Linköping Studies in Science and Technology. Dissertations No. 1090.

Statistical Feature Selection

With Applications in Life Science

Roland Nilsson

Department of Physics, Chemistry and Biology
Linköping University, SE58183 Linköping, Sweden

Linköping, 2007

Statistical Feature Selection

With Applications in Life Science

rolle@ifm.liu.se
Division of Theory and Modelling
Department of Physics, Chemistry and Biology
Linköping University, SE58183 Linköping, Sweden

ISBN 978-91-85715-24-4 ISSN 0345-7524

Cover art by Roland Nilsson 2007. Image of nerve cells (bottom right) kindly provided by the Biomedical Electron Microscope Unit, Newcastle University, United Kingdom. Typeset in LaTeX 2_ε. Printed by LiU-Tryck, Linköping, Sweden 2007.

Abstract

The sequencing of the human genome has changed life science research in many ways. Novel measurement technologies such as microarray expression analysis, genome-wide SNP typing and mass spectrometry are now producing experimental data of extremely high dimensions. While these techniques provide unprecedented opportunities for exploratory data analysis, the increase in dimensionality also introduces many difficulties. A key problem is to discover the most *relevant* variables, or *features*, among the tens of thousands of parallel measurements in a particular experiment. This is referred to as *feature selection*.

For feature selection to be principled, one needs to decide exactly what it means for a feature to be "relevant". This thesis considers relevance from a statistical viewpoint, as a measure of statistical dependence on a given *target variable*. The target variable might be continuous, such as a patient's blood glucose level, or categorical, such as "smoker" *vs.* "non-smoker". Several forms of relevance are examined and related to each other to form a coherent theory. Each form of relevance then defines a different feature selection problem.

The *predictive* features are those that allow an accurate predictive model, for example for disease diagnosis. I prove that finding predictive features is a tractable problem, in that consistent estimates can be computed in polynomial time. This is a substantial improvement upon current theory. However, I also demonstrate that selecting features to optimize prediction accuracy does not control feature error rates. This is a severe drawback in life science, where the selected features *per se* are important, for example as candidate drug targets. To address this problem, I propose a statistical method which to my knowledge is the first to achieve error control. Moreover, I show that in high dimensions, feature sets can be impossible to replicate in independent experiments even with controlled error rates. This finding may explain the lack of agreement among genome-wide association studies and molecular signatures of disease.

The most predictive features may not always be the most relevant ones from a biological perspective, since the predictive power of a given feature may depend on measurement noise rather than biological properties. I therefore consider a wider definition of relevance that avoids this problem. The resulting feature selection problem is shown to be asymptotically intractable in the general case; however, I derive a set of simplifying assumptions which admit an intuitive, consistent polynomial-time algorithm. Moreover, I present a method that controls error rates also for this problem. This algorithm is evaluated on microarray data from case studies in diabetes and cancer.

In some cases however, I find that these statistical relevance concepts are insufficient to prioritize among candidate features in a biologically reasonable manner. Therefore, effective feature selection for life science requires both a careful definition of relevance and a principled integration of existing biological knowledge.

Sammanfattning

Sekvenseringen av det mänskliga genomet i början på 2000-talet tillsammans och de senare sekvenseringsprojekten för olika modellorganismer har möjliggjort revolutionerade nya biologiska mätmetoder som omfattar hela genom. Microarrayer, mass-spektrometri och SNP-typning är exempel på sådana mätmetoder. Dessa metoder genererar mycket högdimensionell data. Ett centralt problem i modern biologisk forskning är således att identifiera de relevanta variablerna bland dessa tusentals mätningar. Detta kallas för *variabelsökning*.

För att kunna studera variabelsökning på ett systematiskt sätt är en exakt definition av begreppet ”relevans” nödvändig. I denna avhandling behandlas relevans ur statistisk synvinkel: ”relevans” innebär ett statistiskt beroende av en *målvariabel*; denna kan vara kontinuerlig, till exempel en blodtrycksmätning på en patient, eller diskret, till exempel en indikatorvariabel såsom ”rökare” eller ”icke-rökare”. Olika former av relevans behandlas och en sammanhängande teori presenteras. Varje relevansdefinition ger därefter upphov till ett specifikt variabelsökningsproblem.

Prediktiva variabler är sådana som kan användas för att konstruera prediktionsmodeller. Detta är viktigt exempelvis i kliniska diagnossystem. Här bevisas att en konsistent skattning av sådana variabler kan beräknas i polynomisk tid, så att variabelssökning är möjlig inom rimlig beräkningstid. Detta är ett genombrott jämfört med tidigare forskning. Dock visas även att metoder för att optimera prediktionsmodeller ofta ger höga andelar irrelevanta varibler, vilket är mycket problematiskt inom biologisk forskning. Därför presenteras också en ny variabelsökningsmetod med vilken de funna variablernas relevans är statistiskt säkerställd. I detta sammanhang visas också att variabelsökningsmetoder inte är reproducerbara i vanlig bemärkelse i höga dimensioner, även då relevans är statistiskt säkerställd. Detta förklarar till viss del varför genetiska associationsstudier som behandlar hela genom hittills har varit svåra att reproducera.

Här behandlas också fallet där alla relevanta variabler eftersöks. Detta problem bevisas kräva exponentiell beräkningstid i det allmänna fallet. Dock presenteras en metod som löser problemet i polynomisk tid under vissa statistiska antaganden, vilka kan anses rimliga för biologisk data. Också här tas problemet med falska positiver i beaktande, och en statistisk metod presenteras som säkerställer relevans. Denna metod tillämpas på fallstudier i typ 2-diabetes och cancer.

I vissa fall är dock mängden relevanta variabler mycket stor. Statistisk behandling av en enskild datatyp är då otillräcklig. I sådana situationer är det viktigt att nyttja olika datakällor samt existerande biologisk kunskap för att för att sortera fram de viktigaste fynden.

Publications

The scientific publications underlying this Ph.D. thesis are:

1. Roland Nilsson, José M. Peña, Johan Björkegren, and Jesper Tegnér. *Evaluating feature selection for SVMs in high dimensions.* In Proceedings of the 17th European Conference on Machine Learning, pages 719-726, 2006.

2. José M. Peña, Roland Nilsson, Johan Björkegren, and Jesper Tegnér. *Identifying the relevant nodes before learning the structure.* In Proceedings of the 22nd Conference on Uncertainty in Artificial Intelligence, pages 367-374, 2006.

3. Roland Nilsson, José M. Peña, Johan Björkegren, and Jesper Tegnér. *Consistent Feature Selection for Pattern Recognition in Polynomial Time.* Journal of Machine Learning Research 8:589-612, 2007.

4. Roland Nilsson, José M. Peña, Johan Björkegren, and Jesper Tegnér. *Detecting Multivariate Differentially Expressed Genes.* BMC Bioinformatics, 2007 (in press).

5. Roland Nilsson, Johan Björkegren, and Jesper Tegnér. *Reliable discovery of predictive gene lists using the bootstrap.* Manuscript.

Acknowledgments

This thesis could never have been written without the joint effort of a small but bright and dedicated team — my supervisors at Linköping University, Karolinska Institutet and Clinical Gene Networks AB.

First, my main supervisor, professor Jesper Tegnér, who amazingly always manages to understand arbitrarily complicated problems in five minutes. More than once have a solution to an elusive puzzle dawned on me only when discussing the problem with Jesper.

Second, my main supervisor at Clinical Gene Networks AB, associate professor Johan Björkegren, who consistently provides a fresh "what-is-it-good-for?" perspective and a seemingly unlimited supply of creative (and crazy) ideas.

Third, a special acknowledgement to my co-supervisor Dr. José M. Peña, who first introduced me to the world of graphical models and who has been instrumental in many of the developments in this work. Your thorough knowledge, diligence and patience has been crucial at many points in the developments herein.

Finally, to the Computational Medicine team at Karolinska Institutet and Linköping University and the Bioinformatics group at Linköping University, for inspiration and discussions and also for kindly preventing me from starvation by telling me when it's time to stop staring at my theorems and go to lunch.

This work has been supported by the Swedish Knowledge Foundation through the Industrial PhD programme in Medical Bioinformatics at the Strategy and Development Office at Karolinska Institutet, Linköping University, and Clinical Gene Networks AB.

Contents

Notation

X	Feature vector; a vector-valued random variable
n	The dimension of X.
V_n	The set {1,. . . ,n}
$\mathcal{X}$	Domain (event space) of the random variable X
X_i	Feature; a component of the vector X, a random variable
X_S	For $S \subseteq \{1, \ldots, n\}$, the sub-vector $X_{\{i \in S\}}$ of X
$X_{\neg S}$	The sub-vector $X_{\{i \notin S\}}$ of X
$X_{1:n}$	Equal to X_S with $S = \{1, \ldots, n\}$
Y	Target variable; a random variable
Z	A pair of features and target, $Z = (X, Y)$
x	Observation of the random variable X
$x_i^{(j)}$	Observation j the random variable X_i in a sample
$x_j^{(1:l)}$	A sample (vector) of l observations of the random variable X_i
$X^{(1:l)}$	A vector of l independent, identical random variables X
$p(x)$	Probability mass function
$f(x)$	Probability density function
$P(\xi)$	Probability of an event $\xi \subset \mathcal{X}$
$p(y \mid x)$	Conditional probability of $Y = y$ given $X = x$.
$Y \perp X \mid Z$	Y is conditionally independent of X given Z
$Y \not\perp X \mid Z$	Y is conditionally dependent of X given Z
$g(x)$	Predictor; a function $\mathcal{X} \mapsto \mathcal{Y}$
g^*	The Bayes predictor
$\mathcal{G}$	A set (domain, class) of predictors
$I(Z^{(1:l)})$	Inducer; a map $\mathcal{Z}^l \mapsto \mathcal{G}$
$h(\hat{y} \mid y)$	Loss function
$R(g)$	Risk functional for classifier g
$\hat{R}(g)$	Empirical risk estimate for classifier g
$\rho(I)$	Expected risk for inducer I
S^*	The Bayes-relevant feature set (Definition 3.4)
$S^\dagger$	An expectation-optimal feature set (Definition 3.9)
$S^\ddagger$	Min-features set (Definition 3.10)
S^A	The set of all relevant features (Definition 3.11)
M^*	The Markov boundary of Y
$\mathbb{E}[X]$	Expectation value of X
$\mathcal{O}(f(n))$	Order of $f(n)$ (Landau notation)

1

Introduction

In the past decade, molecular biology has undergone something of a revolution due to the sequencing of the human genome. Two decades ago, a researcher seeking to discover molecular mechanisms behind a human disease was largely confined to explore variants or close relatives of already known genes and pathways, able to extend biological knowledge only in the immediate vicinity of already established facts. As a result, molecular biology has largely concentrated on detailed studies of a fairly small number of well-characterized mechanisms, rather than exploration of completely new terrain. For example, a few well-known protein families form the basis of most of the known pharmaceutical compounds, while the majority of human proteins are unexplored for this purpose [80].

The human genome project [30, 48] and the subsequent sequencing projects in mouse [31], rat [29] and other common model organisms is changing this situation drastically. The genome projects did not by far provide complete "maps" of biology, but knowledge of complete genome sequences for these important organisms has been crucial for the development of massively parallel measurement technologies. Today, microarrays and mass spectrometry-based methods allows measuring transcript levels [151], protein abundance or phosphorylation states [2] and DNA mutations [115], covering entire genomes. Such measurements are herein referred to as *genome-wide* in lack of a better term, although strictly speaking, components other than genes are often being measured.

With these new tools, biologists can now search for mechanisms behind biological processes — for example those contributing to human disease — in a much more objective, unbiased fashion. Correctly used, genome-wide technology can reveal novel genes, transcripts, proteins and entire signalling pathways. In this thesis, those genes, transcripts, proteins and pathways, or whatever the unit of information may be, are called *features*. The task of finding these pieces of information is called *feature selection*. With successful feature selection, genome-wide techniques holds the promise to open up entire new arenas of research.

At first glance, the genome-wide strategy is astoundingly simple: a researcher interested in discovering completely new biological mechanisms (and thereby publishing papers that will be cited for years to come) need only measure as many genes as possible, somehow "weed out" the genes that correlate with the process of interest, and compile a list of suspects. Unfortunately, this "high-throughput biology" idea suffers from one major problem: since measurements are always to some extent noisy, increasing the number of measured features will drastically increase the risk of finding "significant" features simply by chance. Thus, measurement noise effectively imposes a limit on how much information one can discover from high-dimensional data with a limited number of samples. A trade-off takes place: with more features, there are more potential features to discover; but at the same time, the power to discover each feature is reduced.

In the early days of genome-wide technology — around the mid-90's — this problem was not adequately appreciated by experimental researchers. Often, existing statistical methods developed for one-dimensional data were directly transferred to the new high-dimensional domain, usually meaning that statistical hypothesis tests were simply repeated thousands of times, whereafter the significant findings were selected. In some cases, experimental replication and statistical treatment was absent altogether [106]. As a result, many methodologically incorrect papers were published with alleged novel findings which were hard to reproduce and validate.

In the past few years, statisticians have turned their attention to these problems. As a result, more principled analysis methods have now emerged [5]. However, these developments have mostly treated the highly multivariate genome-wise data as a large number of univariate measurements, each considered more or less in isolation. This is natural, as this is the domain where statistical theory is most fully developed, but it is also rather restrictive. Simultaneously, data analysis methods from the field of *machine learning* has attracted considerable attention

in genome-wide data applications [40]. A large body of machine learning methods have been applied to genome-wide data in various forms, often for prediction problems such as cancer diagnosis [62], protein structure prediction [83] or gene/intron/exon prediction [188], but also for feature selection, for example in elucidating gene expression "signatures" of cancer cells [22, 142, 179].

However, while the machine learning field has developed a large body of theory and an impressive array of techniques for prediction problems, the purpose of feature selection in this context is traditionally different from that in biology: in machine learning, feature selection is a means to an end, a kind of pre-processing step applied to data with the ultimate goal of deriving better predictors [85]. The actual *identity* of the features (*e.g.*, which genes, proteins, or pathways are used by your predictor to diagnose leukemia?) is here less important. Broadly speaking, in machine learning, prediction accuracy is the goal. As a result, a statistical perspective which ensures that reasonably correct features are selected has been notoriously lacking. Indeed, it is often not clear what is a "correct" feature even *means*.

This clashes with the typical goal of the biologist, who is most interested in the mechanisms underlying the observed, predictable change in phenotype, for example, the mechanisms that transform a normal leukocyte into a malignant cancer cell. This is a pity, since machine learning — I believe — has a lot to offer to biology also in this area. This thesis is an attempt to somewhat improve the situation in this intersection between the fields of machine learning and statistics, and perhaps to some extent bridge between the two; hence the thesis title. I address basic questions such as what a "correct" or "relevant" feature is, how these concepts can be described by statistical models, and how to develop inference methods that control error rates in this setting. To introduce these problems and questions, perhaps a brief history of the research presented herein is in order.

1.1 A BRIEF BACKGROUND

The questions and ideas underlying this thesis began to take shape in early 2003, at the time of writing my Master's thesis at Stockholm Bioinformatics Center, also then supervised by Prof. Jesper Tegnér. We had briefly investigated some feature selection methods for the classification problems, and discovered some unsettling facts. Not only did the various methods tested select widely different genes for a given problem (Fig-

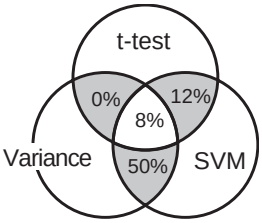

Figure 1.1: Overlap between the top 200 features selected by three ranking methods tested on the leukemia gene expression data set from Golub et al. [62]. For all methods, the corresponding predictor accuracy is around 95% [125].

ure 1.1), but they also tended to change their selections quite drastically when some of the samples in the data set in question was removed [125]. In other words, the methods were unstable and seemed to have different "goals", and we could not find any theory to to explain these goals or support one particular method over another.

Our intention at this time was to apply feature selection methods to a microarray gene expression data set being collected by Prof. Tegnér and Dr. Johan Björkegren's team, known as the the Stockholm Atherosclerosis Gene Expression (STAGE) study [70]. Our aim was to discover genes associated with different aspects of *atherosclerosis*, a complex inflammatory disease underlying clinical cardiovascular complications such as heart infarction and stroke [113]. The apparent lack of reliability in the feature selection methods tested was therefore was deeply troubling for our group. The underlying idea was that genes selected by these methods should form a "subsystem" which would be more amenable (due to its lesser dimensionality) to more advanced analysis such as network reconstruction [59, 168]. Thus, all subsequent work hinged upon the correctness of the initial feature selection step. We therefore decided to investigate feature selection more thoroughly, and attempt to find a theoretical justification that satisfied our needs. This became the motivation for my Ph.D. work.

We had several questions concerning feature selection which we could not resolve within available literature. The most important were the following:

1. What does "relevance" mean? When is a feature relevant to a

given target variable? Are there perhaps several perspectives on relevance, and do we need to choose a particular perspective in a subjective fashion?

2. What is the relation between a good predictive model and the features "relevant" to that model?

3. Why are feature selection methods so unstable? Is there any way to "stabilize" feature selection?

4. What types of feature selection methods are feasible given the limited sample sizes we have access to? (Presumably, too complex methods involve too many parameters to be useful with small samples sizes.)

To answer the first question, I undertook a theoretical study of the notion of relevance [126]. I decided that a statistical data model (where features are random variables following some underlying distribution) was reasonable in our context (Chapter 2). I found that there indeed were different notions of relevance, and that the choice between these notions largely depends on whether the end goal is prediction accuracy or feature error rate control (Chapter 3). The theoretical study laid the foundation of for this thesis by defining the "ground truth" for feature selection. This allows for more principled methods, avoiding *ad hoc* heuristics. Also, existing feature selection methods can be analyzed with respect to these relevance notions in order to better understand their function (Chapter 4).

In the course of this study I also found the first clues to the second question by characterizing the set of features relevant for prediction. As a rather unexpected side-development, this characterization also showed that, under some mild assumptions, it is possible to perform such feature selection in polynomial time. This is a strong improvement upon previous theory, which holds that the problem is intractable (Chapter 6). Similarly, I also studied the problem of discovering *all* relevant features (a larger set than the features relevant for prediction). While this problem was proven to be intractable in the general case, I found a set of reasonable conditions which again admit correct, polynomial-time algorithms (Chapter 8).

However, these theoretical results were mostly asymptotic (valid in the limit of infinite samples), and finding results for the small sample case proved difficult. I therefore conducted an extensive simulation study [127] to investigate the second question in more detail. This verified our

suspicions from my Master's thesis: at small samples, there is little correlation between obtaining good predictive models and selecting "relevant" features (Chapter 5). This phenomenon was especially pronounced for the best predictive methods, such as Support Vector Machines [32]: these often revealed few relevant features and large numbers of false positives, while nevertheless delivering highly accurate predictors. From this study I had to conclude that the typical predictive models were nearly useless for our purposes.

The simulation study also suggested an answer to the third question: many feature selection seem to be unstable because, for typical high-dimensional data, the feature selection problem is *under-determined*: there are many feature sets which are useful for making accurate predictions, and which subset happens to be selected given the data at hand is more less random. At this time (about mid-2005), a key paper by Ein-Dor et al. appeared, which promoted similar conclusions [46]. I therefore set out to investigate this problem more closely and attempt to find a remedy. Again turning towards simulations, I found that the instability was not only due to the under-determinedness of the problem, but in part also derived from low power due to small samples. I developed a simple feature selection framework based on the bootstrap, which for the first time allowed general feature selection with proper control of error rates (Chapter 7). This method was still to some extend unstable, but this instability is harmless and unavoidable. I also developed a method of error rate control for the problem of finding all relevant features (Chapter 8).

The fourth question was also answered, at least in part, by the simulation studies and the work on error control. By measuring or controlling error rates, respectively, one may simply assess the number of discoveries of any method for a given acceptable error rate as a function of the sample size. This number may then be compared to the actual number of true features (in simulations) or to an estimate thereof (in the bootstrap method. This is not an entirely satisfactory answer, since simulations always entail relevance problems (does the distribution used resemble real data?) and the error controlling methods are only approximate (the bootstrap method) or limited to particular distributions (Chapter 8), but it is currently the best I am aware of.

This is as far as my knowledge has reached at the time of writing this thesis. Along the way I also found that in some cases, feature selection solely based on experimental (*e.g.*, microarray) data sometimes renders *too large* gene sets to be readily interpretable, even with stringent error rate thresholds (Chapter 8). I therefore concluded — as have many oth-

ers have by now [60, 150] — that integration of other data types would be essential to identify the genes we were most interested in. Fortunately, the theory of feature selection presented herein is quite general and is in no way limited to particular data types. Therefore, I hope that this thesis will afford a useful framework also for discovering biological knowledge using multiple data sources. Work along these lines is currently in progress. I also plan to apply the methodology developed herein to more complicated inference problems such as network reconstruction, which also can be cast in the form of a feature selection problem. In conclusion, I anticipate that the theory and results presented herein should be of broad interest.

1.2 A guide to the thesis

While this work originated in biological scientific questions, the development of sound feature selection strategies for genome-wide data quickly became a rather theoretical task. Thus, most of the material is mathematical. I have tried to make the content as self-contained as possible by providing a fairly detailed background on machine learning and statistical concepts in Chapter 2, but nevertheless the text is probably not very accessible to readers without a fair background in mathematics.

For the reader with a more practical/biological background, I would recommend first reading the summaries at the end of each chapter, which I have strived to make more accessible. Also, some recommendations are provided in Section 9.2 which might be helpful to the newcomer. An overview of the remaining chapters follows.

In Chapter 2 I introduce statistical data models, the setting for the remainder of the thesis. I consider parametric and graphical/axiomatic distribution classes. I briefly review some principles of for statistical inference, with particular emphasis on methods for learning predictive models from data. I explain key theoretical concepts in machine learning such as over-fitting and regularization, and very briefly survey some popular methods in supervised learning.

In Chapter 3 I examine the feature selection problems and the concept of feature relevance in detail. I argue that feature selection can be used for different purposes, and that the end goal must be specified carefully before one can choose a method in a rational way. Thus, a number of different feature selection problems

are defined, describing the "ground truth" against which we may evaluate feature selection methods.

In Chapter 4 a number of existing feature selection methods for are reviewed. Importantly, I attempt to determine which of the problems defined in Chapter 3 each method tries to solve. This analysis is to my knowledge novel in many cases. Hopefully, this helps to bring some order to the multitude of available methods, which can be quite confusing.

In Chapter 5 I present a benchmark study in which the performance of some of the methods from Chapter 4 is assessed in the context of high-dimensional data. Importantly, building on the definitions established in Chapter 3, this also includes a thorough assessment of feature error rates.

In Chapter 6 some important theoretical results are presented which explain how the different feature selection problems relate to each other. This chapter also establishes *predictive* features can be found consistently in polynomial time (*i.e.*, with a reasonable amount of computations). This result is a major improvement over the long-standing consensus in the feature selection field which holds that the problem is intractable.

In Chapter 7 I consider the issue of feature selection instability. I show that instability derives may result from low power to detect truly relevant features, so that it does not imply excessive amounts of false positives. I develop a general framework for error control for feature selection methods based on the bootstrap. This methodology is shown to be sound in simulations studies, and some results on gene expression data is presented.

In Chapter 8 I consider the problem of discovering *all* relevant features, as opposed to only the most predictive ones. This problem is shown to be intractable in the general case, but feasible in a somewhat restricted class of data distributions. I propose two algorithms which I show to be consistent, and develop a methods for error control also for this problem. Case studies indicate that this approach is useful for genome-wide applications with high noise levels.

In Chapter 9 I present my overall conclusions, provide some recommendations for practical applications, and outline some possible future developments.

2

Statistical Data Models

In a statistical data model, we think about experimental systems as statistical distributions. A biological experimental system might be human patients or biological model organisms such as *mus musculus* (mouse), or perhaps a cell culture. We observe the system by making measurements, hoping that these measurements can be used to derive facts or at least corroborate hypotheses about the system. A schematic of this perspective is given in figure 2.1. The model is "statistical" in that, when experiments are repeated, there is some random variation in the measurements which we cannot explain by the experimental design.

For example, we might study the blood cholesterol level in mice on different diets, as illustrated in Figure 2.2. While we expect — or rather, hope — to find variation in the measurement (cholesterol level) related to the diet, we probably also realize that across different individuals, there will also be variation unrelated to that diet. For example, the cholesterol level might vary with age, or depend on unknown genetic factors. We might be able to control some of these "nuisance" variables by experimental design (choose mice of the same age for the experiment), but this is not always possible. For example, even with careful breeding, genotypes are never completely identical. Moreover, many factors that influence the cholesterol levels are probably unknown to us since our knowledge of biology is incomplete, and these are of course impossible to control. Also, the measurements themselves may be more or less corrupted with noise from various physical or chemical factors.

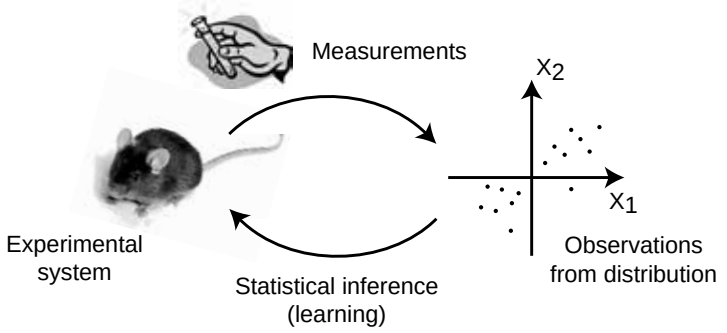

Figure 2.1: An schematic of the statistical data model. Left, an experimental system, here represented by the mouse. Top, taking measurements amounts to observing the system. Right, repeated measurements yields observations following some statistical distribution. Bottom, from a set of such distributions (a sample), statistical inference (learning) is used to learn new facts about the system.

Therefore, one will inevitably observe variation that cannot be explained. We describe this variation using statistical models, using probability distributions to capture the fact that measurements are always slightly uncertain. In this view, we assume that there indeed exists a true distribution, the properties of which is determined by some *parameters*, but that noise and limited samples prevents us from determining those parameter exactly.

In a statistical perspective, we often speak about any variation that cannot be explained by the chosen model as *noise*. However, it should be understood that this does not imply that the variation is truly random. Much of the variation one observes between individuals in biology is probably deterministic, and could in principle be explained if our knowledge of biology was more complete. But biological systems are very complex and our knowledge about them is merely partial. Therefore, the variation we cannot explain must at present be regarded as noise, in absence of any better alternative.

Let us establish some definitions. Throughout, I will represent experimental measurements by a vector $X = (X_i, \ldots, X_n)$ of random variables. Its components X_i are called *features*, following machine learning terminology. I also refer to X as the feature vector. The domain or *event space* of X is the set of possible observations, denoted by calligraphic $\mathcal{X}$. The domains of the individual features are correspondingly denoted

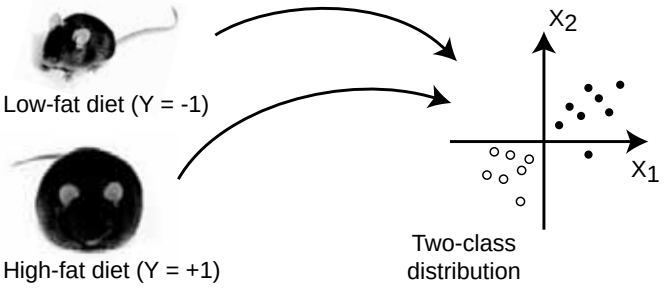

Figure 2.2: An example of a two-class experiment. Left, the classes are represented by mice on low-fat (lean) and high-fat (obese) diets. Taking measurements together with the class variable results in a two-class distribution (right), different classes indicated by open or filled circles.

by $\mathcal{X}_i$, etc. Since the features we are considering are mostly physical measurements, we will typically take $\mathcal{X}_i = \mathbb{R}$, so that $\mathcal{X} = \mathbb{R}^n$. In general, we will take $\mathcal{X}$ to be a cartesian product of the domains of the features, $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_n$. In this thesis, I will in particular study the relation between X and a *target variable* Y. A target variable is some known, well-understood factor such as gender, age, body weight, whether a person has a particular type or cancer or not, *etc.* In contrast, the features X_i are typically less well understood variables, for example indicating the presence of genetic mutations at certain points in the genome, so that by discovering relations between these less known features and the target variable, we may learn something about the features. At this point, an example of these rather abstract notions might be helpful. Let us return again to the mouse example of Figure 2.2.

Example 2.1 *Consider an experiment where two populations of mice are given two different diets with high and low fat content, respectively (Figure 2.2). This is called a two-class experiment; each of the populations is referred to as a class, and the target variable is discrete, e.g.,* $\mathcal{Y} = \{+1, -1\}$*, with* $+1$ *denoting the high-fat diet population and* -1 *the low-fat diet population. Now consider some measurements: for example, it might be interesting to measure the low density lipoprotein (LDL) and high density lipoprotein (HDL) blood cholesterol levels in each population. This yields a two-dimensional* X *(two features representing LDL and HDL) and thus a* $f(x, y)$*. Interesting parameters of this distribution could be the difference in the mean (expected) values for each class. This would determine how the HDL and LDL levels are*

related to the difference in diet.

When Y is discrete, we refer to the model as a *classification* model. One could of course extend the above example to several classes (consider, for example, using several different strains of mice). In this case we usually take $Y = \{1, 2, \ldots, K\}$ to represent K classes. In the case of two-class problems we will often use $\{+1, -1\}$ as it is a mathematically convenient notation; of course, the particular values of Y serve only as indicators and have no meaning *per se*). Most of the examples in this thesis concerns two-class problems. However, generalizations to multiple classes is often straightforward.

The target variable could also be continuous. In Example 2.1, we might let the target variable be the weight or age of the mice. When studying how the features relate to continuous variables, we speak of a *regression* model. For regression, we typically have $\mathcal{Y} = \mathbb{R}$. Although I treat regression problems at some points in this thesis, my main focus will be on classification.

In the statistical data model, both the features X and the target variable Y are random (stochastic) variables with a joint distribution over (X, y), specified by the probability density function $f(x, y)$, or, if X is discrete, by the probability mass function $p(x, y)$. This *data distribution* contains all information about the features X_i, their statistical relations to each other and their relations to the target variable Y. Learning from experiments is therefore equivalent to learning properties of this distribution.

In practise, the data distribution $f(x, y)$ is of course unknown, and one can obtain information about it only indirectly, through observations $(x^{(i)}, y^{(i)}) \in \mathcal{X} \times \mathcal{Y}$ from this distribution. I will denote a set of such pairs of observations by $z^{(1:l)} = \{z^{(1)}, \ldots, z^{(l)}\} = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(l)}, y^{(l)})\}$, where l is the sample size. This process has many names in the literature: statistical *learning*, statistical *inference*, *induction* or *estimation*. Through inference, we learn new facts about the experimental system. Depending on what properties of $f(x, y)$ we are interested in, different inference methods can be used.

The main topic of this thesis is a type of inference that attempts to determine *which* of the features X_i are "related" to the target variable Y. This type of inference is called *feature selection*. Feature selection is a special inference problem in that the question "is X_i related to Y" is a discrete, binary problem; there are only two possible answers, "yes" or "no". This discrete nature sometimes gives rise to hard combinatorial

problems; this is one reason why feature selection is difficult. The feature selection problem will be treated in detail in Chapter 3. The remainder of this chapter will be devoted to a brief survey of methods for statistical inference. This chapter is intended as a reference and as a way of relating the remainder of the thesis to other techniques which perhaps are better known to the reader. It is not necessary to read this entire chapter to appreciate the main contributions of the thesis; one may skip ahead at this point if desired, and come back to the material below when needed.

Before delving into details, for completeness it should be noted that while the statistical perspective is probably the most common in any kind of data analysis, not all inference problems are suitable to be handled statistically. For example, in applications that involve logical inference from observed facts, the "measurements" are truly noise-free, and a deterministic view of data may be more appropriate [177]. Such problems are outside the scope of the present work, however. For biological data, the statistical model is usually appropriate, and in this thesis I will make use of it exclusively.

2.1 Parametric models

2.1.1 The exponential family

To make any kind of statistical inference, it is necessary to introduce some assumptions. In classical statistics, this is done by assuming that the data comes from a particular distribution family (a fix set of distributions). Among all distributions in such a family, a particular one can be identified by a set of *parameters*. The problem of identifying the distribution from data thus becomes equivalent to of identifying these parameters.

A particularly tractable family of distributions which one often encounters in statistics is the *exponential family*, consisting of distributions with densities of the form

$$f(x) = \exp\left\{\sum_i \theta_i \phi_i(x) - g(\theta)\right\}. \tag{2.1}$$

Here $\theta = \{\theta_i\}$ is a parameter vector; a particular value of θ identifies a particular member of the family. Thus we may casually equate a distribution with a parameter value θ. The $\phi_i(x) : \mathcal{X} \mapsto \mathbb{R}$ are known as *sufficient statistics*; I will explain the meaning of this term shortly. The

$g(\theta)$ is a normalization term, which ensures that the density integrates to 1 (as it must, being a probability). From the identity $\int_{\mathcal{X}} f(x)dx = 1$ we immediately find

$$g(\theta) = \ln \int \exp\left\{\sum_i \theta_i \phi_i(x)\right\} dx, \tag{2.2}$$

which may be used to compute the normalization term for a particular choice of ϕ and θ.

The most well-known member of the exponential family is probably the Gaussian distribution, which we will make use of quite often.

Example 2.2 *The univariate Gaussian distribution is given by*

$$f(x) = N(x \,|\, \mu, \sigma) = (2\pi)^{-1/2}\sigma^{-1} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}, \quad x \in \mathbb{R}. \tag{2.3}$$

To put this in the form (2.1), write

$$\begin{aligned}
\ln f(x) &= -\frac{(x-\mu)^2}{2\sigma^2} + \frac{1}{2}\ln(2\sigma\pi^2) \\
&= -\frac{x^2 + \mu^2 - 2x\mu}{2\sigma^2} + \frac{1}{2}\ln(2\sigma\pi^2) \\
&= \frac{x\mu}{\sigma^2} - \frac{x^2}{2\sigma^2} - \left(\frac{\mu^2}{2\sigma^2} - \frac{1}{2}\ln(2\sigma\pi^2)\right) \\
&= \sum_i \theta_i \phi_i(x) - g(\theta),
\end{aligned}$$

here identifying $\phi(x) = (x, x^2), \theta = (\mu\sigma^{-2}, -\sigma^{-2}/2)$ *and* $g(\theta) = \mu^2\sigma^{-2}/2 - \ln(2\sigma\pi^2)/2$. *Solving the latter for* θ_1, θ_2 *yields*

$$g(\theta) = -\frac{1}{4}\theta_1^2\theta_2^{-1} + \frac{1}{2}\ln(2\pi^2) + \frac{1}{4}\ln(-2\theta_2).$$

While the parametrization θ may not be the most convenient in this case, the above demonstrates that that the Gaussian distribution is indeed of the form (2.1). Similar derivations can be made for the multivariane Gaussian distribution and for a number of other distributions. Table 2.1 lists some distributions in the exponential family, covering a wide variety of common statistical models. Many inference methods throughout this chapter can be related to members of this family.

Name	$\mathcal{X}$	$\phi(x)$
Bernoulli	$\{0, 1\}$	x
Gaussian	$\mathbb{R}$	(x, x^2)
Exponential	$(0, \infty)$,	$-x$
Poisson	$\{0, 1, \dots\}$	x
Laplace	$[0, \infty)$	x
Gamma	$[0, \infty)$	$(\ln x, x)$
Beta	$[0, 1]$	$(\ln x, \ln(1 - x))$

Table 2.1: Some distributions in the exponential family.

2.1.2 Maximum likelihood estimation

Learning a distribution from data means to estimate the parameters θ, and given a vector of sufficient statistics ϕ, this is particularly simple in exponential families using the maximum likelihood (ML) principle. In ML estimation, assuming independence of the given observations $x^{(1:l)} = \{x^{(1)}, \dots, x^{(l)}\}$, we simply maximize the joint likelihood

$$L(\theta, x^{(1:l)}) = f(x^{(1)}, \dots, x^{(l)} \mid \theta) = \prod_i f(x^{(i)} \mid \theta).$$

Although the *likelihood function* $L(\theta, x^{(1:l)})$ is identical to the density, we sometimes distinguish between the two because their interpretation is different. In the likelihood function, the observations x_i (the data from an experiment) are treated as constants and we study its dependence on θ to find the most parameter value most *likely* to have generated the data. In the probability distribution, on the other hand, the parameter θ is a (possibly unknown) constant, and x is the variable. The maximum likelihood principle and maximum likelihood estimation is by far the most common statistical inference method in use. It has been studied extensively for nearly a century [4] and has very strong support [21].

Since the logarithm function is monotone, for any distribution in the exponential family one may obtain the ML parameter estimate by maximizing

$$\ln L(\theta, x^{(1:l)}) = \sum_i \sum_k \theta_k \phi_k(x^{(i)}) - l g(\theta). \tag{2.4}$$

Setting its derivatives to zero,

$$\frac{\partial(\ln L(\theta, x^{(1:l)}))}{\partial \theta_k} = \frac{\partial}{\partial \theta_k} \sum_i \theta_k \phi_k(x^{(i)}) - l \frac{\partial}{\partial \theta_k} g(\theta) = 0,$$

we conveniently obtain the maximum likelihood solution

$$\frac{\partial}{\partial\theta_k} g(\theta) = \frac{1}{l}\sum_k \phi_k(x^{(i)}) \tag{2.5}$$

Thus, the estimate for parameter θ_k is obtained by averaging the sufficient statistic $\phi_k(x^{(i)})$ over the samples. This explains the term "sufficient statistic": the values of the functions ϕ_k contain all information we need for learning, so that, having computed these ϕ_k, all other parts of the data is irrelevant and can be discarded.

In some cases, depending on the form of $g(\theta)$, equation (2.5) can be solved analytically for θ. In the Example 2.2, we identified the sufficient statistics $\phi_1 = x$ and $\phi_2 = x^2$. Equation (2.5) then becomes

$$\begin{aligned} -\frac{1}{2}\theta_1\theta_2^{-1} &= \frac{1}{l}\sum x^{(i)} \\ \frac{1}{4}\theta_1\theta_2^{-2} - \frac{1}{4\theta} &= \frac{1}{l}\sum (x^{(i)})^2 \end{aligned}$$

which in the (μ, σ) parametrization gives the familiar ML estimates $\mu^* = \bar{x} = \sum_i x^{(i)}/l$ and $(\sigma^*)^2 = \sum_i (x^{(i)} - \bar{x})^2/l$. The most important fact about the ML method however, is that the problem of maximizing the likelihood is always numerically tractable, even when no analytic solution is available. To see this, note that

$$\begin{aligned} \frac{\partial g(\theta)}{\partial\theta_i} &= \frac{\partial}{\partial\theta_i}\left(\ln\int \exp\left\{\sum_i \theta_i\phi_i(x)\right\} dx\right) \\ &= \frac{\int \phi_i(x)\exp\left\{\sum_i \theta_i\phi_i(x)\right\} dx}{\int \exp\left\{\sum_i \theta_i\phi_i(x)\right\} dx} \\ &= \mathbb{E}\left[\phi_i(X)\right], \end{aligned}$$

and similarly,

$$\frac{\partial^2 g(\theta)}{\partial\theta_i\partial\theta_j} = \mathrm{Cov}(\phi_i(X), \phi_j(X))$$

Since the covariance matrix is always positive definite, it follows that $g(\theta)$ is always convex. Therefore, the global maxima of (2.4) can always be found by numeric optimization (for example using Newton's method). This feature makes the exponential family very useful in practise. Several machine learning methods can be seen as variants of maximum likelihood estimation; see Sections 2.7.2 and 2.7.3.

