = 0.00000
GenotypingErrorVisible     = 'n'
RandomSeed                 = 500


# Separator characters #
[Separators]
BehindStatus               = ','
BetweenHaplotypes          = ' '
BetweenBlocks              = ','
BetweenSNPs                = ','


# Specification of block structures and pre-settings #
[Blocks]
NumberOfBlocks             = 20

(Block)                                (Block)
Number                     = 1         Number                     = 2
```

```
Size                      = 4       Size                      = 6
SuscLocusPosition         = 0       SuscLocusPosition         = 0

{NoSuscHaplotypes}                  {NoSuscHaplotypes}
HtNumber                  = 5       HtNumber                  = 7
HtBlock                   = 1111    HtBlock                   = 111111
HtBlock                   = 2112    HtBlock                   = 211112
HtBlock                   = 2121    HtBlock                   = 111122
HtBlock                   = 1211    HtBlock                   = 111211
HtBlock                   = 1222    HtBlock                   = 212111
HtFrequ                   = 0.80000 HtBlock                   = 112112
HtFrequ                   = 0.16000 HtBlock                   = 122221
HtFrequ                   = 0.02000 HtFrequ                   = 0.25000
HtFrequ                   = 0.01000 HtFrequ                   = 0.38000
HtFrequ                   = 0.01000 HtFrequ                   = 0.05000
                                    HtFrequ                   = 0.05000
                                    HtFrequ                   = 0.11000
                                    HtFrequ                   = 0.07000
                                    HtFrequ                   = 0.09000

(Block)                              (Block)
Number                    = 3        Number                    = 4
Size                      = 4        Size                      = 6
SuscLocusPosition         = 2        SuscLocusPosition         = 0
SuscAlleleFrequ           = 0.35000
                                     {NoSuscHaplotypes}
{NoSuscHaplotypes}                   HtNumber                  = 6
HtNumber                  = 5        HtBlock                   = 111111
HtBlock                   = 1111     HtBlock                   = 111212
HtBlock                   = 2112     HtBlock                   = 211122
HtBlock                   = 1121     HtBlock                   = 112111
HtBlock                   = 1122     HtBlock                   = 221212
HtBlock                   = 2112     HtBlock                   = 222112
HtFrequ                   = 0.62000  HtFrequ                   = 0.49000
HtFrequ                   = 0.31000  HtFrequ                   = 0.23000
HtFrequ                   = 0.05000  HtFrequ                   = 0.20000
HtFrequ                  = 0.01000  HtFrequ                  = 0.04000
HtFrequ                  = 0.01000  HtFrequ                  = 0.02000
                                    HtFrequ                  = 0.02000



{SuscHaplotypes}
HtNumber                 = 4
HtBlock                  = 1211
HtBlock                  = 1212
HtBlock                  = 1222
HtBlock                  = 2221
HtFrequ                  = 0.49000
HtFrequ                  = 0.34000
HtFrequ                  = 0.02000
HtFrequ                  = 0.15000

(Block)                             (Block)
Number                   = 5        Number                   = 6
Size                     = 7        Size                     = 4
SuscLocusPosition        = 0        SuscLocusPosition        = 0

{NoSuscHaplotypes}                  {NoSuscHaplotypes}
HtNumber                 = 6        HtNumber                 = 3
HtBlock                  = 1111112  HtBlock                  = 1111
HtBlock                  = 1111121  HtBlock                  = 2212
HtBlock                  = 2111221  HtBlock                  = 2121
HtBlock                  = 2122111  HtFrequ                  = 0.48000
HtBlock                  = 2211122  HtFrequ                  = 0.44000
HtBlock                  = 1222122  HtFrequ                  = 0.08000
HtFrequ                  = 0.29000
HtFrequ                  = 0.15000
HtFrequ                  = 0.29000
HtFrequ                  = 0.25000
HtFrequ                  = 0.01000
HtFrequ                  = 0.01000
```

```
(Block)
Number                    = 7
Size                      = 4
SuscLocusPosition         = 0

{NoSuscHaplotypes}
HtNumber                  = 4
HtBlock                   = 1111
HtBlock                   = 2111
HtBlock                   = 1222
HtBlock                   = 2121
HtFrequ                   = 0.59000
HtFrequ                   = 0.24000
HtFrequ                   = 0.15000
HtFrequ                   = 0.02000
```

```
(Block)
Number                    = 8
Size                      = 5
SuscLocusPosition         = 0

{NoSuscHaplotypes}
HtNumber                  = 6
HtBlock                   = 11111
HtBlock                   = 21121
HtBlock                   = 11222
HtBlock                   = 22211
HtBlock                   = 12112
HtBlock                   = 12121
HtFrequ                   = 0.52000
HtFrequ                   = 0.32000
HtFrequ                   = 0.06000
HtFrequ                   = 0.08000
HtFrequ                   = 0.01000
HtFrequ                   = 0.01000
```

```
(Block)
Number                    = 9
Size                      = 4
SuscLocusPosition         = 0

{NoSuscHaplotypes}
HtNumber                  = 3
HtBlock                   = 1111
HtBlock                   = 1121
HtBlock                   = 2212
HtFrequ                   = 0.24000
HtFrequ                   = 0.55000
HtFrequ                   = 0.21000
```

```
(Block)
Number                    = 10
Size                      = 6
SuscLocusPosition         = 4
SuscAlleleFrequ           = 0.29000

{NoSuscHaplotypes}
HtNumber                  = 7
HtBlock                   = 111111
HtBlock                   = 111112
HtBlock                   = 121112
HtBlock                   = 122112
HtBlock                   = 211111
HtBlock                   = 211121
HtBlock                  = 221111
HtFrequ                  = 0.24000
HtFrequ                  = 0.11000
HtFrequ                  = 0.25000
HtFrequ                  = 0.11000
HtFrequ                  = 0.09000
HtFrequ                  = 0.10000
HtFrequ                  = 0.10000

{SuscHaplotypes}
HtNumber                 = 6
HtBlock                  = 111211
HtBlock                  = 121212
HtBlock                  = 111221
HtBlock                  = 212211
HtBlock                  = 212212
HtBlock                  = 121221
HtFrequ                  = 0.29000
HtFrequ                  = 0.22000
HtFrequ                  = 0.32000
HtFrequ                  = 0.14000
HtFrequ                  = 0.02000
HtFrequ                  = 0.01000
```

```
(Block)
Number                   = 11
Size                     = 3
SuscLocusPosition        = 0

{NoSuscHaplotypes}
HtNumber                 = 4
HtBlock                  = 211
HtBlock                  = 112
HtBlock                  = 121
HtBlock                  = 222
```

```
(Block)
Number                   = 12
Size                     = 8
SuscLocusPosition        = 0

{NoSuscHaplotypes}
HtNumber                 = 8
HtBlock                  = 21111111
HtBlock                  = 11121211
HtBlock                  = 22212222
HtBlock                  = 11112211
```

```
HtFrequ                    = 0.42000  HtBlock                    = 11211121
HtFrequ                    = 0.16000  HtBlock                    = 21211111
HtFrequ                    = 0.31000  HtBlock                    = 22112121
HtFrequ                    = 0.11000  HtBlock                    = 12212111
                                      HtFrequ                    = 0.46000
                                      HtFrequ                    = 0.34000
                                      HtFrequ                    = 0.13000
                                      HtFrequ                    = 0.03000
                                      HtFrequ                    = 0.01000
                                      HtFrequ                    = 0.01000
                                      HtFrequ                    = 0.01000
                                      HtFrequ                    = 0.01000



(Block)                               (Block)
Number                     = 13       Number                     = 14
Size                       = 4        Size                       = 5
SuscLocusPosition          = 0        SuscLocusPosition          = 0

{NoSuscHaplotypes}                    {NoSuscHaplotypes}
HtNumber                   = 4        HtNumber                   = 4
HtBlock                    = 1112     HtBlock                    = 11211
HtBlock                    = 1121     HtBlock                    = 21112
HtBlock                    = 2211     HtBlock                    = 12121
HtBlock                    = 2212     HtBlock                    = 12112
HtFrequ                    = 0.29000  HtFrequ                    = 0.36000
HtFrequ                    = 0.50000  HtFrequ                    = 0.18000
HtFrequ                    = 0.17000  HtFrequ                    = 0.27000
HtFrequ                    = 0.04000  HtFrequ                    = 0.19000

(Block)                               (Block)
Number                     = 15       Number                     = 16
Size                       = 3        Size                       = 7
SuscLocusPosition          = 0        SuscLocusPosition          = 0

{NoSuscHaplotypes}                    {NoSuscHaplotypes}
```

```
HtNumber                  = 3        HtNumber                  = 8
HtBlock                   = 211      HtBlock                   = 1121111
HtBlock                   = 112      HtBlock                   = 1111112
HtBlock                   = 121      HtBlock                   = 2111221
HtFrequ                   = 0.31000  HtBlock                   = 1111212
HtFrequ                   = 0.23000  HtBlock                   = 1212111
HtFrequ                   = 0.46000  HtBlock                   = 1112212
                                     HtBlock                   = 2211211
                                     HtBlock                   = 1211222
                                     HtFrequ                   = 0.39000
                                     HtFrequ                   = 0.40000
                                     HtFrequ                   = 0.11000
                                     HtFrequ                   = 0.02000
                                     HtFrequ                   = 0.02000
                                     HtFrequ                   = 0.01000
                                     HtFrequ                   = 0.04000
                                     HtFrequ                   = 0.01000

(Block)                               (Block)
Number                    = 17        Number                    = 18
Size                      = 5         Size                      = 5
SuscLocusPosition         = 0         SuscLocusPosition         = 0

{NoSuscHaplotypes}                    {NoSuscHaplotypes}
HtNumber                  = 4         HtNumber                  = 5
HtBlock                   = 11111     HtBlock                   = 11111
HtBlock                   = 21112     HtBlock                   = 22112
HtBlock                   = 21222     HtBlock                   = 11221
HtBlock                   = 12221     HtBlock                   = 11212
HtFrequ                   = 0.52000   HtBlock                   = 22221
HtFrequ                   = 0.23000   HtFrequ                   = 0.52000
HtFrequ                   = 0.17000   HtFrequ                   = 0.35000
HtFrequ                   = 0.08000   HtFrequ                   = 0.03000
                                      HtFrequ                   = 0.07000
                                      HtFrequ                   = 0.03000
```

```
(Block)
Number                     = 19
Size                       = 4
SuscLocusPosition          = 0

{NoSuscHaplotypes}
HtNumber                   = 4
HtBlock                    = 2111
HtBlock                    = 1112
HtBlock                    = 1121
HtBlock                    = 2222
HtFrequ                    = 0.58000
HtFrequ                    = 0.14000
HtFrequ                    = 0.05000
HtFrequ                    = 0.23000
```

```
(Block)
Number                     = 20
Size                       = 6
SuscLocusPosition          = 0