2.2 Graphical models

An important aspect of a probabilistic model is *independence* between variables.

Definition 2.1. *Two discrete variables X, Y are said to be independent if*

$$p(x, y) = p(x)p(y) \quad \forall x, y.$$

This is denoted $X \perp Y$. Two variables X, Y are said to be conditionally independent given the observation $Z = z$ if

$$p(x, y \mid Z = z) = p(x \mid Z = z)p(y \mid Z = z) \quad \forall x, y.$$

If the above holds for all z,

$$p(x, y \mid z) = p(x \mid z)p(y \mid z) \quad \forall x, y, z,$$

we say that X is conditionally independent of Y given Z, denoted $X \perp Y \mid Z$.

The above definitions are appropriate for discrete X, Y, Z; for continuous variables, the definition should be modified so that the factorizations must hold *almost surely*, that is

$$P\left(f(X, Y) = f(X)f(Y)\right) = 1,$$

and similar for the conditional case.

Independencies are useful because these factorizations simplify the mathematics of parameter inference. To represent (conditional) independencies we use *graphical models*. A graphical model over n variables can be defined as a graph G over the vertices $\{1, \ldots, n\}$ taken together with a *criterion* for reading dependencies or independencies from that graph. Graphical models can be constructed using both directed and undirected graphs. The undirected graphs are in some sense simpler, but also less powerful. They give rise to *Markov networks*, while directed graphs yield *Bayesian networks*.

2.2.1 Markov networks

An undirected graph G can be used as a graphical probability model using the following criterion for reading independencies.

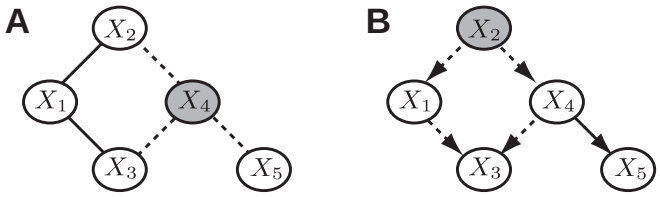

Figure 2.3: Examples of graphical independence criteria. A: An undirected graph. By the U-separation criterion of Definition 2.2 we can identify for example $X_{\{2,3\}} \perp_G X_5 \,|\, X_4$ (highlighted node and dashed edges). B: A directed graph. By the D-separation criterion (Definition 2.6) we here find that $X_1 \perp X_4 \,|\, X_2$, since X_3 has two parents in the path between X_1 and X_4 (that is, here $X_1 \not\perp X_4 \,|\, X_{2,3}$).

Definition 2.2 (U-separation)**.** *For an undirected graph G and three disjoint subsets R, S, T of V_n, we say that R is U-separated from S by T in G, denoted $R \perp_G S \,|\, T$, if and only if there is a node $k \in T$ in every path from each $i \in R$ to each $j \in S$.*

This criterion has the intuitive interpretation that two nodes are conditionally independent if the conditioning set "blocks" all paths between the two. An illustration is given in Figure 2.3A.

Naturally, to be useful for inference the graph G must be such that the criterion $\perp_G$ does indeed identify independencies that hold true in the actual distribution P which G is supposed to model. This requirement is embodied in the following definition.

Definition 2.3 (Independence map)**.** *A graphical model G is an independence map (I-map) of a distribution $f(x)$ over X if it satisfies*

$$R \perp_G S \,|\, T \implies X_R \perp X_S \,|\, X_T \tag{2.6}$$

for all disjoints sets R, S, T.

Note that the I-map property is only "one-way": if G is an I-map, then we can use $\perp_G$ to find independencies in P, but we cannot identify *dependencies*, because it is not clear that $R \not\perp_G S \,|\, T$ implies $X_R \not\perp X_S \,|\, X_T$ (the converse of (2.6) need not hold). Thus, one cannot in general interpret edges in a graphical model as dependencies. A graphs where this is possible is called a *dependence map* (D-map). A graph which is both

an I-map and a D-map is called a *perfect map*. Thus, in a perfect map, $R \perp_G S \,|\, T$ is equivalent to $X_R \perp X_S \,|\, X_T$. However, many distributions do not have perfect undirected maps, because their independence structure is not possible to represent with undirected graphs.

Example 2.3 *Consider the distribution over $\mathcal{X} = \{0,1\}^3$ given by*

$$P(X_3 = 1 \,|\, x_1, x_2) = \begin{cases} 1/5, & x_1 = 1 \ \wedge \ x_2 = 1 \\ 4/5, & \textit{otherwise} \end{cases}$$

$$p(x_1, x_2) = 1/4$$

This distribution clearly satisfies the marginal independence $X_1 \perp X_2$, but

$$p(x_1, x_2 \,|\, X_3 = 1) = \begin{cases} 4/7, & x_1 = 1 \ \wedge \ x_2 = 1 \\ 1/7, & \textit{otherwise} \end{cases}$$

So that $X_1 \not\perp X_2 \,|\, X_3$. It is easy to see that no undirected graph can represent both of these statements simultaneously.

In cases like this, an alternative to a perfect map is the following.

Definition 2.4 (Minimal I-map)**.** *A graph G is a minimal I-map if of a distribution P if G is an I-map of P while no $G' \subset G$ is an I-map of P.*

Here, "minimality" essentially means that there are no "unnecessary" edges in G; if we were to remove any edge from G, then the criterion $\perp_G$ we would give us additional independencies that do hold true in the actual distribution. The minimal I-map is the "best" alternative in the sense that it allows one to identify as many independencies as possible. This is highly desirable because independencies simplify statistical inference by "uncoupling" features from each other, so that one may solve a number of small inference problems instead of one large problem, in a "divide-and-conquer" fashion.

Relying on the I-map property, we now define the Markov network model.

Definition 2.5. *A Markov network for a given distribution $f(x)$ is an undirected graph G which is a minimal I-map of $f(x)$ and satisfies*

$$f(x) = \prod_k \psi_k(x_{C_k}), \tag{2.7}$$

where $\{C_k\}$ is the set of maximal cliques of G. The factors $\psi_k(x_{C_k})$ are called potential functions.

Markov networks are also known as Markov Random Fields, especially in physics applications. I will not make use of the actual factorization of $f(x)$ into potential functions in this thesis; I provide the above definition for completeness. It should be noted that the factors ψ_k are not themselves probabilities. A thorough treatment of Markov networks is given by Chellappa and Jain [24].

2.2.2 BAYESIAN NETWORKS

The most popular graphical model is probably the *Bayesian network*, a directed graph which can be used to encode causality as well as statistical dependence [129]. Bayesian networks are powerful models, but they are also somewhat more complicated than Markov networks. This is evident already in the definition of the the independence criterion, which is more involved than for undirected graphs.

Definition 2.6 (D-separation)**.** *For an directed, acyclic graph G and three disjoint subsets R, S, T of $\{1, \ldots, n\}$, we say that R is D-separated from S by T in G, denoted $R \perp_G S \,|\, T$, if and only if there is a node k in every undirected path from each $i \in R$ to each $j \in S$, such that either (i) k has two parents in the path and neither k nor any descendant of k is in T, or (ii) k has less than two parents in the path and k is in T.*

This independence criterion is illustrated in Figure 2.3B. Using the D-separation criterion, we define a Bayesian network as follows.

Definition 2.7. *A Bayesian network (BN) for a distribution $f(x)$ is a directed acyclic graph G which is a minimal I-map of of $f(x)$ and satisfies*

$$f(x) = \prod_i f(x_i \,|\, x_{\Pi_i}), \tag{2.8}$$

where Π_i is the parents of i in G. The factors $f(x_i \,|\, x_{\Pi_i})$ are called local distributions.

As with the undirected models of the previous section, the graph structure G of a Bayesian network is useful because it simplifies statistical inference: instead of inferring parameters for the full n-dimensional $f(x)$, we may now divide the inference problem into n smaller inference problems for the local distributions $f(x_i \,|\, x_{\Pi_i})$, and then compute the full $f(x)$ from Equation (2.8).

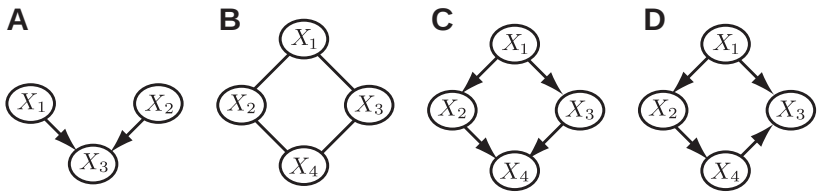

Figure 2.4: A: A Bayesian network which cannot be represented as a Markov network. B: A Markov network cannot be represented as a Bayesian network.

Example 2.4 *Any distribution $p(x)$ over $\mathcal{X} = \{0,1\}^n$ can be described by a set of 2^n parameters*

$$\theta_k = P\left(\sum_{i=1}^{n} 2^{X_i} = k\right), \quad k = 1, \ldots, 2^n.$$

For a given sample $x^{(1:l)}$, these θ^k have the straightforward ML estimates

$$\hat{\theta}_k = \frac{1}{l}\left|\left\{x^{(j)} : \sum_{i=1}^{n} 2^{x^{(j)}{}_i = k}\right\}\right|.$$

Clearly, the number of samples l required for an accurate estimate of θ is on the order of 2^k. However, if $p(x)$ can be represented by a Bayesian network such that each node i has at most $K < n$ parents, then each local distributions $p(x_i \,|\, x_{\Pi_i})$ involve no more than 2^K parameters. Thus, for such a Bayesian network, no more than $n2^K \ll 2^n$ are non-zero, simplifying the estimation problem considerably.

An advantage with the Bayesian network representation is that the local distributions are ordinary conditional probabilities, which are usually easier to interpret than the potential functions of Markov networks. Moreover, Bayesian networks are capable of representing some distributions which cannot be represented by a Markov network: the BN in Figure 2.4A is a perfect map of the distribution in Example 2.3. However, there are also examples of the opposite: the Markov network in Figure 2.4B has no directed perfect map, since no matter how the edges are oriented, we always get at least one node with two parents, which leads to an extra dependency not present in the undirected map. For example, in Figure 2.4C we get $X_2 \not\perp X_3 \,|\, X_{1,4}$ and in Figure 2.4D we

get $X_1 \not\perp X_4 \,|\, X_{2,3}$. Nevertheless, Bayesian networks are generally considered to be more "expressive" (be able to represent a wider class of distributions) than Markov networks [130].

Since by definition a BN is an independence map, we know that it can be used to identify independencies. However, the above example shows that in general, we cannot use Bayesian networks to identify dependencies. This is problematic in genomic applications, where one is often very interested in identifying interacting nodes (*e.g.*, proteins or mRNA transcripts). To avoid this problem, it is common to simply assume that the data distribution has a perfect directed map.

Definition 2.8. *A distribution that has a perfect directed map is said to be faithful to a Bayesian network (DAG-faithful).*

In biology, this "faithfulness" assumption can be motivated by the view that genes, proteins, metabolites, *etc.* interact with each other in a "network", that is, in pairwise interactions, thus forming a graph structure [95]. See Section 8.2.3 for a discussion on this issue.

When the graph G is unknown, which is typically the case, then one must first infer it from data. This is a difficult computational problem. Inference of Bayesian is known to be asymptotically NP-hard [25, 26]. However, several more or less heuristic algorithms exist, which have been shown to be effective in particular cases [122]. All of these assume that the data distribution is DAG-faithful. I will not consider methods for inferring graphical models in this thesis; rather, I will use graphical models as theoretical tools.

2.2.3 PROBABILITY AXIOMS

Conditional independence provides a different way of defining classes of distributions, by define a set of "probability axioms", properties that members of a class must satisfy. This manner of defining classes of distributions is sometimes called *axiomatic characterization*. The approach was pioneered by Pearl [130], and is closely related to graphical models of probability distributions. I here briefly review some commonly used probability axioms. These will be needed for various proofs later on.

The following theorem due to Pearl [130] establishes the basic properties of conditional independence that all probability distributions satisfy. Below, for brevity I use juxtaposition of sets as a shorthand for union, *i.e.*, $X_{ST} = X_{S \cup T}$.

Theorem 2.9. *Let R, S, T, U denote any disjoint subsets of V_n. Any probability distribution over X satisfies the following properties:*

$$\begin{aligned}
&\textit{Symmetry:} \quad X_S \perp X_T \,|\, X_R \implies X_T \perp X_S \,|\, X_R \\
&\textit{Decomposition:} \quad X_S \perp X_{TU} \,|\, X_R \implies X_S \perp X_T \,|\, X_R \\
&\textit{Weak union:} \quad X_S \perp X_{TU} \,|\, X_R \implies X_S \perp X_T \,|\, X_{RU} \\
&\textit{Contraction:} \quad X_S \perp X_T \,|\, X_{RU} \wedge X_S \perp X_U \,|\, X_R \implies X_S \perp X_{TU} \,|\, X_R
\end{aligned}$$

Since these properties are "universal", they do not identify any particular class of distributions, but they are useful for proving other results. If we assume any further properties, we will effectively restrict attention to the set of distributions which satisfy the properties we require. I will make use of this technique in Chapters 6 and 8 to simply feature selection problems. The following important property is satisfied by the set of distributions that are everywhere strictly positive.

Theorem 2.10. *Let R, S, T, U denote any disjoint subsets of V_n. Any distribution such that $f(x_{RST}) > 0$ satisfies the following property:*

$$\textit{Intersection:} \quad X_U \perp X_R \,|\, X_{ST} \wedge X_U \perp X_T \,|\, X_{SR} \implies X_U \perp X_{RT} \,|\, X_S$$

Proof. The statement $X_U \perp X_R \,|\, X_{ST}$ is equivalent to

$$\begin{aligned}
f(x_{RSTU}) &= f(x_U \,|\, x_{RST}) f(x_{RST}) \\
&= f(x_U \,|\, x_{ST}) f(x_{RST}),
\end{aligned}$$

for every $x \in \mathcal{X}$. Similarly, $X_U \perp X_T \,|\, X_{SR}$ is equivalent to

$$f(x_{RSTU}) = f(x_U \,|\, x_{RS}) f(x_{RST}),$$

also for every $x \in \mathcal{X}$. Since $f(x_{RST}) > 0$, it now follows that

$$f(x_U \,|\, x_{ST}) = f(x_U \,|\, x_{RS})$$

Therefore both of these probabilities must be constant with respect to both R and T, that is,

$$f(x_U \,|\, x_{ST}) = f(x_U \,|\, x_{RS}) = f(x_U \,|\, x_S).$$

Hence, $X_U \perp\!\!\!\perp X_R \,|\, X_S$ and $X_U \perp X_T \,|\, X_S$ holds. The intersection property then follows using the contraction property togeher with the assumptions,

$$X_U \perp X_R \,|\, X_{ST} \wedge X_U \perp X_T \,|\, X_S \implies X_U \perp X_{RT} \,|\, X_S.$$

□

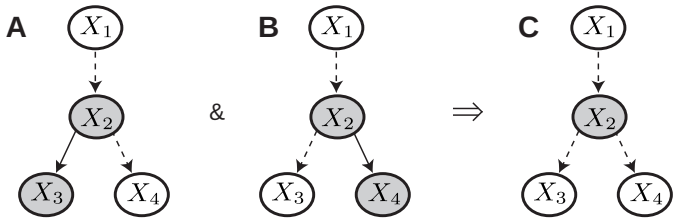

Figure 2.5: An illustration of the intersection property. A: Conditioning on the set $\{X_2, X_3\}$ (gray nodes) "blocks the information flow" from X_1 to X_4 (dashed arrows), yielding the conditional independence $X_1 \perp X_4 \,|\, X_{2,3}$. B: Similarly, Conditioning on $\{X_2, X_4\}$ gives $X_1 \perp X_4 \,|\, X_{2,3}$. C: If the probability distribution satisfied intersection, (A) and (B) implies that $X_1 \perp X_{3,4} \,|\, X_2$, *i.e.*, X_2 must be the node that blocks the paths.

This intersection property essentially states that if both of the independencies to the right hold, then it must be the variables (S) that are responsible for "blocking the flow of information" from R and T to U, rendering U independent from both R and T conditioned on S (Figure 2.5).

The following properties are useful when they hold true, but they are not satisfied by all distributions.

Definition 2.11. *Let R, S, T, U denote any disjoint subsets of V_n and also take $i \in V_n$.*
Composition:

$$X_S \perp X_T \,|\, X_R \;\wedge\; X_S \perp X_U \,|\, X_R \implies X_S \perp X_{TU} \,|\, X_R$$

Strong transitivity:

$$X_R \perp X_T \,|\, X_U \implies X_R \perp X_S \,|\, X_U \;\vee\; X_S \perp X_T \,|\, X_U$$

Weak transitivity:

$$X_S \perp X_T \,|\, X_R \wedge X_S \perp X_T \,|\, X_{R\cup\{i\}} \implies X_S \perp X_i \,|\, X_R \vee X_i \perp X_T \,|\, X_R$$

The strong transitivity property is perhaps easier recognized in its contrapositive form,

$$X_R \not\perp X_S \,|\, X_U \;\wedge\; X_S \not\perp X_T \,|\, X_U \implies X_R \not\perp X_T \,|\, X_U.$$

A common misconception is to assume that strong transitivity holds true in all distributions. This is disproved by Example 2.3, where $X_1 \not\perp X_3$ and $X_3 \not\perp X_2$, but $X_1 \perp X_2$. The composition and weak transitivity properties are satisfied by all DAG-faithful distributions [130]. I will make use of these properties in Chapter 8.

2.3 Conditional probability models

In our setting, we are often not interested in learning *everything* about the data distribution. Moreover, for whole-genome data sample sizes are typically small compared to the number of features (the dimension of X), so that estimating the entire distribution is not realistic. Instead, given a joint data distribution of $f(x, y)$ over features and the target variable Y, we will focus on the conditional density

$$f(y \,|\, x) = \frac{f(x, y)}{f(x)}. \tag{2.9}$$

This is referred to as the *posterior* probability of Y given (conditioned on) X, or just "the posterior", for short. In words, the posterior is the distribution of the target variable conditioned on having observed $X = x$. This is the only distribution that matters for prediction: all information we can possibly obtain about Y from the observation $X = x$ is contained in $f(yx)$. Learning the posterior is often referred to as *supervised learning*, the idea being that the target variable acts as a "supervisor" which provides the "correct" value y_i for each observation x_i.

If our end goal is to find the posterior, then it seems reasonable to estimate it directly, rather than "taking a detour" by first estimating the full density $f(x, y)$ and then computing $f(y \,|\, x)$ from (2.9). This intuition is correct, and in fact, the posterior is often much easier to estimate than the full $f(x, y)$. This is seen in the following very common distribution example.

Example 2.5 *A two-class multivariate Gaussian mixture with* $\mathcal{Y} = \{+1, -1\}$ *is defined by*

$$f(x, y) = p(y)N(x \,|\, \mu_y, \Sigma) + p(-y)N(x \,|\, \mu_{-y}, \Sigma).$$

Here $p(y)$ *denotes the marginal class probability. Importantly, the covariance matrix* Σ *is here set to be equal for both classes. Without loss*

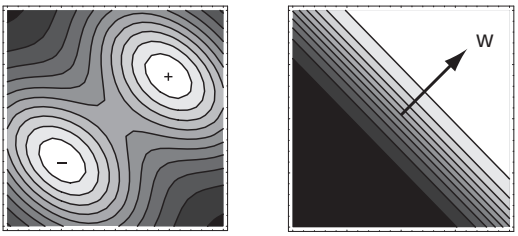

Figure 2.6: A two-class Gaussian distribution in $\mathbb{R}^2$. Left, the mixture $f(x) = \sum_y f(x \,|\, y)$ distribution, with plus/minus signs indicating the mixture components $f(x \,|\, Y = +1)$ and $f(x \,|\, Y = -1)$, respectively. Right, the posterior $p(y \,|\, x)$. Arrow indicates the parameter vector w.

of generality we may also assume $\mu = \mu_{+1} = -\mu_{-1}$, since all problems can be reduced to this case by the translation $X' \leftarrow X - (\mu_{+1} + \mu_{-1})/2$. We rewrite the posterior as

$$\begin{aligned} p(y \,|\, x, \mu, \Sigma) &= \frac{p(y)p(x \,|\, y, \theta)p_(y)}{p(x \,|\, \theta)} \\ &= \frac{p(y)N(x \,|\, y\mu, \Sigma)}{p(y)N(x \,|\, y\mu, \Sigma) + p(-y)N(x \,|\, -y\mu, \Sigma)} \\ &= \left[1 + \frac{p(-y)N(x \,|\, -y\mu, \Sigma)}{p(y)N(x \,|\, y\mu, \Sigma)}\right]^{-1} \end{aligned}$$

After substituting the normal densities and some further simplifications, we obtain

$$p(y \,|\, x, \mu, \Sigma) = \left[1 + \frac{p(-y)}{p(y)} \exp\left\{-2yx^T\Sigma^{-1}\mu\right\}\right]^{-1} \tag{2.10}$$

A two-dimensional example of a two-class Gaussian distribution is shown in figure 2.6 for an example two-dimensional X. Note that in the posterior, the parameters Σ, μ occur only as the vector $w = \Sigma^{-1}\mu$. Therefore, we may re-parameterize the posterior as

$$p(y \,|\, x, w) = \left[1 + \frac{1 - p(y)}{p(y)} \exp\{-2yx^T w\}\right]^{-1}. \tag{2.11}$$

It now suffices to estimate w to completely determine this distribution. Geometrically, w is the direction of change in the posterior; $p(y \,|\, x)$ is

constant in all direction orthogonal to w (figure 2.6). Intuitively, this direction should be along the line joining the two class-conditional means. This intuition is correct when Σ is the identity matrix, while in the general case this direction is transformed to $w = \Sigma^{-1}\mu$, to account for the covariance structure.

While the full distribution $f(x, y)$ has $n(n+3)/2$ free parameters (ignoring $p(y)$), the posterior (2.11) clearly has only n parameters. Therefore, estimating the posterior should be an "easier" inference problem than estimating the full $f(x, y)$. The posterior can of course be found by ML estimation of the parameters Σ, μ (here treating $p(y)$ as known, for simplicity) of the full distribution $f(x, y)$, resulting in a "plug-in" estimate

$$\hat{p}(y \mid x) = \frac{\hat{f}(x, y)}{\hat{f}(x)}.$$

This approach results in the popular *linear discriminant* method, first introduced by Fisher [50].

A different approach to estimating the posterior is to maximize the joint *conditional* likelihood $L(y^{(1:l)} \mid x^{(1:l)}, w) = \prod_i p(y^{(i)} \mid w, x^{(i)})$. This leads to a *logistic regression* model [73, pp. 95]. From (2.11) we have

$$\begin{aligned} L(y^{(1:l)} \mid x^{(1:l)}, w) &= \prod_i \left[1 + c^{(i)} \exp\{-2y^{(i)}(x^{(i)})^T w\}\right]^{-1} \\ &= \left[\prod_i \left(1 + c^{(i)} e^{-2y^{(i)}(x^{(i)})^T w}\right)\right]^{-1}, \end{aligned}$$

where $c^{(i)} = (1 - p(y^{(i)})/p(y^{(i)}$. This likelihood can be shown to be convex for $n \leq l$ [144], so that numerical optimization is feasible. However, a degenerate case occurs if the training data is separable, the likelihood will not be bounded away from zero, *i.e.*, the posterior is not directly identifiable. Much has been written about the relative merits of logistic regression *vs.* the fisher discriminnat; see for example Press and Wilson [137] and Efron [43].

One may simplify the posterior estimation by adding some conditional independence assumptions. The following is know as the "Naive Bayes" method, which is quite common in machine learning.

Example 2.6 (Naive Bayes) *Let the features X_i are conditionally independent given the (discrete) target variable Y, that is, $f(x \mid y) =$*

$\prod_i f(x_i \mid y)$. *A posterior with parameter* θ *is then given by*

$$\begin{aligned} p(y \mid x, \theta) &= \frac{f(x \mid y)p(y)}{f(x)} \\ &= \frac{p(y)\prod_i f(x_i \mid y)}{f(x)} \end{aligned}$$

Then the joint conditional likelihood is given by

$$\prod_j p(y^{(j)} \mid x^{(j)}, \theta) = \prod_j \frac{p(y^{(j)})\prod_i f(x_i^{(j)} \mid y^{(j)})}{f(x^{(j)})}$$

Using this form of $p(y \mid x)$, it suffices to estimate the one-dimensional $f(x_i \mid y)$, which is considerably easier. For example, if X, Y is a Gaussian mixture $X \mid Y \sim N(\mu_y, \Sigma_y)$, then the Naive Bayes assumptions imply that $p(y \mid x)$ is a product of univariate Gaussians, $p(y) = \prod_i N(x_i \mid \mu_{yi}, \sigma_{yi}^2)$. We are then left with only $2|\mathcal{Y}|n$ parameters $\{\mu_{yi}, \sigma_{yi}^2\}$, which are easily estimated. For comparison, the full Gaussian mixture without Naive Bayes assumptions has $|\mathcal{Y}|n(n+3)/2$ parameters.

This predictive model is called "naive" because the conditional independence assumption is rarely true in practise. Nevertheless, one should note that this assumption is considerably weaker than *marginal* independence of the X_i, *i.e.*, it does allow for correlations between X_i due to the target variable. For instance, in terms of Example 2.1 the Naive Bayes model does allow for both types of LDL cholesterol to be affected by the diet, but it assumes that when the diet is kept constant, the two are independent. Further, it has repeatedly been shown that the Naive Bayes predictor often gives good performance on real problems. It should therefore not be dismissed, and it often serves as a useful "benchmark" for gauging the performance of more sophisticated methods [67].

2.4 PREDICTORS AND INDUCERS

Having estimated a posterior, we are usually interested in using it to predict the target variable y for new examples x. The practical utility of this should be obvious: consider for example the case of Y being a clinical diagnosis such as "poor-prognosis, malignant breast cancer".

Definition 2.12. *A predictor is a function*

$$y = g(x) : \mathcal{X} \mapsto \mathcal{Y}$$

which assigns (predicts) a value $y \in \mathcal{Y}$ for every possible data point $x \in \mathcal{X}$.

For classification problems, I will often use the term *classifier* instead. Of course, we would prefer predictors that are able to predict as accurately as possible the true value of Y. An intuitive way of constructing a predictor is to predict the y with maximal posterior probability. This is called the Bayes predictor.

Definition 2.13 (Bayes predictor)**.** *For a given posterior $p(y \mid x)$, the Bayes predictor is given by*

$$g^*(x) = \begin{cases} y, & \text{if } \forall y' p(y \mid x) > p(y' \mid x) \\ g'(x), & \text{otherwise} \end{cases} \tag{2.12}$$

Here we have introduced the function $g'(x)$ merely to "break ties", that is, to ensure a unique decision in cases where several values of y are equally likely. The name "Bayes predictor" (or Bayes classifier, for discrete $\mathcal{Y}$; or sometimes "Bayes' rule"), stems from the connection with Bayes' theorem. It is easy to show that the Bayes predictor is optimal; see Theorem 2.18. The Bayes predictor is a *property of the data distribution*: it is determined by $p(y \mid x)$, which in turn is determined by $f(x, y)$.

Example 2.5 (Continued) *The Bayes predictor is given by solving $p(y \mid x, w) > 1/2$ for y. With $c = p(Y = -1)/p(Y = 1)$ we find*

$$\begin{aligned} p(Y = 1 \mid X = x) = \left[1 + c \exp\left\{-2x^T w\right\}\right]^{-1} &> 1/2 \\ \iff c \exp\left\{-2x^T w\right\} &< 1 \\ \iff 2x^T w - \ln c &> 0, \end{aligned}$$

yielding the predictor

$$g^*(x) = \operatorname{sign}\left(2x^T w - \ln c\right). \tag{2.13}$$

The critical region $\{x : x^T w = 0\}$ where $p(y \mid x, w) = 1/2$ and the Bayes classifier changes sign is called the decision boundary*. For $p(y) = 1/2$, $c = 0$ and this boundary is a hyperplane through the origin, and w is its normal vector. For $p(Y = 1) > p(Y = -1)$, the decision boundary is translated towards the center of class -1, reflecting that $g^*(x)$ is more inclined to predict $+1$ in this case, and vice versa.*

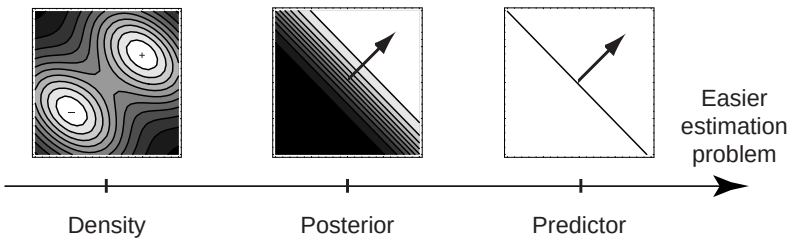

Figure 2.7: Comparison of the estimation problems discussed in this chapter. The density estimation problem is the hardest, the posterior estimation problem is intermediate, and the predictor estimation problem is easiest.

In linear classification, c is referred to as the *bias* term. For convenience, I will mostly consider linear predictors with $c = 0$. This is does not incur any loss of generalization since one may always let w subsume the bias term by introducing an additional, constant feature $X_{n+1} = 1$, so that $w^T x + c = (w^T, c)(x, x_{n+1}) = (w')^T x'$.

As seen in Equation (2.12), g^* is easily derived from $p(y \mid x)$, but it considers only for the maxima of $p(y \mid x)$ at each point x. Therefore, g^* retains only the part of the information contained in $p(y \mid x)$ that determines its maxima, and discards the information required for assessing the *confidence* (*e.g.*, variation) in this prediction. In the above example, this is evident from the fact that in (2.13) the magnitude of the vector w is irrelevant to the sign of $g^*(x)$, even though it does affect the posterior (2.11). Thus, g^* is again over-parameterized: setting $\|w\|_2 = 1$ reduces the number of parameters from n to $n - 1$ while retaining the same classifier.

In general, $p(y \mid x)$ is more difficult to learn from data than g^* is. Therefore, it may be better to estimate g^* directly from data than attempting to first estimate $p(y \mid x)$ and then computing g^* from (2.12). Note that this reasoning is in analogue with the previous section, where we found that estimating the posterior directly is easier than estimating the full distribution. To summarize the discussion so far, the estimation problems become easier as we move from full distribution to posterior to predictor (Figure 2.7). (By "easier" I here mean that the estimates obtained will in general be more accurate, not that the associated equations will be easier to solve.)

Thus, assuming that we care mainly about the prediction itself and not about its variance, we will be interested in procedures that directly estimate predictors from given data. Such estimation procedures are referred to as *inducers*.

Definition 2.14 (Inducer)**.** *For a given set of classifiers $\mathcal{G}$ and data $z^{(i:l)}$, an inducer I is a mapping*

$$g = I(z^{(i:l)}) : (\mathcal{X} \times \mathcal{Y})^l \mapsto \mathcal{G} \tag{2.14}$$

This type of inference is probably the most common in machine learning, and the theory on inducers (supervised learning) is comparatively well developed, especially for the classification case [37].

2.5 Loss and risk

Beyond learning predictors, it is of course important to have a sensible way of measuring their accuracy in making predictions. How to measure accuracy depends on the *loss function*, which is used to measure the gravity of a prediction error.

Definition 2.15 (Loss function)**.** *A loss function is a function $h(\hat{y} \,|\, y)$ on $\mathcal{Y} \times \mathcal{Y}$ satisfying*

$$\forall \hat{y} : h(\hat{y} \,|\, y) \geq h(y \,|\, y).$$

$h(\hat{y} \,|\, y)$ measures the "loss" (cost, penalty) of making the prediction $\hat{y}$ when the true target variable is y.

The loss function determines what types of predictors we prefer. The choice of loss function is subjective; it reflects the experimenter's opinion on how severe particular errors are. This subjectivity is usually not a problem however, since typically one has a fairly good understanding of the target variable and what a reasonable loss function should look like. For classification problems, the following loss function is a very common choice.

Example 2.7 (0-1 loss) *For classification problems the "0-1 loss" is given by*

$$h(\hat{y} \,|\, y) = \begin{cases} 1, & \hat{y} \neq y \\ 0, & \hat{y} = y \end{cases}$$

The 0-1 loss function considers both types of prediction errors as equally severe. In some applications however, these may differ. For example, if our classifier is a medical diagnosis system where $+1$ corresponds to "disease" whereas -1 means "no disease", then $h(-1 \mid +1)$ is the loss associated with failing to diagnose (and treat) a disease, while $h(+1 \mid -1)$ is the loss when one wrongly diagnoses a disease when there is none. The latter is often more acceptable: this is the case for example with melanoma (skin cancer), where physicians prefer to perform surgery in uncertain cases since the cost of not doing so could be fatal, while the procedure itself is fairly harmless [187]. For such "asymmetric" cases, the following loss may be more reasonable.

Example 2.8 *A generalization of 0-1 loss is given by*

$$h(\hat{y} \mid y) = \begin{cases} a, & \hat{y} \neq y, \quad y = +1 \\ b, & \hat{y} \neq y, \quad y = -1 \\ 0, & \hat{y} = y \end{cases}$$

If we here take $a > b$, errors of the type $\hat{y} = -1$ when $y = +1$ are penalized more than the opposite case.

For regression problems ($\mathcal{Y} = \mathbb{R}$), common loss functions are $h(\hat{y} \mid y) = (\hat{y} - y)^2$ or $h(\hat{y} \mid y) = |\hat{y} - y|$. Thus, while for classification problems the loss is usually bounded, for regression this may not be the case. For two-class problems, it is also common to deviate somewhat from Definition 2.15 by using an "intermediate" function with continuous range $g'(x) : \mathcal{X} \mapsto \mathbb{R}$, such that the predictor is given by $g(x) = \text{sign}(g'(x))$, and defining the loss function on g' as

$$h(g'(x) \mid y) : \mathbb{R} \times \mathcal{Y} \mapsto \mathbb{R}.$$

This is useful since loss functions on the discrete $\mathcal{Y} \times \mathcal{Y}$ are difficult to optimize (Section 2.7.2). A number of common loss functions are shown in Table 2.2.

Having chosen a loss function, we measure the accuracy of a predictor on a given data distribution $f(x, y)$ as follows.

Definition 2.16 (Risk)**.** *The risk $R(g)$ of a predictor g on a data distribution $f(x, y)$ is defined as the expected value of the loss function,*

$$R(g) = \mathbb{E}\left[h(g(X) \mid Y)\right]. \tag{2.15}$$

Note that risk is the "inverse" of accuracy: low risk means high accuracy and *vice versa*. For classification problems with discrete Y and

$\mathcal{Y}$	Name(s)	$h(\hat{y} \mid y)$
$\{0,1\}$	0-1	$1 - \delta_{\hat{y},y}$
$\{0,1\}$	Hinge*	$(1 - g(x)y)_+$
$\{0,1\}$	BNLL*	$\ln(1 - e^{-g(x)y})$
$\mathbb{R}^n$	L_2/Gaussian	$(\hat{y} - y)^2$
$\mathbb{R}^n$	L_1/Laplacian	$\lvert\hat{y} - y\rvert^2$
$\mathbb{R}^n$	ϵ-insensitive	$(\lvert\hat{y} - y\rvert - \epsilon)_+$
$\mathbb{R}^n$	Huber's	$\begin{cases} (\hat{y} - y)^2, & \lvert\hat{y} - y\rvert \le a \\ 2\lvert\hat{y} - y\rvert + a(a-2), & \lvert\hat{y} - y\rvert > a \end{cases}$

Table 2.2: Examples of common loss functions. BNLL, Binomial Negative Log-Likelihood. *defined on $\mathbb{R} \times \mathcal{Y}$

.

continuous X, the risk functional can be written as

$$R(g) = \sum_{y \in \mathcal{Y}} \int_{\mathcal{X}} h(g(x) \mid y) f(x, y) dx$$

If we use 0-1 loss, this simplifies to

$$\begin{aligned} R(g) &= \sum_{y \in \mathcal{Y}} p(y) \int_{\mathcal{X}} \delta(g(x), y) f(x \mid y) dx \\ &= \sum_{y \in \mathcal{Y}} p(y) P(g(X) \neq Y \mid Y = y) \\ &= P(g(X) \neq Y), \end{aligned}$$

so that risk is equivalent to the overall probability of making an error in this case. For the two-class Gaussian case of Example 2.5, this probability has a particularly simple form.

Example 2.9 *For a two-class Gaussian $f(x, y \mid \mu, \Sigma)$ and any classifier $g_w(x) = \text{sign}(w^T x)$, consider the projection of the class-conditional variable $(X \mid Y = 1)$ onto the normal vector $w = \Sigma^{-1}\mu$,*

$$t = \frac{w^T X_y}{w^T w}$$

Since X_y is a multivariate Gaussian, all linear combinations of X_y are themselves Gaussian. Thus $t \sim N(\mu_t, \sigma_t^2)$, with

$$\mu_t = \mathbb{E}[t] = \frac{w^T \mathbb{E}[X_y]}{w^T w} = \frac{w^T \mu_y}{w^T w}$$

and

$$\sigma_t^2 = \mathbb{E}\left[(t-\mu_t)^2\right] = \frac{\mathbb{E}\left[(w^T(X_y-\mu))^2\right]}{(w^Tw)^2} = \frac{w^T\Sigma w}{(w^Tw)^2}$$

Now $g_w(x) = 1 \iff t > 0$, *and by symmetry we can write* $R(g_w)$ *as*

$$\begin{aligned} R(g) &= P(w^Tx > 0 \,|\, Y = -1) \\ &= \int_{-\infty}^{0} N(t \,|\, \mu_t, \sigma_t)dt \\ &= \int_{-\infty}^{0} N\left(t \,\middle|\, \frac{w^T\mu}{w^Tw}, \frac{w^T\Sigma w}{(w^Tw)^2}\right)dt \\ &= \frac{1}{2}\left[1 - \operatorname{erf}\left(\frac{w^T\mu}{\sqrt{2w^T\Sigma w}}\right)\right] \end{aligned} \tag{2.16}$$

where $\operatorname{erf}(x) = 2\pi^{-1/2}\int_0^x e^{-t^2}dt$ *is the error function.*

Using the risk measure, we now define an optimal predictor.

Definition 2.17. *A predictor* g^* *is optimal if it has minimal risk,*

$$\forall g : R(g^*) \leq R(g).$$

It is fairly easy to show that the Bayes predictor (2.12) is optimal.

Theorem 2.18. *For any data distribution* $f(x, y)$*, the Bayes predictor* g^* *attains the lowest possible risk.*

Proof. We here give a proof for the case of discrete $\mathcal{Y}$ and 0-1 loss; this can be generalized to other $\mathcal{Y}$ and any loss function with minor modifications. Take any classifier g. We have for every $x \in X$ that

$$\begin{aligned} P(g(X) \neq Y \,|\, X = x) &= 1 - P(g(X) = Y \,|\, X = x) \\ &= 1 - \sum_y P(Y = y \ \wedge \ g(X) = y \,|\, X = x) \\ &= 1 - \sum_y 1_{\{g(x)=y\}} p(y \,|\, x). \end{aligned}$$

Thus, for any $y_0 \in \mathcal{Y}$,

$$\begin{aligned}
&P(g(X) \neq Y \,|\, X = x) - P(g^*(X) \neq Y \,|\, X = x) \\
&\quad = \sum_y 1_{\{g^*(x)=y\}} p(y \,|\, x) - \sum_y 1_{\{g(x)=y\}} p(y \,|\, x) \\
&\quad = p(y_0 \,|\, x)(1_{\{g^*(x)=y_0\}} - 1_{\{g(x)=y_0\}}) \\
&\qquad + (1 - p(y_0 \,|\, x))(1_{\{g^*(x)=y_0\}} - 1_{\{g(x)=y_0\}}) \\
&\quad = (2p(y_0 \,|\, x) - 1)(1_{\{g^*(x)=y_0\}} - 1_{\{g(x)=y_0\}}) \geq 0.
\end{aligned}$$

The last inequality follows directly from the definition of g^*. Integrating with respect to $p(x)dx$ now yields

$$P(g(X) \neq Y) - P(g^*(X) \neq Y) \geq 0.$$

□

From here on, I will use the terms "optimal predictor" and "Bayes predictor" interchangeably. To gauge the "difficulty" of a particular prediction problem (*i.e.*, of a particular data distribution) it is also useful to define the *Bayes risk*, the risk of the Bayes predictor $R(g^*)$. This is clearly the lowest risk achievable, and is determined solely by $f(x, y)$. Again, for example 2.5 we obtain a particularly simple expression.

Example 2.9 (Continued) *2.9 Inserting the Bayes classifier* $g^*(x) = \text{sign}(\mu^T \Sigma^{-1} x)$ *into (2.16), we obtain*

$$\begin{aligned}
R(g^*) &= \frac{1}{2}\left[1 - \text{erf}\left(\frac{w^T \mu}{\sqrt{2 w^T \Sigma w}}\right)\right] \\
&= \frac{1}{2}\left[1 - \text{erf}\left(\sqrt{\mu^T \Sigma^{-1} \mu / 2}\right)\right] \leq 1/2. \qquad (2.17)
\end{aligned}$$

This simple expression and that of (2.16) makes the two-class Gaussian case attractive for evaluating inducers on simulated data: one may then calculate the accuracy of each predictor directly from these expressions. This can be done efficiently even for high-dimensional X. I take advantage of this property in the simulation studies in Chapter 5.

The risk functional measures the accuracy of a particular classifier $g(x)$ on a given data distribution. This is relevant when the exact classifier to use for a certain problem is fixed. For example, we may be interested in evaluating the performance of an existing diagnosis scheme for some

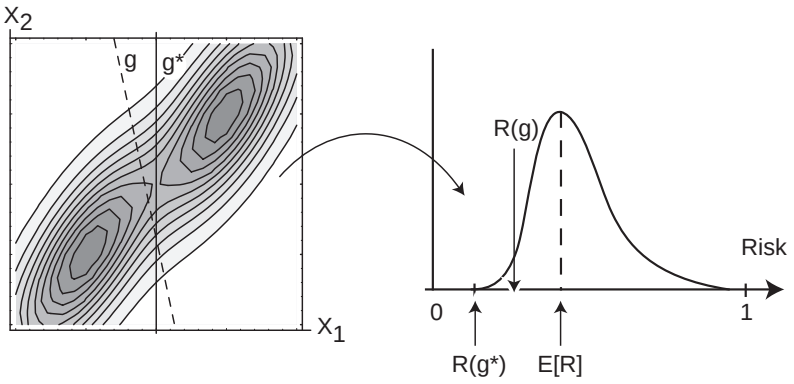

Figure 2.8: Illustration of a risk distribution (right) for a two-class classification problem (left). The risk distribution is bounded below by $R(g^*)$.

disease. Often, however, one is more interested in comparing the performance of *inducers* rather than particular classifiers [38]. This is of relevance for example to ensure that an inducer is reasonably accurate before applying it to experimental data.

In this case, the risk $R = R(I(Z^{(i:l)}))$ is itself a random variable since the data set $Z^{(i:l)}$ is a random variable, drawn from the data distribution. For a particular realized data set $z^{(i:l)}$, the risk of an induced classifier $g = I(z^{(1:l)})$ is merely one observation from this distribution. The performance of an inducer I is therefore described by the distribution of $R(I(Z^{(i:l)}))$. For classification and 0-1 loss, this is a distribution over $[0, 1]$. Figure 2.8 illustrates this situation. Usually, it is convenient to summarize the risk distribution into a single number. For this, I will use the expected risk.

Definition 2.19 (Expected risk)**.** *For a given data distribution, inducer I and sample size l, the expected risk is defined as*

$$\rho(I) = \mathbb{E}\left[R(I(Z^{(i:l)}))\right]. \tag{2.18}$$

In simulation studies, the expectation in (2.18) can be estimated simply by averaging the risks $R(g)$ of particular classifiers $g = I(z^{(1:l)})$ obtained over a number of independent data sets. Confidence intervals are likewise easy to obtain by standard methods.

2.6 Nonparametric methods

So far, we have assumed that we can model the experimental system using some convenient distribution family. This is not always an appropriate assumption in biology. In many applications of classical statistics, sample sizes are large, and one is often interested in inferences about the sample mean $\bar{x}$. In this advantageous setting, the central limit theorem ensures that $\bar{x}$ is Gaussian to a good approximation [21]. This is definitely not the typical situation for high-throughput measurements such as microarray profiling or proteomics. In many practical situations in modern biology, we have high dimensionality, which is often counterintuitive, and due to experiment costs sample size is small, so that we are not protected by the central limit theorem when making Gaussian assumptions.

For these reasons, many researchers believe that parametric approaches are dangerous, as we have no way of assessing that the model assumptions are reasonable. In this section I review some *nonparametric* or "distribution-free" methods for learning predictors, which attempt to avoid such assumptions. In the nonparametric approach, we still assert that there exists some distribution $f(x, y)$ that generates our data, although we hold that it is entirely unknown to us. We may use this unknown distribution and its properties in definitions and theoretical arguments, but we avoid making assumptions about the particular form of this distribution for inference purposes.

2.6.1 Empirical risk minimization

Instead of maximizing a likelihood function which always depends on distribution assumptions, we might consider to learn a predictor by directly minimize the risk $R(g)$ over some set $\mathcal{G}$ of possible predictors. Naturally, since $R(g)$ depends on the unknown data distribution, it cannot be computed in practise. However, we might attempt to estimate the risk of a given predictor directly from the observed data $\{z^{(1:l)} = (x^{(1)}, y^{(1)}), \ldots, (x^{(l)}, y^{(l)})\}$, and then minimize the estimate. An intuitive risk estimate is the following.

Definition 2.20 (Empirical Risk)**.** *The empirical risk $\hat{R}$ of a predictor g is defined as*

$$\hat{R}(g, z^{(1:l)}) = \frac{1}{l} \sum_{i=1}^{l} h(g(x^{(i)}) \,|\, y^{(i)}),$$

where h is some given loss function.

For classification problems with 0-1 loss, the empirical risk is simply the fraction of errors on the training data. Thus, a direct strategy for learning a predictor g from data is to simply minimize $\hat{R}(g, z^{(1:l)})$ over all $g \in \mathcal{G}$. This procedure is known as *Empirical Risk Minimization*.

Definition 2.21 (Empirical Risk Minimization (ERM))**.** *In empirical risk minimization, the inducer estimates a classifer from a set of possible classifiers $\mathcal{G}$ as*

$$I(z^{(1:l)}) = \arg\min_{g \in G} \hat{R}(g, z^{(1:l)}).$$

Empirical risk minimization has a long history in machine learning. One of the first successful machine learning algorithms, the "perceptron" devised by Rosenblatt [148], performs a kind of ERM by adjusting itself to accommodate the training examples (*i.e.*, minimizing the training error).ERM was studied in depth by Vladimir Vapnik in a series of influential papers in the 1970's. Importantly, Vapnik and co-workers derived conditions that determine when empirical risk minimization will work (yield a good predictor) and when it will not [181]. While this theory is far too extensive and complicated to be covered here, some informal remarks may nevertheless be in order. The crucial point in ERM is the set $\mathcal{G}$ of possible classifiers. If this set is too "large" in comparison with the amount of data available for estimating g, meaning that it contains to "complex" g, then the estimated classifier $\hat{g}$ will "over-fit", and generalization ability will be poor. Several methods that implement variations on ERM are discussed in Section 2.7.2.

2.6.2 NEAREST-NEIGHBOR METHODS

Early classification methods were based on simply matching a new x to the closest example in the training data, so-called "template matching" [87]. The *k-nearest-neighbor* (k-NN) classifier, introduced by Fix and Hodges, Jr. [51] is by far the most well-known and successful example. This classifier uses a distance metric d on $\mathcal{X}$ (which must be defined in advance) and classifies each x by "voting" among the k observations x_i nearest to x in the metric d. For $Y \in \{+1, -1\}$,

$$g_k(x) = \text{sign}\left(\sum_{i \in N_k(x)} y_i\right),$$

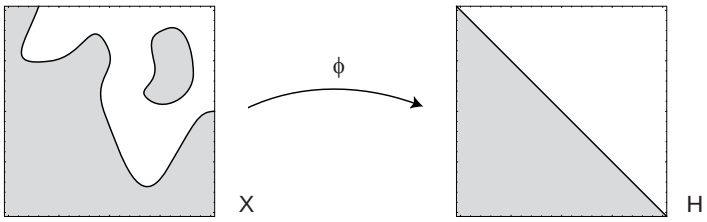

Figure 2.9: Illustration of the feature space concept. A complex decision surface (solid line) between two classes (white *vs.* gray areas) in the input space $\mathcal{X}$ (left) can be mapped to a higher-dimensional feature space $\mathcal{H}$ by a feature map $\phi(x)$.

where

$$N_k(x) = \{i : |\{x_j : d(x, x_j) > d(x, x_i)\}| \geq k - 1\}$$

is simply the set of the k nearest observations in the training data. This is simple to generalize to multiple classes. For discrete $\mathcal{X}$, one also needs some strategy for breaking ties (cases where two x_i, x_j are at an equal distance from x).

Despite its simplicity, the k-nearest neighbor rule is quite powerful, and has many attractive properties. Remarkably, it can be shown to be consistent for *any* underlying data distribution $f(x, y)$, provided that one chooses k dependent on the sample size l such that $k \to \infty$ and $k/l \to 0$ as $l \to \infty$ [37, pp. 170]. Thus, this method is "distribution-free". The k-nearest neighbor method is often used as a "baseline" against which to compare other methods. A drawback is that the computation of $N(x)$ can be computationally expensive, and that performance tends to degrade with increasing dimensionality [103]. A wealth of theoretical results on the k-NN classifier can be found in [37].

2.6.3 Kernel methods

Linear predictors such as $g_\theta(x) = \theta^T x$ can be generalized with the use of *kernel functions*. While I do not make explicit use of kernel functions in this thesis, many of the results I present can be lifted to higher generality by this mechanism, so that a brief explanation of the concept may nevertheless be in order. A comprehensive introduction is given by Schölkopf

and Smola [153]. A kernel function $K(x, x') : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ has the special property that it corresponds to a second function $\phi(x) : \mathcal{X} \mapsto \mathcal{H}$ called a *feature map*, which maps each $x \in \mathcal{X}$ to some Hilbert space $\mathcal{H}$, called a *feature space*, and satisfies

$$K(x, x') = \langle \phi(x), \phi(x') \rangle.$$

In other words, $K(x, x')$ *implicitly* computes an inner product in the space $\mathcal{H}$ without explicitly computing the map ϕ. This is often referred to as the "kernel trick". The main idea here is that we can implement various non-linear predictors through kernel functions by computing scalar products (linear functions) in some space $\mathcal{H}$ that corresponds to a non-linear functions in $\mathcal{X}$ (Figure 2.9). For example, predictors which are linear in the coefficients of a polynomial of x are easily represented by kernels.

Example 2.10 *A polynomial kernel of degree d on $\mathcal{X} = \mathbb{R}^n$ is defined by*

$$K(x, x') = (1 + x^T x')^d.$$

To find the feature map corresponding to this kernel, note that, for the case $d = 2$,

$$\begin{aligned} K(x, x') &= (1 + \sum_i x_i x'_i)^2 \\ &= 1 + 2 \sum_i x_i x'_i + (\sum_i x_i x'_i)^2 \\ &= 1 + 2 \sum_i x_i x'_i + \sum_i \sum_j (x_i x'_j)^2 \\ &= (1, x_1, \ldots x_n, x_1^2, \ldots, x_n^2)^T (1, x'_1, \ldots x'_n, (x'_1)^2, \ldots, (x'_n)^2). \end{aligned}$$

Thus, we can write $K(x, x') = \langle \phi(x), \phi(x') \rangle$ with

$$\phi(x) = (1, x_1, \ldots x_n, x_1^2, \ldots, x_n^2),$$

So in this case $\mathcal{H} = \mathbb{R}^{2n+1}$. Similarly, one can show that for arbitrary d, this kernel computes polynomials of (x, x') of degree d, corresponding to a feature space of dimension $\dim \mathcal{H} = dn + 1$.

It is also possible to compute kernel functions that correspond to infinite-dimensional $\mathcal{H}$. One way to think about this is to consider $\phi(x) \in \mathcal{H}$ as points in a function, space, *i.e.*, $\phi : x \mapsto K(\cdot, x)$. A prominent example of this is the Gaussian kernel, suggested by Boser et al. [18].

Example 2.11 *The Gaussian kernel with kernel parameter γ is given by*

$$K_\gamma(x, x') = e^{-\gamma\|x-x'\|_2^2}.$$

The corresponding feature map $\phi : x \mapsto e^{-\gamma\|x-(\cdot)\|_2^2}$ is a Gaussian function centered at x. Some properties of $\phi(x)$ are immediate: for example,

$$\|\phi(x)\|_{\mathcal{H}}^2 = K(x, x) = 1,$$

so that all $\phi(x)$ are on a unit sphere in $\mathcal{H}$. Similarly, $\phi(x)^T\phi(x') \geq 0$ for all x, x', which means that no two points in $\mathcal{H}$ make obtuse angles.

The Gaussian kernel is "universal" in that *any* function $K(x, x')$ can be approximated with arbitrary precision by a Gaussian kernel $K_\theta(x, x')$ provided that γ is small enough. Importantly, this property can be exploited to construct universally consistent inducers based on Gaussian kernels by choosing a sample size-dependent sequence of $\gamma_l \to 0$ as $l \to \infty$ [160]. Therefore, the Gaussian kernel allows distribution-free methods.

When the feature space $\mathcal{H}$ is very high-dimensional, it is impractical to represent a solution vector $w \in \mathcal{H}$ explicitly. However, it turns out that in many cases, the predictors themselves can also be represented implicitly as a linear combination of kernel functions,

$$g(x) = \sum_{i=1}^{l} \alpha_i K(x_i, x).$$

This result is known as the representer theorem [153, pp. 89]. Due to this theorem, it is possible to perform inference with kernels without ever computing the high-dimensional feature space vectors.

In general, any inference method for which computations can be expressed as scalar products of the training data can be "kernelized" by replacing $x^T x'$ by $K(x, x')$ where appropriate. This applies to a rich family of inference methods. Consequently, researchers in kernel methods have been quite busy during the last decade with kernelizing many well-known methods for classification, regression, clustering, dimensionality reduction, and more. A good source of literature for these developments is the Neural Information Processing Systems (NIPS) conference proceedings. Some examples will be discussed in Section 2.7.2.

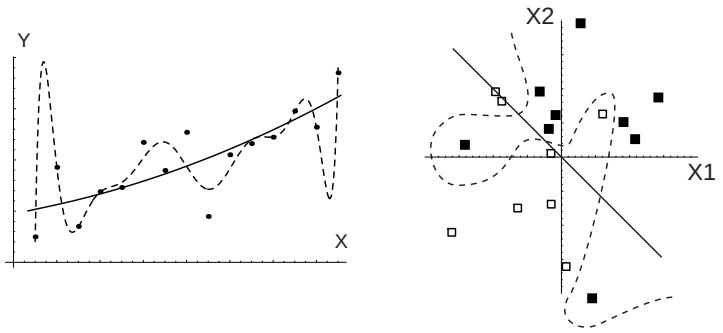

Figure 2.10: Two examples of the over-fitting phenomenon. Left, a regression model. Solid line shows the true model; dotted line, an over-fitting polynomial. Right, two-class model with $\mathcal{Y} = \{+1, -1\}, \mathcal{X} = \mathbb{R}^2$. Filled boxes denote points x for which $y = 1$, open boxes $y = -1$. Solid line shows the true decision boundary, dotted line, an over-fitting model.

2.7 PRIORS, REGULARIZATION AND OVER-FITTING

2.7.1 OVER-FITTING

The phenomenon of *over-fitting* has been a core issue of machine learning research since its inception. In essence, over-fitting means that the inference method "mistakes" random variation in the training for meaningful information about the target variable. This results in an overly complex ("over-fit") predictive model, which will have poor predictive accuracy (high risk) when applied to new observations. The over-fitting phenomenon appears in a wide variety of settings; examples of classification and regression problems are given in Figure 2.10. In classification and regression alike, the problem appears when one tries to learn too "complex" models from limited data in the presence of noise. The opposite problem is "under-fitting", which occurs when the model is not sufficiently complex to fit the data well.

The over-fitting phenomenon is perhaps best appreciated by an example. To this end, I here reproduce a classic example due to Trunk [174].

Example 2.12 (Trunk) *Consider a two-class problem with data* $X \in$

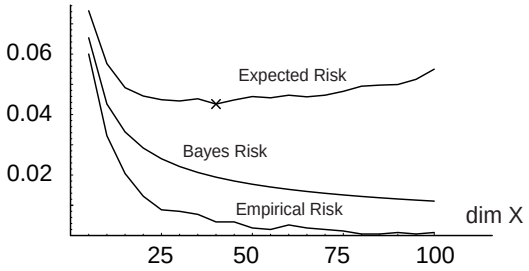

Figure 2.11: Comparison of the Bayes risk, expected risk and empirical risk estimate for Trunk's example, plotted *vs.* data dimension $n = \dim X$. Topmost line, estimate of the expected risk $\rho = \mathbb{E}\left[R(\hat{w})\right]$ at sample size $l = 20$, averaged over 100 training sets. Cross marks estimated minima of ρ. Middle line, Bayes risk $R(g^*)$ (strictly decreasing). Bottom line, empirical risk $\hat{R}$, also at sample size $l = 20$ and averaged over 100 training sets.

$\mathbb{R}^n$, targets $Y \in \{-1, +1\}$, with each feature distributed as a Gaussian

$$f(x_i \,|\, y) = N(x_i \,|\, y/\sqrt{i}, 1).$$

All features X_i are independent (identity covariance matrices for both classes), and we set the class probabilities $p(y) = 1/2$, so that

$$f(x, y) = p(y) \prod_i N(x_i \,|\, y/\sqrt{i}, 1).$$

The Bayes classifier g^* for this problem can then be expressed as

$$g^*(x) = \operatorname{sign}(w^T x)$$

with $w = (1, 1/\sqrt{2}, \ldots, 1/\sqrt{n})$, that is, the hyperplane normal is the same as the $+1$ class mean. Using equation (2.17), the Bayes risk is found to be

$$R(g^*) = \frac{1}{2}\left[1 - \operatorname{erf}\left(\sqrt{w^T w/2}\right)\right] = \frac{1}{2}\left[1 - \operatorname{erf}\left(\sqrt{\sum_{i=1}^{n} \frac{1}{2i}}\right)\right]$$

The Bayes risk is strictly decreasing in n (Figure 2.11). Hence, more features improves the optimal prediction performance, as every new feature contributes some extra information about the target variable, even

though this added information becomes smaller for large n. As n tends to infinity, $R(g^*) \to 0$.

Now, consider that w is unknown, so that we have to estimate this parameter from a data set $z^{(1:l)}$ (we assume that $\Sigma = I$ is known, though). The complete likelihood is then

$$\begin{aligned} f(z^{(1:l)} \,|\, w) &= \prod_i N(x^{(i)} \,|\, y^{(i)}, w, I) \\ &\propto \prod_i \prod_j \exp\left\{-(x_j^{(i)} - y^{(i)} w_j)^2/2\right\} \end{aligned}$$

So the log-likelihood is

$$\ln f(x, y \,|\, w) = -\frac{1}{2} \sum_i \sum_j (x_j^{(i)} - y^{(i)} w_j)^2 + C$$

Maximizing this, we obtain the ML estimates

$$\hat{w}_j = \frac{1}{l} \sum_{i=1}^{l} y^{(i)} x_j^{(i)} \tag{2.19}$$

Note that the ML estimate is simply the mean of the data points "weighted" by their label, which seems reasonable considering the symmetry of the distribution.

Since $\hat{w}$ is now estimated from data, the corresponding classifier $\hat{g}$ and the risk measure risk $R(\hat{g})$ are now random variables, both dependent on the training data (see Figure 2.8). To estimate the overall performance of this inducer, we use the expected risk (Definition 2.19). Figure 2.11 compares the expected risk $\mathbb{E}[R(\hat{g})]$ of the ML estimate to the expected empirical risk $\mathbb{E}\left[\hat{R}(\hat{g})\right]$ of the same. As n increases, the empirical estimate becomes increasingly optimistic. The expected risk first decreases until $n \gtrsim 40$; after this point, over-fitting ensues, and the expected risk begins to increase. In fact, is it easy to show that $\mathbb{E}[R(\hat{g})] \to 1/2$ as $n \to \infty$, while $\mathbb{E}\left[\hat{R}(\hat{g})\right] \to 0$ [174].

The problem in the above example is that, as the number of features n increases, the predictive model becomes too complex for reliable inference to be feasible from limited training data. Even though the information about Y actually increases with increasing n (as evidenced by the strictly decreasing Bayes risk), the increase in model "complexity" soon becomes overwhelming, and the actual performance beings to deteriorate. Thus, for high-dimensional data it is necessary to somehow constrain the complexity of the predictive model. This is known as *regularization*.

2.7.2 Regularization

Regularization, the idea of avoiding "too complex" models, has appeared in literature in many guises, and has many names. To name but a few, in phylogenetic analysis, it is known as *maximum parsimony* [100]; in statistics, it appears in the form of Bayesian *prior distributions* (Section 2.7.3); and in coding theory, it appears in the *minimal description length* principle [146]. A classical reference to regularization is the principle known as "Occam's razor", attributed to the 14th century English logician William of Ockham:

> "*entia non sunt multiplicanda praeter necessitatem*"
> (entities should not be multiplied beyond necessity)

In essence, this means "other things being equal, the simplest solution is usually the best one". One way of simplifying a predictive model is to constrain the number of features; this is one motivation for feature selection.

In various inference problems, including that of learning predictors, regularization is implemented by constraining the "complexity" of the solution. This method was introduced by Tikhonov and Arsenin [171] and has been used in a number of problem settings. In empirical risk minimization, the method amounts to adding a *regularization* term to the empirical risk estimate,

$$\tilde{R}(g) = \hat{R}(g) + \lambda\gamma(g). \tag{2.20}$$

The term $\gamma(g)$ in essence measures the complexity of a predictor g. The regularization parameter λ controls how much "weight" is given to $\gamma(g)$ compared to the risk estimate. Of course, various choices of γ are possible, and the question now becomes how to make this choice in a rational way. In the next section I will discuss one possible solution to this problem, motivated by a parametric data model. Often, the choice of γ is dictated by practical requirements. In particular, one would like to choose γ so that the optimization problem $\min_g \tilde{R}(g)$ is convex. The following case is probably the best known.

Example 2.13 (Support vector machine) *For $Y = \{+1, -1\}$, let $g'(x) = w^T x$, $h(g'(x), y) = (1 - g'(x)y)_+$ and $\gamma(w) = w^T w$. For a sample $z^{(1:l)}$, The linear Support Vector Machine (SVM) induces a classifier $g_w(x) = \text{sign}(g'(x))$ by minimizing*

$$\tilde{R} = \frac{1}{l}\sum_i h(g'(x^{(i)})y^{(i)}) + \lambda\gamma(w). \tag{2.21}$$

$\mathcal{Y}$	$\{0,1\}$			$\mathbb{R}$	
$\gamma \setminus h$	$\lvert\xi\rvert$	ξ^2	$(1-\xi)_+$	$(\lvert\xi\rvert-\epsilon)_+$	ξ^2
$\lVert w\rVert_2^2$		LS-SVM,KFD	SVM	SVR	RR, RVM
$\lVert w\rVert_1$		S-KFD	LP-SVM		LR
$\lVert w\rVert_0$			AROM		

Table 2.3: Some popular combinations of loss functions and regularizers. Here, $\xi = w^T xy$ and $g(x) = \text{sign}(w^T x)$. SVM, Support Vector Machine [18]; LS-SVM, Least Squares-SVM [166]; SVR, Support Vector Regression [158]; RR, Ridge Regression [76]; RVM, Relevance Vector Machine [172]; LR, Lasso Regression [170]; S-KFD, Sparse Kernel Fisher Discriminant [118]; AROM, Approximation of the zero norm [186].

The SVM has become exceedingly popular during the last decade. The name "support vector" derives from the fact that the minima of (2.21) turns out to depend only on a subset of the observations $x^{(i)}$, namely those that are closest to the decision boundary. These were named "support vectors" by Boser et al. [18], from the mechanistic analogy that they "support" the decision boundary.

The formulation (2.20) appears in a multitude of linear methods, each with a particular choice of loss function h and regularization term γ. Some examples are shown in Table 2.3. All of these methods can be "kernelized" by setting

$$g'(x) = \sum_i \alpha_i y^{(i)} K(x^{(i)}, x).$$

There are many other ways of implementing regularization schemes. For example, one may simplify inference problems by "tying together" related parameters. This idea was explored in the context of Bayesian network inference by Segal et al. [154]. Also, the probability density factorization represented by Bayesian networks is itself a means of reducing the set of solutions. The parametric assumptions explored at the beginning of this chapter may also be viewed as a regularizing technique; loosely speaking, parametric distribution classes are smaller than their non-parametric counterparts.

2.7.3 Priors and Bayesian statistics

One way to understand the problem arising in Trunk's example is by studying the properties of $\hat{w}^T X$, since the risk is given by $R(g_{\hat{w}}) = P(\hat{w}^T X > 0 \,|\, Y = -1)$. While the density of $\hat{w}^T X$ is difficult to calculate, a fairly straightforward calculation [174] shows that, for a training data set of l samples,

$$\mathbb{E}\left[\hat{w}^T X\right] = \sum_{i=1}^{l} \frac{1}{i}$$

and

$$\mathrm{Var}[\hat{w}^T X] = \left(1 + \frac{1}{l}\right) \sum_{i=1}^{n} \frac{1}{i} + \frac{n}{m}$$

Thus, while the ML estimates are unbiased, their variance increases with n. Intuitively, with larger variance the risk of errors $\hat{w}^T X > 0 \,|\, Y = -1$ may increase. Therefore, if one could reduce this variance, one might alleviate the over-fitting problem. One method for doing this is to use a *prior* distribution over w, denoted $\pi(w)$. The prior distribution is thought to describe our beliefs about w *before* (prior to) we make the experiment and obtain the data. We then augment the ML estimate to take into account both the data and the prior. Specifically, we define a *posterior* distribution over the *parameters* using Bayes' theorem,

$$\pi(w \,|\, x) = \frac{f(x \,|\, w)\pi(w)}{p(x)} = \frac{f(x \,|\, w)\pi(w)}{\int f(x \,|\, w)\pi(w)dw}. \tag{2.22}$$

This distribution is then used to estimate w. Some remarks are in order:

- Note that the posterior distribution over the parameter $\pi(w \,|\, x)$ has nothing to do with the posterior distribution over the target $p(y \,|\, x)$ defined in (2.9). In general, the term "posterior" is used for any distribution derived through Bayes theorem. Hopefully, the meaning will be clear from the context throughout this thesis.

- The prior and posterior distributions use probability to describe a "belief" (uncertainty) about a variable rather than randomness (the parameters w_i are really constants, not random variables). The use of the symbol π rather than p or f is meant to emphasize this difference. This is a characteristic feature of Bayesian statistics, the appropriateness of which has been a long-standing debate in statistics; in "classical" or "frequentist" statistics, this interpretation of probability is not "allowed". For the interested reader, a good exposition of this topic is given by Berger [13].

For inference from a sample $x^{(1:l)}$, one may maximize the posterior under the usual independence assumptions,

$$\pi(w \,|\, x^{(1:l)}) = \frac{f(x^{(1:l)} \,|\, w)\pi(w)}{f(x^{(1:l)})} = \frac{\prod_i f(x^{(i)} \,|\, w)\pi(w)}{f(x^{(1:l)})}$$

The resulting estimate is called the *maximum a posteriori* (MAP) estimate. Other possible estimates include the mean or median of $p(w \,|\, x^{(1:l)})$. The factor $f(x^{(1:l)}) = \int f(x^{(1:l)} \,|\, w)\pi(w)$ is often difficult to calculate, but can be ignored in MAP estimation since it does not depend on w. For Trunk's example, consider a prior belief that most components of w_i are positive (as is actually the case). A reasonable prior might then be a spherical Gaussian distribution over w centered at unity,

$$\pi(w) = N(w \,|\, 1, \nu^2 I),$$

Having chosen a prior, we can calculate the posterior for w according to (2.22),

$$\begin{aligned}\pi(w \,|\, x, y) &\propto f(x, y \,|\, w)\pi(w) \\ &\propto e^{-(1-w_j)^2/(2\nu^2)} \prod_j e^{-(x_j - y w_j)^2/2}.\end{aligned}$$

For data $z^{(1:l)}$, the joint log-posterior is

$$\ln \pi(w \,|\, z^{(1:l)}) = -\frac{1}{2}\sum_i \sum_j (x_j^{(i)} - y^{(i)} w_j)^2 + \frac{1}{2\nu^2}(1 - w_j)^2 + C.$$

Maximizing this, we obtain the MAP estimate as

$$\tilde{w_j} = \frac{\nu^2 \sum_i y^{(i)} x_j^{(i)} + 1}{l\nu^2 + 1}. \tag{2.23}$$

Note that as $\nu \to \infty$, this approaches the ordinary ML estimate (2.19); this represents a "completely uncertain" prior, so that the data dictates the estimate. Conversely, for small ν the estimate will be close to the prior mean $w_j = 1$ regardless of the data. Moreover, if one increases the sample size l while ν is held constant, the MAP estimate again approaches the ML estimate. A comparison of the original and regularized estimators is given in Figure 2.12. As expected, the prior-regularized estimate gives lower risk.

It is interesting to note that many of the regularized, non-parametric methods discussed in the previous section (Table 2.3) can be intepreted as Bayesian models. For example, consider the Lasso regression scheme.

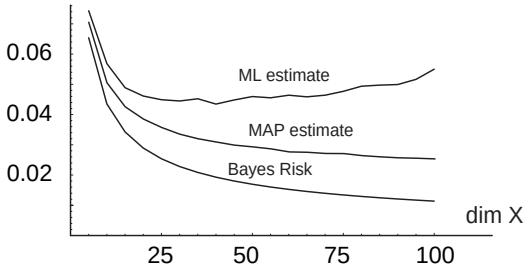

Figure 2.12: Effect of a prior distribution in Trunk's example. Risks plotted *vs.* data dimension $n = \dim X$ as in Figure 2.11. Topmost line, risk of the ML estimate $\mathbb{E}\left[R(\hat{w})\right]$ at sample size $l = 20$. Middle line, risk of prior-regularized estimate $\mathbb{E}\left[R(\tilde{w})\right]$ given by Equation (2.23). Bottom line, Bayes risk $R(g^*)$.

Example 2.14 (Lasso regression) *For $Y = \mathbb{R}$, let $g(x) = w^T x$, $h = (\hat{y} - y)^2$ and $\gamma(g) = \|w\|_1 = \sum_j |w_j|$. The Lasso estimate is then given by*

$$\min_w \sum_i h(g(x^{(i)}), y^{(i)}) + \lambda\gamma(g).$$

Lasso regression (or "the lasso", for short) was first introduced by Tibshirani [170] as an alternative to the commonly used "ridge regression", which uses the regularizer $\gamma(g) = \|w\|_2$ instead [76]. A feature of the Lasso is that the estimates of w is sparse due to the use of the L_1 regularizer. While the Lasso might be considered non-parametric since no distribution arguments are used to derive the loss and regularization terms, it is easy to see that the method can also be obtained in a parametric, Bayesian setting.

Example 2.14 (Continued) *Consider a regression situation with $Y \mid X \sim N(w.X, \sigma), X \in \mathbb{R}^n$. For a sample $z^{(1:l)}$, assuming conditional independence of the $y^{(i)}$ as usual, the conditional likelihood is*

$$f(y^{(1:l)} \mid x^{(1:l)}, w) = \prod_i N(y^{(i)} \mid w^T x^{(i)}, \sigma^2)$$

Choosing a prior over w given by

$$\pi(w) \propto e^{-\lambda' \|w\|_1},$$

we obtain the posterior

$$\pi(w \mid z^{(i)}) \propto \pi(w) \prod_i f(y^{(i)} \mid x^{(i)}, w)$$
$$\propto e^{-\lambda' \|w\|_1} \prod_i e^{-(y^{(i)} - w^T x^{(i)})^2/(2\sigma^2)}.$$

The negative log-posterior becomes

$$\frac{1}{2\sigma^2} \sum_i (y^{(i)} - w^T x^{(i)})^2 / + \lambda' \|w\|_1.$$

Taking $\lambda = 2\sigma^2\lambda'$ and minimizing the above gives the Lasso regression scheme.

Thus, the loss function in Lasso regularization corresponds to a Gaussian conditional distribution $Y \mid X$, while the L_1 regularizer corresponds to a a Laplace prior $\pi(w) = e^{-\lambda \|w\|_1}$, expressing a prior belief that many w_i are zero. Note that the above reasoning applies to any regularized empirical risk of the form (2.20), and that the corresponding conditional and prior distributions are always from the exponential family. Whether these assumptions are reasonable or not is a separate question, but this type of interpretation of regularization methods is useful in that it sheds light on the underlying model.

2.8 SUMMARY

In this chapter I have reviewed a number of concepts related to the developments in later chapters. I introduce a statistical model where experimental data is considered as observation from an underlying but unknown data distribution, which contains all information about the experiment. The data distribution can be described at different levels of detail by parametric models (Section 2.1) or graphical models (Section 2.2). Various learning problems, are then posed as statistical inference problems. In particular, we discuss the problem of learning a *predictor* the target variable (Sections 2.3 and 2.4). I then discuss how to estimate the accuracy of predictors (Section 2.5) and discuss some popular learning methods based on minimizing such accuracy estimates (Section 2.6). I consider the central problem of *over-fitting* and various methods of *regularization* that address this problem. Feature selection is one possible solution to this over-fitting problem.

An important idea in this chapter is that it is important to identify the assumptions that underlie learning methods, because this makes it easier to *interpret* the models. Also, making assumptions explicit helps the practitioner to choose among methods, according to what assumptions can be judged reasonable in each particular application. An example of this is the Bayesian interpretation of regularized linear methods discussed in Section 2.7.3. This idea is central also to the remainder of the thesis.

There are of course many interesting methods and aspects of statistical inference and machine learning that cannot possibly be covered in this brief introduction. Important examples are decision trees [120], boosting methods [53, 54] and neural networks [75]. Further, I do not consider the density estimation problems such as data clustering [84]. However, recent developments in the machine learning field increasingly finds that these methods, while originating in separate ideas and often motivated by heuristics, are in fact related in intricate ways [39, 143]. Thus, a more coherent theory of these predictive models is now emerging, which again is useful for the practitioner since it allows a more principled choice of methodology.

3

Feature Selection Problems

In this chapter I consider in some detail various definitions of feature selection problems, assuming a statistical data model as discussed in the previous chapter. The main issue here is to choose a suitable definition of *relevance*: when is a feature relevant to the target variable? The feature selection problem is then to discover the relevant features from data as accurately as possible.

The choice of a relevance definition depends on the end goal of the analysis. There may be several reasons for performing feature selection:

- Reducing dimension may improve the accuracy of an inferred predictive model.
- Each feature may be expensive to measure.
- Inference and/or prediction may be computationally costly in high dimensions.
- Reducing the number of features may give insights into the working of the system itself.

As I will argue in this chapter, these criteria are not wholly compatible with each other; it may not be possible to satisfy all of them simultaneously. Therefore there does not exist a single feature selection problem,

but rather several different feature selection problems, depending on the precise end goal. Hence the title of this chapter.

I will devote this section to establishing rigorous mathematical definitions of feature selection problems. The material in this section is purely theoretical — I will discuss no real data and propose no algorithms. The next chapter provides a survey of a number of feature selection algorithms that attempt to solve the problems defined here.

3.1 PREDICTIVE FEATURES

As we learned in the previous chapter, in machine learning, feature selection is usually intended to facilitate learning of an accurate predictor. This is motivated by the fact that features which do not carry any information about the target variable may hinder inference of accurate predictors due to over-fitting, as described in Section 2.7.1. A natural point of view is then that only the *predictive* features should be considered to be "relevant" for learning predictors, while the remaining features should be ignored (the corresponding observations discarded). However, upon careful examination it turns out that "predictive" can be interpreted in different ways, depending on whether we are estimating the posterior $p(y \,|\, x)$ or merely a predictor $g(x)$.

3.1.1 THE MARKOV BOUNDARY

In Section 2.3 we discussed predictive models based on the posterior $p(y \,|\, x)$, which is useful when one is interested not merely in a prediction of the most likely y, but also in some estimate of the confidence in this prediction. For example, one may require not merely a prediction of the type "patient X has breast cancer", but rather a probabilistic statement such as "there is a 93% probability that patient X has breast cancer". In this case, the "predictive" features is the set of features that influences the the posterior $p(y \,|\, x)$. This feature set is known as the *Markov boundary* of Y.

Definition 3.1 (Markov boundary)**.** *The Markov boundary X_M of a variable Y is the smallest set $S \subseteq V_n$ that satisfies*

$$Y \perp X_{\neg S} \,|\, X_S \tag{3.1}$$

Here $\perp$ denotes conditional independence (Section 2.2). By the definition

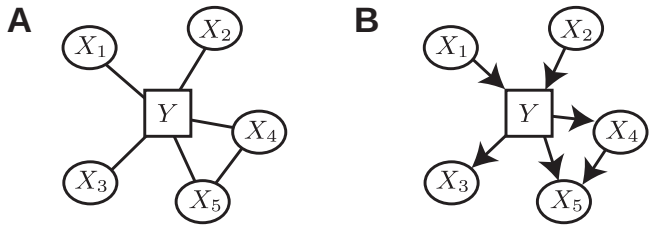

Figure 3.1: Illustration of the Markov boundary (circles) of a target variable Y (square) using graphical models. A: Markov (undirected) network representation. B: Bayesian (directed) network representation.

of conditional independence, equation (3.1) is equivalent to the identity

$$P\left(p(Y \mid X) = p(Y \mid X_M)\right) = 1. \tag{3.2}$$

From this is should be clear that the Markov boundary indeed consists exactly of the features which affect the posterior. Note that here $p(Y \mid X)$ and $p(Y \mid X_M)$ are themselves random variables, representing the true probability (density) functions evaluated at X, Y. The outer $P\left(\cdots\right) = 1$ is a technicality required for continuous X to ensures that events $\xi \in \mathcal{X}$ with zero probability do not affect the statement of conditional independence. For discrete $\mathcal{X}$, the above can also be written as

$$\forall y, x : p(y \mid x) = p(y \mid x_M).$$

An equivalent definition for discrete X and Y given by Koller and Sahami [101] is

$$\sum_{x \in \mathcal{X}, y \in \mathcal{Y}} p(y \mid x) \ln \frac{p(y \mid x)}{p(y \mid x_M)} = 0.$$

For continuous X or Y, the respective sum is replaced by an integral. The left hand side here is the well-known Kullback-Leibler divergence between $p(Y \mid X)$ and $p(Y \mid X_M)$, which is always nonnegative and equals zero if and only if the distributions coincide [104].