{NoSuscHaplotypes}
HtNumber                   = 7
HtBlock                    = 111121
HtBlock                    = 111112
HtBlock                    = 111211
HtBlock                    = 112111
HtBlock                    = 212122
HtBlock                    = 212212
HtBlock                    = 121112
HtFrequ                    = 0.35000
HtFrequ                    = 0.10000
HtFrequ                    = 0.05000
HtFrequ                    = 0.13000
HtFrequ                    = 0.16000
HtFrequ                    = 0.04000
HtFrequ                    = 0.17000
```

A.2 Results of Simulation

The different empirical distributions between cases and controls can be seen in Figure A.1 for the scenario containing three causative two-way interactions.

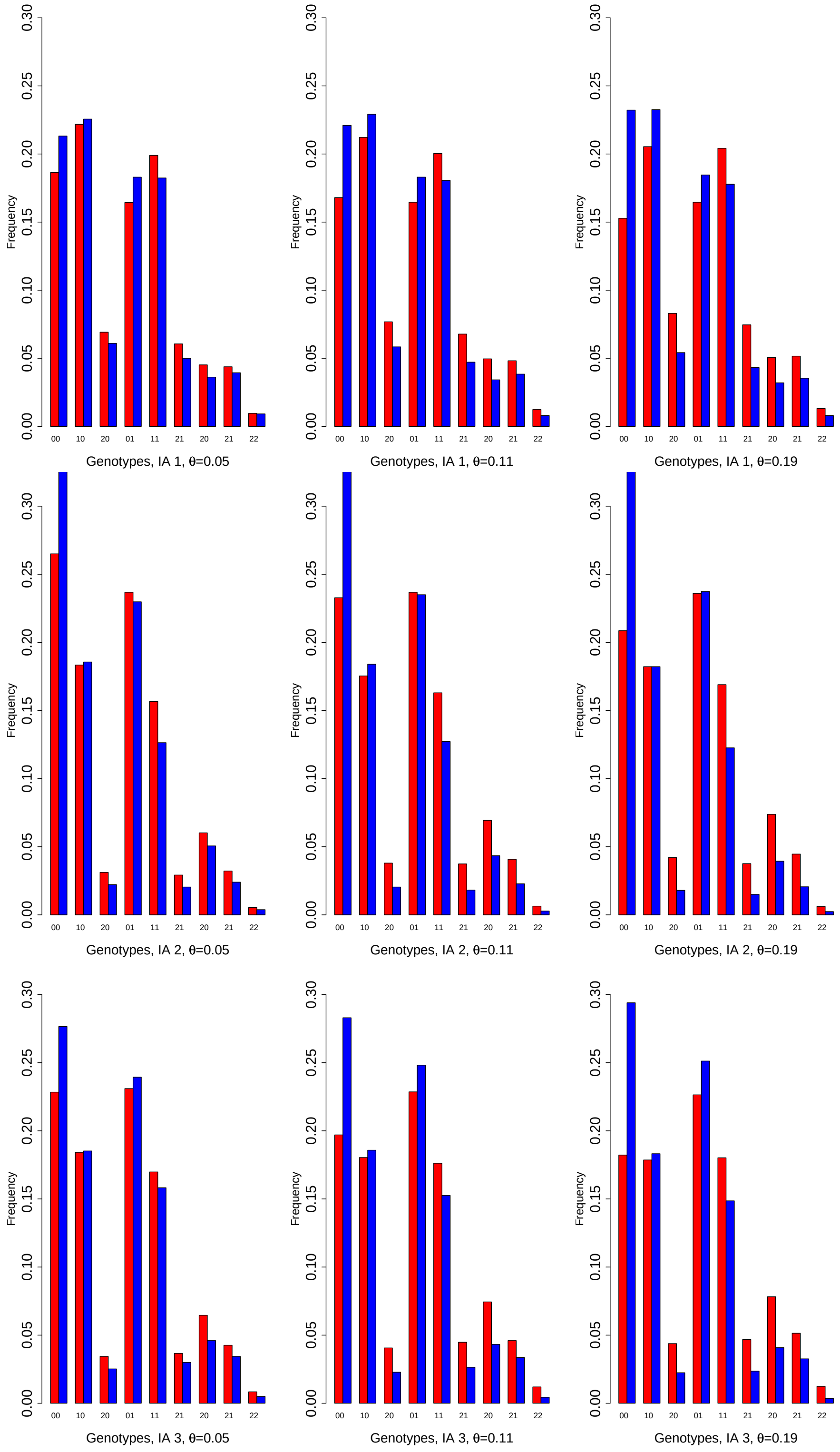

Figure A.1: Different genotype distributions for causative SNPs between cases (red) and controls (blue) for different effect sizes, given three causative SNP interactions.

APPENDIX B

Clustering

The simulated data was used to find the best number of clusters according to the desirability index. To achieve a generalisable result, we split the ten data sets from the same scenario that share the same effect size into two groups and obtained best cluster numbers in both groups separately. The results can be found in Tables B.1 - B.6.

The results for the quality measures and the desirability index were similar for all simulation scenarios. Figure 6.2 is displayed in Chapter 6. Figures B.1 and B.2 are given in the following.

effect size	SMC	JC	FMC1	FMC2	PCC
effect size 0.5	32	34	44	47	25
effect size 0.7	42	44	48	46	13
effect size 0.9	49	50	49	45	13
effect size 1.1	20	46	47	43	26
effect size 1.3	46	45	47	40	19
effect size 1.5	35	45	47	41	20
effect size 1.7	35	50	50	41	19
effect size 1.9	39	44	50	41	24

Table B.1: Optimal number of clusters for the different similarity measures obtained on the first five data sets with effect size θ and one causative two-way-interaction.

effect size	SMC	JC	FMC1	FMC2	PCC
effect size 0.5	39	39	50	41	24
effect size 0.7	39	44	50	45	25
effect size 0.9	39	44	50	45	24
effect size 1.1	39	50	50	41	24
effect size 1.3	39	44	50	41	24
effect size 1.5	39	49	50	41	24
effect size 1.7	39	48	50	41	24
effect size 1.9	39	50	48	41	24

Table B.2: Optimal number of clusters for the different similarity measures obtained on the second five data sets with effect size θ and one causative two-way-interaction.

effect size	SMC	JC	FMC1	FMC2	PCC
effect size 0.5	42	37	42	39	14
effect size 0.7	40	37	43	39	12
effect size 0.9	40	34	41	39	15
effect size 1.1	39	33	41	39	15
effect size 1.3	39	35	44	41	15
effect size 1.5	37	38	43	41	14
effect size 1.7	36	38	44	41	14
effect size 1.9	36	34	43	41	14

Table B.3: Optimal number of clusters for the different similarity measures obtained on the first five data sets with effect size θ and two causative two-way-interaction.

effect size	SMC	JC	FMC1	FMC2	PCC
effect size 0.5	39	38	43	39	13
effect size 0.7	37	36	43	41	13
effect size 0.9	36	37	43	41	14
effect size 1.1	36	37	43	39	12
effect size 1.3	36	37	43	41	13
effect size 1.5	36	38	43	39	13
effect size 1.7	36	38	43	41	13
effect size 1.9	36	35	43	41	13

Table B.4: Optimal number of clusters for the different similarity measures obtained on the second five data sets with effect size θ and two causative two-way-interaction.

effect size	SMC	JC	FMC1	FMC2	PCC
effect size 0.5	32	29	42	38	15
effect size 0.7	31	29	36	35	13
effect size 0.9	30	35	42	38	12
effect size 1.1	30	33	41	35	12
effect size 1.3	31	32	42	36	13
effect size 1.5	32	35	42	33	13
effect size 1.7	32	34	39	33	12
effect size 1.9	31	35	43	35	16

Table B.5: Optimal number of clusters for the different similarity measures obtained on the first five data sets with effect size θ and three causative two-way-interaction.

effect size	SMC	JC	FMC1	FMC2	PCC
effect size 0.5	31	29	34	35	13
effect size 0.7	32	28	32	35	13
effect size 0.9	35	29	34	35	14
effect size 1.1	31	28	44	35	14
effect size 1.3	32	30	36	35	16
effect size 1.5	35	30	37	36	15
effect size 1.7	35	32	39	35	15
effect size 1.9	35	34	39	36	16

Table B.6: Optimal number of clusters for the different similarity measures obtained on the second five data sets with effect size θ and three causative two-way-interaction.

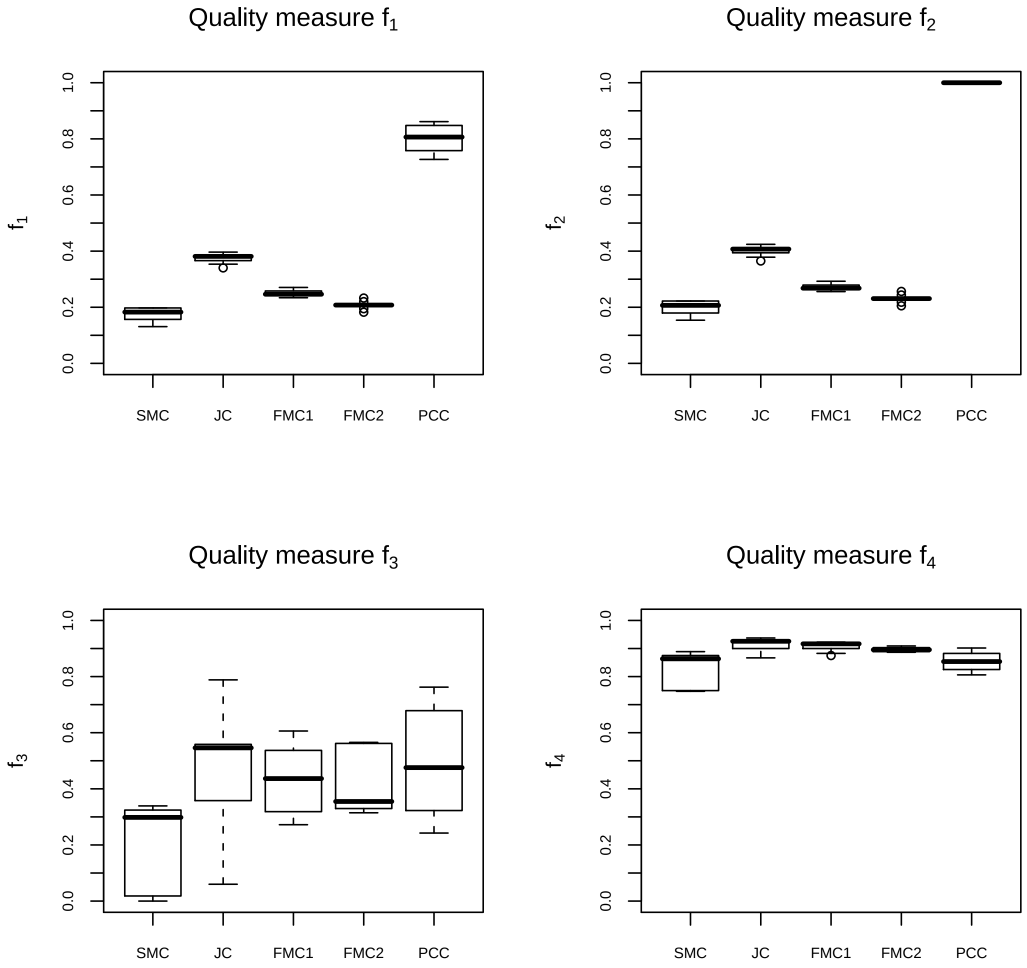

Figure B.1: Quality measure values for the different similarity measures for data set with two two-way interactions and $\theta = 1.1$.

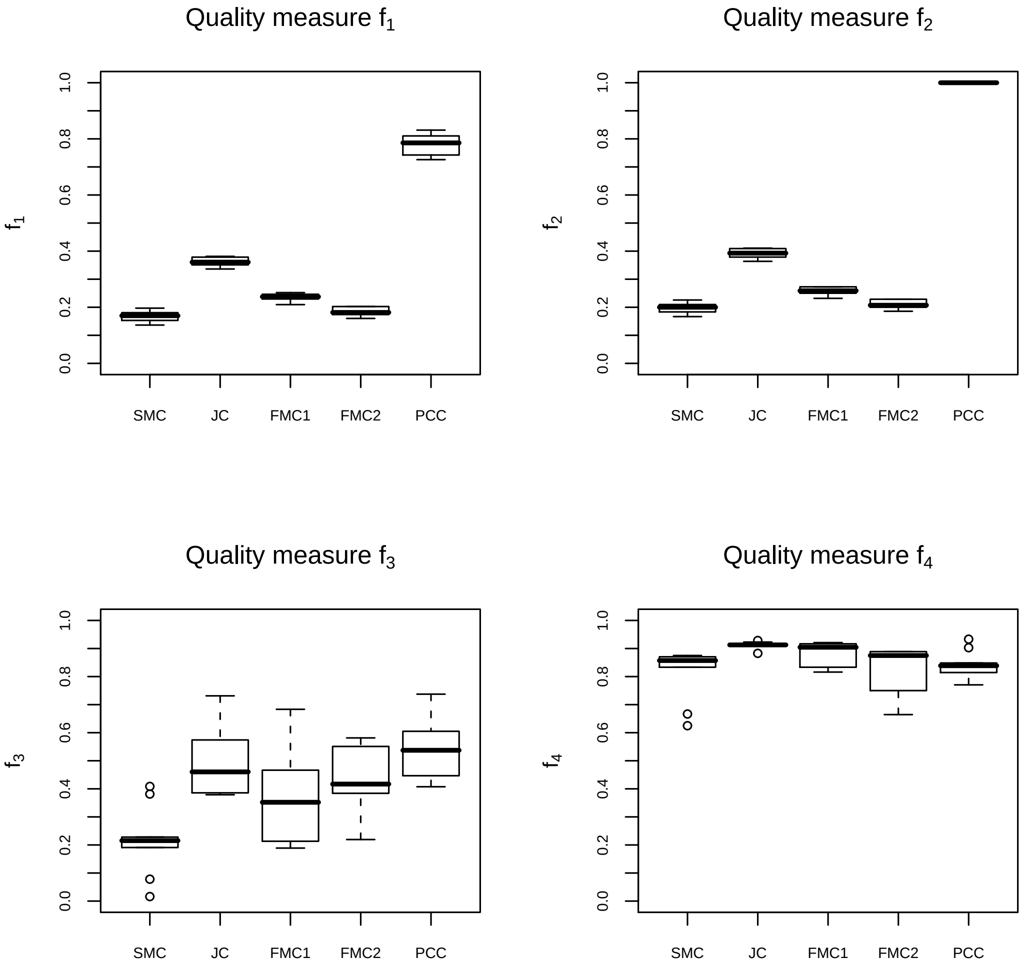

Figure B.2: Quality measure values for the different similarity measures for data set with three two-way interactions and $\theta = 1.1$.

APPENDIX C

Classification

C.1 Interest Measures

We want to show that the ranking of association rules does not differ between rankings according to confidence, lift and conviction of the study design is balances.
Proofs (see also Bayardo and Agrawal (1999)) :
Let B be the rule's body and H the rule's head. The necessary notation for confidence and support is given in terms of probabilities: $= con(B \rightarrow H) = P(H|B) = \frac{P(H \cup B)}{P(B)}$ and $supp(H) = P(H)$.

Lift:

$$lift(B \rightarrow H) = \frac{con(B \rightarrow H)}{P(H)} = c \cdot con(B \rightarrow H),$$

for $P(H)$ = constant for all rules.

Conviction:

$$\begin{aligned} conviction(B \rightarrow H) &= \frac{P(B)P(\bar{H})}{P(B \cup \bar{H})} = \frac{P(B)P(\bar{H})}{P(\bar{H}|B)P(B)} = \frac{P(\bar{H})}{P(\bar{H}|B)} \\ &= P(\bar{H})(\frac{P(B \cup \bar{H})}{P(B)})^{-1} = P(\bar{H})(1 - \frac{P(B \cup H)}{P(B)})^{-1} \\ &= c \cdot (1 - con)^{-1} \end{aligned}$$

This term is monotonously increasing in con for $con \in [0,1)$, but it is undefined for

con = 1.

The equivalences derived above holds if we deal with association rules for which we allow for two different items in the consequent, but the proportion of cases in the study equals the proportion of controls, i.e. $P(H) = P$(status = case) = $P(\bar{H}) = P$(status = control) = 0.5. For studies that show large differences in sample sizes between the two collectives, the analysis would have to be adapted.

C.2 Classification Methods

C.2.1 Local class

The complete algorithm of Local Class is given in the following.

Algorithm 4 local class

```
The data base $\mathcal{D}$ is divided into training and test data bases $\mathcal{D}^{Tr}$ and $\mathcal{D}^{Te}$, respectively.

for $k \in \{1, 2, 3\}$ do
  Set a minimum support $supp_{min_k}$ for $k$-itemsets in $\mathcal{D}^{Tr}$
  Find set of frequent $k$-itemsets $F_k$ with $n_k = |F_k|$ elements
  Order all $f^k \in F_k, k = 1, \ldots, n_k$ according to support:
  $supp(f_1^k) \geq supp(f_2^k) \cdots \geq supp(f_{n_k}^k)$
end for
$F = F_3 \cup F_2 \cup F_1$, regarding the existing order.
Label all $f \in F$ with increasing index $g = 1, \ldots, n_F (= n_1 + n_2 + n_3)$ .
Each $f_g, g = 1, \ldots, n_F$ corresponds to a future group $G_g$

Let $G_{T_i^{Tr}}$ be class label of $T_i^{Tr} \in \mathcal{D}^{Tr}$ and $G_{T_l^{Te}}$ be class label of $T_l^{Te} \in \mathcal{D}^{Te}$.
Initially, $G_{T_i^{Tr}} = G_{T_l^{Te}} = 0$
for all $T_i^{Tr} \in \mathcal{D}^{Tr}$ do
  $g = 1$
  while $g \leq n_F$ do
    if $f_g \not\subseteq T_i^{Tr}$ then
      $G_{T_i^{Tr}} = 0, g = g + 1$
    else
      $G_{T_g^{Tr}} = g$, stop
    end if
  end while
end for
Redo for-loop for test data to determine $G_{T_l^{Te}}$
if $G_{T_i^{Tr}} = 0$ or $G_{T_i^{Te}} = 0$ or $G_{T_i^{Tr}} = g$ with $|G_g| < 20$ or $G_{T_i^{Te}} = g$ with $|G_g| < 20$ then
  $G_{T_i^{Tr}} = G_{T_i^{Te}} = misc$ (miscellaneous group).
end if
```

C.2.2 Classification results

We presented the results of the classification methods on the simulated data as boxplots in Chapter 6. All results in detail are given in the following Tables C.1 - C.3.

	effect size = 0.5						
	minMCR	meanMCR	sdMCR	meanSens	sdSens	meanSpecs	sdSpecs
FC	0.4750	0.5000	0.0157	0.4436	0.3243	0.5564	0.3243
LC	0.4427	0.4809	0.0341	0.0574	0.0275	0.5016	0.0450
NC	0.4580	0.4902	0.0170	0.0570	0.0609	0.9626	0.0359
ACV	0.4690	0.4871	0.0155	0.5272	0.4116	0.4986	0.4157
ACSC	0.4460	0.4763	0.0218	0.3566	0.2452	0.6908	0.2367
LR	0.4500	0.4751	0.0235	0.5106	0.0691	0.5392	0.0601
LLR	0.4840	0.4958	0.0092	0.4966	0.0412	0.5112	0.0414
CART	0.4540	0.4797	0.0210	0.5030	0.0745	0.5376	0.0790
RF	0.4560	0.4779	0.0163	0.5220	0.0268	0.5222	0.0173

	effect size = 0.7						
	minMCR	meanMCR	sdMCR	meanSens	sdSens	meanSpecs	sdSpecs
FC	0.4860	0.5020	0.0116	0.3984	0.3837	0.5976	0.3886
LC	0.4238	0.4658	0.0197	0.0623	0.0228	0.5228	0.0545
NC	0.4160	0.4677	0.0500	0.3770	0.1289	0.6876	0.1205
ACV	0.4300	0.4843	0.0231	0.1340	0.2233	0.8974	0.1809
ACSC	0.4120	0.4376	0.0192	0.4528	0.0858	0.6720	0.1083
LR	0.3830	0.4170	0.0228	0.4826	0.0791	0.6834	0.1080
LLR	0.4640	0.4891	0.0165	0.4994	0.0547	0.5210	0.0401

CART	0.4060	0.4339	0.0182	0.5560	0.0502	0.5762	0.0724
RF	0.4390	0.4534	0.0103	0.5530	0.0204	0.5402	0.0173
	effect size = 0.9						
	minMCR	meanMCR	sdMCR	meanSens	sdSens	meanSpecs	sdSpecs
FC	0.4790	0.4996	0.0136	0.4066	0.3275	0.5942	0.3369
LC	0.4299	0.4686	0.0329	0.0650	0.0254	0.5165	0.0500
NC	0.4030	0.4400	0.0169	0.3274	0.0916	0.7926	0.0851
ACV	0.4000	0.4509	0.0291	0.1900	0.1666	0.9082	0.1155
ACSC	0.3950	0.4215	0.0227	0.4980	0.1635	0.6590	0.1946
LR	0.3680	0.3866	0.0162	0.4834	0.0150	0.7434	0.0243
LLR	0.4800	0.5029	0.0133	0.3664	0.0261	0.6278	0.0209
CART	0.3870	0.4099	0.0234	0.5474	0.0509	0.6328	0.0492
RF	0.4100	0.4342	0.0158	0.5598	0.0247	0.5718	0.0316
	effect size = 1.1						
	minMCR	meanMCR	sdMCR	meanSens	sdSens	meanSpecs	sdSpecs
FC	0.4930	0.5078	0.0113	0.5380	0.2476	0.4464	0.2457
LC	0.4188	0.4724	0.0398	0.0825	0.0283	0.5215	0.0695
NC	0.3910	0.4201	0.0188	0.3614	0.1068	0.7984	0.0821
ACV	0.3710	0.4515	0.0350	0.1438	0.1423	0.9532	0.0760
ACSC	0.3780	0.3951	0.0108	0.4842	0.0934	0.7256	0.1043
LR	0.3570	0.3716	0.0138	0.5074	0.0121	0.7494	0.0228
LLR	0.4830	0.4994	0.0132	0.4856	0.0362	0.5136	0.0305
CART	0.3570	0.3953	0.0298	0.5556	0.0506	0.6538	0.0748

RF	0.4050	0.4205	0.0101	0.5608	0.0269	0.5982	0.0234
	effect size = 1.3						
	minMCR	meanMCR	sdMCR	meanSens	sdSens	meanSpecs	sdSpecs
FC	0.4870	0.5052	0.0113	0.4150	0.4065	0.5746	0.4062
LC	0.4115	0.4679	0.0441	0.0889	0.0341	0.5263	0.0627
NC	0.3860	0.4364	0.1067	0.4298	0.0860	0.6974	0.2576
ACV	0.3450	0.4084	0.0617	0.3188	0.2457	0.8644	0.1234
ACSC	0.3450	0.3707	0.0133	0.5074	0.0622	0.7512	0.0759
LR	0.3430	0.3558	0.0139	0.5326	0.0158	0.7558	0.0213
LLR	0.4700	0.4913	0.0123	0.4904	0.0342	0.5230	0.0415
CART	0.3430	0.3713	0.0186	0.5840	0.0457	0.6734	0.0672
RF	0.3830	0.4006	0.0114	0.5812	0.0283	0.6176	0.0280
	effect size = 1.5						
	minMCR	meanMCR	sdMCR	meanSens	sdSens	meanSpecs	sdSpecs
FC	0.4690	0.4970	0.0182	0.5302	0.2968	0.4758	0.2927
LC	0.4066	0.4540	0.0310	0.0876	0.0337	0.5424	0.0646
NC	0.3750	0.3993	0.0194	0.4108	0.0723	0.7906	0.0466
ACV	0.3500	0.4419	0.0503	0.1564	0.1661	0.9598	0.0721
ACSC	0.3290	0.3493	0.0150	0.5472	0.0516	0.7542	0.0666
LR	0.3290	0.3423	0.0118	0.5522	0.0185	0.7632	0.0183
LLR	0.4620	0.4845	0.0174	0.4858	0.0332	0.5416	0.0313
CART	0.3360	0.3626	0.0178	0.5872	0.0512	0.6876	0.0508
RF	0.3760	0.3905	0.0135	0.5820	0.0278	0.6370	0.0354

	effect size = 1.7						
	minMCR	meanMCR	sdMCR	meanSens	sdSens	meanSpecs	sdSpecs
FC	0.4750	0.4933	0.0160	0.5346	0.1734	0.4788	0.1815
LC	0.4209	0.4638	0.0361	0.1053	0.0330	0.5350	0.0690
NC	0.3300	0.4014	0.1036	0.5466	0.0600	0.6506	0.2302
ACV	0.3240	0.4101	0.0616	0.3056	0.2499	0.8742	0.1310
ACSC	0.3280	0.3692	0.0433	0.5248	0.1326	0.7368	0.1146
LR	0.3160	0.3314	0.0128	0.5682	0.0188	0.7690	0.0179
LLR	0.4750	0.4970	0.0144	0.3732	0.0156	0.6328	0.0221
CART	0.3160	0.3455	0.0191	0.5906	0.0498	0.7184	0.0654
RF	0.3490	0.3713	0.0144	0.5990	0.0331	0.6584	0.0298

	effect size = 1.9						
	minMCR	meanMCR	sdMCR	meanSens	sdSens	meanSpecs	sdSpecs
FC	0.4780	0.4985	0.0145	0.5544	0.3045	0.4486	0.2950
LC	0.3473	0.4380	0.0534	0.1176	0.0381	0.5608	0.0765
NC	0.3210	0.3603	0.0232	0.5668	0.0472	0.7126	0.0356
ACV	0.3070	0.4159	0.0659	0.2266	0.2103	0.9416	0.0810
ACSC	0.3240	0.3610	0.0218	0.4596	0.1223	0.8184	0.1101
LR	0.3070	0.3210	0.0125	0.5826	0.0178	0.7754	0.0185
LLR	0.3390	0.3843	0.0235	0.5596	0.0420	0.6716	0.0380
CART	0.3070	0.3341	0.0137	0.5876	0.0586	0.7442	0.0622
RF	0.3420	0.3603	0.0136	0.6092	0.0307	0.6702	0.0242

Table C.1: Classification results on simulated data set with one causative two-way interaction. For each classification methods (rows), the minimum of the observed MCR is given. Mean values for MCR, sensitivity (Sens) and specificity (specs) are given with the respective standard deviations.