The Markov boundary concept can be visualized using graphical models (Section 2.2). For Markov networks (undirected graphs), the Markov boundary of a node Y can be shown to consist exactly of the neighbors of Y in the graph (Figure 3.1A). The intuitive interpretation is that these neighbors "captures" all "information flow" to Y, and hence contains all

information needed to predict the state of Y. For Bayesian networks (directed graphs), the Markov boundary consists of the parents of Y, the children of Y, and the parents of the children of Y (Figure 3.1B).

A related concept is a *Markov blanket*, defined as *any* set S satisfying $Y \perp X_{\neg S} \,|\, X_S$ [130]. Hence, the Markov boundary is the minimal Markov blanket. A Markov blanket is clearly not unique, since any superset of the Markov boundary is a Markov blanket. The trivial Markov blanket is of course X itself. Unfortunately, this terminology is not standardized: some authors instead refer to the Markov boundary as "the Markov blanket" [175], so care must be taken when consulting the literature. Guyon et al. [69] also refer to the Markov boundary as a "surely sufficient feature subset".

An important fact for devising feature selection algorithms is that the Markov boundary is unique for any strictly positive data distribution. The following theorem is given by Pearl [130].

Theorem 3.2. *For any data distribution $f(x, y)$ satisfying $f(x) > 0$, the Markov boundary M^* is unique.*

Proof. Let S be the set of all Markov blankets of Y,

$$S = \{T \subseteq V_n : Y \perp X_{\neg T} \,|\, X_T\}.$$

Let T_1, T_2 be any two Markov blankets in S. Since $f(x) > 0$, by Theorem 2.10 the intersection property holds, so with $T' = T_1 \cap T_2$ we obtain

$$\begin{cases} Y \perp X_{\neg T_1} \,|\, X_{T' \cup (T_1 \setminus T')} \\ Y \perp X_{\neg T_2} \,|\, X_{T' \cup (T_2 \setminus T')} \end{cases} \implies Y \perp X_{\neg T'} \,|\, X_{T'}$$

Hence T' is a Markov blanket of Y. Continuing in this fashion for all members of S, we obtain the unique Markov boundary $M^* = T_1 \cap T_2 \cap \cdots \cap T_{|S|}$. □

Feature selection methods for inferring the Markov boundary of Y from data will be discussed in Section 4.1.4.

3.1.2 The Bayes-relevant features

From equation (3.2) it is clear that the Markov boundary contains exactly the features that affect the posterior distribution $p(y \,|\, x)$. However, we saw in section 2.4 that often, one does not estimate the posterior directly, but rather directly estimates the Bayes predictor $g^*(x)$. In some

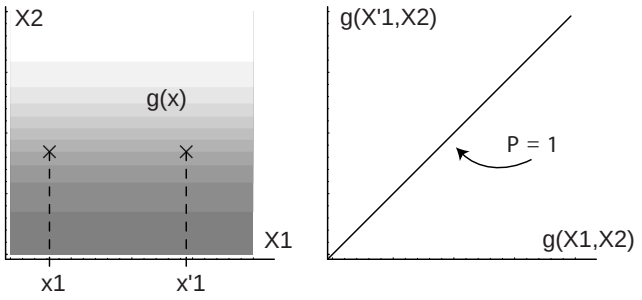

Figure 3.2: The concept of relevance to a predictor g for a two-dimensional case. Left, a function $g(x_1, x_2)$ which is constant with respect to x_1. Crosses indicate two observations (x_1, x_2) and (x'_1, x_2) where x_2 is held constant. Taking samples in this manner defines two random variables $g(X_1, X_2)$ and $g(X'_1, X_2)$, where X'_1 and X_1 and independent and identically distributed. If these are equal with probability 1 (right), then X_1 is not relevant to g.

cases, this an easier estimation problem (Section 2.4). This suggests a different definition, which considers as relevant only the features that affect the Bayes predictor. First we define this relevance concept for any predictor.

Definition 3.3. *A feature X_i is relevant to a predictor g iff*

$$P\left(g(X_i, X_{\neg i}) \neq g(X'_i, X_{\neg i})\right) > 0, \tag{3.3}$$

where X_i, X'_i are independent and identically distributed., and $X_{\neg i}$ denotes the vector of all features except X_i.

A definition of relevance very similar to the above was first suggested by Blum and Langley [16], although not in the probabilistic form used above. The left-hand side measures the probability that g will change its value (its prediction) merely due to a change in X_i. Note that in the above, X_i and X'_i are two identical variables corresponding to two samplings from the feature X_i, while the values of the remaining features are held constant; an illustration is given in Figure 3.2. Thus, the probability measure is over the domain $\mathcal{X} \times \mathcal{X}_i$.

As with definition 3.1, the criterion (3.3) can be put into integral form

$$\int_{\mathcal{X}\times\mathcal{X}_i} p(x, x_i') \left[g(x_i, x_{\neg i}) - g(x_i', x_{\neg i})\right] d(x, x_i') > 0.$$

The Kullback-Leibler divergence is not appropriate here though, since g^* is not a probability.

The definition of relevance to the Bayes predictor is now simply a special case of the above.

Definition 3.4 (Bayes-relevant)**.** *A feature X_i is said to be Bayes-relevant if it is relevant to the Bayes predictor g^* in the sense of definition 3.3. The set of Bayes-relevant features is denoted by S^*.*

Note that we here implicitly assume that the Bayes predictor is unique; otherwise definition 3.3 does not define a unique set S^*. Specifically, we will require uniqueness of g^* in the following sense.

Definition 3.5. *The Bayes rule g^* is said to be unique if for every g with $R(g) = R(g^*)$,*

$$P\left(g(X) = g^*(X)\right) = 1$$

In words, by "unique" we mean that every classifier attaining the Bayes risk makes the same predictions with probability 1. This issue is not entirely straightforward, because the definition of the Bayes predictor (Equation (2.12)) does not define a unique function; it leaves open the choice of the function g' for breaking ties $p(y \mid x) = p(y' \mid x)$. Hence, if "ties" occur with nonzero probability, then one could devise two optimal classifiers with are *not* identical in the above sense. This is remedied when necessary by the following assumption on $f(x, y)$, which simply prohibits "ties" to occur.

Assumption 3.6.

$$P\left(\exists y \neq y' : p(y \mid X) = p(y' \mid X)\right) = 0$$

Under this assumption, we now establish the required uniqueness.

Theorem 3.7. *For any distribution $f(x, y)$ satisfying assumption 3.6, the Bayes classifier g^* is unique in the sense of definition 3.5.*

Proof. We consider the case of discrete $\mathcal{Y}$ and 0-1 loss. From Theo-

rem 2.18 we have that, for any $y_0 \in \mathcal{Y}$,

$$\begin{aligned} P\left(g(X) \neq Y \,|\, X = x\right) - P\left(g^*(X) \neq Y \,|\, X = x\right) \\ = (2p(y_0 \,|\, x) - 1)(1_{\{g^*(x)=y_0\}} - 1_{\{g(x)=y_0\}}) \\ = |2p(y_0 \,|\, x) - 1| 1_{\{g^*(x)\neq g(x)\}} \end{aligned}$$

Integrating with respect to $p(x)dx$,

$$\begin{aligned} R(g) - R(g^*) &= P\left(g(X) \neq Y\right) - P\left(g^*(X) \neq Y\right) \\ &= \int_{\mathcal{X}} |2p(y_0 \,|\, x) - 1| 1_{\{g^*(x)\neq g(x)\}} f(x)dx. \end{aligned}$$

For discrete $\mathcal{Y}$, Assumption 3.6 implies $|2p(y_0 \,|\, x) - 1| > 0$ with probability 1. Therefore, the above integral is positive if and only if

$$\int_{\mathcal{X}} 1_{\{g^*(x)\neq g(x)\}} f(x)dx = P\left(g^*(X) \neq g(X)\right) > 0.$$

□

This result immediately establishes uniqueness of S^*.

Corollary 3.8. *For any distribution $f(x, y)$ satisfying assumption 3.6, the set S^* of Bayes-relevant features is unique.*

In some particular cases the Markov boundary and the Bayes-relevant features coincide. An important example of this is the Gaussian mixture in example 2.5. This is easy to see directly from the form of the posterior and Bayes classifier in this case. If $S = \{i : w_i \neq 0\}$, then

$$\begin{aligned} p(y \,|\, x, w) &= \left[1 + \frac{p(-y)}{p(y)} \exp\left\{-2yx^T w\right\}\right]^{-1} \\ &= \left[1 + \frac{p(-y)}{p(y)} \exp\left\{-2yx_S^T w_S\right\}\right]^{-1} \\ &= p(y \,|\, x_S, w_S), \end{aligned}$$

so that S is the Markov boundary of Y. Also, $g^*(x) = \text{sign}(w^T x) = \text{sign}(w_S^T x_S)$, so that (3.3) holds for g^* iff $i \in S$, and therefore $S = S^*$.

3.2 Small sample-optimal features

In the previous section we identified two sets of predictive features, those that influence the posterior (Markov boundary) or the Bayes predictor

(Bayes-relevant features). Both of these concepts are idealizations, since in practise, given limited training data, we cannot determine the posterior or Bayes predictor exactly. One might say that these feature sets are asymptotic cases, as they are optimal in the large sample limit. In this section I consider the small-sample case.

We saw in 2.7.1 that the performance of an inferred predictor may degrade with the number of features, even if the Bayes error is decreasing, due to over-fitting. Thus, in practise the feature set optimal for *inferring* an accurate predictor may be smaller than S^*. A more appropriate definition of the optimal feature set for small samples is the following.

Definition 3.9 (Expectation-optimal feature set)**.** *For a given data distribution, a sample size l, and an inducer I_S, an expectation-optimal feature set $S^\dagger$ satisfies*

$$\forall S : \mathbb{E}\left[R(I_{S^\dagger})\right] \leq \mathbb{E}\left[R(I_S)\right].$$

To be precise, we here require a *family* of inducers $\{I_S(Z_S^{(1:l)}) : S \subseteq V_n\}$, *i.e.*, one inducer I_S for each possible subset S, so that the expected risk can be evaluated for every S. This is not a problem in practise since inducers are typically well-defined for any input space $\mathcal{X}_S$ (otherwise, feature selection would not work). Consequently, when speaking of "the inducer", it is implicitly that we have a family of I_S.

By definition, $S^\dagger$ is the best possible choice for a particular inducer, data distribution and sample size. Unfortunately, this feature set is difficult to analyze theoretically, precisely because it depends on the inducer. For this reason, I will focus on the more tractable Bayes-relevant features in chapter 6. Under some conditions on the data distribution and the inducer I, it can be shown that $S^\dagger \subseteq S^*$. I will defer the discussion of this issue to Section 6.1.3.

If two or more features X_i, X_j are identically distributed, then the set $S^\dagger$ may not be unique (even though S^* is). In Chapter 7, we will find that this leads to "instability" for methods that attempt to approximate $S^\dagger$; that is, different feature sets are obtained when one tries to replicate experiments. Finally, it should be noted that one could consider optimality with respect to other measures than the expectation value. For example, one might want to minimize the median risk, or perhaps the 5%-percentile of the distribution of $R(I_S)$. Thus, the above definition of $S^\dagger$, while reasonable, is somewhat subjective.

Tsamardinos and Aliferis [176] recently raised the issue that many feature selection papers (notably those concerning "filter" methods, Sec-

tion 4.1) unfortunately do not recognize that $S^\dagger$ depends on the choice of inducer. This dependence was also the major motivation for the "wrapper" approach introduced by Kohavi and John [97], which I review in Section 4.2.

3.2.1 The min-features bias

In many applications a small set of predictive features is preferable. One reason may be that the computational cost of inference and/or prediction is too high for large feature sets; this is important in real-time applications such as computer vision [28], but probably less important in biological applications. Another plausible reason is that a smaller feature set may be easier to interpret intuitively. Thus, many methods attempt to optimize the feature set with an explicit bias towards small sets, even if this means sacrificing some predictive performance. This is referred to as the *min-features* bias [6]. It may be formulated as follows.

Definition 3.10. *The min-features problem is defined as*

$$\min_{S \subseteq X} \mathbb{E}\left[R(I_S)\right] + \lambda |S|. \tag{3.4}$$

A set which is a solution to this problem is called a min-features set and is denoted $S^\ddagger_\lambda$.

This feature set is also referred to as "minimal approximately sufficient" by Guyon et al. [69]. Like $S^\dagger$, this set clearly depends on the inducer and the sample size, and is in general not unique. It also depends on the parameter λ, which controls the strength of the bias towards small S. This problem formulation may be viewed is a form of regularization, with $|S|$ as the regularization term (*cf.* Section 2.7.2); I will discuss this connection further in Section 4.3. Note also that one obtains $S^\dagger$ as $\lambda \to 0$ in (3.4). Hence, it must hold that $S^\ddagger_\lambda \subseteq S^\dagger$ for all λ. A number of techniques that address the min-features problem are discussed in Section 4.3.

3.2.2 k-optimal feature sets

A variant of the above is to simply fix the feature set size k in advance and attempt to solve the optimization problem

$$\min_S \mathbb{E}\left[R(I_S)\right] \quad \text{subject to } |S| = k.$$

We call the solution to this problem a k-optimal feature set. This problem was studied extensively in early feature selection research [33, 94, 121]. This problem formulation was motivated in an optimization setting rather than in a statistical setting, and is therefore somewhat difficult to relate to the other formulations examined above. If we know that $k \leq |S^\dagger|$, then intuitively a k-optimal feature set corresponds to a min-features set $S^\ddagger_\lambda$ for some value of λ, and in this case the latter formulation may be used instead. However, if $k > |S^\dagger|$, then the meaning of a k-optimal feature set is less clear.

3.3 All relevant features

In the previous section we considered the most predictive features, motivated by the problem learning accurate predictors. This has also been the primary motivation for performing feature selection in machine learning literature [67, 97]. However, for biological data we are primarily interested in features of biological importance: examples include genes with a genetic association to a particular disease, key players in biological pathways of interest, residues in an enzyme that mediate its mechanism of action, DNA sequence features that affect gene regulation, etc. Naturally, "biological interest" is a rather fuzzy concept — often in biology, one cannot know in advance exactly what one is searching for, and so precise definitions are unfortunately lacking. Consequently, in analysis of biological prediction problems, researchers often use relevance for prediction as a substitute for biological relevance, in want of a better alternative. However, it is not obvious that the most predictive variables are always the most "interesting". Consider the following example.

Example 3.1 *A target variable Y is affecting the expression level of a transcription factor T_1, which in turn is affecting the expression level T_2 of some effector protein. Both T_1 and T_2 are observed only indirectly, through measurements X_1, X_2. A Bayesian network representing this situation is given in Figure 3.3A. Here $Y \in \{+1, -1\}$ and*

$(T_1, T_2, X_1, X_2) \in \mathbb{R}^4$. *Let the local probabilities be*

$$\begin{aligned} p(y) &= 1/2, \\ f(t_1 \mid y) &= N(t_1 \mid y, \sigma^2) \\ f(t_2 \mid t_1) &= N(t_2 \mid \beta t_1, \sigma^2) \\ f(x_1 \mid t_1) &= N(x_1 \mid t_1, \nu^2) \\ f(x_2 \mid t_2) &= N(x_2 \mid t_2, \nu^2). \end{aligned}$$

Here σ represents noisy transcriptional regulation while ν represents measurement noise The marginal densities for x_1 and x_2 are easily found by integrating out t_1 and t_2,

$$\begin{aligned} f(x_1 \mid y) &= p(y)N(x_1 \mid y, \nu^2 + \sigma^2) \\ f(x_2 \mid y) &= p(y)N(x_1 \mid \beta y, \nu^2 + \sigma^2 + \beta^2\sigma^2). \end{aligned}$$

Clearly, the best individual predictors are $g_1(x_1) = \text{sign}(x_1)$ and $g_2(x_2) = \text{sign}(x_2)$, respectively. From Equation (2.17) we find that $R_1(g_1) > R_2(g_2)$ if and only if

$$\frac{1}{\nu^2 + \sigma^2} < \frac{\beta^2}{\nu^2 + \sigma^2 + \beta^2\sigma^2}$$

For $\sigma = 1$, Figure 3.3B depicts this region the parameter space (β, ν).

In this example, marginalizing out the unobservable true transcript levels can result in a distribution $f(x_1, x_2, y)$ where X_2 is a better predictor of Y than X_1 even though T_1 is a direct effect of the target Y, while T_2 is a downstream effect. This happens because T_1 is expressed in small amounts compared to T_2, *i.e.*, when the amplification β is large (figure X), while measurements X_1, X_2 have some constant additive noise level.

Indeed, transcription factors are often present in very small amounts and are therefore difficult to detect with microarrays, leading to poor signal-to-noise ratios [78]. Yet, these genes are biologically very important as they are involved in regulation of other genes. For example, they are often implicated in for example cancer development [35]. Therefore, to avoid missing out on important but minutely expressed genes like T_1, we might prefer a definition of "relevance" which includes *all* genes that are somewhat statistically dependent of Y, as opposed to only the most predictive ones. The notion of statistical dependency we require here is the following.

Definition 3.11 (Relevance)**.** *A feature X_i is relevant to Y iff*

$$\exists S \in V_n : Y \not\perp X_i \mid X_S. \tag{3.5}$$

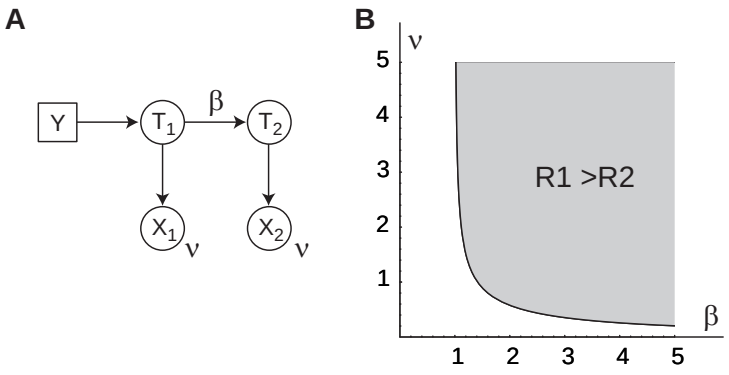

Figure 3.3: Graphical illustration of Example 3.1. **A:** Network structure for the problem. β denotes the amplification factor while ν denotes measurement noise. **B:** Plot over the parameter space (β, ν). In the shaded region, the risk R_1 for the predictor using X_1 is greater than R_2, using X_2.

A feature X_i is irrelevant to Y iff it is not relevant to Y. The set of all relevant features is denoted S^A.

Unfortunately, the term "relevance" is often used casually in feature selection literature, without any clear definition. As often as not, some other definition than the above may be intended, so care must be taken when comparing literature. From the negation of (3.5),

$$\forall S \in V_n : Y \perp X_i \,|\, X_S, \tag{3.6}$$

it is clear that if X_i is irrelevant, then it is statistically independent of Y no matter which features are conditioned on or marginalized out. As with the Markov boundary, this definition can also be expressed using a distribution divergence measure. The following formulation is given by Guyon et al. [69]. Since (3.6) can be written as $p(y, x_i \,|\, x_S) = p(y \,|\, x_S)p(x_i \,|\, x_S)$, then for discrete X and Y it is equivalent to

$$\forall S \in V_n : \sum_{x_i \in \mathcal{X}_i, x_S \in \mathcal{X}_S, y \in \mathcal{Y}} p(y, x_i \,|\, x_S) \ln \frac{p(y, x_i \,|\, x_S)}{p(y \,|\, x_S)p(x_i \,|\, x_S)} = 0.$$

Since the irrelevant features contain no information about Y whatsoever, S^A is the maximal set of features with some association to Y. Therefore, if one could learn this set, that would ensure that no feature of potential (biological) importance goes unnoticed.

3.3.1 The univariate case

In many applications to biological data, it is common to consider the particular subset of S^A which are *marginally* dependent on Y, that is, the features X_i which satisfy $Y \not\perp X_i \,|\, \emptyset$, or equivalently $p(Y \,|\, X_i) \neq p(Y)$. This is a rather narrow special case of (3.5), with the empty conditioning set $S = \emptyset$. It is often further constrained to particular forms of dependence, such as difference in class-conditional expectations $\mathbb{E}\left[X_i \,|\, Y = y\right]$ for classification problems, or linear correlation for the regression case. The primary reason for restricting analysis to the univariate case is of course that univariate problems are comparatively simple. A survey of methods for solving this case is found in Section 4.1.1.

3.3.2 The multivariate case

It is in no way apparent from biology that the above univariate case should be the most "interesting" one, even though it certainly is the most tractable and (therefore) well-studied situation. While we in Example 3.1 considered a simple network consisting of only two genes, in reality gene expression is known to be governed by large, complex networks of regulatory interactions involving many transcription factors and effector proteins [95, 167]. For example, the tumor suppressor p53, one of the most studied cancer-related genes, is currently known to interact with more than 200 other genes according to the HPRD database [133]. This suggests that the statistical dependencies among gene expression levels and similar biological measurements may be highly multivariate.

I therefore consider the general problem of inferring S^A in chapter 8. It turns out that identifying S^A from data is in many ways much harder than identifying the Markov boundary or the Bayes-relevant features. Yet, I find that there exists fairly general distribution classes where the problem is tractable. However, in some cases the set S^A may be very large, and therefore not very useful for identifying "interesting" genes. In such situations, other methods of prioritizing among the genes in S^A are needed. One may then resort back to the predictive features, or perhaps apply some external criteria motivated by biology.

3.4 FEATURE EXTRACTION AND GENE SET TESTING

A different way to reduce dimension and thus overcome the over-fitting problem (Section 2.7.1) is to somehow transform the input space $\mathcal{X}$ into some lower-dimensional space $\mathcal{X}'$. For example, we might consider projecting the points in $\mathcal{X}$ onto a linear subspace of dimension $m < n$ [71]. This process is called *feature extraction*. It may be seen as a generalization of feature selection, since feature selection is the particular transformation $x \mapsto x_S$ for some set $S \subset V_n$. While feature extraction methods can be effective for inferring predictors [11, 136], the results are not easy to interpret in terms of the original features since each transformed feature is a "mix" of the original features X_i. For this reason, I do not consider feature extraction methods in this thesis.

A related problem which has recently been given much attention in gene expression analysis is *gene set testing* [61]. Here, prior knowledge of functional groups of genes is used to define a number of feature sets $S_1, \ldots, S_K$ independent of data. Then, for each $k = 1, \ldots, K$, a measure of (multivariate) dependence between X_{S_k} and Y is constructed. Similar to feature extraction methods, this may be effective in terms of deciding whether each S_k is associated with Y, but it says nothing about the individual features X_i in each S_k. As with feature extraction methods, dependence is now measured against some mixture of the X_i, and is therefore more difficult to interpret. Nevertheless, gene set testing still yields biologically informative results because the S_k were defined from domain knowledge. Some methods for feature extraction and gene set testing are briefly surveyed in Section 4.4.

3.5 SUMMARY

In this chapter I have defined a number of feature selection problems. In my opinion, a major impediment to feature selection research is that the particular problem considered is often not stated clearly, with a formal and operational definition. As a result, the various feature selection problems are easily confounded. Perhaps this tendency is part of a more general but unfortunate trend in machine learning, to focus on algorithm development without first carefully considering what problem one is trying to solve. Quoting from Almuallim and Dietterich [6],

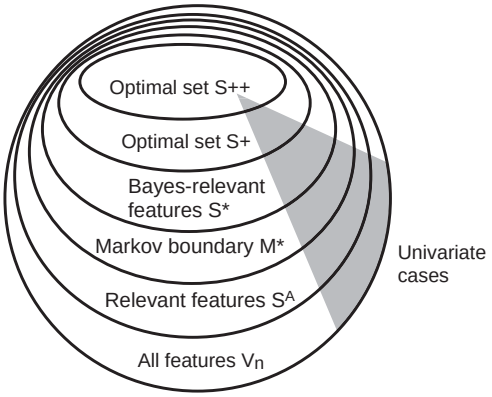

Figure 3.4: An overview of the feature sets defined in this chapter. Each smaller set is included in the larger, surrounding set. Gray area represents the univariate case of each set. Note that the inclusion $S^{\dagger} \subseteq S^{*}$ is subject to some constraints; see Section 6.1.3.

> "there is no separation between a *specification* of the desired learning behavior of the algorithm and its *implementation* ... consequently, it is difficult to tell in advance whether the [algorithm] is appropriate for a new problem."

This quote is from 1991; as described in the previous chapter, the situation has improved since then, particular for prediction methods. For feature selection, however, this problem is still present. In this chapter, I have tried to address this issue by rigorously defining a set of common feature selection problems and the corresponding feature sets. Figure 3.4 summarizes the relations between these sets. In summary, feature selection problems can be roughly divided into two rather different types: (i) finding the *predictive* features, which are important for building accurate predictors, and (ii) finding *all* features relevant to the target variable.

Problem (i) can be further subdivided depending on the type of prediction. If one desires a measure confidence in the prediction (posterior probability), then in general more features are required than if one does not. Also, for large samples (the asymptotic case) more features are useful than for small samples, as a consequence of the over-fitting problem (Section 2.7.1). Most methods from the machine learning field are of this type. In practise, this problem formulation is useful when (1) the end goal is the predictor and the relevant features are of limited interest, or

(2) when the set of relevant features is too large to analyze, and it seems sensible to focus on analyzing the predictive features. In Chapter 6 I consider this problem in detail.

Problem (ii) is relevant whenever one wants to ensure that no feature associated with the target variable is missed. This problem formulation can also be divided into several simplified subproblems by considering particular classes of distributions. In particular, statistical methods such as differential expression testing treat the one-dimensional special case of problem (ii). I believe that this problem is often relevant in genome-wide experiments, as illustrated in Example 3.1. I will treat this problem in more detail in Chapter 8.

4

Feature Selection Methods

In this chapter I review existing methods for feature selection. Formally, I define a feature selection method as follows.

Definition 4.1. *For a given data set $z^{(1:l)}$, a feature selection method is a mapping*

$$\Phi(z^{(1:l)}) : \mathcal{Z}^l \mapsto 2^{V_n},$$

where 2^{V_n} is the power-set of V_n, $2^{V_n} = \{S : S \subseteq V_n\}$.

In words, a feature selection method selects a subset S of V_n for each given data set $z^{(1:l)}$. As with inducers, when the data set is viewed as a random variable $Z^{(1:l)}$, then naturally $\Phi(Z^{(1:l)})$ is also a random variable on 2^{V_n}. In machine learning literature, feature selection methods are traditionally divided into *filter methods* (or simply "filters"), which perform feature selection independent of any particular inducer, and *wrapper methods* (or "wrappers"), which try to optimize the feature set for a given inducer. More recently a third category, the *embedded* feature selection methods, has been added to this system. This chapter is organized according to this system. In addition, I provide a brief summary of feature extraction and gene set testing in Section 4.4.

For each section and each method, I try to relate to the corresponding feature selection problems defined in the previous chapter. This is not

always trivial, since many methods are not grounded in any such theory, but rather motivated by intuition and heuristics. Often, determining exactly what problem each FS method really tries to solve is something of a "reverse engineering" task. In my opinion, this is unfortunate for the interpretability of the results from these methods; more about this issue in chapter 9. Therefore I will throughout this chapter strive to identify the *target set* for each for each feature selection method discussed. Asymptotically, this amounts to analyzing the *consistency* of each method.

Definition 4.2. *A feature selection algorithm* $\Phi(Z^{(1:l)})$ *is consistent with respect to a feature set* S *if*

$$\Phi(Z^{(1:l)}) \xrightarrow{P} S.$$

The set S *is called the target set of* Φ*.*

The target set S to which a given method Φ can be shown to converge will in general be various subsets of the sets defined in the previous chapter ($S^A, S^*, S^\dagger$, *etc.*). Often, it is hard to obtain convergence results to a precisely determined target set S, but it may still be feasible to state a set which must *enclose* the target set to which the method converges. To my knowledge, this type of analysis is novel for many of the methods herein. Hopefully, this will lead to better understanding of the available feature selection methods and their relations to each other, and to some extent explain which methods are suitable for which problems. An overview of these (fairly complicated) relations is given in Figure 4.1.

Results for for small samples are much harder to derive and is known only in some very simple situations. I will state such results when possible.

4.1 FILTER METHODS

The distinction between filter and wrapper methods was first described in a seminal paper by Kohavi and John [97]. The definition of a *filter method* is that it "attempts to assess the merits of features from the data alone" [99], that is, without considering any particular inducer I. Admittedly, this definition is somewhat vague. Nevertheless, it allows a few observations: since filter methods are oblivious to the choice of predictor (if any), they must be derived from, or at least motivated by, properties of the data distribution itself.

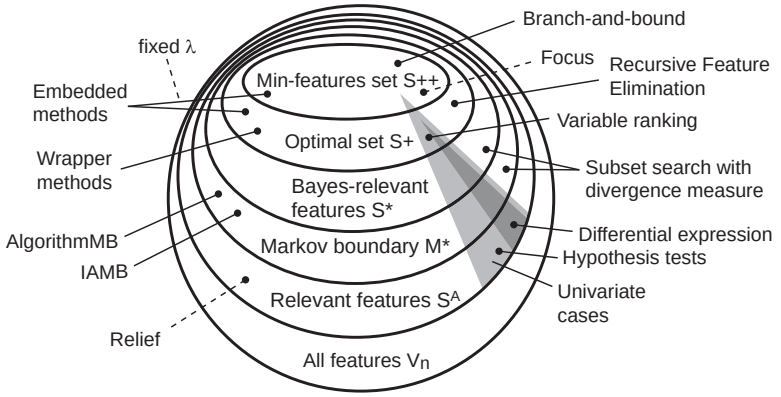

Figure 4.1: The feature sets established in the previous chapter, indicating the "target" set for methods presented herein.

Hence, filter methods cannot estimate the expected-optimal set $S^\dagger$, since this set depends on a particular inducer I. More precisely, for a given filter method Φ, it may be possible to construct an inducer I such that its corresponding $S^\dagger$ coincides with the target set of Φ; however, this is then the only I for which Φ is optimal. In other words, there is no "universal" filter method that is optimal at small samples regardless the predictor chosen. This fact was stated as a "no free lunch" theorem by Tsamardinos and Aliferis [176]. Typically, the analysis in this section will show that the target set of each filter method is some particular subset of S^* or S^A.

4.1.1 Statistical hypothesis tests

Statistical *hypothesis tests* measure the dependency of individual features X_i on the target variable, and hence concern themselves with the marginal distributions $f(x_i, y)$ only. These tests will identify any feature with a statistically significant dependence, and do not attempt single out the most predictive features. Therefore, their target set is always the subset of the relevant features S^A which are "marginally relevant", that is, X_i such that $Y \not\perp X_i \,|\, \emptyset$ (Figure 4.1).

The ideas underlying hypothesis testing are part of classical statistics; a comprehensive treatment is given by Lehmann [107]. Briefly, the hypothesis tests we are interested in assume two complementary hypotheses: the

null hypothesis H_0^i, which states that the feature X_i is irrelevant to (independent of) Y, and the alternative hypothesis H_1^i, which states that it is relevant (dependent). Clearly, every feature belongs to either H_0^i or H_1^i. A statistical test can then be defined a function $\phi_i : \mathcal{Z}^l \mapsto \{1, 0\}$ of the observations for feature X_i, which decides ("calls") that H_0^i is true if $\phi_i(z_i^{(1:l)}) = 0$ and H_1 if $\phi_i(z_i^{(1:l)}) = 1$. This decision may or may not be correct, of course. How often the test is correct is measured by two *error rates*: the "type I" error rate

$$P\left(\phi_i = 1 \,|\, H_0^i\right),$$

which measures how often feature X_i is called relevant by ϕ_i when it in fact is not, and the "type II" error rate

$$P\left(\phi_i = 0 \,|\, H_1^i\right),$$

which measure the opposite situation. The type I error rate is also referred to as the *false positive rate*, and the type II error rate is referred to as the *false negative rate*. I prefer these latter terms since their meaning is clearer and easier to remember. The *power* of a test is defined as 1 minus the false negative rate, that is, power is the probability that false null hypotheses are rejected, $P\left(\phi_i = 1 \,|\, H_1^i\right)$.

Classical statistical tests always have a single parameter α, referred to as the *level* of the test. Ideally, tests are constructed so that, under some appropriate distribution assumptions, the false positive rate can be proven to be less than α,

$$P\left(\phi_i = 1 \,|\, H_0^i\right) \leq \alpha. \tag{4.1}$$

When the above is satisfied, we say that ϕ_i is a level α test, or that the test is *exact*. If equality holds in (4.1), then we say that ϕ_i is a *size* α test [21]. (Some statistical textbooks do not distinguish between size and level, however.) If a test does not satisfy (4.1), we distinguish between the *nominal* level α and the *realized* level, which is the actual value of $P\left(\phi_i = 1 \,|\, H_0^i\right)$. Of course, one would always like to construct tests so that the nominal and realized levels coincide, but in practise this is not always possible.

From a set of hypothesis tests ϕ_i, $i = 1, \ldots, n$, it is straightforward to define a feature selection method

$$\Phi(z^{(1:l)}) = \{i : \phi_i(z_i^{(1:l)}) = 1\}. \tag{4.2}$$

Since this Φ depends only on the marginal distributions $f(x_i, y)$, it clearly ignores any multivariate effects. If we may further assume some

particular density for these marginals, we obtained so-called *parametric* tests; if we refrain from making such distribution assumptions, the tests are *nonparametric*. But in either case, it should be remembered that we are assuming that the relevant features can be detected merely by observing their marginals. This is correct only in a rather narrow class of data distributions $f(x, y)$.

There are also many multivariate statistical tests which do consider the joint distribution $f(x, y)$. However, these are not interesting for feature selection, since they test a single hypothesis for all features, and so constitute a single function $\phi(z^{(1:l)}) \mapsto \{0, 1\}$. This function cannot be used directly as a feature selection method according to (4.2). However, multivariate tests can be used together with search methods (Section 4.1.5) and are also useful for gene set testing; a brief survey is given in Section 3.4.

Testing for marginal dependence

For continuous Y, the most popular tests for dependence concern linear correlations between an X_i and Y,

$$r = \frac{\mathrm{Cov}(X_i, Y)}{\sqrt{\mathrm{Var}\, X_i \,\mathrm{Var}\, Y}}.$$

Fisher's t-test is a parametric test for correlation assuming a bivariate Gaussian $f(x_i, y)$, with null hypothesis $H_0^i : r = 0$. Given the Pearson sample correlation estimate

$$\hat{r} = \frac{\sum_j (x_i^{(j)} - \bar{x}_i)(y^{(j)} - \bar{y})}{\sqrt{\sum_j (x_i^{(j)} - \bar{x}_i)^2 \sum_i (y^{(j)} - \bar{y})^2}},$$

The Fisher t-test computes a statistic

$$t = \frac{\hat{r}\sqrt{l-2}}{\sqrt{1-\hat{r}^2}},$$

which under the Gaussian assumptions has a Student t distribution with $l-2$ degrees of freedom [138]. Under these assumptions, this test is exact for small samples, and is also known to have optimal power [21].

Another common test based on Pearson's correlation measure is the *Fisher z-test* (often confused with the above). This test computes the statistic

$$t = \tanh^{-1} \hat{r} = \frac{1}{2} \ln \frac{1+\hat{r}}{1-\hat{r}},$$

which is asymptotically Gaussian, $f(t) = N(\tanh^{-1} r, (l-3)^{-1})$ for *any* $f(x_i, y)$ due to the central limit theorem [49]. The test is therefore asymptotically correct.

For small samples however, both tests may be misleading since Pearson's $\hat{r}$ can be very inaccurate when $f(x_i, y)$ is not Gaussian. Notably, "outliers" can have dramatic effects and may result in large false positive rates. For such situations, tests based on rank correlation measures such as Spearman's may be used instead [108]. These are much more robust against outliers and still retain most of the power of the parametric tests, unless sample size is very small ($l < 20$). Note that, while these tests are often referred to as "distribution-free", they still assume that we are testing a linear correlation, so they are not general independence tests.

General independence tests for continuous Y exist, although they are less well known. Some tests based on kernel methods are discussed by Gretton et al. [65]. General independence measures such as the mutual information [34] may be also be used, for example with together with permutation tests [52] (see below). However, these independence measures generally require more data than the simpler tests discussed above in order to be informative.

Differential expression tests

For binary $\mathcal{Y} = \{+1, -1\}$, testing the null hypothesis $H_0^i : Y \perp X_i \,|\, \emptyset$ is equivalent to testing whether the conditional distributions $f(x_i \,|\, Y = +1)$ and $f(x_i \,|\, Y = -1)$ differ. A well-known non-parametric statistical test for this is the Kolmogorov-Smirnov test [138], which is known to be consistent for any distribution $f(x_i, y)$ by the Glivenko-Cantelli theorem [181, pp. 42].

An important special case here are features which differ in expectation between the two classes, *i.e.*, features X_i that satisfy

$$\mathbb{E}\left[X_i \,|\, Y = +1\right] \neq \mathbb{E}\left[X_i \,|\, Y = -1\right]. \tag{4.3}$$

In gene expression analysis, this type of dependence is termed *differential expression*. The most popular test in this case (and probably the most well-known of all) is Student's t-test [164]. This test is exact for small samples if both $f(X_i \,|\, Y = +1)$ and $f(y_i \,|\, Y = -1)$ are Gaussian with equal variances. For unequal variance, the correction by Welch [184] yields a conservative, nearly exact test. However, like Pearson's $\hat{r}$, Student's t-test is unreliable when the true marginal distributions

are non-Gaussian. A variety of general non-parametric tests also exist, including Wilcoxon's rank sum test and the Whitney-Mann U test [108].

In microarray analysis, testing for differential expression has been studied extensively, and a plethora of methods tailored to the particular distributions observed in microarray data have been proposed. Recently, Bayesian hypothesis testing methods [10, 45, 124] have been especially popular, as these include correction for multiple testing (see Section 4.1.2) and appear to have good power. For a recent review of this topic, see Allison et al. [5].

Permutation tests

The permutation test, originally introduced by [135], is a general method for obtaining a statistical test from any statistic T that measures dependence between two variables. Permutation tests are non-parametric and provide exact confidence levels under mild assumptions. A modern treatment is given by [63].

Let $T_i = T(X_i^{(1:l)}, Y^{(1:l)}) \in \mathbb{R}$ be a measure of dependency such that, under H_0^i, $T_i \perp Y_i$. Then, under H_0^i it is clear that the distribution of T_i will not change if one permutes (re-orders) the vector $Y^{(1:l)} = (Y^{(1)}, \ldots, Y^{(l)})$. Thus, under H_0^i, every statistic T_i' computed from permutations of $Y^{(1:l)}$ is distributed identically to T_i, so that a large number of such permuted statistics may be used as an estimate of the distribution of T_i. Therefore, an observed t_i which is "extreme" with respect to this distribution, that is $P(|T_i| > |t_i|) \leq \alpha$ indicates that H_0^i should be rejected. The key result is the following [63].