	effect size = 0.5						
	minMCR	meanMCR	sdMCR	meanSens	sdSens	meanSpecs	sdSpecs
FC	0.4800	0.5078	0.0148	0.5934	0.2951	0.3910	0.2841
LC	0.4262	0.5080	0.0592	0.0909	0.0486	0.4782	0.0905
NC	0.4900	0.4981	0.0034	0.0266	0.0455	0.9772	0.0399
ACV	0.4830	0.4977	0.0084	0.8036	0.1303	0.2010	0.1407
ACSC	0.4490	0.4860	0.0199	0.5592	0.1688	0.4688	0.1918
LR	0.4320	0.4833	0.0305	0.5058	0.0556	0.5276	0.0638
LLR	0.4630	0.4957	0.0170	0.4624	0.0302	0.5462	0.0389
CART	0.4450	0.4828	0.0170	0.5114	0.0819	0.5230	0.0769
RF	0.4620	0.4935	0.0182	0.4958	0.0214	0.5172	0.0331
	effect size = 0.7						
	minMCR	meanMCR	sdMCR	meanSens	sdSens	meanSpecs	sdSpecs
FC	0.4660	0.5012	0.0164	0.5090	0.1475	0.4886	0.1480
LC	0.4595	0.5126	0.0533	0.0819	0.0573	0.4527	0.0997
NC	0.4750	0.4962	0.0093	0.0608	0.0804	0.9468	0.0735
ACV	0.4470	0.4868	0.0163	0.1150	0.1684	0.9114	0.1430
ACSC	0.4290	0.4573	0.0122	0.4926	0.1607	0.5928	0.1532
LR	0.4270	0.4607	0.0168	0.5556	0.0395	0.5230	0.0332
LLR	0.4630	0.4938	0.0154	0.4896	0.0567	0.5218	0.0513

CART	0.4460	0.4743	0.0157	0.5272	0.0357	0.5242	0.0423
RF	0.4400	0.4783	0.0174	0.5168	0.0184	0.5266	0.0342
	effect size = 0.9						
	minMCR	meanMCR	sdMCR	meanSens	sdSens	meanSpecs	sdSpecs
FC	0.4800	0.5023	0.0152	0.5082	0.0800	0.4872	0.0788
LC	0.4498	0.4838	0.0265	0.0623	0.0426	0.4963	0.0503
NC	0.4420	0.6098	0.1677	0.4738	0.1233	0.3066	0.3276
ACV	0.4540	0.4867	0.0160	0.1938	0.2907	0.8328	0.2709
ACSC	0.4240	0.4389	0.0130	0.4618	0.1215	0.6604	0.1055
LR	0.4220	0.4495	0.0146	0.5746	0.0329	0.5264	0.0350
LLR	0.4910	0.5004	0.0117	0.4608	0.0339	0.5378	0.0458
CART	0.3960	0.4435	0.0244	0.5426	0.0700	0.5704	0.0767
RF	0.4420	0.4641	0.0113	0.5334	0.0197	0.5384	0.0238
	effect size = 1.1						
	minMCR	meanMCR	sdMCR	meanSens	sdSens	meanSpecs	sdSpecs
FC	0.4740	0.5031	0.0186	0.4972	0.1258	0.4966	0.1316
LC	0.4255	0.4962	0.0447	0.1043	0.0616	0.4776	0.0569
NC	0.4210	0.4411	0.0170	0.3196	0.1315	0.7982	0.1062
ACV	0.4880	0.4976	0.0039	0.0078	0.0125	0.9970	0.0058
ACSC	0.4140	0.4220	0.0095	0.4346	0.0643	0.7214	0.0670
LR	0.3520	0.3945	0.0309	0.5666	0.0818	0.6444	0.0477
LLR	0.4660	0.4955	0.0163	0.4738	0.0721	0.5338	0.0585
CART	0.3670	0.4161	0.0269	0.5868	0.0475	0.5810	0.0645

RF	0.4170	0.4450	0.0170	0.5500	0.0233	0.5600	0.0291
	effect size = 1.3						
	minMCR	meanMCR	sdMCR	meanSens	sdSens	meanSpecs	sdSpecs
FC	0.4700	0.5004	0.0204	0.5654	0.2109	0.4338	0.1968
LC	0.3748	0.4573	0.0420	0.0862	0.0344	0.5130	0.0415
NC	0.4040	0.4766	0.1258	0.5170	0.0695	0.5298	0.2817
ACV	0.4600	0.4883	0.0108	0.0334	0.0332	0.9900	0.0133
ACSC	0.3850	0.4456	0.0461	0.2666	0.2289	0.8422	0.1388
LR	0.3430	0.3665	0.0186	0.6364	0.0585	0.6306	0.0308
LLR	0.4610	0.4920	0.0203	0.4762	0.0529	0.5388	0.0604
CART	0.3570	0.3991	0.0249	0.6004	0.0670	0.6014	0.0613
RF	0.4040	0.4263	0.0142	0.5694	0.0217	0.5780	0.0261
	effect size = 1.5						
	minMCR	meanMCR	sdMCR	meanSens	sdSens	meanSpecs	sdSpecs
FC	0.4560	0.4958	0.0148	0.5236	0.3258	0.4848	0.3254
LC	0.4138	0.4672	0.0325	0.0813	0.0240	0.5126	0.0754
NC	0.3840	0.4191	0.0208	0.4542	0.1936	0.7076	0.1710
ACV	0.4120	0.4754	0.0283	0.0980	0.1617	0.9512	0.1104
ACSC	0.3990	0.4209	0.0131	0.4744	0.2048	0.6838	0.2061
LR	0.3220	0.3623	0.0302	0.6526	0.0799	0.6228	0.0587
LLR	0.4190	0.4573	0.0207	0.4542	0.0471	0.6312	0.0630
CART	0.3420	0.3783	0.0287	0.6262	0.0815	0.6172	0.0553
RF	0.3910	0.4143	0.0151	0.5902	0.0288	0.5812	0.0254

	effect size = 1.7						
	minMCR	meanMCR	sdMCR	meanSens	sdSens	meanSpecs	sdSpecs
FC	0.475	0.4967	0.0169	0.5182	0.0593	0.4884	0.0759
LC	0.413	0.4667	0.0476	0.1065	0.0377	0.4816	0.1139
NC	0.380	0.3948	0.0154	0.4648	0.0715	0.7456	0.0511
ACV	0.482	0.4937	0.0067	0.0160	0.0179	0.9966	0.0071
ACSC	0.371	0.3881	0.0130	0.4766	0.0665	0.7472	0.0583
LR	0.320	0.3509	0.0221	0.6528	0.0740	0.6454	0.0363
LLR	0.474	0.4995	0.0138	0.4686	0.0567	0.5310	0.0492
CART	0.317	0.3569	0.0230	0.6504	0.0560	0.6358	0.0388
RF	0.381	0.3960	0.0135	0.5998	0.0278	0.6082	0.0269

	effect size = 1.9						
	minMCR	meanMCR	sdMCR	meanSens	sdSens	meanSpecs	sdSpecs
FC	0.4490	0.4893	0.0220	0.4918	0.0885	0.5296	0.0791
LC	0.3876	0.4554	0.0470	0.1135	0.0323	0.4736	0.1122
NC	0.3500	0.3733	0.0158	0.5688	0.0452	0.6846	0.0456
ACV	0.4890	0.4971	0.0038	0.0074	0.0094	0.9984	0.0025
ACSC	0.3500	0.3772	0.0162	0.4682	0.0660	0.7774	0.0591
LR	0.3000	0.3342	0.0213	0.6744	0.0925	0.6572	0.0675
LLR	0.4650	0.4914	0.0184	0.4594	0.0597	0.5578	0.0445
CART	0.3160	0.3455	0.0256	0.6652	0.0533	0.6438	0.0551
RF	0.3660	0.3870	0.0151	0.6148	0.0282	0.6112	0.0241

Table C.2: Classification results on simulated data set with two causative two-way interactions. For each classification methods (rows), the minimum of the observed MCR is given. Mean values for MCR, sensitivity (Sens) and specificity (specs) are given with the respective standard deviations.

	effect size = 0.5						
	minMCR	meanMCR	sdMCR	meanSens	sdSens	meanSpecs	sdSpecs
FC	0.4740	0.4968	0.0132	0.5592	0.3655	0.4472	0.3694
LC	0.2956	0.4621	0.1043	0.1370	0.1085	0.4290	0.1136
NC	0.5000	0.5555	0.1050	0.2830	0.3134	0.6060	0.4279
ACV	0.4900	0.5035	0.0119	0.8916	0.2209	0.1014	0.2000
ACSC	0.4640	0.4943	0.0159	0.4804	0.1492	0.5310	0.1473
LR	0.4540	0.4861	0.0163	0.5352	0.1039	0.4926	0.1135
LLR	0.4700	0.4988	0.0143	0.4836	0.0169	0.5184	0.0335
CART	0.4660	0.4917	0.0125	0.5440	0.0701	0.4726	0.0669
RF	0.4730	0.4887	0.0137	0.5124	0.0202	0.5102	0.0256

	effect size = 0.7						
	minMCR	meanMCR	sdMCR	meanSens	sdSens	meanSpecs	sdSpecs
FC	0.4790	0.5021	0.0127	0.4424	0.3339	0.5534	0.3346
LC	0.3163	0.4823	0.1162	0.1303	0.1022	0.4573	0.1335
NC	0.4720	0.5833	0.1431	0.2352	0.2392	0.5982	0.4129
ACV	0.4370	0.4825	0.0218	0.3214	0.2910	0.7136	0.2722
ACSC	0.4560	0.4921	0.0225	0.5556	0.1756	0.4602	0.1944
LR	0.4440	0.4754	0.0308	0.5292	0.0841	0.5200	0.0862
LLR	0.4610	0.4978	0.0196	0.5090	0.0569	0.4944	0.0630

CART	0.4280	0.4859	0.0278	0.4916	0.0480	0.5366	0.0606
RF	0.4370	0.4786	0.0210	0.5236	0.0193	0.5192	0.0266
	effect size = 0.9						
	minMCR	meanMCR	sdMCR	meanSens	sdSens	meanSpecs	sdSpecs
FC	0.4940	0.5026	0.0059	0.3944	0.4286	0.6004	0.4270
LC	0.4642	0.5063	0.0366	0.0766	0.0298	0.4822	0.0432
NC	0.4700	0.4940	0.0110	0.0414	0.0556	0.9706	0.0416
ACV	0.4260	0.4869	0.0246	0.2578	0.3462	0.7684	0.3355
ACSC	0.4390	0.4652	0.0203	0.5470	0.1615	0.5226	0.1927
LR	0.4420	0.4735	0.0279	0.5510	0.0806	0.5020	0.1031
LLR	0.4770	0.5037	0.0156	0.4980	0.0589	0.4930	0.0539
CART	0.4090	0.4792	0.0313	0.4832	0.0688	0.5584	0.0494
RF	0.4300	0.4733	0.0179	0.5236	0.0246	0.5298	0.0292
	effect size = 1.1						
	minMCR	meanMCR	sdMCR	meanSens	sdSens	meanSpecs	sdSpecs
FC	0.4790	0.4972	0.0110	0.4600	0.3123	0.5456	0.3117
LC	0.4028	0.5030	0.0576	0.0669	0.0392	0.4270	0.0720
NC	0.4110	0.4660	0.0293	0.2516	0.1734	0.8164	0.1231
ACV	0.4500	0.4947	0.0157	0.0396	0.1231	0.9710	0.0917
ACSC	0.4160	0.4516	0.0248	0.5594	0.1575	0.5374	0.1751
LR	0.3980	0.4560	0.0346	0.5312	0.0858	0.5568	0.0810
LLR	0.4780	0.4986	0.0123	0.4878	0.0656	0.5146	0.0563
CART	0.4370	0.4607	0.0197	0.5306	0.0596	0.5480	0.0555

RF	0.4360	0.4610	0.0209	0.5354	0.0281	0.5426	0.0249
	effect size = 1.3						
	minMCR	meanMCR	sdMCR	meanSens	sdSens	meanSpecs	sdSpecs
FC	0.4850	0.5006	0.0068	0.2952	0.4178	0.7036	0.4184
LC	0.3764	0.4840	0.0484	0.0753	0.0341	0.5226	0.0607
NC	0.4470	0.4743	0.0171	0.1446	0.0706	0.9068	0.0522
ACV	0.4360	0.4797	0.0222	0.2110	0.2689	0.8296	0.2551
ACSC	0.3940	0.4464	0.0346	0.5088	0.0985	0.5984	0.1182
LR	0.4220	0.4470	0.0272	0.5344	0.1068	0.5716	0.1233
LLR	0.4670	0.4996	0.0152	0.5038	0.0596	0.4952	0.0555
CART	0.3970	0.4416	0.0349	0.5280	0.0823	0.5888	0.0792
RF	0.4170	0.4481	0.0148	0.5532	0.0201	0.5506	0.0275
	effect size = 1.5						
	minMCR	meanMCR	sdMCR	meanSens	sdSens	meanSpecs	sdSpecs
FC	0.4910	0.4972	0.0049	0.5022	0.3834	0.5034	0.3792
LC	0.3852	0.4603	0.0382	0.0704	0.0380	0.5447	0.0405
NC	0.4040	0.4312	0.0264	0.4890	0.0722	0.6486	0.1091
ACV	0.4240	0.4922	0.0240	0.0462	0.1440	0.9694	0.0961
ACSC	0.4000	0.4280	0.0234	0.4946	0.1283	0.6494	0.1426
LR	0.3410	0.3970	0.0356	0.6128	0.0986	0.5932	0.0536
LLR	0.4890	0.5031	0.0136	0.4884	0.0383	0.5026	0.0340
CART	0.3830	0.4268	0.0251	0.5092	0.0473	0.6372	0.0375
RF	0.4210	0.4359	0.0136	0.5678	0.0259	0.5604	0.0291

	effect size = 1.7						
	minMCR	meanMCR	sdMCR	meanSens	sdSens	meanSpecs	sdSpecs
FC	0.4580	0.4986	0.0226	0.4828	0.1002	0.5200	0.1121
LC	0.4184	0.4785	0.0496	0.0821	0.0236	0.4799	0.0790
NC	0.4060	0.4387	0.0307	0.3614	0.1714	0.7612	0.1208
ACV	0.4890	0.4988	0.0035	0.0040	0.0120	0.9984	0.0051
ACSC	0.4030	0.4202	0.0206	0.5046	0.1146	0.6550	0.1450
LR	0.3370	0.3752	0.0328	0.6236	0.0797	0.6260	0.0317
LLR	0.4850	0.4978	0.0079	0.5068	0.0377	0.4966	0.0330
CART	0.3880	0.4104	0.0248	0.5650	0.0390	0.6142	0.0368
RF	0.4010	0.4293	0.0161	0.5638	0.0172	0.5776	0.0285

	effect size = 1.9						
	minMCR	meanMCR	sdMCR	meanSens	sdSens	meanSpecs	sdSpecs
FC	0.4650	0.4989	0.0162	0.4978	0.2118	0.5044	0.1996
LC	0.3076	0.4767	0.0972	0.1556	0.0903	0.4882	0.1080
NC	0.3920	0.4355	0.0244	0.2750	0.0685	0.8540	0.0424
ACV	0.4650	0.4944	0.0116	0.0182	0.0386	0.9930	0.0155
ACSC	0.3950	0.4590	0.0422	0.7510	0.2599	0.3310	0.3383
LR	0.3450	0.3863	0.0325	0.5900	0.0986	0.6374	0.0694
LLR	0.4790	0.4923	0.0122	0.4908	0.0643	0.5232	0.0622
CART	0.3740	0.3993	0.0275	0.5872	0.0665	0.6142	0.0793
RF	0.3970	0.4185	0.0167	0.5746	0.0202	0.5884	0.0267

Table C.3: Classification results on simulated data set with three causative two-way interactions. For each classification methods (rows), the minimum of the observed MCR is given. Mean values for MCR, sensitivity (Sens) and specificity (specs) are given with the respective standard deviations.

The classification result for three two-way interactions was not displayed in Chapter 6. It can be found in Figure C.1.

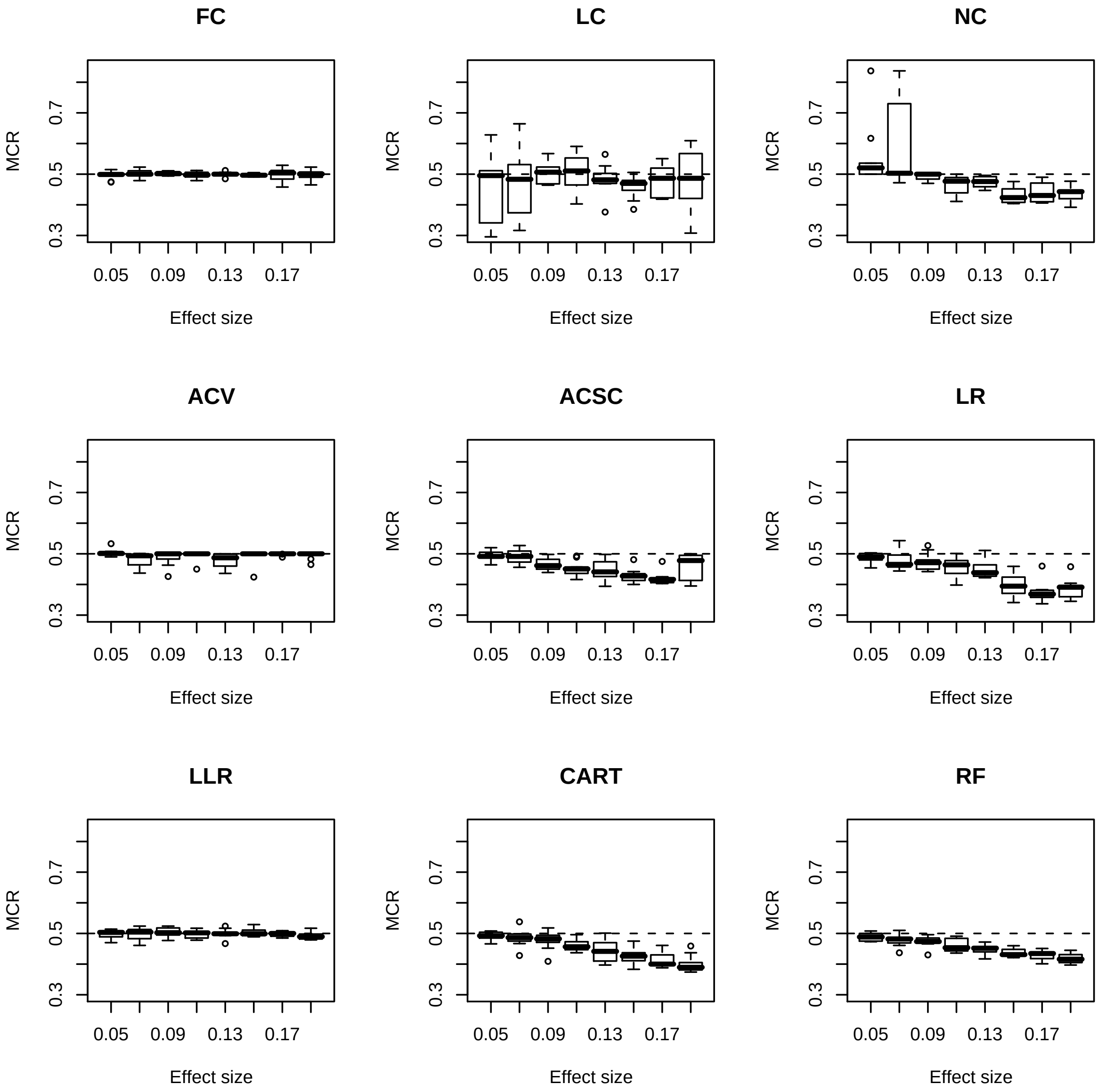

Figure C.1: Misclassification rates for the simulated data data achieved by the different classification methods for three causative two-way interactions.

Bibliography

Affymetrix (2006): Brlmm: An improved genotype calling method for the genechip human mapping 500k array set. *Technical Report*, Affymetrix, Santa Clara, CA.

Affymetrix (2007): Affymetrix genome-wide human SNP array 6.0 data sheet. Affymetrix, Santa Clara, CA.

Agrawal, H., Mannila, R., Srikant, H., Toivonen, and Verkamo, A. I. (1996): Fast discovery of association rules. In: U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy (eds.) *Advances in Knowledge Discovery and Data Mining*, 307–328. AAAI Press.

Agrawal, R., Imielinski, T., and Swami, A. (1993): Mining association rules between sets of items in large databases. In: P. Buneman and S. Jajodia (eds.) *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, 207–216.

Ali, K., Manganaris, S., and Srikant, R. (1997): Partial classification using association rules. In: D. Heckerman, H. Mannila, D. Pregibon, and R. Uthurusamy (eds.) *Knowledge Discovery and Data Mining*, 115–118. AAAI Press.

Anderberg, M. (1973): *Cluster Analysis for Applications*. New York: Academic Press.

Baralis, E. and Garza, P. (2003): Majority classification by means of association rules. In: *PKDD*, 35–46.

Bayardo, R. J. and Agrawal, R. (1999): Mining the most interesting rules. In: *KDD '99: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, 145–154. ACM Press.

Bellman, R. (2000): *Adaptive Control Processes: A Guided Tour*. Princeton, N.J.: Princeton University Press.

Besson, J., Robardet, C., and Boulicaut, J.-F. (2004): Constraint-based mining of formal concepts in transactional data. In: H. Dai, R. Srikant, and C. Zhang (eds.) *PAKDD 2004 LNAI 3056*, 615–624. Springer-Verlag Berlin Heidelberg.

Borgelt, C. and Kruse, R. (2002): Induction of association rules: apriori implementation. In: W. Härdle and B. Rönz (eds.) *COMPSTAT 2002, 15th Conference on Computational Statistics, 2002*, 395–400.

Bornkamp, B. (2006): *Comparison of Model-Based and Model-Free Approaches for the Analysis of Dose-Response Studies*. Master Thesis, Universität Dortmund, Fachbereich Statistik.

Breiman, L. (2001): Random forests. *Machine Learning*, 45, 5–32.

Breiman, L., Friedman, J., Olshen, R., and Stone, C. (1984): *Classification and Regression Trees*. New York: Chapman and Hall.

Breiter, D. (2008): *Vergleich verschiedener Algorithmen zur Erzeugung von Assoziationsregeln*. Bachelor Thesis, Universität Dortmund, Fakultät Statistik.

Clark, A., Boerwinkle, E., Hixson, J., and Sing, C. (2005): Determinants of the success of whole genome association testing. *Genome Research*, 15, 1463–1467.