Theorem 4.3. *For a set of permuted statistics $T_i^{(1)}, \ldots, T_i^{(N)}$, if H_0^i is true, then*

$$\frac{1}{N}\mathbb{E}\left[|\{j : |T_i^{(j)}| > |t_i|\}|\right] = P\left(T_i > t \,|\, H_0^i\right).$$

The above implies that the permutation test given by

$$\phi_i(t_i) = \begin{cases} 1, & |\{j : |t_i^{(j)}| > |t_i|\}| < N\alpha \\ 0, & \text{otherwise} \end{cases}$$

has level α. However, to obtain a reasonably powerful test, N has to be quite large. For a single test, $N = 1,000$ might give good power at $\alpha = 0.05$, while for multiple tests or more stringent α we might need much larger N, perhaps on the order of 10^6. Thus, unless the

statistic T is easy to compute, permutation tests may require a lot of computations. Nevertheless, permutation tests are very useful and quite popular in biological applications [82].

4.1.2 The multiple testing problem

A pervasive problem in hypothesis testing for high-dimensional data is *test multiplicity*, also known as the *multiple testing problem*. This issue arises as soon as more than one test is made, in the following fashion. Consider a feature selection method (4.2), which makes n tests $\phi_1, \dots \phi_n$, each of level α. By definition, the expected number of errors is then

$$\mathbb{E}\left[\left|\Phi(Z^{(1:l)}) \cap \{i : H_0^i\}\right|\right] = \mathbb{E}\left[\sum_{i:H_0^i} \phi_i(Z^{(1:l)})\right].$$

In the worst case, if all H_0^i are true, this expectation equals $n\alpha$, which for a typical genome-wide data set with $n = 10^4$, $\alpha = 0.05$ yields 500 errors and no true positives. If the fraction of true H_1^i is small, we still obtain about 500 errors and only a few true positives. We might attempt to compensate for this by lowering α (see next section), but when doing so we will inevitably lose power. This is the multiple testing problem. It is yet another manifestation of the problem that, with increasing data dimension, statistical inference becomes more difficult.

The family-wise error rate

The most immediate technique for compensating for multiplicity is to adjust the tests to be more stringent. Instead of constructing level α tests, which control the false positive rate, a first intuitive idea is to construct tests which control the *family-wise error rate*.

Definition 4.4. *For a set of null hypotheses $H_0^1, \dots, H_0^n$ and a corresponding set of tests $\phi_1, \dots, \phi_n$, the family-wise error rate (FWER) is defined as*

$$P\left(\exists i : \phi_i(Z^{(1:l)}) = 1 \mid H_0^{(1:n)}\right),$$

where $H_0^{(1:n)}$ is the complete null hypothesis

$$H_0^{(1:n)} = H_0^1 \wedge H_0^2, \wedge \cdots \wedge H_0^n.$$

In words, the FWER is the probability that a single error is made by any of the tests $\phi_1, \ldots, \phi_n$. Typical methods for controlling the FWER are based on the p-value concept.

Definition 4.5 (P-value)**.** *A p-value for a given null hypothesis H_0 is a random variable $p \in [0,1]$ satisfying*

$$P\left(p \leq \alpha \,|\, H_0\right) \leq \alpha$$

for all $\alpha \in [0,1]$.

The traditional notation here is a rather confusing: p is test statistic, a random variable computed from data as $p = p(Z^{(1:l)})$, and yet it is denoted by a lowercase symbol. Moreover, it is easy to confuse p-values with probability functions such as $p(x)$. It should therefore emphasized that the p-value is *not* a probability; it is a random variable. This is a source of much confusion, but unfortunately the notation is now fixed by convention.

The p-value statistic is useful because a level α test is immediately obtained from p as

$$\phi(p) = \begin{cases} 1, & p < \alpha \\ 0, & p \geq \alpha \end{cases} .$$

The p-value also has the fairly intuitive interpretation that an observed value p is the lowest level for which the test $\phi(p)$ would reject the null hypothesis. Therefore, the p-value effectively measures the "confidence" in the rejection. However, I again emphasize that it must *not* be interpreted as the "probability of H_0".

Returning to the family-wise error rate, for n independent level α tests, the FWER can be controlled as

$$\begin{aligned} P\left(\exists i : \phi_i(Z^{(1:l)}) = 1 \,|\, H_0^{(1:n)}\right) &= P\left(\exists i : p_i \leq \alpha \,|\, H_0^{(1:n)}\right) \\ &= 1 - P\left(\forall p_i \geq \alpha \,|\, H_0^{(1:n)}\right) \\ &= 1 - \prod_i P\left(p_i \geq \alpha \,|\, H_0^{(1:n)}\right) \\ &= 1 - (1-\alpha)^n \end{aligned}$$

Inverting this equation, one finds that if the corresponding p-value is adjusted as

$$p_i' = 1 - (1 - p_i)^n, \tag{4.4}$$

then one obtains $P\left(\exists i : p_i' \leq \alpha \,|\, H_0^{(1:n)}\right) \leq \alpha$ as desired. This idea of "adjusting" p-values is common since the resulting p_i' again serves as a

measure of "confidence" in the test result, but now with respect to the FWER, which is easy to interpret. The above correction is sometimes referred to as the Šidàk correction [155].

Often, assuming independence between the tests ϕ_i is not reasonable with biological data, given that the features X_i are often dependent. In this case, the Šidàk correction may be too liberal (or too conservative; but as long as the nature of the dependencies is unknown, this cannot be determined). A simple method that does not require independence is obtained directly from the union bound,

$$P\left(\exists i : p_i \leq \alpha \,|\, H_0^{(1:n)}\right) \leq \sum_i P\left(p_i \leq \alpha \,|\, H_0^{(1:n)}\right) = n\alpha$$

This yields the correction $p_i' = \min\{np_i, 1\}$. This is known as the Bonferroni correction, named after Carlo Emilio Bonferroni, who first described the union bound [1]. This correction always controls the FWER, since we have made no additional assumptions other than that the test are indeed level α. For large n, it is also close to the Šidàk correction.

The Bonferroni correction is unnecessarily conservative, however. More powerful alternatives are the so-called *step-down* approaches, based on ordered p-values $p_{(1)} \leq \cdots \leq p_{(n)}$. The following correction is due to Holm [79], where the corrected, ordered p-value is given by

$$p'_{(i)} = \max_{j=1,\ldots,i} \left(\min\{(n-j+1)p_{(j)}, 1\}\right).$$

Note that the quantity $(n-i+j)p_{(j)}$ need not be monotone in j; hence the outer maxima over $j = 1, \ldots, i$. This correction controls the FWER under the same assumption as the Bonferroni correction but is more powerful, and so is preferable. Another, yet more powerful correction was proposed by [77]. However, this requires independent p_i to be correct, like the Šidàk correction. Hence, this correction is probably unreasonable in many applications, and should be used with caution.

The false discovery rate

For high-dimensional data and small sample sizes, it is often the case that all known procedures for controlling the FWER have very low power, typically yielding no significant tests at reasonable levels [159]. This is not primarily because the above correction methods are inefficient. Rather, it is an unavoidable consequence of performing tens of thousands of tests with only a handful of observations. In typical genome-wide

data, there is simply too much multiplicity to be able to guarantee that no single error is made during inference, as the FWER requires.

Luckily, in most situations one is willing to settle for something less stringent than FWER control. In genome-wide biology, data analysis is often explorative, and we expect that findings must be validated in follow-up experiments. Therefore, a small proportion of errors in a selected set of features is typically acceptable. If we thus settle for a type of control where the *proportion* of errors among the list of findings is kept below some given threshold, then we obtain the *false discovery rate*.

Definition 4.6. *For a given feature selection method $S = \Phi(Z^{(1:l)})$ and a set of null hypotheses $H_0^1, \ldots, H_0^n$, the false discovery rate (FDR) is defined as*

$$\mathrm{FDR} = \mathbb{E}\left[\frac{\left|S \cap \{i : H_0^i\}\right|}{|S|} \,\middle|\, |S| > 0\right] P\left(|S| > 0\right). \tag{4.5}$$

This error measure was introduced by Benjamini and Hochberg [12] and has since then become something of a *de facto* standard for genome-wide data analysis. While (4.5) may look complicated, it basically represents the expected fraction of false positives in the set S selected by the method $\Phi(Z^{(1:l)})$. The conditioning on the event $|S| > 0$ is simply needed to avoid division by zero, and the factor $P\left(|S| > 0\right)$ is a technicality required to obtain a quantity which is possible to control (bound from above) [12]. Benjamini and Hochberg also introduced a simple correction method

$$p'_{(i)} = \min_{j=i,\ldots,n} \left(\min\{np_{(j)}/j\}\right),$$

which they proved controls the FDR under the assumption that the tests ϕ_i corresponding to the true null hypotheses are independent. Note that this assumption is substantially weaker than complete independence of *all* ϕ_i.

A variant of the FDR was later proposed by Storey [161], who argued that one is really only interested in preventing false discoveries *conditional* on the event $|S| > 0$. Consequently, Storey suggested the factor $P\left(|S| > 0\right)$ should be dropped from (4.5). The result is called the *positive false discovery rate* (pPDR),

$$\mathrm{pFDR} = \mathbb{E}\left[\frac{\left|S \cap \{i : H_0^i\}\right|}{|S|} \,\middle|\, |S| > 0\right]$$

While seemingly similar to the FDR, the pFDR has different properties. As noted already by Benjamini and Hochberg [12], the conditioning makes it impossible to actually *control* the pFDR; that is, we cannot obtain an upper bound on this quantity. However, under some assumptions we can still *estimate* the pFDR from data in a consistent, unbiased way [161].

While the FDR was devised in a traditional hypothesis testing setting, the pFDR has intimate connections with Bayesian hypothesis testing [162]. Under a Bayesian framework, it can be shown that the pFDR equals the Bayesian posterior probability that a rejected hypothesis H_0^i is true,

$$\text{pFDR} = \pi\left(H_0^i \,|\, \phi_i = 1\right).$$

Assuming that we have observed a p-values p_i for each test ϕ_i, Storey defines the *q-value* (a pFDR counterpart of the traditional p-value) to be the posterior probability of H_0

$$q_i = \pi\left(H_0^i \,|\, p_i\right). \tag{4.6}$$

The q-values may be viewed as "local" false discovery rates for each feature X_i. A procedure for estimating q-values is given in Storey and Tibshirani [163].

The FDR paradigm, and in particular Bayesian methods for FDR estimation have become very popular recently, particularly in microarray analysis [5]. This is understandable since they appear to be more powerful than other methods (rendering more significant findings). However, lately some criticism concerning the validity of this Bayesian approach has arisen. Mainly, the problem with most FDR estimation methods is that they often require independent (or at least weakly dependent) test statistics to be correct. As genome-wide data often display very correlated features, this may be a serious problem, giving rise to highly variable and sometimes even systematically optimistic FDR estimates [96]. It thus appears that there is reason for caution also in this paradigm. Any solution to these problems — short of reverting back to more stringent methods that do not impose independence assumptions — is yet to be found.

4.1.3 VARIABLE RANKING

In addition to the above, several measures of marginal dependence have been suggested in machine learning literature for the purpose of ranking (ordering) features according to strength of the dependency on Y [67].

These methods typically do not perform statistical tests, but merely compute a measure of dependence. Examples include the signal-to-noise ratio [62], the Gini index [141] and information-theoretic measures such as mutual information [69, ch. 5]. The distributions of these measures are often unknown, but since the ordering is the primary interest here, this is not an issue.

Typically, the ordered list obtained from the particular ranking criterion chosen is then used to estimate a set of predictive features. This is usually done by estimating the risks of a number of predictors $g_S = I_S(z_S^{(1:l)})$, where S is the top K features from the ranked list, and finally choosing the set S that minimizes $R(g_S)$ (Figure 4.1). As always when dealing with empirical risk estimates, this method is prone to some amount of over-fitting (section 2.7.1), but since the possible "candidate" sets S are quite constrained, this over-fitting is hopefully not that severe (see also Chapter 5).

Since this method of choosing $|S|$ involves an inducer, variable ranking is not a "pure" filter method — this is an example where the distinction is somewhat blurred. Since risk estimation is involved, the target set of a variable ranking method must be some subset of $S^\dagger \cap \{i : Y \not\perp X_i \,|\, \emptyset\}$. The exact goal depends on the nature of the dependence measure used. See Figure 4.1.

Recursive Feature Elimination (RFE) is a variation on the variable ranking theme introduced by [68]. Here, an ordering of features is computed, whereafter a given fraction of the worst-ranked features are removed; the ordering is then re-computed for the remaining features, and the process is repeated. RFE is intended for multivariate criteria functions, for which the ordering is a function of all features. Also here, a risk estimate is typically used to select the final feature set [68].

4.1.4 Multivariate filters

Multivariate filters attempt to take into account dependencies between the X_i. This is a very difficult problem in general, so most methods simplify the situation by various heuristics and restrictions to particular domains $\mathcal{X} \times \mathcal{Y}$. Unfortunately, the heuristics are often difficult to translate to statistical assumptions, so that it is unclear precisely for which problems each of these methods is appropriate.

Focus

Focus is one of the earliest multivariate algorithms, devised for binary data ($X_i, Y \in \{0,1\}$), introduced by Almuallim and Dietterich [6]. It was designed to solve the "min-features" problem (Section 3.2.1), which was introduced in the same paper. Almuallim and Dietterich state this problem as follows.

Definition 4.7 (Min-Features)**.** *Given a set of examples* $(x^{(i)}, y^{(i)})$*, find the smallest set* S *such that, for all* (i,j) *with* $y^{(i)} \neq y^{(j)}$*, it holds that* $x_S^{(i)} \neq x_S^{(j)}$*.*

In other words, the min-features problem is to find a minimal feature set S which is able to separate the classes on the *training* data. If the training set is inseparable, no solution exists. This problem is known to be NP-complete [36], and all known algorithms (including Focus) have exponential complexity.

The Focus algorithm performs an exhaustive search over all subsets of size $k = 1, 2, \ldots$ until a solution is found (assuming that one exists). If the minimal set has size k, then the algorithm complexity is $\mathcal{O}(l(2n)^k)$ [6]. This is polynomial in the dimension n, but tractable only for small k. Focus is known to be problematic when data is noisy, and is prone to over-fitting: as Kohavi and John [97] pointed out, if the social security number of a person is included as a feature, Focus will happily select that feature as predictive of any target variable.

In fact, Focus is motivated by a more deterministic view of data, which is rather incompatible with the statistical data model adopted in this thesis. Under a statistical model with noisy data, Focus will fail eventually as sample size increases, since the probability of inseparable training data will converge to 1. For the same reason, Focus is not a well-defined inducer in the sense of definition 2.14, since some $z^{(1:l)}$ cannot be mapped to any set S. Hence, the target set of Focus cannot be determined exactly; intuitively, one might expect that it estimates some subset of $S^{\ddagger}$. I have indicated this in Figure 4.1 by a dashed line.

Relief

Relief is a multivariate method for classification problems introduced by Kira and Rendell [94]. While it was first introduced based on heuristic considerations, Kononenko [102] and Robnik-Sikonja and Kononenko [147] later provided more a detailed analysis which identified some prop-

erties of the method. I describe the two-class case here; the generalization to $|\mathcal{Y}| > 2$ is generally done by transformation to a series of two-class problems [94].

Relief is derived from the nearest neighbor classifier, described in Section 2.6.2. For each sample $x^{(i)}$, define the *nearest hit* as the nearest point of the same class,

$$h_i = \arg \min_{j: y^{(j)} = y^{(i)}} d(x^{(i)}, x^{(j)}).$$

Similarly, define the *nearest miss*

$$m_i = \arg \min_{j: y^{(j)} \neq y^{(i)}} d(x^{(i)}, x^{(j)}).$$

Here d is some pre-specified distance metric on $\mathcal{X}$, usually taken to be the Euclidean distance. We then compute a measure for each feature X_j

$$\hat{c}(j) = \frac{1}{l} \sum_{i=1}^{l} (|x_j^{(i)} - x_j^{(m_i)}| - |x_j^{(i)} - x_j^{(h_i)}|).$$

Intuitively, $\hat{c}(j)$ should be positive when the class-conditional distributions $f(x_j \,|\, Y = 1)$ and $f(x_j \,|\, Y = -1)$ differ. Further, because h_i, m_i are defined based on the distance on the full space $\mathcal{X}$, it may be possible to detect multivariate dependencies as well. Thus, it seems that the target set of Relief is (a subset of) S^A [97]. However, to my knowledge no theoretical results of its correctness are known. As with Focus, I have indicated this in Figure 4.1 by a dashed line.

Markov boundary methods

Several authors have proposed methods for feature selection which aim to discover the Markov boundary M^* of Y. All of these methods assume that the data distribution is faithful to a Bayesian network (see Section 2.2).

The *Incremental Association Markov Blanket* (IAMB) algorithm computes an estimate M of the Markov boundary in two stages. In the first stage, IAMB tests for the dependency $Y \not\perp X_i \,|\, \emptyset$ and heuristically accepts the strongest significant association as a member of M; it then tests the remaining features for $Y \not\perp X_i \,|\, X_M$, and so on, always accepting the one strongest association in each iteration, until no more significant dependencies are found. In the second stage, for each each

$i \in M$, IAMB tests $Y \perp X_i \,|\, X_{M\setminus\{i\}}$ and removes any X_i that turns out to be independent.

Tsamardinos and Aliferis [176] proved that IAMB is consistent ($M \xrightarrow{P} M^*$ as $l \to \infty$), provided that the independence tests used are consistent. However, the power of the conditional tests $Y \perp X_i \,|\, X_{M\setminus\{i\}}$, can be shown to decrease exponentially with the size of the conditioning set $M \setminus \{i\}$ [175]. Therefore, the required training set size l increases exponentially in $|M|$, resulting in small power for limited sample sizes.

A potentially more powerful approach is the Max-Min Markov Blanket (MMMB) algorithm, due to Tsamardinos et al. [175]. The gain in power is due to a more efficient strategy which uses smaller conditioning sets by taking advantage of the local structure of the Bayesian network. MMMB was claimed to be consistent by Tsamardinos et al. [175], but this was later refuted by Peña et al. [131], who found errors in the original proofs. Peña et al. also provided a corrected version named *AlgorithmMB* for which consistency was proven, while retaining better power than IAMB. Thus, both IAMB and AlgorithmMB estimate M^* in a consistent fashion, while AlgorithmMB has better power.

4.1.5 MULTIVARIATE SEARCH METHODS

A common multivariate type of filter method is based on the subset search approach. Here, a "criterion" function $c(S)$ is chosen that for a given subset $S \subseteq V_n$ is supposed to estimate the strength of the dependence $Y \not\perp X_S$. Then, a search procedure is used to maximize $c(S)$ over the possible subsets of V_n. A bewildering array of multivariate criteria exist for measuring a dependence $Y \not\perp X_S$ for a given feature set S. Several multivariate statistical tests can be used; some are reviewed in Section 4.4. For two-class problems, several measures of distance between the class-conditional densities $f(x_S \,|\, Y = 1)$ and $f(x_S \,|\, Y = -1)$ can be used. Examples include the Mahalanobis distance and the Jeffreys' and Kullback-Leibler divergences [104]. A thorough discussion of these is found in [37, ch. 2]. Information-theoretic methods such as the mutual information between X_S and Y can also be used [67, ch. 5]. From the discussion in sections 3.1.1 and 3.1.2 it can be seen that, depending on the divergence measure, the target set of subset search is M^* or S^*. However, a common problem with the more ambitious, non-parametric divergence measures is that they are difficult to estimate reliably from limited data.

Which search procedure to use has been the subject of much research

[85]. Clearly, exhaustive subset search is not feasible for moderately large dimensions, as the number of subsets of V_n grows as 2^n. Greedy search procedures either start with $S = \emptyset$ and iteratively adds the elements that give maximal increase in $c(S)$ ("forward" search), or start with $S = V_n$ and remove elements that give minimal decrease in $c(S)$ ("backward" search). These and more elaborate variants with "back-tracking" mechanisms are discussed by Pudil and Novovičová [140].

Often, the search procedures presuppose an optimal feature set size $|S|$. In this case, the target set is $S^\ddagger$. The size of S may be dictated by practical considerations such as the computational complexity of a subsequent predictor learning procedure, or one may desire to keep S small to be interpretable. In this case, the branch-and-bound algorithm by Narendra and Fukunaga [121] is known to be optimal when the criterion function is *monotone*, that is, satisfies $c(S) < c(S') \iff S \subset S'$. This assumption is rather strong however. A number of variations on this theme, some of which are claimed to be robust against violations of monotonicity, is surveyed by Kudo and Sklansky [103].

A general problem with all subset search strategies is that they tend to ignore the fact that for finite samples, the criterion function $c(S)$ is necessarily an *estimate* which exhibits random variation. Therefore, searching for its optimum over different S inevitably causes over-fitting [145]. More about this in the next section.

4.2 Wrapper methods

Wrapper methods, or "wrappers" for short, were first introduced by Kohavi and John [97]. As with multivariate filters, wrappers make use of subset search methods, but here the criterion function is based on an empirical risk estimate for a particular inducer I, that is, $c(S) = 1 - \hat{R}(I_S(z_S^{(1:l)})$. Thus, wrapper methods perform a kind of empirical risk minimization (see Section 2.6.1) over the the function class $\mathcal{G} = \{g_S : S \subseteq V_n\}$. Therefore, wrapper methods attempt to estimate $S^\dagger$. Often, a cross-validation estimate of $\hat{R}$ is used, although theoretic risk bounds [89, 181] are also possible.

Like any methods that attempts to minimize an empirical risk estimate, the wrapper approach suffers from over-fitting problems [98]. Strictly speaking, the estimate $\hat{R}(g)$ is biased downwards (over-optimistic) whenever two or more different predictors g are tried on the same (or even dependent) data sets. Strictly speaking, the search scheme will begin to

over-fit (slightly) already by the second evaluation of the risk estimate. Thus, minimizing the number of feature sets tested by the search procedure is critical, not only for computational reasons, but also to avoid over-fitting [145]. On the other hand, the search procedure must be allowed to explore enough feature sets to be able to discover true minima of $R(g)$.

Generally speaking, the wrapper approach seems to have declined in popularity in recent years, although it is still in use. The reason is probably that most subset search methods are computationally feasible only for problems of moderate dimensionality ($n \approx 10 \ldots 100$), while in many recent applications of feature selection, including genome-wide data analysis, dimensionality is much higher [67].

4.3 EMBEDDED METHODS

A different approach to feature selection originates in the min-features problem described in Section 3.2.1. This can be formulated as an optimization problem if one approximates the unobservable expected risk with a risk estimate $\hat{R}$. One then obtains

$$\min_{S \subseteq X} \hat{R}(I(z_S^{(i:l)})) + \lambda |S|. \tag{4.7}$$

This can be interpreted as a form of regularization, with $|S|$ being the regularizer (see Section 2.7.2). From this viewpoint, feature selection regularizes predictor estimation by constraining the dimension of the input space.

Some predictors are parameterized in a way that immediately reveals how each feature influences the predictions. The most obvious case is perhaps linear classifiers $g_w(x) = \text{sign}(w^T x)$ or linear regression $g_w(x) = w^T x$, where each "weight" w_i corresponds to feature X_i. For such predictors, Equation (4.7) can be written as

$$\min_{w} \hat{R}(g_w) + \lambda \|w\|_0, \tag{4.8}$$

where the "zero norm" $\|w\|_0 = |\{i : w_i \neq 0\}|$ simply counts the number of non-zero elements [186]. (Although strictly speaking $\|w\|_p$ is not a norm for $p < 1$, the term is motivated by the fact that the L_p norm $\|w\|_p$ approaches $\|w\|_0$ as $p \to 0$.) In this case, minimizing over w both induces a classifier g_w and a set of features $\{i : w_i \neq 0\}$. This is referred to as "embedded" feature selection, since feature selection happens "inside"

the inducer and is therefore inseparable from it. A key fact here is that, due to the choice of regularizer, the solution w to the above problem is *sparse*, that is, contain few non-zero elements.

4.3.1 Sparse linear predictors

The computational complexity of problem (4.8) is known to be exponential [8], and is therefore intractable in practise. Consequently, the L_0 norm $\|w\|_0$ is often approximated by an L_1 norm such as $\|w\|_1$ to obtain feasible optimization problems. In principle, any L_p norm with $p \leq 1$ will yield a sparse solution, but the L_1 norm occupies a unique position since it is the only L_p norm which gives both sparse solutions and a convex optimization problem, so that a global optimum can be guaranteed [134]. Thus, the L_1 norm is a popular choice, and several authors have explored different variants of L_1 regularized linear problems for feature selection; see for example Bi et al. [15], Bradley and Mangasarian [19], Fung and Mangasarian [57], Mika et al. [118]. Non-convex ($p < 1$) approximations to (4.8) are certainly also possible, although the optimization problems consequently become more difficult. A comprehensive treatment of this topic is given by Weston et al. [186].

Sparse linear methods can also be derived within the Bayesian paradigm. The Relevance Vector Machine due to Tipping [172] is a Bayesian approach applicable to both regression and classification problems, which yields sparse solutions θ that can be exploited for feature selection. Also, the Kernel Fisher Discriminant [117] technique with L_1 regularization can be interpreted as a Bayesian regression model with a Laplace prior [118], as does the Lasso regression method [170]. See also Example 2.14.

At first glance, these regularization methods may appear to have a built-in mechanisms for choosing the size of the selected feature set S. It is true that for a fixed $\lambda > 0$, the set S is completely determined. In this case, embedded methods employ the min-features bias, so that the target set is $S^{\ddagger}_{\lambda}$. In practise however, the λ parameter is usually selected based on a cross-validation risk estimate for the resulting predictor. In this case, the target set is $S^{\dagger}$ (Figure 4.1).

4.3.2 Non-linear methods

For non-linear classifiers, the above approach breaks down since there is no longer a natural mapping between the features X_i and the parameters θ. This is a problem especially for kernel methods, where the parameter

vector θ is implicit and may be very high-dimensional or even infinite (see section 2.6.3). A solution for kernel methods is to define a "scaled" kernel

$$K_\theta(x^{(i)}, x^{(j)}) = K_\theta(\theta^T x^{(i)}, \theta^T x^{(j)})$$

which essentially evaluates any chosen kernel K with feature X_i scaled by a factor θ_i [185]. One may then include an optimization step for θ in the inducer, thus effectively estimating both the classifier $\hat{g}$ and the scaling parameters θ_i simultaneously. For support vector machines, it turns out that an efficient gradient search procedure can be used for this purpose. Unfortunately, the resulting optimization problem may be severely non-convex, so there are in general no guarantees for the accuracy of the estimate [23].

A recent alternative approach which avoids non-convex optimization is the *feature vector machine* proposed by Li et al. [110]. This method is essentially a modification of Lasso regression (section 2.7.2) where a kernel function K is applied to the *feature vectors* $x_j^{(1:l)}$ rather than to the sample vectors $x_{1:n}^{(i)}$. While method was proposed for regression problems (continuous Y), it may be adapted to classification problems as well, for example through logistic regression, as with the relevance vector machine [172].

4.4 FEATURE EXTRACTION AND GENE SET TESTING METHODS

In addition the feature selection methods described so far, there are also many methods that perform feature *extraction*, that is, transform the features $X_1, \ldots, X_n$ into new ("extracted") features $X'_1, \ldots, X'_m$, typically with $m \ll n$ (Section 3.4). For example, dimensionality reduction methods such as Principal Component Analysis [73, pp. 485] can be used prior to predictor induction [136]. Dimensionality reduction techniques are also closely related to data clustering [39, 191]. Also, kernel methods (Section 2.6.3) perform a kind of feature extraction in that the the training data $x^{(1:l)}$ is implicitly mapped into the l-dimensional subspace of $\mathcal{H}$ spanned by $\phi(x^{(1)}), \ldots, \phi(x^{(l)})$, and we typically have $l \ll n$. For more information on feature extraction, see for example Liu and Motoda [111].

I here briefly consider a related class of methods of particular interest to applications in genome-wide data. These are called *gene set testing*

methods since they originate in gene expression data analysis, although one could easily conceive of applications for other data types, so that *feature set testing* may be a more appropriate term. For a given set of (disjoint) feature sets $\{S_k \subseteq V_n\}$, we are here interested in multivariate tests for the null hypotheses

$$H_0^k : X_{S_k} \perp Y \tag{4.9}$$

for each gene set S_k. Many options for such multivariate testing are available. Under the assumption of multivariate Gaussian X, a method based on Hotelling's T^2 distribution (a generalization of Student's t [81]) is described by Lu et al. [112]. More general methods based on correlations in feature space are described by Gretton et al. [64] and Borgwardt et al. [17].

The currently most popular method for gene set testing is probably Gene Set Enrichment Analysis (GSEA), introduced by Mootha et al. [119]. This method can utilize any statistic $T_i = T(X_i, Y)$ to order the features X_i. GSEA then tests for an enrichment of high-ranking features in set S_k by a permutation test based on a Kolmogorov-Smirnov statistic [119, 165]. Note that this method performs a "competitive" test, in that each set S_k is evaluated against all other sets. The null hypothesis is thus not the same as (4.9). A good discussion on this topic and a comprehensive review of gene set testing methods is given by Goeman and Buhlmann [61].

4.5 SUMMARY

Feature selection methods come in all shapes and sizes. For newcomers, the multitude of available methods can be daunting. In order to bring some structure to this large collection of methods and to better understand their relations to each other, I have in this chapter reviewed a number of methods of different types and attempted to relate them to the feature selection problems defined in the previous chapter.

In machine learning, feature selection methods are typically divided into *filter* (Section 4.1), *wrapper* (Section 4.2), and more recently *embedded* methods (Section 4.3). This division largely depends on the algorithmic form of each method (*e.g.*, filters can be defined as "algorithms that do not invoke the inducer procedure") rather than statistical arguments. Consequently, I find that members of each class target various feature selection problems. The details are often important — two methods cannot

be concluded to target the same problem because the algorithms appear similar. For example, the "target set" of a given variable ranking methods (Section 4.1.3) may vary considerably (from S^A to $S^\ddagger$) depending of the precise criterion used to choose the number of selected features.

Figure 4.1 provides an overview of this chapter, where the various methods are fitted into the framework described in Chapter 3. This defines a more fine-grained structure on the collection of feature selection methods, so that one may easier choose a method appropriate for a given task, *i.e.*, for one of the feature sets in Figure 4.1. This again emphasizes the importance of clearly defining the data model and the feature selection problem *before* choosing a method.

In this chapter I have also described a number of concepts relating to hypothesis testing (Section 4.1.1) and the multiple testing problem (Section 4.1.2) which is an important issue for high-dimensional data. These also perform a type of feature selection, and in contrast to most methods from the machine learning field they provide control over error rates. In summary, I hope that this chapter may be useful to the reader as a reference and guide to the extensive literature on feature selection methods.

5

A benchmark study

Many of the features selection methods described in the previous chapters have been designed for and tested with data distributions of moderate dimensionality, on the order $10-100$ features. It is therefore not clear how the performance of these methods translates to modern applications such as microarray data, where dimension is at least an order of magnitude higher and sample size is often very small. Moreover, it is not clear whether feature selection is advantageous in this domain compared with other regularization mechanisms employed by modern inducers such as the support vector machine. Finally, the accuracy of feature selection methods with respect to the features selected (*i.e.*, feature error rates) has not been systematically evaluated.

To assess these questions, in this chapter I present a "benchmark" study, a systematic evaluation based on simulated data of a number of feature selection methods coupled with the linear support vector machine (SVM) as a predictor. We designed the simulations used to be as representative as possible for the data distributions encountered in microarray data analysis. We chose the support vector machine as the "reference" classifier for this study since it is very popular for microarray classification problems [58, 68].

To my knowledge, this study is the first systematic evaluation of feature set accuracy, and the first feature selection evaluation to simulate high-dimensional data of the order encountered in microarray data analysis (up to 5,000 dimensions are considered). Most of the material in this

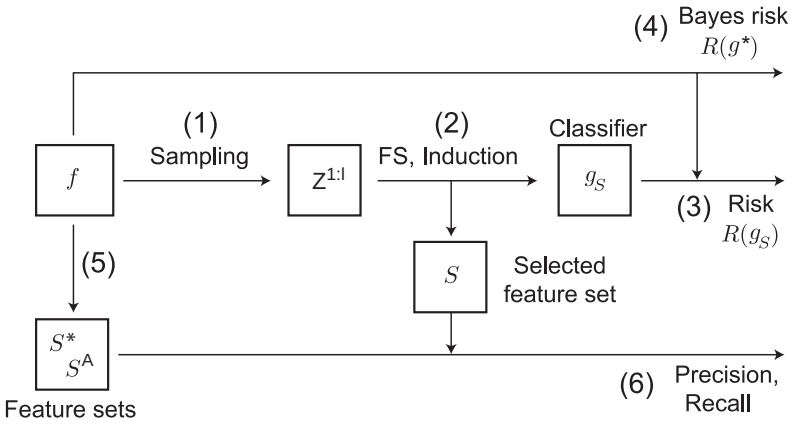

Figure 5.1: A schematic view of the evaluation process.

chapter can also be found in Nilsson et al. [127].

5.1 Evaluation system

An overview of our simulation and evaluation procedure is shown in Figure 5.1. For a given data distribution $f(x, y)$, we first take a sample $Z^{(1:l)}$ to be used as training data (step 1). We then perform feature selection and classifier (SVM) induction (step 2) as discussed in section 5.2. We calculate classifier error probabilities (steps 3,4) by direct numerical evaluation of the risk functional $R(g)$, which is feasible here since the data distribution is known (see below). For assessing the accuracy of selected feature sets, we first calculate S^A and $S^\dagger$ for the given distribution (step 5; see below) and then measure the precision and recall (step 6) of the selected feature sets as follows.

Definition 5.1. *For a selected feature set S, we define the precision and recall with respect to a "true" set T as*

$$\text{Precision}(S) = \frac{|S \cap T|}{|S|} \quad \text{Recall}(S) = \frac{|S \cap T|}{|T|}$$

In our evaluations, we consider as the "true" feature set T either S^A (all

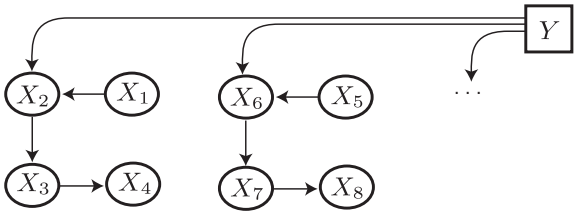

Figure 5.2: A Bayesian network representation of the distribution used in the simulations in this chapter. For $m < n$ relevant features, there are $m/4$ of the connected 4-blocks of features.

relevant features) or $S^\dagger$ (the optimal feature set).

We designed a Bayesian network (see Section 2.2) to represent a high-dimensional data distribution. We used a "block structure" consisting of small sub-networks of 4 nodes, chosen to contain both (i) features which are in S^* but are marginally irrelevant, *e.g.*, $X_i \perp Y \,|\, \emptyset$, and (ii) features which are relevant by Definition 3.11 but yet are not in S^*. This graph structure is given in Figure 5.2. We then simply repeated the number of "blocks" to obtain any desired number of relevant features m. The local (conditional) distributions of the Bayesian network was chosen to achieve reasonable expected risks for the SVM, as follows.

$$\begin{aligned} f(x_1) &= N(x_1 \,|\, 0, 3^2) \\ f(x_2 \,|\, x_1, y) &= N(x_2 \,|\, y - x_1, 1) \\ f(x_3 \,|\, x_2) &= N(x_3 \,|\, x_2, 1) \\ f(x_4 \,|\, x_3) &= N(x_4 \,|\, x_3, 1) \end{aligned}$$

A number of irrelevant features distributed as $N(x_i \,|\, 0, 1)$ were then chosen, for a total dimension of $n > m$. We set $p(y) = 1/2$ throughout. For this data distribution and the linear SVM, we found in preliminary simulations that the optimal set $S^\dagger$ coincides with the Bayes-relevant features S^* regardless of dimension. I will therefore use these two feature sets interchangeably in this chapter.

Since the $f(x, y)$ defined above is two-class Gaussian, the risk $R(g)$ and Bayes risk $R(g^*)$ can be calculated as described in Example 2.9. Further, since the structure of the Bayesian network yields a very sparse precision matrix Σ^{-1}, these quantities are efficiently calculated using sparse linear algebra for large n even when the full matrix Σ is prohibitively large.

Method	Type	Output	Ref.
PC	Filter	Ranking	[67]
WR	Embedded	Ranking	[68]
RFE	Wrapper	Ranking	[68]
LP-SVM	Embedded	Set	[57]
AROM	Wrapper	Set	[186]
IAMB*	Filter	Set	[176]
R2W2*	Embedded	Set	[23]

Table 5.1: Feature selection methods tested in this study. The two methods marked by an asterisk were excluded early in the study due to very high computational complexity.

For evaluating the expected performance of a given algorithm and obtaining statistical confidence measures, the simulation process is repeated a number of times, yielding a number of independent observations of each error measure. Standard non-parametric statistical methods can then be used to obtain correct confidence intervals for each error measure [105]. In preliminary experiments we found that confidence intervals were sufficiently narrow for our purposes at $l = 100$ samples, and hence fixed $l = 100$ for all subsequent evaluations.

5.2 Feature selection methods tested

We initially chose seven well-known feature selection methods for evaluation. Three of these were variable ranking methods (Section 4.1.3), two were filter methods (Section 4.1), two were wrappers (Section 4.2) and three were embedded (Section 4.3). A summary is found in Table 5.1. However, the Radius-Margin Descent method (R2W2) [23] and the Incremental Associative Markov Blanket (IAMB) [176] were unfortunately too computationally demanding in high dimensions, wherefore they were excluded from further analysis. We did not consider any feature extraction methods (Section 3.4). The remaining methods are described briefly below; for details, please see the respective section of Chapter 4.

Pearson correlation (PC) is a simple filter method, ranking features by the Pearson correlation with the target Y. Although PC is intended to be used with continuous Y, it was included since it is commonly used also for discrete Y in feature selection literature

[67, 73].

SVM Weight Ranking This method ranks features by the absolute value of the corresponding weight of a linear SVM hyperplane. See Section 4.3.

Recursive Feature Elimination (RFE) This method was proposed by Guyon et al. [68]. For a given ranking method, in each iteration the RFE algorithm removes a fraction of the lowest-ranking features to produce a sequence of feature sets $S_1 \supset S_2 \supset S_3, \ldots$, and evaluates a goodness criterion c for each S_i (typically a classifier risk estimate). The set that maximizes c is then taken as the final feature set.

Linear Programming-SVM This method induces a sparse linear classifier and simply selects the features with nonzero weights. See Section 4.3.1. The regularization parameter was set to 1 throughout.

Approximation of the nero norm (AROM) This method approximates a "zero norm"-regularized linear classifier and selects the features with nonzero weights (Section 4.3.1). We used the L_2 version of the algorithm, which amounts to learning a linear SVM in each iteration, as described by Weston et al. [186].