Cox, T. and Cox, M. (2001): *Multidimensional Scaling*. London: Chapman and Hall, 2nd edition.

Cramer, J. (2003): *Logit Models*. Cambridge: Cambridge University Press.

Devlin, B. and Risch, N. (1995): A comparison of linkage disequilibrium measures for fine-scale mapping. *Genomics*, 29, 311–322.

Dietterich, T. G. (1998): Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10, 1895–1923.

Dong, G., Zhang, X., Wong, L., and Li, J. (1999): CAEP: Classification by aggregating emerging patterns. In: S. Arikawa and K. Furukawa (eds.) *Proceedings of the 2nd International Conference on Discovery Science, Lecture Notes in Artificial Intelligence*, volume 1721, 30–42. Springer-Verlag.

Fahrmeir, L., Hamerle, A., and Tutz, G. (1996): *Multivariate statistische Verfahren*. Berlin: Walter de Gruyter, 6th edition.

Flach, P. A. and Lavrac, N. (2000): The role of feature construction in inductive rule learning. In: L. D. Raedt and S. Kramer (eds.) *Proceedings of the ICML2000 workshop on Attribute-Value and Relational Learning: Crossing the boundaries*, 1–11.

Freund, Y. and Schapire, R. (1997): A decision-theoretic generalization of online learning and an application to boosting. *Journal of Computer and System Science*, 55, 119–139.

Fürnkranz, J. (2005): From local to global patterns: Evaluation issues in rule learning algorithms. In: *Lecture Notes in Computer Science: Local Pattern Detection*, volume 3539, 20–38. Springer-Verlag.

Gabriel, S., Schaffner, S., Nguyen, H., Moore, J., Roy, J., Blumenstiel, B., Higgins, J., DeFelice, M., Lochner, A., Faggart, M., LiuCordero, S., Rotimi, C., Adeyemo, A., Cooper, R., Ward, R., Lander, E., Daly, M., and Altshuler, D. (2002): The structure of haplotype blocks in the human genome. *Science*, 296, 2225–2229.

Garte, S. (2001): Metabolic susceptibility genes as cancer risk factors: Time for a reassessment? *Cancer Epidemiology Biomarkers & Prevention*, 10, 1233–1237.

Goldstein, D. and Cavalleri, G. (2005): Genomics: Understanding human diversity. *Nature*, 437, 1241–1242.

Hahsler, M., Gruen, B., and Hornik., K. (2009): *arules: Mining Association Rules and Frequent Itemsets*. R package version 1.0-0.

Hahsler, M., Grun, B., and Hornik, K. (2005): arules - a computational environment for mining association rules and frequent item sets. *Journal of Statistical Software*, 14, 1–25.

Haines, J. and Pericak-Vance, M. A. (2006): *Genetic Analysis of Complex Disease*. Hoboken, New Jersey: Wiley, 2nd edition.

Halkidi, M., Batistakis, Y., and Vazirgiannis, M. (2001): On clustering validation techniques. *Journal of Intelligent Information Systems*, 17, 107–145.

Harrington, E. (1965): The desirability function. *Industrial Quality Control*, 21, 494–498.

Hastie, T., Tibshirani, R., and Friedman, J. H. (2001): *The Elements of Statistical Learning*. New York: Springer.

Hill, W. and Robertson, A. (1968): Linkage disequilibrium in finite populations. *Theoretical and Applied Genetics*, 38, 226 – 231.

Hoh, J. and Ott, J. (2003): Mathematical multi-locus approaches to localizing complex human trait genes. *Nature Review Genetics*, 4, 701–709.

Ickstadt, K., Müller, T., and Schwender, H. (2006): Analyzing SNPs: Are there needles in the haystack? *Chance*, 19, 21–26.

Illumina (2009): Genome-wide DNA analysis beadchips data sheet. Illumina, San Diego, CA.

Jain, A. and Dubes, R. (1988): *Algorithms for Clustering Data*. New Jersey: Prentice Hall.

Jorde, L. B., Carey, J. C., Bamshad, M. J., and White, R. L. (1995): *Medical Genetics*. St. Louis: Mosby, 2nd edition.

Justenhoven, C., Hamann, U., Pesch, B., Harth, V., Rabstein, S., Baisch, C., Vollmert, C., Illig, T., Ko, Y., Brüning, T., and Brauch, H. (2004): ERCC2 genotypes and a corresponding haplotype are linked with breast cancer risk in a German population. *Cancer Epidemiology Biomarkers & Prevention*, 13, 2059–2064.

Justenhoven, C., Hamann, U., Schubert, F., Zapatka, M., Pierl, C., Rabstein, S., Selinski, S., Mueller, T., Ickstadt, K., Gilbert, M., Ko, Y., Baisch, C., Pesch, B., Harth, V., Bolt, H., Vollmert, C., Illig, T., Eils, R., Dippon, J., and Brauch, H. (2008): Breast cancer: a candidate gene approach across the estrogen metabolic pathway. *Breast Cancer Research and Treatment*, 108, 137–149.

Kooperberg, C. and Ruczinski, I. (2008): *LogicReg: Logic Regression*. R package version 1.4.8.

Kooperberg, C., Ruczinski, I., LeBlanc, M. L., and Hsu, L. (2001): Sequence analysis using logic regression. *Genetic Epidemiology*, 21, 626–631.

LaFramboise, T. (2009): Single nucleotide polymorphism arrays: a decade of biological, computational and technological advances. *Nucleic acids research*, 37 (13), 4181–4193.

Larsen, B. and Aone, C. (1999): Fast and effective text mining using linear-time document clustering. In: *KDD '99: Proceedings of the fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 16–22. New York, NY, USA: ACM.

Lewontin, R. C. (1964): The interaction of selection and linkage. I. General considerations; Heterotic models. *Genetics*, 49, 49–67.

Liu, B., Hsu, W., and Ma, Y. (1998): Integrating classification and association rule mining. In: R. Agrawal, P. Storloz, and G. Piatetsky-Shapiro (eds.) *Knowledge Discovery and Data Mining*, 80–86. AAAI Press.

Loader, C. (1999): *Local Regression and Likelihood*. New York: Springer.

Marchini, J., Donnelly, P., and Cardon, L. (2005a): Genome-wide strategies for detecting multiple loci influencing complex diseases. *Nature Genetics*, 37, 413–417.

Marchini, J., Donnelly, P., and Cardon, L. (2005b): Genome-wide strategies for detecting multiple loci influencing complex diseases. *Nature Genetics*, 37, (Supplementary Notes).

McCullagh, P. and Nelder, J. A. (1989): *Generalized Linear Models*. London: Chapman & Hall, 2nd edition.

Mielikäinen, T. (2005): *Summarization Techniques for Pattern Collections in Data Mining*. PhD Thesis, University of Helsinki, Faculty of Science, Department of Computer Science.

Müller, T. (2004): *Clusteranalyse von SNP-Daten: Verschiedene Ähnlichkeitsmaße im Vergleich*. Master Thesis, Universität Dortmund, Fachbereich Statistik.

Müller, T., Schwender, H., and Ickstadt, K. (2008): Finding SNP Interactions. In: M. Ahdesmäki, K. Strimmer, N. Radde, J. Rahnenführer, K. Klemm, H. Lähdesmäki, and O. Yli-Harja (eds.) *Proceedings of the Fifth International Workshop on Computational Systems Biology (WCSB)*, 109–112. Leipzig, Germany.

Müller, T., Selinski, S., and Ickstadt, K. (2005): Cluster analysis: A comparison of different similarity measures for SNP data. *Technical Report*, Statistics Department University of Dortmund, Germany.

Neumann, C. (2007): *Einsatz von Clusterverfahren zur Produktfamilienbildung*. Master Thesis, Universität Dortmund, Fachbereich Statistik.

Nothnagel, M. (2002): Simulation of LD block-structured SNP haplotype data and its use for the analysis of case-control data by supervised learning methods. *American Journal of Human Genetics*. 71, (Suppl.)(4): A2363.

Nunkesser, R., Bernholt, T., Schwender, H., Ickstadt, K., and Wegener, I. (2007): Detecting high-order interactions of single nucleotide polymorphisms using genetic programming. *Bioinformatics*, 23, 3280–3288.

Piatetsky-Shapiro, G. (1991): Discovery, analysis and presentation of strong rules. In: G. Piatetsky-Shapiro and W. J. Frawley (eds.) *Knowledge Discovery in Databases*, 229–248. AAAI Press.

Quinlan, J. (1992): *C4.5: Programs for Machine Learning*. San Mateo, CA: Morgan Kaufmann Publishers.

R Development Core Team (2008): *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. URL `http://www.R-project.org`. ISBN 3-900051-07-0.

Rao, C. and Wu, Y. (2001): On model selection. *IMS Lecture notes - Monograph Series*, 38, 1–64.

Risch, N. and Merikengas, K. (1996): The future of genetic studies of complex human diseases. *Science*, 273, 1516–1517.

Rocca, W., Amaducci, L., and Schoenberg, B. (1986): Epidemiology of clinically diagnosed Alzheimer's disease. *Annals of Neurology*, 19, 415–424.

Ruczinski, I. (2000): *Logic regression and statistical issues related to the protein folding problem*. PhD Thesis, University of Washington, Seattle.

Ruczinski, I., Kooperberg, C., and LeBlanc, M. L. (2003): Logic regression. *Journal of Computational and Graphical Statistics*, 12, 475–511.

Ruczinski, I., Kooperberg, C., and LeBlanc, M. L. (2004): Exploring interactions in high-dimensional genomic data: an overview of logic regression, with applications. *Journal of Multivariat Analysis*, 90 (1), 178–195.

Schiffner, J., Szepannek, G., Monthé, T., and Weihs, C. (2009): Localized logistic regression for categorical influential factors. In: A. Fink, B. Lausen, W. Seidel, and A. Ultsch (eds.) *Advances in Data Analysis, Data Handling and Business Intelligence*, 185–195. Heidelberg: Springer.

Schwender, H. (2003): Modifying microarray analysis methods for categorical data - SAM and PAM for SNPs. In: C. Weihs and W. Gaul (eds.) *Classification - The Ubiquitous Challenge*, 370 – 377. Heidelberg: Springer.

Schwender, H. (2007): *Statistical analysis of genotype and gene expression data*. PhD Thesis, Universität Dortmund, Fachbereich Statistik.

Schwender, H. and Fritsch, A. (2009): *scrime: Analysis of High-Dimensional Categorical Data such as SNP Data*. R package version 1.1.2.

Selinski, S. and Ickstadt, K. (2005): Similarity Measures for Clustering SNP Data. *Technical Report*, Statistics Department University of Dortmund, Germany.

Selinski, S. and Ickstadt, K. (2008): Cluster analysis of genetic and epidemiological data in molecular epidemiology. *Journal of Toxicology and Environmental Health, Part A*, 71.

Steinbach, M., Karypis, G., and Kumar, V. (2000): A comparison of document clustering techniques. *Technical Report*.

Steuer, D. (2005): *Statistische Eigenschaften der Multikriteriellen Optimierung mittels Wünsch-barkeiten*. Phd Thesis, Universität Dortmund, Fachbereich Statistik.

Storey, J. D. and Tibshirani, R. (2003): Statistical significance for genomewide studies. *Proceedings of the National Acadamy of Science USA*, 100 (16), 9440–9445.

Tan, P., Kumar, V., and Srivastava, J. (2002): Selecting the right interestingness measure for association patterns. In: *Proceedings of the Eight A CM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 32–41.

Thabtah, F. (2005): A review on associative classification mining. *Journal of Knowledge Engineering Review*, 22 (1), 37–65.

The International HapMap Consortium (2003): The International HapMap Project. *Nature*, 426, 789–796.

The International HapMap Consortium (2007): A second generation human haplotype map of over 3.1 million SNPs. *Nature*, 449, 851–861.

Therneau, T. M. and Atkinson., B. (2008): *rpart: Recursive Partitioning*. URL `http://mayoresearch.mayo.edu/mayo/research/biostat/splusfunctions.cfm`. R package version 3.1-42.

Trautmann, H. and Weihs, C. (2006): On the distribution of the desirability index using Harrington's desirability function. *Metrika*, 63 (2), 207–213.

Tusher, V. G., Tibshirani, R., and Chu, G. (2001): Significance analysis of microarrays applied to the ionizing radiation response. *Proceedings of the National Academy of Sciences USA*, 98 (9), 5116–5121.

Tutz, G. and Binder, H. (2005): Localized classification. *Statistics and Computing*, (15), 155–166.

Wolpert, D. H. (1996): The lack of a priori distinctions between learning algorithms. *Neural Computation*, 8, 1341–1390.

Wu, X., Jin, L., and Xiong, M. (2008): Composite measure of linkage disequilibrium for testing interaction between unlinked loci. *European Journal of Human Genetics*, 16, 644–651.

Xinhua, Y., Kuan, Y., and Wu, D. (2006): A k-means clustering algorithm based on self-adoptively selecting density radius. In: *IJCSNS International Journal of Computer Science and Network Security*, volume 6, 43–46.

Zaki, M. J., Parthasarathy, S., Ogihara, M., and Li, W. (1997): New algorithms for fast discovery of association rules. In: *In 3rd Intl. Conf. on Knowledge Discovery and Data Mining*, 283–286. AAAI Press.

Zhao, J., Boerwinkle, E., and Xiong, M. (2005): An entropy-based statistic for genomewide association studies. *American Journal of Human Genetics*, 11 (1), 27–40.