Throughout, we used a linear SVM [18, 32] as the inducer I_S for each selected feature set S. Briefly, the SVM estimates the hyperplane normal vector by solving the optimization problem

$$\min_w \sum_i (1 - w^T x^{(i)} y^{(i)}) + \lambda w^T w.$$

This can be written equivalently as

$$\begin{aligned} \min_w \quad & w^T w + C \sum_i \xi^{(i)} \\ \text{s.t.} \quad & y^{(i)} w^T x^{(i)} \geq \xi^{(i)} \\ & \xi^{(i)} \geq 0 \end{aligned}$$

where $C = 1/\lambda$ is the regularization parameter. See also Example 2.13. A comprehensive introduction is found in, *e.g.*, Christianini and Shawe-Taylor [27]. For SVM training, we used an in-house Java implementation based on the Generalized Sequential Minimal Optimization (GSMO) algorithm [93].

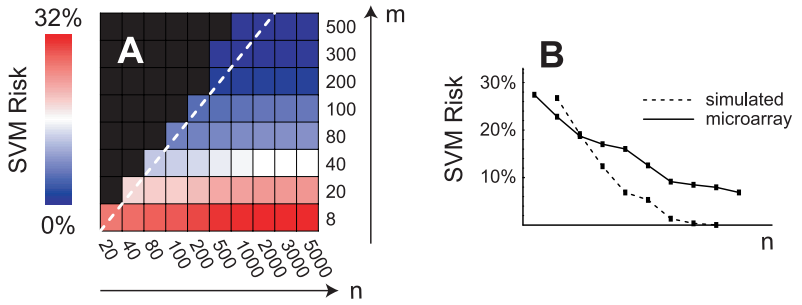

Figure 5.3: A: Plot of expected SVM risk $\rho(I)$ *vs.* number of relevant features m and total number of features n. **B:** Dotted line, plot of SVM risk *vs.* n for simulations, corresponding to the dotted diagonal in (A). Solid line, SVM risk *vs.* n for microarray data. Here m is unknown but proportional to n.

5.3 Results

By the design of the data distribution, features $X_1, \ldots, X_m$ were relevant to Y, while $X_{m+1}, \ldots, X_n$ were irrelevant. Of the m relevant features, $m/2$ were in the optimal feature set; further, half these $(m/4)$ were *marginally* independent of Y and thus undetectable by univariate filter methods like PC. We sampled $l = 100$ training data points from this distribution and normalized data to zero mean and unit variance before applying each method, following standard practise for SVMs [68]. The key parameters to the "difficulty" of the learning problems represented by this data distribution are m and n. We chose a parameter grid $8 \leq m \leq 500$, $20 \leq n \leq 5000$, with values evenly spaced on a logarithmic scale (Figure 5.3A). For each (m, n) we repeated the simulations 100 times.

5.3.1 Robustness against irrelevant features

The expected risk $\rho(I) = \mathbb{E}\left[R(I(Z^{(1:l)}))\right]$ for the SVM without feature selection on the (m, n) parameter grid is shown in Figure 5.3A (we set the regularization parameter C to 1; see next section). We find that $\rho(I)$ increases slightly with n, but decreases rapidly with respect to m. Thus, more features is in general better for the SVM: as long as we can obtain a few more relevant features, we can afford to include many irrelevant

ones. Therefore, improving SVM performance by feature selection is very difficult. In particular, a feature selection method must provide very good recall (preserve the relevant features), or SVM performance will degrade quickly.

To validate our results, we also tested the feature selection methods on a microarray data set from prostate cancer biopsies [156]. This data set consists of 12,600 features (each feature corresponding to a microarray probe interrogating a particular gene) and 136 samples. To obtain dimensionality and sample size comparable with our simulations, we first pre-filtered this data by extracting the 5000 features with largest variance and then selected random subsets of sizes $10, \ldots, 5000$ from these. Although m is unknown in this case, in expectation this procedure gives a constant m/n ratio, since we sample features with uniform probability. This roughly corresponds to a diagonal in Figure 5.3A. We used random subsets of $l = 100$ samples for feature selection and classifier induction and estimated $\hat{R}(g)$ by the hold-out error on the remaining 36 samples. This procedure was repeated 300 times for each n. The resulting risk estimate was found to agree qualitatively with our simulations (Figure 5.3B), suggesting that the simulated data distribution is reasonable.

5.3.2 Regularization in high dimensions

The value of the regularization parameter C has been found to strongly impact SVM classification accuracy for low-dimensional data [92]. To study the effect of this parameter in high dimensions, we optimized C over a range $10^{-4}, \ldots, 10^{4}$ for each position (m, n) on our parameter grid. We found that C was no longer important in higher dimensions (Figure 5.4A), as classification accuracy was approximately constant over the tested range of C, regardless of the value of m. At lower dimensions, C did affect classification accuracy, in accordance with the results by Keerthi [92]. We found that $C \approx 1$ provided good performance (Figure 5.4B) at low dimensions, so we fixed $C = 1$ for the remainder of this study.

The same (and even stronger) trend was observed for the microarray data (Figure 5.4C), again suggesting that our high-dimensional simulated distribution is a reasonable evaluation tool. It therefore seems that the regularization parameter C has virtually no impact on classification accuracy in high dimensions. A satisfactory explanation for this phenomenon is yet to be found. I note that somewhat similar results

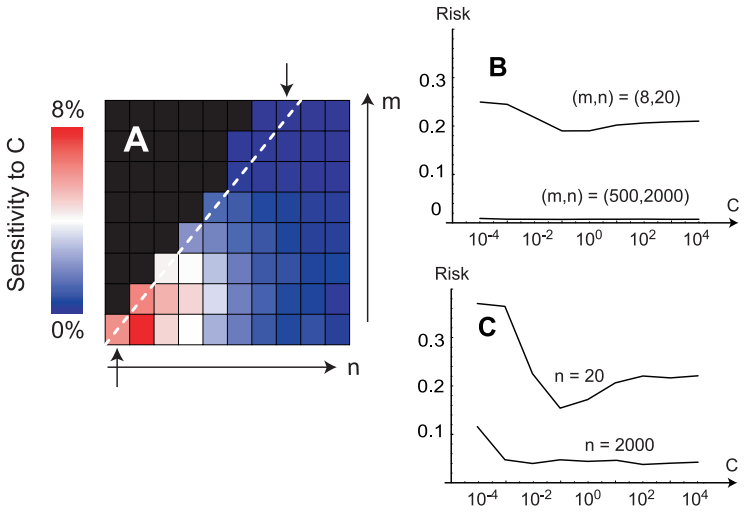

Figure 5.4: Sensitivity to the SVM C-parameter. **A:** Sensitivity defined as $\max_C \hat{R} - \min_C \hat{R}$ plotted against m and n. **B:** For simulated data, detailed plot of $\hat{R}$ against C for the cases (m, n) marked by arrows in (A), roughly corresponding to the cases in (C). **C:** For microarray data, plot of $\hat{R}$ against C for $n = 20$ and $n = 2000$.

have been observed by Hastie et al. [74].

5.3.3 Rankings methods are comparable in high dimensions

Next, we investigated the accuracy of the feature rankings produced by PC, WR and RFE. To address this question without involving the problem of choosing $|S|$ at this stage, we measured precision and recall over the range $|S| = 1, \ldots, n$ and visualized the results using Receiver Operator Characteristic (ROC) curves (Figure 5.5) [73, pp. 277]. We found that RFE outperforms WR in this respect, which in turn outperforms PC, in agreement with Guyon et al. [68]. This was expected, since 1/4 of the relevant features are not detectable by PC. However, these differences diminished with increasing n. At $n = 5000$, the simpler WR method was as accurate as RFE. Thus, more sophisticated method motivated by experiments in lower dimensions [68] may not be as advantageous in high

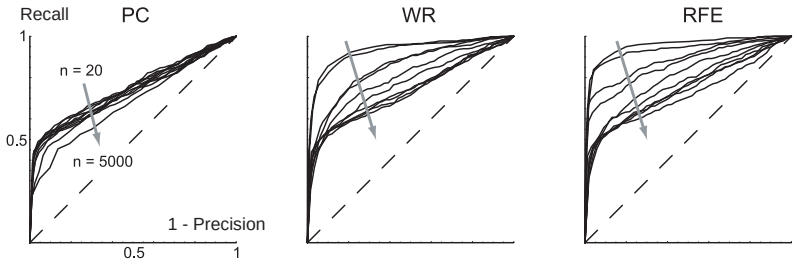

Figure 5.5: ROC-curves for the PC, WR and RFE methods. Here we fixed $m = 8$ relevant features and varied $n = 20, \ldots, 5000$ as indicated by grey arrows. Dashed diagonals indicate expected result for randomly selected features.

dimensions.

5.3.4 Number of selected features increases with dimension

To use ranking methods for feature selection, a critical issue is how to determine $|S|$ (for LPSVM and AROM, this is determined by heuristics that favor small feature sets [57, 186]). For this purpose, we used the radius-margin risk bound [181, section 10.7] as a proxy for $R(g_S)$, and minimized this over a number of possible choices of $|S|$ as described in Section 4.1.3. In preliminary experiments, we also tried the risk bound due to Joachims [89] for this purpose; however, we found the radius-margin bound to be more accurate, and hence chose this alternative for the remainder of the study.

Using this method to determine $|S|$, we found that the ranking methods tend to select more features than AROM or LPSVM (Figure 5.6A). Especially AROM was extremely "sparse" and selected very few features. We also found that $|S|$ tends to *increase* with n. This might seem counter-intuitive at first glance; as n increases, the inference problem becomes more difficult, and one would therefore expect fewer features to be selected. However, this phenomenon can be explained by noting that (i) the ranking problem is harder for larger n, and that (ii) with less accurate rankings (i.e., smaller area-under-curve in Figure 5.5), we will need a larger $|S|$ to include enough informative features to produce an accurate classifier. Conversely, as n decreases and the rankings im-

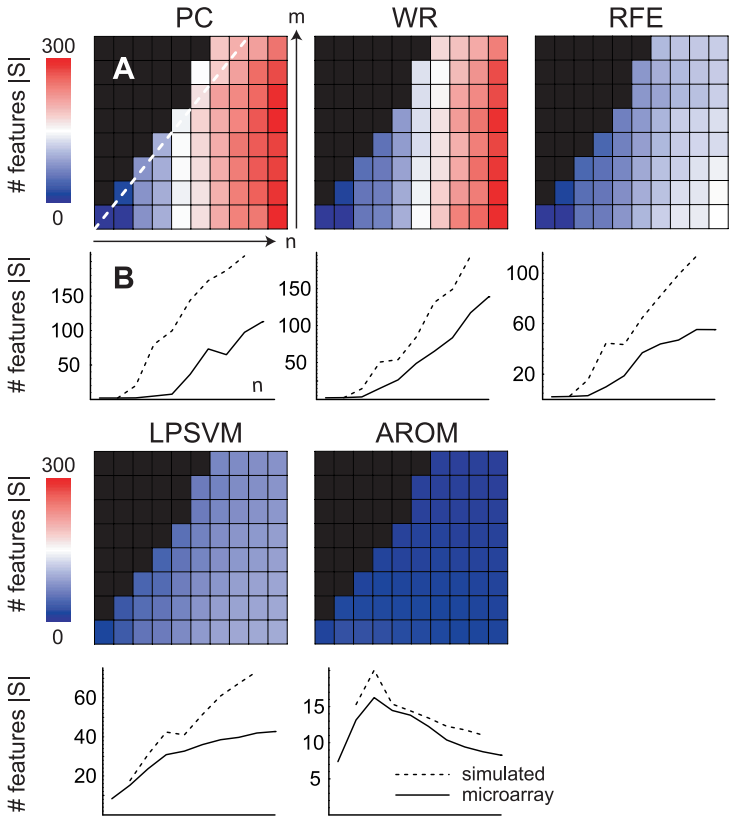

Figure 5.6: A: Number of selected features $|S|$ for each (m, n) for simulated data. All plots are scaled equally to $(0, 300)$. **B:** Number of selected features $|S|$ *vs.* n, corresponding to the dashed diagonal in (A), for simulated and microarray data. Plots are not on equal scales.

prove, more informative feature are "on the top list" and thus a smaller $|S|$ suffices to optimize $R(g_S)$. Accordingly, the PC method which is the least accurate (Figure 5.5) chooses the largest $|S|$, and RFE the smallest. This was also verified for the microarray data (Figure 5.6B).

More surprisingly, there was also a tendency for $|S|$ to *decrease with* m (most evident for RFE). This can be understood in the same fashion: the feature selection problem becomes harder as m decreases, so that the rankings become more inaccurate, and a larger $|S|$ must be used. These results already suggest that by selecting $|S|$ to minimize risk, we obtain methods that attempt to preserve a high recall but sacrifice precision. More about these issues in Section 5.3.6.

LPSVM produced smaller feature sets than RFE, but otherwise exhibited the same tendencies as above. It should be noted that for LPSVM, $|S|$ depends on the regularization parameter C, which we fixed at 1. However, since the risk estimates were quite insensitive to C, it is difficult to choose this parameter in a principled manner.

AROM gave the smallest $|S|$ of all methods, but here the dependence on n was more complicated: $|S|$ was maximal at $n \approx 100$ (similar to the data set used in the original paper [186]) and then decreased for $n > 100$. We are not certain as to the cause of this behavior, but we note that AROM merely guarantees convergence to a *local* maxima of its heuristic (the approximation of the "zero norm"). This might become problematic in higher dimensions, so that a different behavior is observed in different regimes. This underlines the importance of performing simulations in a setting as close as possible to the real application data. Again, the simulation results were consistent with microarray data (Figure 5.6B).

5.3.5 No method improves SVM accuracy

In principle, if the risk estimate (in our case the radius-margin bound) is accurate, then optimizing this estimate over $|S|$ should at least guarantee that ranking methods do not increase classifier risk; in the worst-case scenario, we should reach an optimum at $|S| = n$. Our simulations verified this intuition. In Figure 5.7A, the difference $\rho(I) - \rho(I_S)$ is close to 0 (where I is the SVM classifier without feature selection). We did find $\rho(I_S) > \rho(I)$ at some (m, n) for all ranking methods, but the difference was not substantial. These small discrepancies may be because our procedure favored smaller $|S|$ in cases where $\hat{R}(g_S)$ was constant over a range of $|S|$; thus, a possible improvement to this scheme is to choose a $|S|$ in the middle of such "flat" ranges. Overall, the radius-margin bound

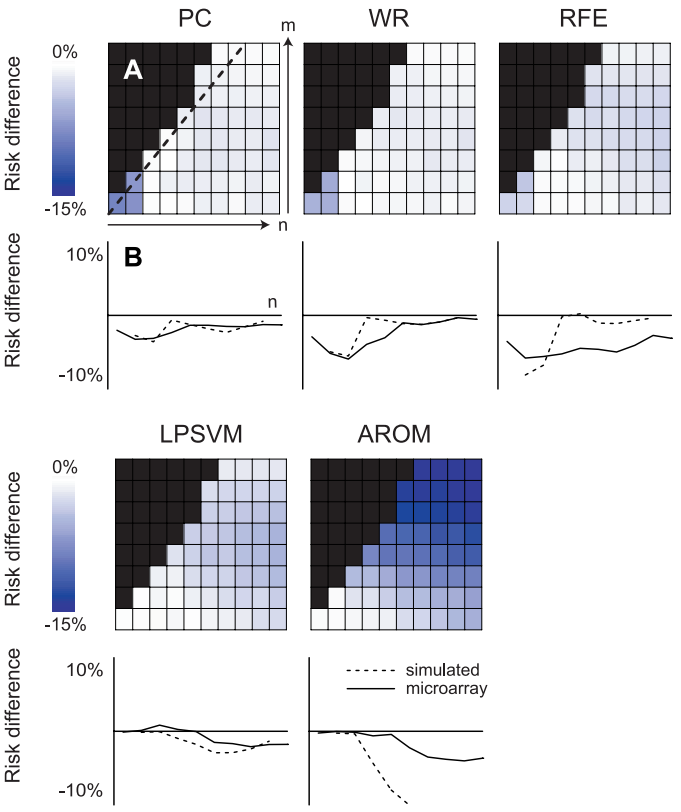

Figure 5.7: **A:** Risk difference $\rho(I) - \rho(I_S)$, using each respective feature selection method to obtain S (negative means worse accuracy with feature selection), simulated data. **B:** Estimated risk *vs.* n, corresponding to the dashed diagonal in (A), for simulated and microarray data.

seems to be accurate. We therefore hold that the results concerning the ranking methods can be attributed to the ranking methods themselves.

LPSVM and AROM gave the best accuracy around $n \approx 100$ (Figure 5.7A,B), which is very close to the data sets used in the original publications [57, 186]. In higher dimensions however, we found that these methods tend to increase the SVM risk. AROM performed particularly bad in this respect, increasing $\rho(I)$ by up to 15%, probably because it insisted on very small feature sets. Results on microarray data were similar (Figure 5.7B) except possibly for RFE, which was less accurate on microarray data. In summary, none of the feature selection methods managed to improve the SVM accuracy.

A reasonable interpretation for this negative finding is that the L_2 norm regularization employed by the SVM is more effective in high dimensions if considering feature selection as an approximate minimization of the L_0 norm of a vector of feature weights, while the SVM minimizes the L_2 norm [134]. From this perspective, our results on simulated and real data imply that in high dimensions, the L_2 norm is simply the better choice. In accordance with this interpretation, the LPSVM method which explicitly minimizes the L_1 norm seems to be closer to SVM performance (Figure 5.7B).

5.3.6 Feature set accuracy

For the simulated data, we measured the accuracy of selected feature sets by precision and recall *vs.* $S^\dagger$ (Figure 5.8) and S^A (Figure 5.9). There are interesting differences between these two feature sets (recall that for our data distribution, $S^\dagger = S^*$ constitutes half of S^A). Concerning recall *vs.* S^A, the best method appears to be PC, followed by WR, RFE, LPSVM, and lastly AROM, in that order. The filter method PC presumably performs best here since it captures all marginally relevant features and does not discard features outside $S^\dagger$. Further, PC selects more features outside $S^\dagger$, since it gives lower recall *vs.* $S^\dagger$; this is presumably because half of $S^\dagger$ is not marginally relevant. In contrast, RFE, LPSVM and AROM have higher recall *vs.* $S^\dagger$ than *vs.* S^A. WR seems to lie somewhere in-between RFE and PC in this respect. These results are in accordance with the analysis in Chapter 4: all of these methods involve some form of risk optimization and therefore target $S^\dagger$. Consequently, they tend to miss — or, avoid — many of the relevant features outside $S^\dagger$. Whether this is a problem or an advantage depends on which problem one wants to solve; hence, we conclude that it is important to carefully define the

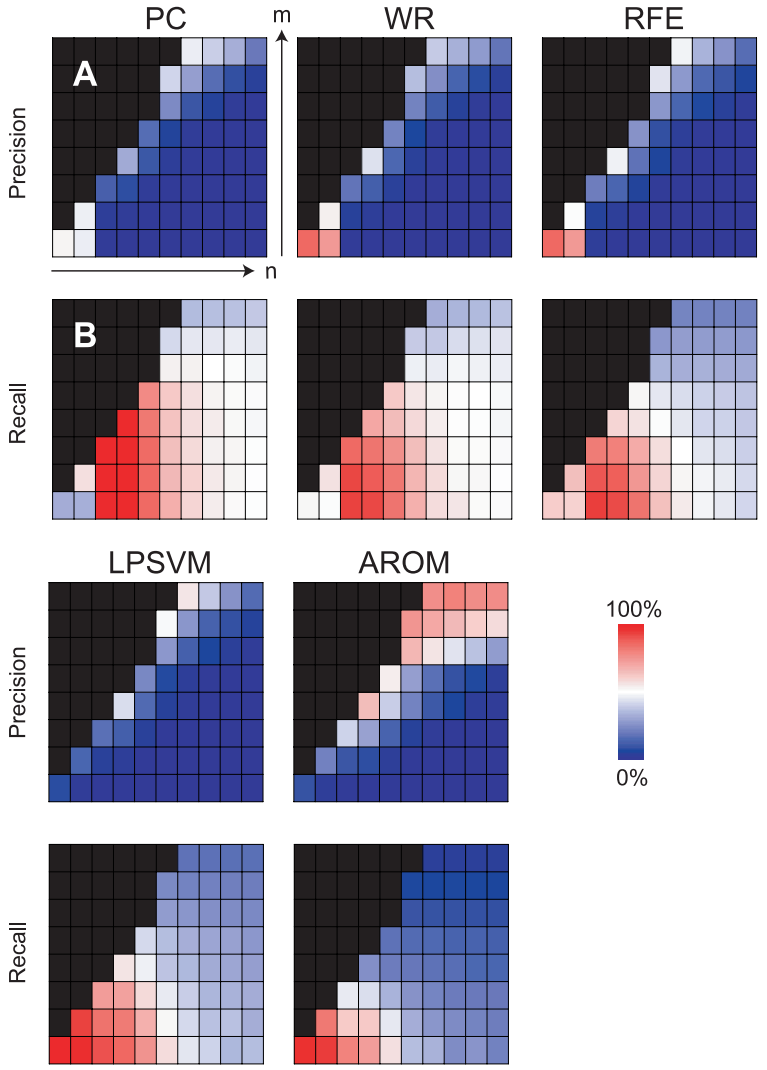

Figure 5.8: Feature set accuracy measured against the optimal set $S^{\dagger}$ (which here coincides with the set of Bayes-relevant features S^{*}).

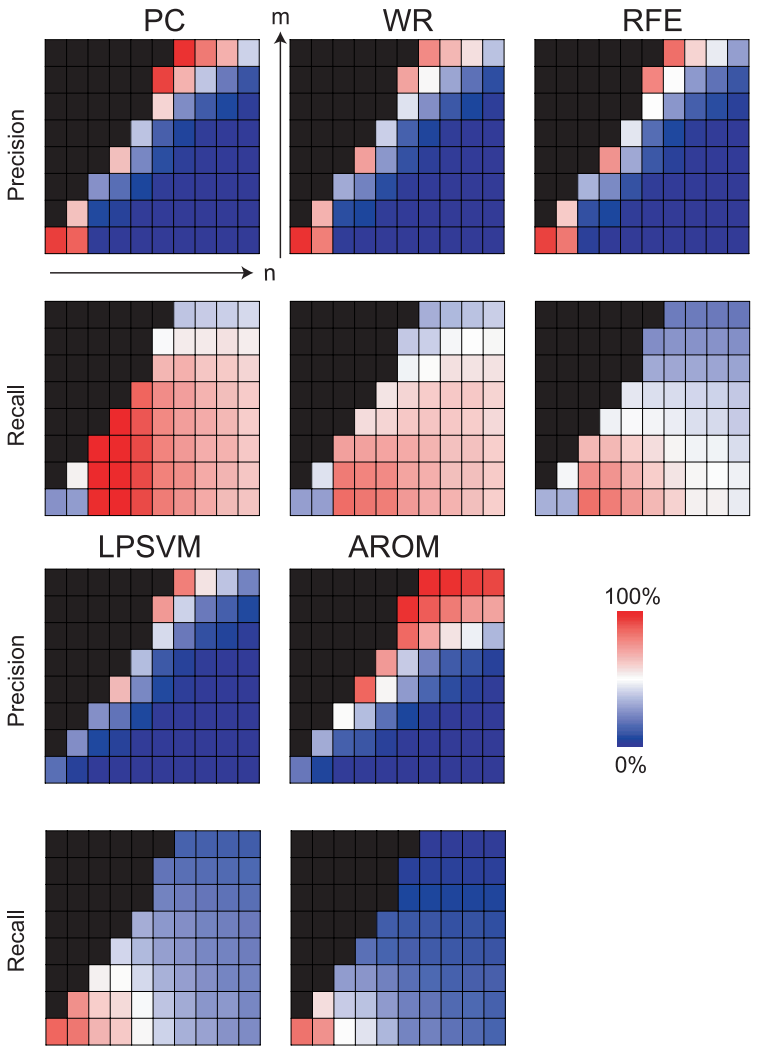

Figure 5.9: Feature set accuracy measured against the set of all relevant features S^A.

feature selection problem before choosing a method.

All methods have low precision, often including more irrelevant feature than relevant ones. AROM provided the best precision, but at the price of lower recall. This is natural since AROM was biased towards very small $|S|$ (Figure 5.6). The remaining methods were comparable.

These results cannot be validated on microarray data since the true feature sets are unknown (which of course is the reason for performing simulation studies in the first place). However, given the good agreement demonstrated in the previous tests (Figures 5.4, 5.6, 5.7), it is plausible that the findings in this section also apply to real data.

5.4 Summary

A striking trend throughout this chapter is that classification and feature selection methods behave very differently in high *vs.* low dimensions. Thus, I would strongly recommend that simulation studies of feature selection methods are performed with dimensions comparable to real data. Since data dimensionality has increased sharply during the last decade, one may suspect that much "folklore" in the feature selection field derived from earlier studies may not hold true for high-dimensional applications such as genome-wide measurements.

While some of the methods tested were previously been found to improve classification accuracy for SVMs at lower dimensions with few relevant features [57, 186], I find no such improvement in high dimensions (Figure 5.7). Thus, the utility of these methods for prediction purposes in seems marginal in high dimensions, unless the fraction of relevant features is very small (on the order of 1 predictive feature out of 1,000) in which case univariate filter methods may be helpful. Also, feature selection may still be useful when sparse models is a goal in itself, since several methods did manage to reduce dimension considerably without compromising predictive accuracy to any large extent.

It is clear from figures 5.8, 5.9 and the discussion in section 5.3.4 that none of the methods tested provide any control over false positives (precision). On the contrary, optimizing classifier risk results in a behavior opposite to what one would expect from a statistical test: here, recall is the most important factor, since this is essential for separating the classes and outweighs the additional noise due to low specificity. This is perhaps particularly true for regularized inducers such as the SVM, which tend to

be noise-tolerant (and thus false positive-tolerant) in themselves. This is probably the most serious problem for biological applications, where the interpretation of the selected feature set is very important.

On the other hand, statistical hypothesis tests (Section 4.1.1) that do control false positives are limited in that they cannot detect multivariate relationships. It is therefore of interest to develop feature selection methods that solves both of these problems. Such methods would be very useful for example in microarray data analysis, as multivariate analogues of the statistical methods currently available. I will consider two approaches to such methods in Chapter 7 (for $S^{\dagger}$) and Chapter 8 (for S^{A}).

6

Consistent Feature Selection in Polynomial Time

In this chapter I will study in detail the problem of finding a set of predictive features. This is the most studied feature selection problem in machine learning, on which a wealth of literature exists. The current consensus in the feature selection field is that the problem in general is intractable [67, 69, 85, 97]. Therefore, the focus has been on developing various heuristic methods, which are typically shown to be effective on various real-world data sets in terms of the predictive accuracy of some particular inducer.

A number results for various feature selection problems are frequently cited as evidence for intractability to motivate this direction of research. Cover and van Campenhout [33] treated the problem of finding a set S of size k with minimal Bayes risk (assuming that $|S^*| < k$, so that the problem is non-trivial). These authors showed that there exists "difficult" Gaussian distributions over *matrix-valued* $X \in \mathbb{R}^{n \times n}$ such that no method except exhaustive search can possibly find the optimal set. This result is very often put forward as an argument for the difficulty of the feature selection problem; however, van Campenhout [178] later showed that this result is specific to matrix-valued X, and does not apply to the typical case of $X \in \mathbb{R}^n$. The paper by Cover and van Campenhout

[33] thus appears to be frequently miscited. Devroye et al. [37], pp. 562 provides a different proof using a data distribution consisting of a complicated mixture of dirac distributions. However, this distribution is zero almost everywhere and thus not very realistic in practise.

Several particular cases have also been shown to be intractable. The min-features problem as defined by Almuallim and Dietterich [6] (see Section 4.1) was proved to be NP-complete by Davies and Russel [36]. This problem concerns deterministic data however, and is not relevant in our setting. Similarly, the "zero-norm" minimization (Section 4.3) was shown to be NP-hard by Amaldi and Kann [8].

Note that all of the above problems concern "noise-free" settings, either asymptotic cases or the optimization of a deterministic quantity, disregarding the statistical properties of data. The common question addressed by these studied might be phrased as "what is the computational complexity of the problem of finding the optimal feature set?". However, under a statistical data model, this question does not make sense: we cannot compute optimal feature sets from noisy, small samples in the first place. What we *can* do is to *estimate* the optimal feature set.

In this chapter I therefore take a different approach to the problem of selecting predictive features. I show that with very mild restrictions on the underlying data distributions, the problem of *estimating* S^* is tractable. Specifically, I prove that one can compute *consistent* estimates of S^* in polynomial time. Along the way, we will also find some interesting characterizations of the Bayes-relevant features S^* and the Markov boundary M^*. The algorithms presented in this chapter are not meant as practical suggestions however; their statistical power on real problems is probably quite low. Most of the material in this chapter appears in Nilsson et al. [126]. However, I herein generalize the exposition somewhat and also include some additional results for the small-sample case.

6.1 Relations between feature sets

6.1.1 The Markov boundary and strong relevance

We saw in Chapter 3 that all predictive features are contained in the Markov boundary M^*. However, the definition of M^* does not by itself suggest any simple method for inference. We will therefore begin this section by a result that relates M^* to the following more tractable

concepts of relevance.

Definition 6.1 (strong and weak relevance)**.** *A feature X_i is strongly relevant to Y iff $Y \not\perp X_i \,|\, X_{\neg i}$. A feature X_i is weakly relevant to Y iff it is not strongly relevant, but satisfies $Y \not\perp X_i \,|\, S$ for some set $S \subset X_{\neg i}$.*

Informally, a strongly relevant feature carries information about Y that cannot be obtained from any other feature. A weakly relevant feature also carries information about Y, but this information is "redundant" — it can also be obtained from other features. Note that in this terminology, Definition 3.11 of relevance is equivalent to the following.

Definition 6.2 (relevance)**.** *A feature X_i is relevant to Y iff it is strongly relevant or weakly relevant to Y.*

Our first major theorem of this chapter proves that the Markov boundary M^* actually is identical to the set of strongly relevant features.

Theorem 6.3. *For any strictly positive distribution $f(x, y)$, a feature X_i is strongly relevant if and only if i is in the Markov boundary M^* of Y.*

Proof. First, assume that X_i is strongly relevant. Then $Y \not\perp X_i \,|\, X_{\neg i}$, which implies $M^* \not\subseteq V_n \setminus \{i\}$, so $i \in M^*$. Conversely, fix any $i \in M^*$ and let $M' = M^* \setminus \{i\}$. If X_i is not strongly relevant, then $Y \perp X_i \,|\, X_{\neg i}$, and by the definition of the Markov boundary, $Y \perp X_{\neg M^*} \,|\, X_{M^*}$. We may rewrite this as

$$\begin{cases} Y \perp X_i \,|\, X_{M' \cup V_n \setminus M^*} \\ Y \perp X_{\neg M^*} \,|\, X_{M' \cup \{i\}}. \end{cases}$$

The intersection property (Theorem 2.10 of Section 2.2) now implies $Y \perp X_{\neg M'} \,|\, X_{M'}$. Hence, M' is a Markov blanket smaller than M^*, a contradiction. We conclude that X_i is strongly relevant. □

This theorem is important for algorithmic complexity when estimating the posterior $p(y \,|\, x)$. The definition of strong relevance immediately suggests a simple algorithm for estimating M^*: one need only test each X_i for strong relevance, that is, test for the conditional independence $Y \perp X_i \,|\, X_{\neg i}$. This procedure is clearly consistent and can be implemented in polynomial time. It is not very practical though, since these tests have very limited statistical power for large n due to the large conditioning sets $V_n \setminus \{i\}$ [131]. However, realistic solutions have recently been devised for the more narrow class of DAG-faithful distributions, yielding

polynomial and consistent algorithms [131, 176]. These algorithms are described in Section 4.1.4.

Tsamardinos and Aliferis [176] also proved a version of this theorem for the class of distributions faithful to a Bayesian Network (see Section 2.2). However, this distribution class is quite narrow and may be unreasonable in many practical applications. Theorem 6.3 puts the Markov boundary methods of Section 4.1.4 on a sound basis for a wide array of practical problems.

6.1.2 The Bayes-relevant features

Next, we will investigate the connection between the concepts of Bayes-relevance and strong relevance. For the proof of the main theorem of this section, we will need the following lemma.

Lemma 6.4. *For any conditional distribution $p(y \mid x)$, it holds that*

$$P\left(p(Y \mid X_i, X_{\neg i}) = p(Y \mid X_i', X_{\neg i})\right) = 1$$

if and only if

$$P\left(p(Y \mid X_i, X_{\neg i}) = p(Y \mid X_{\neg i})\right) = 1$$

provided that X_i, X_i' are independent and identically distributed.

Proof. Assume that the left-hand side holds. Then we must have

$$P\left(p(Y \mid X_i, X_{\neg i}) = p_0\right) = 1$$

for some p_0 constant with respect to X_i. But

$$p(y \mid x_{\neg i}) = \frac{f(x_{\neg i}, y)}{f(x_{\neg i})} = \frac{\int_{\mathcal{X}_i} p(y \mid x) f(x)}{\int_{\mathcal{X}_i} f(x)} = \frac{p_0 \int_{\mathcal{X}_i} f(x)}{\int_{\mathcal{X}_i} f(x)} = p_0$$

with probability 1, which implies the right-hand side. The converse is trivial. □

Theorem 6.5. *If $f(x, y)$ satisfies $P\left(\exists y \neq y' : p(y \mid X) = p(y' \mid X)\right) = 0$ (Assumption 3.6), then every Bayes-relevant feature is strongly relevant.*

Proof. For a Bayes-relevant feature X_i,we have

$$P\left(g^*(X_i, X_{\neg i}) \neq g^*(X_i', X_{\neg i})\right) > 0,$$

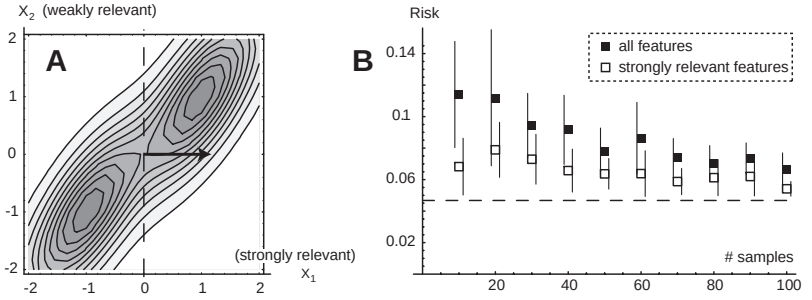

Figure 6.1: A: The example density $f(x_{2i-1}, x_{2i})$ given by (6.1). Here, X_1 is strongly relevant and X_2 weakly relevant. Arrow and dashed line indicates the optimal separating hyperplane. **B:** The risk functional $R(g)$ for a linear SVM trained on all relevant features (filled boxes) *vs.* on strongly relevant features only (open boxes), for the 10-dimensional distribution (6.1). Average and standard deviation over 20 runs are plotted against increasing sample size. The Bayes risk (dashed line) is $R(g^*) = 0.047$.

where X_i, X_i' are independent and identically distributed. From (Definition 2.12), we find that the event $g^*(X_i, X_{\neg i}) \neq g^*(X_i', X_{\neg i})$ implies

$$p(y \,|\, X_i, X_{\neg i}) \neq p(y \,|\, X_i', X_{\neg i}) \vee \exists y \neq y' : p(y \,|\, X) = p(y' \,|\, X).$$

The right alternative has probability 0 due to Assumption 3.6. Therefore,

$$P\left(p(y \,|\, X_i, X_{-i}) \neq p(y \,|\, X_i', X_{-i})\right) \geq P\left(g^*(X_i, X_{-i}) \neq g^*(X_i', X_{-i})\right) > 0.$$

By Lemma 6.4 this is equivalent to $P\left(p(Y \,|\, X_i, X_{-i}) = p(Y \,|\, X_{-i})\right) < 1$, which is the same as $Y \not\perp X_i \,|\, X_{\neg i}$. Hence, X_i is strongly relevant. □

Theorem 6.5 asserts that for predictive purposes, one may safely ignore weakly relevant features. This is important because it leads to more efficient (polynomial-time) algorithms for finding S^*. We will explore this consequence in Section 6.2. At this point it might be instructive to consider some examples.

Example 6.1 *Let $f(x, y)$ be a 10-dimensional Gaussian mixture*

$$f(x_1, \ldots, x_{10}, y) \propto \prod_{i=1}^{5} e^{-\frac{9}{8}((x_{2i-1}-y)^2 + (x_{2i-1}-x_{2i})^2)}. \qquad (6.1)$$

Figure 6.1A shows the joint distribution of (X_{2i-1}, X_{2i}) (all such pairs are identically distributed). Note that, although the shape of the distribution in Figure 6.1 suggests that both features are relevant to Y, it is easy to verify directly from (6.1) that $X_2, X_4, \ldots, X_{10}$ are weakly relevant: considering for example the pair (X_1, X_2), we have

$$\begin{aligned} p(y \mid x_1, x_2) &= \frac{f(x_1, x_2, y)}{f(x_1, x_2)} \\ &= \left[1 + \exp\left\{-\frac{9}{8}((x_1 + y)^2 - (x_1 - y)^2)\right\}\right]^{-1} \\ &= \left[1 + \exp\left\{-\frac{9x_1 y}{2}\right\}\right]^{-1} \end{aligned}$$

which depends only on x_1. Therefore, $Y \perp X_2 \mid X_1$, proving that X_2 is weakly relevant. The Bayes classifier is easy to derive from the condition $p(y \mid x) > 1/2$ of Equation (2.12)) and turns out to be $g^*(x) = sgn(x_1 + x_3 + x_5 + x_7 + x_9)$, so that $S^* = \{1, 3, 5, 7, 9\}$ as expected.

As the next example illustrates, the converse of Theorem 6.5 is false: there exist strictly positive distributions where even strongly relevant features are not relevant to the Bayes classifier.

Example 6.2 *Let $\mathcal{X} = [0, 1], \mathcal{Y} = \{-1, +1\}$, $f(x) > 0$ and $p(y = 1 \mid x) = x/2$. Here X is clearly strongly relevant. Yet, X is not relevant to the Bayes classifier, since we have $p(y=1 \mid x) < 1/2$ almost everywhere (except at $x = 1$). We find that $g^*(x) = -1$ and $R(g^*) = P(Y=1)$.*

Clearly, this situation occurs whenever a strongly relevant feature X_i affects the value of the posterior $p(y \mid x)$ but not the Bayes classifier g^* (because the change in $p(y \mid x)$ is not large enough to alter the decision of $g^*(x)$). In this sense, relevance to the Bayes classifier is *stronger* than strong relevance.

Remarks on strict positivity

The above theorems seemingly contradicts several examples found in the literature which indicate that weakly relevant features may be required by the Bayes classifier [97, 190]. This is because all such examples violate the requirement $f(x) > 0$. For example, a common "counterexample" to theorem 6.5 seen in the literature is the following: let $X_1 \sim N(y, 1)$ and let $X_1 = X_2$, that is, assume a functional, deterministic relation between

X_1 and X_2. Here it is often claimed that both features are weakly relevant, because one of them is "needed" by the Bayes predictor, but not both; which one to select is obviously arbitrary [190]. However, here we have $f(x) = 0$ almost everywhere. For such distributions weak relevance is not even well-defined, because (X_1, X_2) does not have a joint density $f(x_1, x_2)$, so that conditional independence is not well-defined. Thus, the theory of feature relevance does not apply in this case. This situation occurs whenever there exists a some functional relationship $X_i = h(X_j)$ between two features (or two sets of features) because this constrains the entire probability mass to a zero-measure set $\{X : X_i = h(X_j)\}$, so that $f(x)$ is unbounded on this set and zero everywhere else.

In practise, distributions of this type can be excluded from consideration whenever the data modeled is noisy. This is the case with physical measurements in general. For example, consider the additive Gaussian noise model

$$X = x_0 + \epsilon, \quad \epsilon \sim N(0, \sigma).$$

Since the noise component ϵ is strictly positive over the domain of X, we immediately obtain $f(x) > 0$. A similar argument holds for binary data with Bernoulli noise, and indeed for any additive noise model with $f(\epsilon) > 0$. In general, the strictly positive restriction is considered reasonable whenever there is uncertainty about the data [130]. Note that the $f(x) > 0$ criterion by definition only concerns the actual domain $\mathcal{X}$. If the data distribution is known to be constrained for physical reasons to some compact set such as $0 < X < 1$, then naturally $f(x) > 0$ need not hold outside that set.

There are examples of noise-free data in the machine learning literature, however, for example inference of logic propositions [177]. But this type of learning is not well-described by a statistical data model, and hence outside the scope of this thesis. Finally, it should be pointed out that for discrete features, it may happen that for some features the *observed* data satisfies $\forall k : x_i^{(k)} = x_j^{(k)}$ or similar, especially for small data sets. This observation is not incompatible with the notion that in distribution $P(X_i = X_j) < 1$, and it does not invalidate a statistical data model. Of course, if such a relationship should observed for large sample sizes, it would render a noisy statistical model unlikely. I have not encountered this problem in real data sets, however.

The feature set relations established at this point for strictly positive distributions are summarized in figure 6.2. In Section 6.2, I exploit these results to obtain polynomial-time algorithms for estimating M^* and S^*.

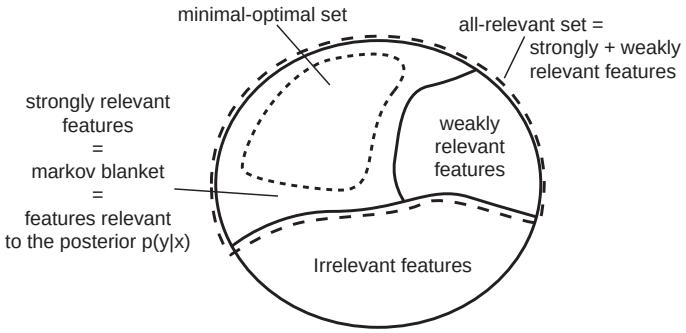

Figure 6.2: The identified relations between feature sets for strictly positive distributions. The circle represents all features. The dotted line (S^*) denotes a subset, while the solid lines denote a partition into disjoint sets.

Related work

The concepts of strong and weak relevance have been studied by several authors. In particular, the relation between the optimal feature set and strong *vs.* weak relevance was treated by Kohavi and John [97], who conjectured from motivating examples that "(i) all strongly relevant features and (ii) some of the weakly relevant ones are needed by the Bayes classifier". As we have seen in example 6.2, part (i) of this statement is not correct in general. Part (ii) is true in general, but theorem 6.5 shows that is not the case for the class of strictly positive f, and I would therefore argue that it is rarely true in practise.

A recent study by Yu and Liu [190] examines the role of weakly relevant features in more detail and subdivides these further into "redundant" and "non-redundant" features, of which the latter is deemed to be important for the Bayes classifier. For strictly positive distributions however, it is easy to see that all weakly relevant features are also "redundant" in their terminology, so that this distinction is not meaningful.

6.1.3 THE OPTIMAL FEATURE SET

The above theorems deal with M^* and S^*, which are properties of the true data distribution. Therefore, the results so far can be considered asymptotic. For finite sample sizes, the situation is more complicated. I next establish some conditions under which an expectation-optimal

feature set $S^\dagger$ is always contained in S^*, so that weakly relevant features are not an issue even for small sample sizes. We will need the following concept.

Definition 6.6. *The design cost of an inducer I_S is the functional*

$$\Delta(I_S) = \rho(I_S) - R(g_S^*) \geq 0.$$

In words, the design cost is the excess risk incurred by the fact that the inducer I cannot estimate the optimal classifier exactly. The design cost generally increases with the size of the feature set S, which is one motivation for using feature selection [86]. The following theorem outlines the conditions under which it there is a simple, intuitive relation between S^* and $S^\dagger$.

Theorem 6.7. *Let $f(x, y)$ be any distribution such that for every $k \leq |S^*|$, and every S of size k,*

$$\exists S' \subseteq S^* : |S'| = k \ \wedge \ R(g_{S'}^*) \leq R(g_S^*), \tag{6.2}$$

and let I be any inducer with range $\mathcal{G}$ containing g^* and design cost $\Delta(I_S)$ depending only on $|S|$. Then there exists an $S^\dagger \subseteq S^*$.

Proof. By (6.2), take an $S' \subseteq S^*$ with $|S'| = |S^\dagger|$ and $R(g_{S'}^*) \leq R(g_{S^\dagger}^*)$. Then

$$\begin{aligned} \rho(I_{S'}) &= \Delta(I_{S'}) + R(g_{S'}^*) \\ &\leq \Delta(I_{S'}) + R(g_{S^\dagger}^*) \\ &= \Delta(I_{S^\dagger}) + R(g_{S^\dagger}^*) = \rho(I_{S^\dagger}) \end{aligned}$$

□

Whether the condition (6.2) is true in general is to my knowledge an open problem. van Campenhout [178] studied the orderings of sets induced by the Bayes risk, but always assumed that $V_n = S^*$ so that this issue does not arise. Note that if the design cost increases only slowly with $|S|$, then we would expect that $S^\dagger \approx S^*$, so that the optimal feature set $S^\dagger$ can be well approximated by S^* for the purpose of minimizing risk. This was the case for the support vector machine in Chapter 5.

To illustrate the small-sample case, consider the following application of a linear support vector machine to the distribution in Example 6.1. Let I be a linear, soft-margin support vector machine (SVM) [32] with

the regularization parameter C fixed at 1, and let training data $Z^{(1:l)}$ be sampled from the density (6.1) with sample sizes $l = 10, 20, \ldots, 100$. Figure 6.1B shows the risk of $g = I(Z^{(1:l)})$ and $g_{S^*} = I_{S^*}(Z_{S^*}^{(1:l)})$. We find that I_{S^*} outperforms I, as expected. The risk functional $R(g)$ was here calculated by numerical integration of Equation (2.15) for each SVM hyperplane g and averaged over 20 training data sets. Clearly, adding the weakly relevant features increases risk in this example.

6.2 CONSISTENT POLYNOMIAL-TIME SEARCH ALGORITHMS

By limiting the class of distributions, I have hitherto simplified the feature selection problem to the extent that weakly relevant features can be safely ignored. In this section, we show that this simplification leads to polynomial-time feature selection algorithms.

For finite samples, the optimal feature selection algorithm Φ (Definition 4.1) depends on both the unknown data distribution f and as well as the inducer I [176]. However, a reasonable necessary condition for a "correct" feature selection algorithm is that it is consistent. Here, consistency means convergence to the Bayes-relevant feature set,

$$\Phi(Z^{(1:l)}) \xrightarrow{P} S^*.$$

Conveniently, the consistency of Φ then depends only on the data distribution f. Next, we propose a polynomial-time feature selection algorithm and show that it is consistent for any strictly positive f. As before, feature sets used as subscripts denotes quantities using only those features.

Theorem 6.8. *Take any strictly positive distribution $f(x, y)$ and let $\hat{c}(Z_S^{(1:l)})$ be a real-valued criterion function such that, for every feature subset S,*

$$\hat{c}(Z_S^{(1:l)}) \xrightarrow{P} c(S), \tag{6.3}$$

where $c(S)$ depends only on the distribution $f(x, y)$ and satisfies

$$c(S) < c(S') \iff R(g_S) < R(g_{S'}). \tag{6.4}$$

Then the feature selection method

$$\Phi(Z^{(1:l)}) = \{i : \hat{c}(Z_{\neg i}^{(1:l)}) > \hat{c}(Z^{(1:l)}) + \epsilon\}$$

where $\epsilon \in (0, \eta)$ with $\eta = \min_{i \in S^*}(c(V_n \setminus \{i\}) - c(V_n))$, is consistent.

Proof. Since f is strictly positive, S^* is unique by Lemma 3.7. By Definition 3.4 and the assumption (6.4) it holds that $i \in S^*$ iff $c(V_n) < c(V_n \setminus \{i\})$. First consider the case $i \in S^*$. Fix an $\epsilon \in (0, \eta)$ and let $\epsilon' = \min\{(\eta - \epsilon)/2, \epsilon/2\}$. Choose any $\delta > 0$. By (6.3) there exist an l_0 such that for all $l > l_0$,

$$P\left(\max_S |\hat{c}(Z_S^{(1:l)}) - c(S)| > \epsilon'\right) \leq \delta/2n$$

Note that since the power-set 2^{V_n} is finite, taking the maxima above is always possible even though (6.3) requires only point-wise convergence for each S. Therefore the events (i) $\hat{c}(Z^{(1:l)}) < c(V_n) + \epsilon'$ and (ii) $\hat{c}(Z_{\neg i}^{(1:l)}) > c(V_n \setminus \{i\}) - \epsilon'$ both have probability at least $1 - \delta/2n$. Subtracting the inequality (i) from (ii) yields

$$\begin{aligned} \hat{c}(Z_{\neg i}^{(1:l)}) - \hat{c}(Z^{(1:l)}) &> c(V_n \setminus \{i\}) - c(V_n) - 2\epsilon' \\ &\geq c(V_n \setminus \{i\}) - c(V_n) - (\eta - \epsilon) \geq \epsilon \end{aligned}$$

Thus, for every $l > l_0$,

$$\begin{aligned} P\left(i \in \Phi(Z^{(1:l)})\right) &= P\left(\hat{c}(Z_{\neg i}^{(1:l)}) - \hat{c}(Z^{(1:l)}) > \epsilon\right) \\ &\geq P\left(\hat{c}(Z^{(1:l)}) < c(V_n) + \epsilon' \ \wedge \ \hat{c}(Z_{\neg i}^{(1:l)}) > c(V_n \setminus \{i\}) - \epsilon'\right) \\ &\geq P\left(\hat{c}(Z^{(1:l)}) < c(V_n) + \epsilon'\right) + P\left(\hat{c}(Z_{\neg i}^{(1:l)}) > c(V_n \setminus \{i\}) - \epsilon'\right) - 1 \\ &\geq 1 - \delta/n \end{aligned}$$

For the converse case $i \notin S^*$, note that since $c(V_n) = c(V_n \setminus \{i\})$,

$$\begin{aligned} P\left(i \in \Phi(Z^{(1:l)})\right) &= P\left(\hat{c}(Z_{\neg i}^{(1:l)}) - \hat{c}(Z^{(1:l)}) > \epsilon\right) \\ &\leq P\left(|\hat{c}(Z_{\neg i}^{(1:l)}) - c(V_n \setminus \{i\})| + |c(V_n) - \hat{c}(Z^{(1:l)})| > \epsilon\right) \\ &\leq P\left(|\hat{c}(Z_{\neg i}^{(1:l)}) - c(V_n \setminus \{i\})| > \frac{\epsilon}{2} \ \vee \ |c(V_n) - \hat{c}(Z^{(1:l)})| > \frac{\epsilon}{2}\right) \\ &\leq P\left(|\hat{c}(Z_{\neg i}^{(1:l)}) - c(V_n \setminus \{i\})| > \epsilon'\right) + P\left(|c(V_n) - \hat{c}(Z^{(1:l)})| > \epsilon'\right) \\ &\leq \delta/n \end{aligned}$$

where in the last line we have used $\epsilon' \leq \epsilon/2$. Putting the pieces together, we obtain

$$\begin{aligned} P(\Phi(Z^{(1:l)}) = S^*) &= P(\Phi(Z^{(1:l)}) \supseteq S^* \ \wedge \ \Phi(Z^{(1:l)}) \subseteq S^*) \\ &= P(\forall i \in S^* : i \in \Phi(Z^{(1:l)}) \ \wedge \ \forall i \notin S^* : i \notin \Phi(Z^{(1:l)})) \\ &\geq |S^*|(1 - \delta/n) + (n - |S^*|)(1 - \delta/n) - (n - 1) \\ &= 1 - \delta \end{aligned}$$

Since δ was arbitrary, the required convergence follows. □

The requirement to choose an $\epsilon < \eta$ may seem problematic, since in practise η depends on the true distribution $f(x, y)$ and hence is unobservable. For convergence purposes, this can be remedied by choosing a sequence $\epsilon = \epsilon(l) \to 0$, so that $\epsilon < \eta$ will become satisfied eventually. In practise, the parameter ϵ controls the trade-off between precision and recall; a small ϵ gives high recall but low precision, and vice versa. With this in mind, one might choose ϵ based on the (estimated) variance of $\hat{c}$, so as to control precision and recall as desired.

The algorithm Φ evaluates the criterion $\hat{c}$ precisely n times, so it is clearly polynomial in n provided that $\hat{c}$ is. The theorem applies to both filter and wrapper methods, which differ only in the choice of $\hat{c}(Z_S^{(1:l)})$ [97]. As an example, let I be the k-NN rule with training data $Z^{(1:l/2)} = \{(X_1, Y_1), \ldots, (X_{l/2}, Y_{l/2})\}$ and let $\hat{R}$ be the usual empirical risk estimate on the remaining samples $\{(X_{l/2+1}, Y_{l/2+1}), \ldots, (X_l, Y_l)\}$. Provided k is properly chosen, this inducer is known to be universally consistent,

$$P\left(R(I_S(Z_S^{(1:l/2)})) - R(g_S^*) > \epsilon\right) \leq 2e^{-l\epsilon^2/(144\gamma_S^2)}$$

where γ_S depends on $|S|$ but not on l [37, pp. 170]. Next, with a test set of size $l/2$, the empirical risk estimate satisfies

$$\forall g : P\left(|\hat{R}(g) - R(g)| > \epsilon\right) \leq 2e^{-l\epsilon^2}$$

[37, pp. 123]. We choose $\hat{c}(Z_S^{(1:l/2)}) = \hat{R}(I_S(Z_S^{(1:l/2)}))$ and $c(S) = R(g_S^*)$ so that (6.4) is immediate. Further, this choice satisfies

$$\begin{aligned}
P\left(|\hat{c}(Z_S^{(1:l)}) - c(S)| > \epsilon\right) &= P\left(|\hat{R}(I_S(Z_S^{(1:l/2)})) - R(g_S^*)| > \epsilon\right) \\
&\leq P\left(|\hat{R}(I_S(Z_S^{(1:l/2)})) - R(I_S(Z_S^{(1:l/2)}))| + |R(I_S(Z_S^{(1:l/2)})) - R(g_S^*)| > \epsilon\right) \\
&\leq P\left(|\hat{R}(I_S(Z_S^{(1:l/2)})) - R(I_S(Z_S^{(1:l/2)}))| > \frac{\epsilon}{2}\right) \\
&\qquad + P\left(|R(I_S(Z_S^{(1:l/2)})) - R(g_S^*)| > \frac{\epsilon}{2}\right) \\
&\leq 2e^{-l\epsilon^2/4} + 2e^{-l\epsilon^2/(576\gamma_S^2)} \to 0
\end{aligned}$$

as required by the theorem, and is polynomial in n. Therefore this choice defines a polynomial-time, consistent wrapper algorithm Φ. Similarly, other consistent inducers and consistent risk estimators could be used, for example support vector machines [160] and the cross-validation error estimate [37, chap. 24].

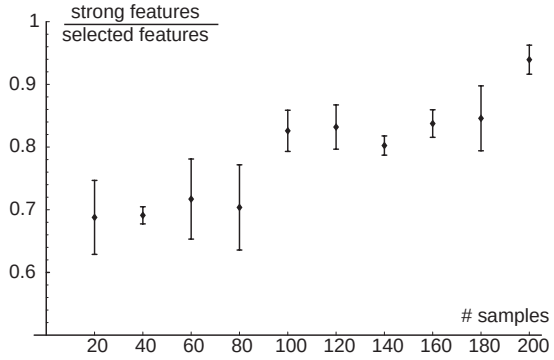

Figure 6.3: A feature selection example on a 10-dimensional density with 5 strongly and 5 weakly relevant features (equation (6.1)). Averaged results of 50 runs are plotted for samples sizes $20, \ldots, 200$. Error bars denote standard deviations.

The feature selection method Φ described in theorem 6.8 is essentially a backward-elimination algorithm. With slight modifications, the above shows that many popular feature selection methods that implement variants of backward-search, for example Recursive Feature Elimination [68], are in fact consistent. This provides important evidence of the soundness of these algorithms.

In contrast, forward-search algorithms are not consistent even for strictly positive f. Starting a with feature set S, forward-search would choose the feature set $S' = S \cup \{i\}$ (that is, add feature X_i) iff $\hat{c}(Z_{S'}^{(1:l)}) < \hat{c}(Z_S^{(1:l)})$. But it may happen that $R(g_{S'}^*) \not< R(g_S^*)$ even though S' is contained in S^*. Therefore, forward-search may miss features in S^*. The "noisy XOR problem" [67, pp. 1116] is an example of a strictly positive distribution with this property.

A simple example illustrating theorem 6.8 is shown in figure 6.3. We implemented the feature selection method Φ defined in the theorem, and again used the data density f from equation (6.1). Also here, we employed a linear SVM as inducer. We used the leave-one-out error estimate [37] as $\hat{R}$. As sample size increases, we find that the fraction of strongly relevant features selected approaches 1, confirming that $\Phi(D^l) \xrightarrow{P} S^*$. Again, this emphasizes that asymptotic results can serve as good approximations for reasonably large sample sizes.

Data set	$l \times n$	No FS	$\Phi_{\epsilon=0}$	RELIEF	FCBF
Breast cancer	569×30	8	9(7)	69(8)	62(2)
Ionosphere	351×34	11	9(14)	16(26)	14(5)
Liver disorder	345×6	36	39(5)	−(0)	43(2)
E.Coli	336×7	36	20(5)	43(1)	57(1)
P.I. Diabetes	768×8	33	36(7)	35(1)	35(1)
Spambase	4601×57	12	17(39)	25(26)	40(4)

Table 6.1: Feature selection on UCI data sets. Test error rates are given in %, number of features selected in parentheses. Significant differences from the classifier without feature selection ("No FS") are underscored (McNemar's test, $p = 0.05$). l denotes number of samples, n number of features.

6.2.1 EXPERIMENTAL DATA

The algorithm Φ is primarily intended as a constructive proof of the fact that polynomial and consistent algorithms exist; we do not contend that it is optimal in practical situations. Nevertheless, we conducted some experiments using Φ on a set of well-known data sets from the UCI machine learning repository [123] to demonstrate empirically that weakly relevant features do not contribute to classifier accuracy. We used a 5-NN classifier together with a 10-fold cross-validation error estimate for the criterion function $\hat{c}$. For each case we estimated the final accuracy by holding out a test set of 100 examples. Statistical significance was evaluated using McNemar's test [38]. We set $\epsilon = 0$ in this test, as we were not particularly concerned about false positives. For comparison we also tried the Relief algorithm [94] and the FBCF algorithm by Yu and Liu [190], both of which are based on the conjecture that weakly relevant features may be needed. We found that Φ never increased test error significantly compared to the full data set, and significantly improved the accuracy in one case (table 6.1). The FCBF and Relief algorithms significantly increased the test error in five cases. Overall, these methods selected very few features (in one case, Relief selected no features at all) using the default thresholds recommended by the original papers (for FCBF, $\gamma = n/\log n$ and for Relief, $\theta = 0$, in the notation of each respective paper; these correspond to the ϵ parameter of the Φ algorithm).

6.3 Discussion

I have here shown that there exists polynomial-time search (wrapper) algorithms which converge to S^* as $l \to \infty$, regardless of the underlying data distribution. It is easy to obtain a similar result in the case of consistent linear classifiers $g(x) = \text{sign}(w^T x)$: since in this case $R(I(Z^{(1:l)})) \xrightarrow{P} R(g^*)$ as $l \to \infty$, we must have $g^* = \text{sign}((w^*)^T x)$ and $i \in S^* \iff w_i^* \neq 0$. Hence, $\phi = 1_{\{w_i \neq 0\}}$ is consistent. However, this presupposes a fairly narrow class of data distributions where g^* indeed is linear. In contrast, the algorithm suggested in Theorem 6.8 is consistent for arbitrary distributions, provided that the inducer used is. A possible generalization here could be to "kernelize" the linear classifier with a universal kernel using the recent method of Li et al. [110].

One consequence of the results in this chapter that possibly is of general theoretical interest is that it is sometimes possible to design tractable, consistent algorithms for the small-sample case even when the asymptotic case is intractable. Often, the tractability of machine learning problems are considered in the asymptotic case; for example, [26] recently proved that Bayesian network inference is NP-hard in the asymptotic case. The results herein suggest that despite difficulties in the asymptotic case, it may be possible to design algorithms for computing estimates at small samples which *converge* to the correct solution as $l \to \infty$. Indeed, a recent paper by Kalisch and Bühlmann [91] establishes such a consistent, polynomial-time procedure for Bayesian network inference.

6.4 Summary

In this chapter, I have explored an alternative approach to the problem of finding predictive features: instead of designing suboptimal methods for the intractable full problem, I propose to use consistent and efficient (polynomial-time) methods for a restricted data distribution class. I find that a very mild restriction to *strictly positive* distributions is sufficient for the problem to be tractable. I argue that the condition of strict positivity is satisfied in almost every experimental setting due to the presence of noise. Therefore, I conclude that finding predictive features is tractable for most practical problems, contrary to the current consensus [67].

The results of this chapter are mainly theoretic. The algorithm outlined in Section 6.2 is intended as a proof-by-example of the existence of cor-

rect algorithms, and is not meant to be directly applicable in practical settings. Nevertheless, these results provide a foundation upon which one can build sound algorithms for practical problems. In the next chapter, I propose a statistical method that addresses these problems.

7

Bootstrapping Feature Selection

In Chapter 6 we found that it is possible to devise algorithms for discovering predictive features which are consistent and yet computationally efficient. However, we left open the question of the accuracy of the feature sets selected when such methods are applied in practise, for small sample sizes: recall that or the algorithm presented in Section 6.2, the parameter ϵ was seen to control the trade-off between false positives and false negatives, but no principled method of adjusting this parameter was devised. In this chapter, I will consider the issue of controlling false positive rates for learning predictive features.

7.1 Stability and error rates

A problem which has recently received much attention in practical applications of feature selection, in particular cancer research, is *instability* of feature selection methods [46, 47, 116]. Stability is defined as the normalized expected overlap between two features S, S' derived from independent, replicate experimental data sets.

Definition 7.1 (Stability)**.** *Let S, S' be two independent, identically distributed random sets, that is, two random variable on 2^{V_n}. Then the*

stability of S is defined as

$$\mathcal{S}(S) = \mathbb{E}\left[\frac{|S \cap S'|}{\max\{|S \cup S'|, 1\}}\right]. \tag{7.1}$$

This stability measure is always between 0 (no expected overlap) and 1 (complete overlap). Clearly, a feature selection method Φ applied to two independent, replicate experimental data sets $Z^{(1:l)}$, $(Z')^{(1:l)}$ yields two such random sets, so that the corresponding stability measure $\mathcal{S}(\Phi(Z^{(1:l)}))$ is well-defined. When this is close to zero, we say that Φ is unstable.

It has been observed experimentally that feature selection methods tend to be unstable in high-dimensional settings [46, 116]. This has caused a lot of concern regarding the reliability of these methods [88]. The consensus among practitioners is naturally that analysis methods should be reproducible, and it is expected that sound methods should yield identical results on independent data sets. Therefore, instability has been interpreted as evidence that the methods are incorrect, *i.e.*, that the fraction of false positives is high.

This is an understandable position, but it is grounded in intuition derived from low-dimensional or even one-dimensional data sets. We will see that — perhaps surprisingly — this notion of replication does not carry over to high-dimensional data. For high-dimensional feature selection problems, it is quite possible for a feature selection method to be both unstable and correct.

To better understand the concept of instability, it seems reasonable to relate it to better understood statistical concepts like false discovery rate (FDR) and power. The false discovery rate was described in Section 4.1.2. Power is here defined as the expected fraction of true positives among the selected features,

$$\mathbb{E}\left[\frac{|S \cap S^*|}{\max\{|S|, 1\}}\right].$$

Here the the Bayes-relevant features S^* is considered as the "true" feature set. To examine these properties, I conducted a simple simulation experiment using a two-class data distribution. Here $n = 1000$ while S^* consisted of 200 (20%) differentially expressed features, $(X_i \,|\, Y = y) \sim N(y\mu, 1)$. The remaining 800 features were distributed as $N(0, 1)$. I then investigated how stability, FDR and power depends on μ. To select feature sets, Student's t-test with the Benjamini-Hochberg correction was used, since this is known to control FDR at nominal levels [12].

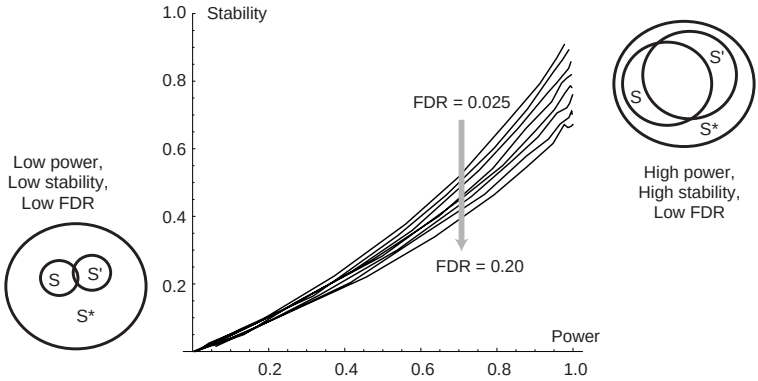

Figure 7.1: Simulation experiment of stability and power for differential expression testing with the t-test. Left and right, Venn diagram illustrations of the instability measure (see text). S^*, Bayes-relevant features; S, S', selected feature sets.

The t-test is in fact optimal in this setting [21]. Nevertheless, our simulations revealed that it can exhibit low stability (Figure 7.1). Specifically, when μ was small and the method exhibited low power, stability was always low, even with a stringent FDR. Conversely, in situations with strong differential expression and high power, stability was high and did not depend much on the chosen FDR. I therefore conclude that instability need not indicate that the feature selection method produces many false positives. Rather, low stability results from low power to detect true positives, so that in each experiment small feature sets are selected at random among the many true positives, with little overlap between experiments (figure 1, left). In essence, this result means that in situations where many features are weakly associated with the target variable and power is low, *one cannot expect to reproduce feature sets in independent experiments*, even with the most stringent and correct methods.

7.2 Feature selection is ill-posed

The above shows that instability can arise merely from low power in high dimensions. With simple univariate statistical tests like the t-test used above, this is a satisfactory explanation for the phenomenon, as the false discovery rate was controlled in this case. However, for feature selection

methods that attempt to optimize the risk of a predictor, there is a second component that contributes to instability. We found in section Section 3.2 that for small samples, the optimal set $S^\dagger$ may not be unique, even if the expected risk could be determined exactly. As in practise we must necessarily measure risk only approximately, it is to be expected that the solution is even more degenerate. This may contribute to the observed instability [116] of such methods.

Selecting features by optimizing predictor risk can thus be seen as an *ill-posed problem* in the classic sense [181]. Finding an expectation-optimal feature set $S^\dagger$ amounts to solving the equation

$$\mathbb{E}\left[R(I_S)\right] = \mathbb{E}\left[R(I_{S^\dagger})\right] \tag{7.2}$$

for a feature set S. This problem is ill-posed in the sense that the selected set S might deviate substantially from $S^\dagger$ even for very small changes in the right-hand side. Conversely, fairly large changes in S may not have any substantial effect on $\mathbb{E}\left[R(I_S)\right]$. Hence, for real data where one merely has access to noisy estimates of these quantities, one should expect unstable feature sets.

In contrast to the situation in the previous section, this degeneracy may contribute to false positives, as seen in Chapter 5. In summary, for many feature selection algorithms there will be one "harmless" component component contributing to instability due to low power, without inflating false positive rates; but there will also be one component due to degeneracy that contributes both instability *and* false positives. Intuitively, this will happen whenever the feature set S is derived through an "inverse" problem such as (7.2), as opposed to the "direct" formulation seen in hypothesis testing.

7.3 The bootstrap approach

From the above result and discussion, I conclude that stability is not an operational measure of feature set accuracy. It would be preferable to be able to perform statistical tests to decide whether to include each feature, as with the t-test used above. This has hitherto not been attempted for the more complex machine learning methods typically used to construct feature sets [67]. This is probably because a rigorous definition of the "true set" has been elusive. Building upon the definitions established in Chapter 3, we are now in a position to design such tests. I next describe a general method for this purpose.

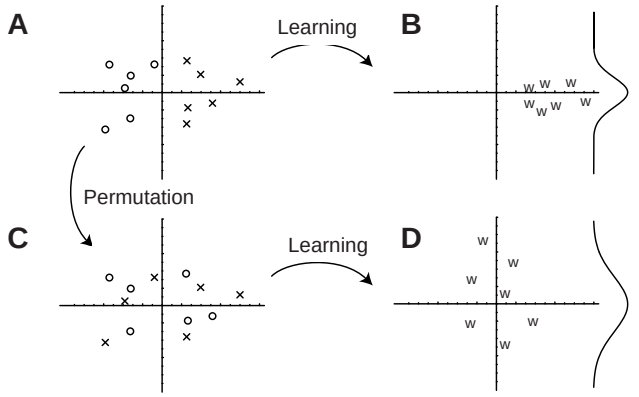

Figure 7.2: Illustration of the problem of permutation testing for predictive models. A: Example data distribution where $X_2 \perp Y$ so that H_0^2 is true. B: The distribution of W_i corresponding to (A) satisfies $\mathbb{E}[W_2] = 0$. C: Distribution of permuted data X'. D: The corresponding distribution of W_i' satisfies $\mathbb{E}[W_i'] = 0$, but the distribution of W_2' may now be different from W_2.

The first step in developing a hypothesis test is to define the null and alternate hypotheses H_0 and H_1. In our case, as we are interested in discovering predictive features, we define the null hypothesis for each feature X_i to be $H_0^i : i \notin S^*$.

Next, for each X_i we need a statistic that can be used to test H_0^i. This statistic depends on the learning method used; as an example we here consider the linear Support Vector Machine (SVM) [32]. If we consider the coefficients w_i estimated from the training data as observations of a statistic W_i, then (for large samples) we may re-write the null hypotheses simply as $H_0^i : \mathbb{E}[W_i] = 0$. In general, any learning procedure that for each feature outputs a parameter estimate $W_i = W_i(Z^{(1:l)})$ satisfying

$$\mathbb{E}[W_i] = 0 \implies i \notin S^*$$

can be used in this fashion. Several inducers are known to satisfy this assumption in the large sample limit; for a proof in the case of the SVM, see for example Hardin et al. [72].

Still, testing this null hypothesis is non-trivial, since the distribution of the statistic W_i is difficult to determine. Given the unknown data distribution and the complex procedure for computing W_i from training data (*e.g.*, the SVM solves a quadratic programming problem), there is

little hope deriving this distribution analytically. Non-parametric permutation tests [63] are unfortunately not applicable here either, since permuting the target variable does not correctly reconstruct the distribution of W_i under H_0^i. This is because W depends on all features, and permuting Y renders every feature independent of Y, which clearly is different from the single independence $X_i \perp Y$ that we want to test. An illustration of this problem is shown in Figure 7.2. A possible remedy for this problem would be to perform n permutation tests by permuting each X_i instead. This is computationally very expensive, however.

I therefore considered a hypothesis testing framework based on the bootstrap [44]. In this approach, one constructs a bootstrap sample consisting of B data sets $D^{(1)}, \ldots D^{(B)}$ (here subscript denote bootstrap replication number, not feature indices) by sampling with replacement from a given data set $D = Z^{(1:l)}$. Applying the inducer to each of these yields the statistic vectors $w^{(1)} = W(D^{(1)}), \ldots, w^{(B)} = W(D^{(B)})$. Then, for each feature i, the corresponding bootstrap observations $w_i^{(1:B)} = w_i^{(1)}, \ldots, w_i^{(B)}$ are used to estimate the distribution of W_i. From this we obtain bootstrap confidence intervals for each $\mathbb{E}[W_i]$, and by inverting this interval (*i.e.*, computing the confidence level at which 0 is covered) we obtain p-values p_i for each null hypothesis H_0^i.

7.3.1 ACCURACY OF THE BOOTSTRAP

The bootstrap methodology is necessarily an approximate technique [42]. It is known that the bootstrap estimates of the distributions $p(w_i)$ are consistent [21, pp. 480]. For finite samples however, one cannot obtain *exact* hypothesis tests using the bootstrap; that is, bootstrap p-values will satisfy $P(p_i \leq \alpha) \leq \alpha$ only approximately. Put differently, the realized level of the bootstrap will only approximately equal the nominal level α. More precisely, under some conditions on the data distribution, it can be established that the bootstrap p-values satisfy

$$P(p_i \leq \alpha) \leq \alpha + \mathcal{O}(1/l).$$

See for example Efron and Tibshirani [44]. Thus, the tests converge towards the nominal level α at a rate of $1/l$. The main issue in practise however, is the precise value of the term $\mathcal{O}(1/l)$ for a given sample size. In the next section I verify by simulations that this term is indeed negligible.

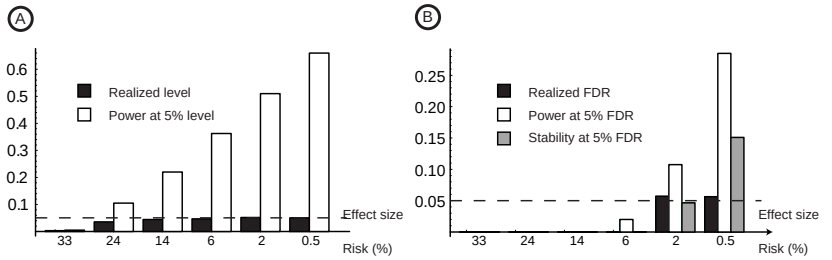

Figure 7.3: Results on simulated data for the bootstrap method using the SVM inducer. **A:** Realized level and power at 5% nominal level, indicated by dashed line. **B:** Realized FDR, power and stability after the Benjamini-Hochberg correction. Here the nominal FDR was set at 5%, indicated by dashed line.

7.3.2 Simulation studies

I conducted a simulation study of the proposed bootstrap procedure for two-class data sets, using 1,000 features and 100 samples. The data distribution was the same as in Section 7.1, that is, 20% of the features were differentially expressed, with varying effect size. The bootstrap method was tested with a linear Support Vector Machine (SVM) [32], the Kernel Fisher Discriminant (KFD) [117] and the Weighted Voting (WV) algorithm of Golub et al. [62]. Since results were highly similar for each of these learning methods, I here present the results for the SVM only. For each learning method and for a range of effect sizes, our bootstrap test produced correct p-values, resulting in test levels very close to the chosen nominal level of 5%, while power increased with increasing effect size (Figure 7.3A). This indicates that the bootstrap test itself is sound. To yield reliable feature sets, we then corrected for multiplicity using the procedure of Benjamini and Hochberg [12], setting nominal FDR at the typical value of 5%. This did indeed control FDR at nominal levels (Figure 7.3B), although power was limited, yielding no significant features for predictors with less than approx. 90% accuracy. I therefore expect that predictors must be quite accurate in order to yield reliable feature sets. It was also observed that the stability of the

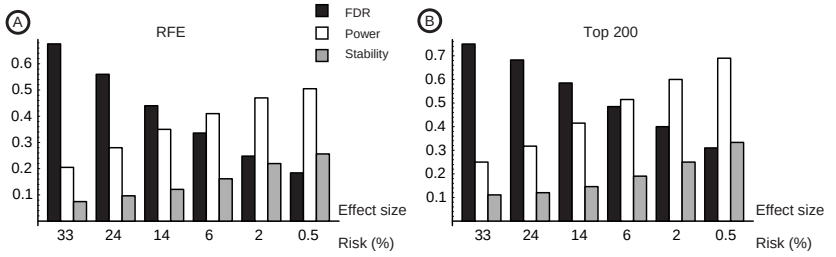

Figure 7.4: Results on simulated data using the SVM inducer. **A:** feature set selected by recursive feature elimination (RFE). **B:** feature set selected as the top 200 features.

resulting feature sets was low (less than 20% in all of our experiments), in agreement with previous result (Chapter 5).

The simulation study was repeated using the popular Recursive Feature Elimination (RFE) method to select feature sets. While this method did produce accurate predictive models (data not shown), the feature sets derived from it contained excessive amounts false positives (Figure 7.4A). This problem was most pronounced for difficult classification problems (small effect sizes), where most of the features chosen were false positives. Since RFE chooses the feature set based on the (estimated) accuracy of the predictor, I believe that this phenomenon is a consequence of predictor regularization, as discussed in the introduction. I conclude that this method is not suitable for generating feature sets. Moreover, this problem is not specific to RFE, as similar results have recently been obtained with other methods that optimize the feature set for prediction accuracy [126].

Similar results was also obtained when choosing as the feature set a "top list" of 200 features (the size of the true feature set in our simulations), ranked by the w_i statistics (Figure 7.4B). I would thus also advise against the common practise of choosing a fix number of "top features" from a ranked list.

The performance of both the RFE and bootstrap methods are summarized using ROC curves in Figure 7.5. The ROC curves were nearly identical, indicating that the bootstrap does not alter the overall performance of the learning method. However, merely summarizing the

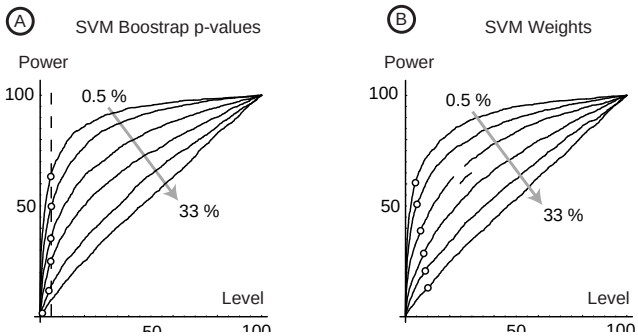

Figure 7.5: ROC curves for simulated data using the SVM inducer. Gray arrow indicates decreasing effect size (harder prediction problem), percentages indicate SVM error rates. **A:** Genes ranked by bootstrap p-values. Circles indicate the points chosen on each ROC curve by the bootstrap method at the 5% level (dashed line). **B:** Genes ranked directly by the SVM weights. Circles indicate the points chosen by recursive feature elimination.

Data set (ref.)	n	MCF(%)	CV(%)
Alon [7]	62	35	19±7.2
Golub [62]	72	32	3.0±4.2
Singh [156]	136	43	7.4±3.0
van't Veer [179]	97	47	38±8.4
Wang [183]	286	37	35±4.3

Table 7.1: Cancer gene expression data sets. n, number of samples; MCF, minority class frequency; CV, cross-validation error with a linear SVM, mean ± std.dev.

performance of a feature selection methods using ROC curves is not sufficient; to control error rates, it is also necessary to be able to choose a point on this ROC correctly, for example at 5% level (dashed line in Figure 7.5A). Existing feature selection methods have no such facility; they choose points on the curve to optimize predictor accuracy, rather than to avoid excessive false positives in the feature set, as in the case of RFE (Figure 7.5B). This leads to loss of error rate control.

Data set (ref.)	BS	BS_0	RFE	RFE_0	DE
Alon [7]	19	0	62	125	316
Golub [62]	537	0	78	156	1092
Singh [156]	97	0	78	312	3822
van't Veer [179]	0	0	76	153	2
Wang [183]	0	0	312	312	110

Table 7.2: Results of features selection on cancer gene expression data. BS, significant genes using the bootstrap with SVM at 5% FDR; RFE, genes chosen by recursive feature elimination; BS_0 and RFE_0, gene chosen by the bootstrap and RFE methods respectively on permuted data. DE, differentially expressed genes using the t-test at 5% FDR.

7.3.3 APPLICATION TO CANCER DATA

We next applied our bootstrap method with the SVM learning method to analyze a number of publicly available cancer gene expression data sets (Table 7.1). The feature selection results are summarized in Table 7.2. Intuitively, a prerequisite for obtaining a good feature set is that the predictor is accurate; a predictor which does not perform better than random should not be expected to be useful for constructing feature sets. For the van't Veer and Wang data sets, the SVM was unable to discriminate effectively between the classes (Table 7.1). Accordingly, for these data sets the bootstrap method did not call any genes significant. For the other data sets, the general trend is that higher predictive accuracy results in greater power for the bootstrap test, with over 500 significant predictive genes at 5% FDR for the data set by Golub et al. [62].

For comparison, we performed a conventional differential expression test for each data set using the t-test statistic with the Benjamini-Hochberg correction (Table 7.2). This identified many more genes than the bootstrap test, the only exception being the van't Veer data. This may indicate that many of the differentially expressed genes are redundant for prediction and therefore does not enter into the Bayes-relevant feature set S^*. In this way, we can identify the genes most predictive of the clinical variable among a much larger set of indirectly related genes. However, it may also be possible that the bootstrap test in some cases has less power since the models fitted by the machine learning methods could be overly complex.

We also tried re-running the bootstrap method on a permuted version

of each original data set. This yielded zero findings in each case, thus confirming that we do not obtain spurious findings where no predictive power exists. In contrast, the RFE method reports even *larger* feature sets with permuted data, and also reports gene lists for the difficult data sets, in accordance with previous results (Chapter 5).

7.4 Discussion

For simplicity, I have in this chapter restricted attention to two-class problems and linear predictors. However, the bootstrap framework outlined herein is in principle applicable to any learning method for which a statistic can be derived to test the feature set null hypothesis. Continuous clinical variables can easily be handled by substituting the classifiers used here for, say, ridge regression [76] or the relevance vector machine [172]. Moreover, non-linear dependencies could be possibly be addressed using the recent method of Li et al. [110].

A number of issues remain to be considered. The statistics W_i used herein are guaranteed to be correct (i.e., satisfy $\mathbb{E}[W_i]$ when $g \notin S^*$) only in the limit of large samples. In our experiments we have noted that this property seems to hold for small samples as well, but formal proofs are lacking. Also, even the multivariate inference methods tend to favor the marginal distributions in high dimensions, and thus resemble the simpler univariate methods. This is likely a consequence of the specific type of regularization used (often, the regularizer strives towards independent features), but it is unsatisfying given that gene expression is governed by complex networks. Further research is needed concerning the small-sample properties of the W_i statistics and possible alternative regularization techniques that better take into account the biological properties of gene expression data.

As discussed above, bootstrap hypothesis testing is known to provide only approximate p-values. The validity of this approximation should be verified in each situation by appropriate simulations. A possible future extension for improved performance could be to also *estimate* the extra term $\mathcal{O}(1/l)$ from simulations and correct the bootstrap p-values accordingly, thus "calibrating" the method for each particular application. As mentioned, it would also be possible to apply a separate permutation test by permuting each feature X_i independently. However, this would vastly increase the computational burden for large n.

7.5 SUMMARY

While much attention has been given to the problem of learning predictors from gene expression data, biology researchers are often more interested in the predictive genes than in the predictor itself. For example, while being able to diagnose a tumor merely from its expression profile is theoretically interesting, the improvement in accuracy compared with existing diagnosis methods is still unclear [41, 128]. The feature set, on the other hand, may supply important clues to the molecular pathology of the tumor and is therefore of considerable interest. It is unfortunate that the accuracy of the feature sets derived from predictive models is poorly understood. Many researchers consider good predictive accuracy as evidence for an accurate feature set. However, with modern machine learning methods, this is not a valid assumption: many predictors have internal mechanisms for suppressing noise and may perform well even with excessive amounts of false positives in the associated feature set [127, 134].

In this chapter, I therefore develop a method for controlling error rates of feature sets directly, as is done with statistical tests for differential expression. The resulting bootstrap method proposed herein can be used to discover predictive genes in a statistically sound manner. In essence, our test identifies the genes directly associated with the target variable, rather than all genes with a significant association. The method is thus primarily useful when expression changes are abundant, as in cancer gene expression, so that prediction is feasible.

Many important inference problems in bioinformatics may be cast as prediction problems. Case-control association studies can be viewed as predicting phenotypes from SNP data, with the feature set corresponding to the associated markers [182]; in protein structure prediction, the feature set corresponds to the features (motifs, active sites, etc) of the protein that determine its functional class [109]; in gene network reconstruction, a model is fit to predict expression values of regulated genes given the expression of their regulators and the feature set corresponds to the interactions in such networks [189]. The bootstrap framework describe here is potentially applicable in all of these situations, thus providing error control for important, complex inference problems where permutation testing is not possible. Many more examples could be mentioned, also outside of the biological sciences. Therefore, I hope that the developments in this chapter could to be of general interest.

8

Finding All Relevant Features

In biological applications of feature selection, many researchers are primarily interested in the "biological significance" of features (genes) that depend on the target variable Y [157], rather than their predictive power. As a rule, biological significance means that a gene is causally involved (either indirectly or indirectly) in the biological process of interest. As previously discussed in Section 3.3, the most predictive features are not in general the biologically most relevant ones, since features may be predictive without necessarily being causally closely related, and may therefore be quite irrelevant to the biologist.

In this section I therefore focus on the problem of identifying the set S^A consisting of *all* genes relevant to the target variable, rather than the set S^*, which may be more determined by technical factors than by biological significance. To the best of my knowledge, this problem has not previously been studied in full generality, although several important special cases have been considered (reviewed in Chapter 4).

8.1 Computational complexity

In Chapter 6, we found that for strictly positive distributions, the problem of finding the Bayes-relevant feature set S^* is tractable in the sense

that there exist consistent estimate computable in polynomial time. The first result of this chapter demonstrates that because S^A includes weakly relevant features (figure 6.2), finding this set is computationally much harder than finding the Bayes-relevant features.

Theorem 8.1. *For a given feature X_i and for every $S \subseteq V_n \setminus \{i\}$, there exists a strictly positive $f(x, y)$ satisfying*

$$Y \not\perp X_i \,|\, X_S \;\wedge\; \forall S' \neq S : Y \perp X_i \,|\, X_{S'}. \tag{8.1}$$

Proof. Without loss of generalization we may take $i = n$ and $S = \{1, \ldots k\}$. Let $X_{S\cup\{k+1\}}$ be distributed as a $k+1$-dimensional Gaussian mixture

$$\begin{aligned} f(x_{S\cup\{k+1\}} \,|\, y) &= \frac{1}{|M_y|} \sum_{\mu \in M_y} N(x_{S\cup\{k+1\}} \,|\, \mu, \Sigma) \\ M_y &= \{\mu \in \{1,0\}^{k+1} : \mu_1 \oplus \cdots \oplus \mu_{k+1} = (y+1)/2\} \end{aligned}$$

where $\oplus$ is the XOR operator (M_y is well-defined since $\oplus$ is associative and commutative). This distribution is a multivariate generalization of the "noisy XOR problem" [67]. It is obtained by placing Gaussian densities centered at the corners of a $k+1$-dimensional hypercube given by the sets M_y, for $y = \pm 1$. It is easy to see that this gives $Y \not\perp X_{k+1} \,|\, X_S$ and $Y \perp X_{k+1} \,|\, X_{S'}$ if $S' \subset S$. Next, let $X_{i+1} = X_i + \epsilon$ for $k < i < n$, where ϵ is some strictly positive noise distribution. Then it holds that $Y \not\perp X_i \,|\, X_S$ for $k < i < n$, and in particular $Y \not\perp X_n \,|\, X_S$. But it is also clear that $Y \perp X_n \,|\, X_{S'}$ for $S' \supset S$, since every such S' contains a better predictor $X_i, k < i < n$ of Y. Taken together, this is equivalent to (8.1), and f is strictly positive. □

This theorem asserts that the conditioning set that satisfies the relation $Y \not\perp X_i \,|\, S$ may be completely arbitrary. Therefore, no search method other than exhaustively examining all sets S can possibly determine whether X_i is weakly relevant. Since discovering S^A requires that we determine this for every X_i, the following corollary is immediate.

Corollary 8.2. *Determining S^A requires exhaustive subset search.*

Exhaustive subset search is widely regarded as an intractable problem, and no polynomial algorithm is known to exist. While the exact complexity class of subset search is not known (Peter Jonsson, Linköping University, personal communication), it is clear that finding S^A is intractable. This fact is illustrative in comparison with Theorem 6.5; the

```
Function RIT(Y, X)
Input: target node Y, features X
    Let S = ∅;
    foreach X_i ∈ X do
        if X_i ⊥̸ Y | ∅ then
            S = S ∪ {X_i};
        end
    end
    foreach X_i ∈ S do
        S = S ∪ RIT(X_i, X \ S);
    end
    return S
end
```

Figure 8.1: The Recursive Independence Test (RIT) algorithm.

problem of finding S^* is tractable for strictly positive distributions precisely because S^* does *not* include weakly relevant features.

Since the restriction to strictly positive distributions is not sufficient in this case, we must look for additional constraints. In the following sections I propose two different polynomial-time algorithms for finding S^A, and prove their consistency.

8.2 The Recursive Independence Test algorithm

8.2.1 Outline

In this section I describe a simple algorithm named Recursive Independence Test (RIT) based on pairwise tests for marginal (in)dependencies. The algorithm pseudocode is given in Figure 8.1. In the first round, RIT tests for the marginal dependencies $X_i \not\perp Y \mid \emptyset$ for each gene X_i and obtains a corresponding gene set S. Next, for each $X_i \in S$ we recursively call RIT to test for the marginal dependencies $X_i \not\perp X_j \mid \emptyset$ against each gene $X_j \notin S$, and add the significant findings to S. We continue in this fashion until no more dependencies are found.

An illustrating example of the RIT algorithm is given in Figure 8.2. Here

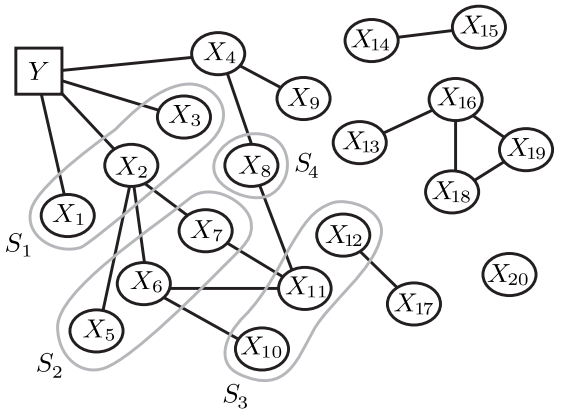

Figure 8.2: Example illustrating the RIT algorithm. Edges (solid lines) denote marginal dependencies between genes X_i (circles) and the class label variable Y (square). Gene sets found in each round of RIT are denoted $S_1, \ldots, S_4$. The final output of the algorithm is the union of these.

we have $S^A = \{1, \ldots, 11\}$ with $X_1, \ldots, X_4$ being marginally dependent on Y. The remaining features are irrelevant to Y. In the first round of RIT we obtain the set S_1. In this contrived example we obtain a S_1 which differs from U, which of course may happen for small sample sizes since the statistical tests used have limited power. In the next recursion, RIT tests every feature in S_1 against $X_{\neg S_1} = X_{4,\ldots,20}$; this discovers the set S_2, which is dependent on X_2. Continuing the recursion, RIT eventually finds two more feature sets S_3, S_4, after which no more significant features are found and the algorithm terminates. In S_3 we obtain a false positive X_{12}, and since $4 \notin S_1$, we also fail to detect X_9 because the required test is never made.

Since the RIT algorithm makes up to $n = \dim X$ tests for each element of S^A found, in total RIT will evaluate no more than $n|S^A|$ tests. Thus, for small S^A the number of tests is approximately linear in n, although the worst-case complexity is quadratic.

There are many possible alternatives as to what independence test to use. A survey of useful tests is found in Section 4.1.1. A popular choice in Bayesian networks literature is Fisher's Z-test, which tests for linear correlations and is consistent within the family of jointly Gaussian distributions [90]. In general, a different test may be needed for testing $Y \perp X_i \,|\, \emptyset$ than for testing $X_i \perp X_j$, since the domains $\mathcal{X}$ and $\mathcal{Y}$ may

differ. For discrete Y and jointly Gaussian X, a reasonable choice is Student's t-test, which is known to be optimal in this special situation [21]. For other possibilities, see Section 4.1.1.

8.2.2 Asymptotic correctness

I will next prove that RIT is consistent for the following class of distributions.

Definition 8.3. *The PCWT distribution class is the set of data distributions which satisfy the probability axioms of strict positivity (P), composition (C) and weak transitivity (WT), as defined in Section 2.2.*

The proof of asymptotic correctness of RIT for PCWT distributions rests on the following theorem.

Theorem 8.4. *For any PCWT distribution, let $R \subseteq V_n$ denote the feature set for which there exists a sequence $Z_{1:m} = \{Z_1, \dots, Z_m\}$ between $Z_1 = Y$ and $Z_m = X_k$ for every $k \in R$, such that $Z_i \not\perp Z_{i+1} \,|\, \emptyset$, $i = 1, \dots, m-1$. Then $R = S^A$.*

Proof. Let $I = V_n \setminus R$ and fix any $k \in I$. Since $Y \perp X_k \,|\, \emptyset$ and $X_i \perp X_k \,|\, \emptyset$ for any $i \in R$, we have $\{Y\} \cup X_R \perp X_I \,|\, \emptyset$ by the composition property. Then $Y \perp X_k \,|\, X_S$ for any $S \subseteq V_n \setminus \{k\}$ by the weak union and decomposition properties, so X_k is irrelevant; hence, $S_A \subseteq R$.

For the converse, fix any $k \in R$ and let $Z_{1:m} = \{Z_1, \dots, Z_m\}$ be a shortest sequence between $Z_1 = Y$ and $Z_m = X_k$ such that $Z_i \not\perp Z_{i+1} \,|\, \emptyset$ for $i = 1, \dots, m-1$. Then we must have $Z_i \perp Z_j \,|\, \emptyset$ for $j > i+1$, or else a shorter sequence would exist. We will prove that $Z_1 \not\perp Z_m \,|\, Z_{2:m-1}$ for *any* such shortest sequence, by induction over the sequence length. The case $m = 2$ is trivial. Consider the case $m = p$. Assume as the induction hypothesis that, for any $i, j < p$ and any chain $Z_{i:i+j}$ of length j, it holds that $Z_i \not\perp Z_{i+j} \,|\, Z_{i+1:i+j-1}$. By the construction of the sequence $Z_{1:m}$ it also holds that

$$Z_1 \perp Z_i \,|\, \emptyset, \quad 3 \le i \le m \qquad \Longrightarrow \qquad Z_1 \perp Z_{3:i} \,|\, \emptyset \tag{8.2}$$

(composition)

$$\Longrightarrow \qquad Z_1 \perp Z_i \,|\, Z_{3:i-1}. \tag{8.3}$$

(weak union)

Now assume to the contrary that $Z_1 \perp Z_p \,|\, Z_{2:p-1}$. Together with (8.3), weak transitivity implies

$$Z_1 \perp Z_2 \,|\, Z_{3:p-1} \;\vee\; Z_2 \perp Z_p \,|\, Z_{3:p-1}.$$

The latter alternative contradicts the induction hypothesis. The former together with (8.2) implies $Z_1 \perp Z_{2:p-1} \,|\, \emptyset$ by contraction, which implies $Z_1 \perp Z_2 \,|\, \emptyset$ by decomposition. This is also a contradiction; hence $Z_1 \not\perp Z_p \,|\, Z_{2:p-1}$, which completes the induction step. Thus X_k is relevant and $R \subseteq S^A$. The theorem follows. □

The above theorem asserts that RIT is correct if each marginal independence test is exact. Assuming consistent tests, we obtain the following corollary.

Corollary 8.5. *For any PCWT distribution and any consistent marginal independence tests ϕ_{ij} for $H_0^{ij} : X_i \perp X_j$ and ϕ_i for $H_0^i : Y \perp X_i$, the RIT algorithm is consistent.*

Proof. The RIT algorithm makes n tests ϕ_i and no more than n^2 tests ϕ_{ij}. Since the tests are consistent, to each $\delta > 0$ we can find an l such that for a data set $Z^{(1:l)}$ it holds that

$$P(\forall i, j : \phi_i = 1 \,|\, H_0^i \;\wedge\; \phi_{ij} = 1 \,|\, H_0^{ij}) < \delta.$$

Thus the RIT algorithm will discover every sequence $Z_{1:m} = \{Z_1, \ldots, Z_m\}$ between $Z_1 = Y$ and $Z_m = X_k$ with probability $1 - \delta$. It then follows from theorem 8.4 that for the set S returned by the RIT algorithm,

$$P\left(S = S^A\right) \geq 1 - \delta.$$

□

Hence, as the sample size l increases, the RIT algorithm will eventually produce a correct estimate of the set S^A with high probability, provided that the PCWT assumption holds for the data distribution. Next, I will demonstrate that the PCWT class is a reasonable model for measurements of biological systems.

8.2.3 Biological relevance of the PCWT class

Since cellular systems are believed to be well described by complex networks [95], it is reasonable to assume that the distribution of *all* variables X' comprising a cellular network (transcripts, proteins, metabolites, etc.) can be assumed to be faithful to a *Bayesian network* [55].

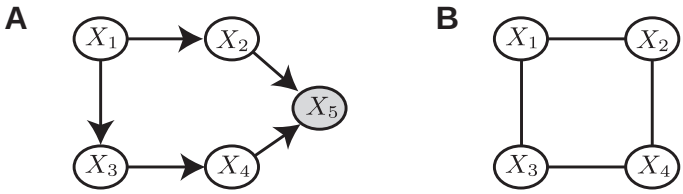

Figure 8.3: DAG-faithful distributions are not closed under conditioning. A: Graphical model of a DAG-faithful distribution over $X_1, \ldots, X_5$. Conditioning on node X_5 in (A) here results in the distribution $X_{1234} \,|\, X_5 = x_5$. B: Markov network (perfect undirected map) of the conditional distribution. No DAG is a perfect map of this distribution (see Section 2.2).

The following theorem, given by Pearl [130], asserts that the PCWT class contains all data distributions associated with such networks.

Theorem 8.6. *Any strictly positive distribution faithful to a Bayesian network is PCWT.*

However, in practise we typically cannot measure all variables X', but merely a subset X; for example, with microarrays we can perhaps measure most transcripts, but certainly no proteins or metabolites. Unfortunately, this means that in many cases the distribution of $X \subset X'$ will *not* be faithful to a Bayesian network, even though that of X' is. An example illustrating this problem is given in Figure 8.3; see also Chickering and Meek [25]. Nevertheless, the next theorem asserts that X is still PCWT.

Theorem 8.7. *Let X' be a random vector with a PCWT distribution and let S, T be any two disjoint subsets of the components of X'. Then the distribution of $X = (X' \setminus \{S, T\} \,|\, T = t)$ is also PCWT.*

A proof is given by Peña et al. [132]. Theorem 8.7 states that for PCWT distributions, we may fix some variables T to constant values t and ignore other variables S, and the remaining variables will still form a PCWT distribution. Thus, given that the distribution of all variables X' comprising the cellular network are PCWT, then any measurements X we make will also have a PCWT distribution, even though we fail to measure many variables of the system and perhaps fix others to constant values by experimental design. From this I conclude that PCWT is a realistic distribution class for biological data.

8.2.4 MULTIPLICITY AND SMALL-SAMPLE ERROR CONTROL

While the consistency is an important necessary condition for a "sound" algorithm, it is still merely an asymptotic result, and far from satisfactory for the small sample sizes typical for microarray data. With small samples and high-dimensional X, large amounts of tests will be made, and it is necessary to properly adjust for multiplicity, or else many findings are likely to be false positives (Section 4.1.2). While the multiplicity problem has been thoroughly investigated for univariate tests [5], our situation is more complicated since RIT performs multiple iterations of testing, and also chooses which tests to make in each round depending on the outcome of previous iteration.

To obtain a version of RIT with multiplicity control, we will require a correction procedure that, for ordered p-values $p_{(1)} \leq p_{(2)} \leq \dots p_{(n)}$ and a given significance level α, produces k (possibly zero) corrected p-values $\tilde{p}_1, \dots, \tilde{p}_k$ that satisfy

$$P(\tilde{p}_i \leq \alpha \,|\, H_0^i) \leq \alpha, \;\; i = 1, \dots, k, \tag{8.4}$$

where H_0^i is the corresponding null hypothesis. That is, we require (8.4) to hold for each of the "top genes" chosen. This is a weaker requirement than FWER control, but slightly stronger that FDR control. We next prove that the procedure of Benjamini and Hochberg [12] satisfies the above.

Lemma 8.8. *Assume $p_1, \dots p_{n_0}$ are independent p-values corresponding to true null hypotheses $H_0^1, \dots H_0^{n_0}$, while $p_{n_0+1}, \dots p_n$ are p-values corresponding to the remaining false null hypotheses, taking any joint distribution on $[0,1]^{n-n0}$. Then the Benjamini-Hochberg procedure*

$$\tilde{p}_{(i)} = \frac{n p_{(i)}}{i}$$

satisfies

$$P(\tilde{p}_{(i)} \leq \alpha \,|\, H_0^i) \leq \alpha$$

Proof. Since the $p_1, \dots p_{n_0}$ are independent, the $p_{(i)} \,|\, H_0^i$ are order statistics of a $U(0,1)$ distribution. These are well known to be beta distributed,

$$p_{(i)} \sim Beta(i, n-i+1),$$

and therefore

$$\begin{aligned} P(\tilde{p}_{(i)} \leq \alpha \,|\, H_0) &= P(p_{(i)} \leq i\alpha/n \,|\, H_0) \\ &= I_{i\alpha/n}(i, n-i+1) \end{aligned}$$

where I_z is the regularized incomplete beta function. For all α, this function takes its largest value for $i = 1$, so it suffices to note that for all n,

$$I_{i\alpha/n}(1, n) = 1 - \left(1 - \frac{\alpha}{n}\right)^n \leq \alpha.$$

□

Other FDR-controlling procedures than the above could probably also be used for this purpose. However, it is currently not known whether (8.4) holds for any procedure that controls the FDR, and I have not attempted to prove it for other procedures.

To establish error rate control, we employ an induction argument. Fix an $\alpha \in [0, 1]$. Assume as the induction hypothesis that in the first **foreach** loop of the RIT algorithm (Figure 8.1) we have tested the null hypotheses $H_0^i = X_i \perp Y \,|\, \emptyset$ for each X_i and obtained corrected p-values $\tilde{p}_i$ satisfying (8.4) and a gene list $S = \{i : \tilde{p}_i \leq \alpha\}$. Now consider the recursive calls $\mathrm{RIT}(X \setminus S, X_i)$. For each $i \in S$, this will test the null hypotheses $H_0^{ij} = X_i \perp X_j \,|\, \emptyset$ for every $j \notin S$, producing the p-values p_{ij}. We now combine the previously obtained $\tilde{p}_i$ with these p_{ij} to obtain a single p-value p_j for each $j \notin S$. To accomplish this, note that by Theorem 8.4, $j \notin S^A$ is possible at this point only if, for *every* $i \in S$, either H_0^i *or* H_0^{ij} holds true. Hence, the null hypothesis for X_j is

$$H_0^j = \bigcap_{i \in S} (H_0^i \cup H_0^{ij}). \tag{8.5}$$

This situation is known in statistics as *intersection-union* testing [14, 149]. By the intersection-union method one can calculate p-values p_j for (8.5); correcting these using the Benjamini-Hochberg procedure then results in $\tilde{p}_j$ satisfying (8.4) and a new gene list. This completes the induction step. As the induction hypothesis is easily satisfied in the first round of testing, it follows by induction that the p-values produced by the RIT algorithm all satisfy (8.4). A formal statement and proof of this fact follows.

Theorem 8.9. *Assume that the distribution of* (X, Y) *is PCWT. For a given set* $S \subseteq V_n$ *and* $i \in S$*, let* p_i *be a p-value for the null hypothesis* $H_0^i : i \notin S^A$*. Choose a* $j \notin S$*, and let* p_{ij} *be p-values for* $H_0^{ij} = X_i \perp X_j \,|\, \emptyset$ *for each* $i \in S$*. Then the null hypothesis*

$$H_0^j = \bigcap_{i \in S} (H_0^i \cup H_0^{ij}) \tag{8.6}$$

holds true if $j \notin S^A$*, and*

$$p_j = |S| \min_{i \in S}(\max\{p_i, p_{ij}\}) \tag{8.7}$$

is a p-value for H_0^j*.*

Proof. Since the data distribution is PCWT, we know from Theorem 8.4 that

$$\exists i : i \in S^A \ \wedge \ X_i \not\perp X_j \,|\, \emptyset \implies j \in S^A.$$

Negating this, we obtain

$$j \notin S^A \implies \forall i : i \notin S^A \ \vee \ X_i \perp X_j \,|\, \emptyset.$$

Thus, equation (8.6) is a null hypothesis for $j \notin S^A$. Further, since p_i and p_{ij} are p-values, it holds that

$$P\left(p_i \leq \alpha \,|\, H_0^i\right) \leq \alpha \quad \text{and} \quad P\left(p_{ij} \leq \alpha \,|\, H_0^{ij}\right) \leq \alpha.$$

Equation (8.7) is now computed using the intersection-union method [14]. We find that

$$\begin{aligned}
P\left(p_j \leq \alpha \,|\, H_0^j\right) &= P\left(|S| \min_{i \in S}(\max\{p_i, p_{ij}\}) \leq \alpha \,|\, \bigcap_{i \in S}(H_0^i \cup H_0^{ij})\right) \\
&\leq \sum_{i \in S} P\left(\max\{p_i, p_{ij}\} \leq \alpha/|S| \,|\, H_0^i \cup H_0^{ij}\right) \\
&\leq \sum_{i \in S} P\left(p_i \leq \alpha/|S| \,|\, H_0^i \ \wedge \ p_{ij} \leq \alpha/|S| \,|\, H_0^{ij}\right) \\
&\leq \sum_{i \in S} \min\left\{P\left(p_i \leq \alpha/|S| \,|\, H_0^i\right), \ P\left(p_{ij} \leq \alpha/|S| \,|\, H_0^{ij}\right)\right\} \\
&\leq |S| \cdot \alpha/|S| = \alpha
\end{aligned}$$

which proves that p_j is a p-value for the null hypothesis (8.6). □

An version of the RIT algorithm that implements these corrections for error rate control is given in Figure 8.4.

8.2.5 Simulated data

To illustrate the above result and also to assess the statistical power of RIT as a function of the sample size, I conducted a simulation study. As

Function RIT(X, S, p, ϕ, α)
Input: data X, set $S \subset X$, p-values p, test ϕ, level α
  foreach $X_j \notin S$ **do**
    foreach $X_i \in S$ **do**
      Let $p_{ij} = \phi(X_i, X_j)$;
    end
    Let $p_j = |S| \min_i \max\{p_i, p_{ij}\}$; `// Theorem 8.9`
  end
  Let $p_j = |X \setminus S| p_j / r_j$; `// Theorem 8.8`
  Let $S' = \{X_j \notin S : p_j \leq \alpha\}$;
  if $S' \neq \emptyset$ **then**
    $p = \text{RIT}(X \setminus S, S', p, \phi, \alpha)$;
  end
  return p ; `// Result is modified vector`
end

Figure 8.4: Modified version of the RIT algorithm, implementing corrections for error control.

in Chapter 5, multivariate Gaussian distributions were used. Here $n = 1,000$, of which $|S^A| = 100$, of which 50% were differentially expressed (differed in the marginal distribution $X_i \not\perp Y \,|\, \emptyset$). To this end, a 4-dimensional Gaussian distribution was used, with class-dependent mean vector $\mu_y = 2y \cdot (0, 0, 1, 1)$ and covariance matrix

$$\Sigma = 4 \cdot \begin{pmatrix} 2 & 1 & 1 & 1 \\ 1 & 2 & 2 & 2 \\ 1 & 2 & 5 & 5 \\ 1 & 2 & 5 & 8 \end{pmatrix},$$

equal for both classes. The full distribution over S^A was then constructed using 25 of these 4-dimensional blocks. The remaining features had the same covariance matrix but had mean $\mu = (0, 0, 0, 0)$. Sample sizes $l = 10, 20, 30, \ldots, 100$ were tested.

We compared the performance of RIT against a typical univariate test, namely Student's t-test [21] with the Benjamini-Hochberg correction [12]. We also compared the method against the popular Recursive Feature Elimination (RFE) feature selection method [68].

Figure 8.5 summarizes the results of this experiment. We find that RIT does indeed control the FDR at the nominal level ($\alpha = 0.05$), in the same way as the univariate t-test. The power of the t-test converges to 0.5

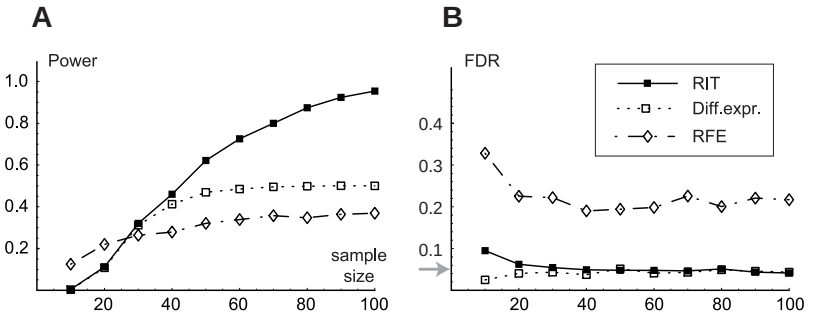

Figure 8.5: Simulation results for the RIT algorithm, differential expression using the t-test, and RFE. Left, statistical power (1− false negative rate) as a function of sample size. Right, false discovery rate (FDR) as a function of sample size. Grey arrow marks the nominal FDR = 0.05.

as expected (since only half of the relevant features S^A were marginally relevant), while the RIT converges to 1.0, in agreement with the above theoretical results. This indicated that, when multivariate effects are present, RIT affords more power than the univariate test at the same FDR level.

In contrast, the RFE method clearly does not control the FDR, choosing many genes unrelated to Y. This is in agreement with the results of Chapter 5. RFE also displays low power, most likely because it considers some of the features in S^A to be "redundant" for prediction and consequently ignores these. Indeed, it can be verified that only half of S^A is in S^* in this case, which roughly agrees with the power curve for RIT. Further, the previous results of Chapter 5 indicates that similar behavior is to be expected from other feature selection methods. I conclude that it is feasible to apply the RIT algorithm to small-sample data while controlling the FDR at the desired level. Note that sample size requirements cannot be inferred from Figure 8.5 however, as this depends the particular data distribution, in particular the fraction of MDE genes and the amount of noise.

8.2.6 MICROARRAY DATA

We next tested the RIT algorithm on two publicly available microarray data sets (see methods section). A diabetes data set was obtained

from the study by [66]. It is publicly available through the Diabetes Genome Anatomy Project, `http://www.diabetesgenome.org`. This data set contrasts human pancreas islets expression from 7 normal *vs.* 15 type 2 diabetic patients. The original data comprises 44,928 probesets from the Affymetrix U133A and B chips. We used only the A chip in our experiments, since we needed to evaluate our results against literature and the A chip contains better annotated sequences. We also filtered genes by variance before any further analysis, keeping only the 5,000 most variable genes.

The second data set derives from a breast cancer study by van't Veer et al. [179]. This consists of $l = 78$ samples from patients divided into one "good prognosis" group ($l = 44$) and one a "poor prognosis" group ($l = 34$) based on the time until relapse [179]. The data set is publicly available at `www.rii.com/publications/2002/vantveer.html`. The microarrays used contains approx. 25,000 transcripts, out of which 4,918 were selected using the same quality filter as in the original publication.

For the diabetes data, the original study identified 370 genes as differentially expressed using the t-test, but this unfortunately did not account for multiplicity [66]. Using the Benjamini-Hochberg correction, only the top three genes of the original study were declared to be significant: *Arnt*, *Cdc14a*, and *Ddx3y*. The RIT algorithm identified an additional 21 transcripts as relevant, of which 5 were unknown EST:s and 16 were known genes according to the NCBI Gene database [114]. Naturally, RIT is an hypotheses-generating method, and the discovered genes may or may not be of *functionally* related to the target variable even though they are statistically dependent. We therefore conducted a literature validation of the 16 known genes (Table 8.1) to search for possible biologically important findings. Five of these (31%) were previously associated with diabetes. Among the remaining 11 novel genes, several give rise to interesting hypotheses: dor example, *Dopey1* was recently shown to be active in the vesicle traffic system, the mechanism that delivers insulin receptors to the cell surface. Four genes encoded transcription factors, as do the majority of previously discovered diabetes-associated genes [66]. The *Usp9y* gene discovered by RIT is associated with male infertility and Sertoli cell-only syndrome. Interestingly, so is the differentially expressed *Ddx3Y* gene; this is unlikely to be a coincidence as only 6 human genes were annotated with this function in NCBI Gene. This is an example of general tendency we have observed in our experiments, that the additional genes with multivariate dependencies discovered by RIT often are functionally related to the marginally dependent genes. This seems reasonable, given that RIT relies on pairwise independence

Gene	Status	Function	Ref. (PMID)
Bcat1	D	Candidate gene for the type I diabetes susceptibility locus Idd6.	14563018
Clca2	N	Chloride channel. Multiple functions, colocalizes with adhesion molecule integrin β_4.	15707651
Clcn1	N	Chloride channel. Regulates electric excitability of the skeletal muscle membrane.	7735894
Cltb	D	Involved in insulin receptor internalization.	7821727
Dopey1	N	Involved in Golgi/ER vesicle traffic.	1630131
Epha7	N	Ephrin receptor subfamily. Ephrin receptors are associated with the pancreatic islets.	15681844
Hcap-G	N	Chromosome condensation protein, member of the condensin complex.	14593730
Kcng1	D	Potassium channel. Potassium channels are involved in regulation of insulin secretion.	16567526
Loc51152	N	Melanoma antigen.	
Psmal	D	Glutamate carboxypeptidase. Inhibition prevents long-term type 1-diabetic neuropathy.	11809162
Sox5P	N,T	Transcription factor activity (GeneOntology). Related to the sex determining region Y-gene.	
Sptlc2	N	Sphingolipid biosynthesis enzyme, upregulated upon transepidermal water loss.	12445191
Ssx2	N,T	Suggested role as transcriptional repressor.	12007189
Tfap2B	D,T	Genetic variations in this gene are associated with type 2 diabetes.	15940393
Usp9Y	N	Associated with male infertility and Sertoli cell-only (SCO) syndrome.	12871878
Zf	N,T	Specific inhibitor of the transcription factor Luman. Inhibits herpes virus replication.	16282471

Table 8.1: Genes with multivariate expression patterns discovered by the RIT algorithm for the diabetes data. Status column: D, diabetes-related; N, novel; T, transcription factor

tests. The chloride channels *Clca2* and *Clcn1* are also highly interesting findings, as ion channels in pancreas islets has been shown to regulate insulin secretion [139]. The diabetes-associated potassium ion channel *Kcng1* was also discovered by RIT, strengthening this hypothesis.

For the breast cancer data set we observed large amounts of pairwise correlations among genes, resulting in a highly connected dependence graph. This is an example where the set S^A is problematic as a hypothesis class. To limit the number of findings in this case, we required significant correlations to exceed a threshold 0.85 to be considered by RIT (see discussion below). The original study identified a set of 70 cancer-related genes using a variable ranking method [179]. In addition to these, the RIT algorithm identified 43 relevant genes (Table 8.2). Literature validation revealed that 23 of these (53%) had a previously known function in cancer development, whereof 6 (14%) were specifically implicated in breast cancer (table 2). An additional 10 transcripts (23%) were cell cycle-related and may also be considered as potential cancer proliferation genes. Our literature validation thus confirmed 39 (77%) of the genes reported by RIT to be cancer-related. The higher percentage in this case as compared to the diabetes data may reflect the fact that more genes are known for cancer than for diabetes. To assess the amount of cancer annotations among the 5,000 genes considered, we examined an additional 43 genes chosen at random. Indeed, we found that about 25% of all genes had some cancer or cell cycle-related annotation. Nevertheless, the above fraction of 77% validated genes is highly significant with a Binomial tail p-value $< 10^{-20}$.

Encouraged by the above result, we set out to investigate the remaining 10 genes that were not previously associated with cancer. We found three histone proteins, which may be related to chromatin remodelling. One novel gene *Gpr116* was recently identified as a G-protein with a potential role in immune response. The novel gene *Prr11* had predicted binding sites for the transcription factor E2F, which in turn is known to be crucial in the control of tumorigenesis. *Ube2s* is an essential component of the ubiquitin-protein conjugation system, which is implicated in various cancer forms. This gene is also functionally related to the known cancer gene *Ube2c*, which also was discovered by RIT. Also interesting were the novel proteins *Depdc1* and *Depdc1b*, both containing RhoGAP domains. This may implicate them in the regulation of various Rho GTPases, which are currently being investigated as cancer-therapy targets [56].

Gene	Status	Function	Ref. (PMID)
Anln	C	Over-expressed in diverse common human tumors, may have potential as biomarker.	16203764
Aurka	B	Cell cycle-regulated kinase, possible prognostic indicator for patients with breast tumors.	12833450
Aurkb	C	Highly expressed in high-grade gliomas, correlated with malignancy and clinical outcomes.	15072448
Birc5	B	Prevents apoptotic cell death, differentially expressed in breast cancer.	16142334
Blm	C	Cooperates with p53 in regulation of cell growth, associated with colorectal cancer.	11781842, 12242432
Brrn1	CC	Required for the conversion of interphase chromatin into condensed chromosomes.	
Bub1	C	Mutations in this gene have been associated with aneuploidy and several forms of cancer.	15931389
Ccnb1	B	Detected in various human breast cancer cell lines and breast tumor tissues.	11779217
Ccnb2	CC	Essential component of the cell cycle regulatory machinery.	
Cdc20	C	Regulatory protein in the cell cycle, associated with gastric cancer.	15701830
Cdc25A	C	Known oncogene, required for progression from G1 to the S phase of the cell cycle.	14673957
Cdc45L	CC	Important for early steps of DNA replication in eukaryotes, loss may affect cell proliferation.	9660782
Cdca8	CC	Required for stability of the bipolar mitotic spindle.	
Depdc1	N	Contains domain of unknown function often present together with the RhoGAP domain.	
Depdc1B	N	Unknown function, has conserved RhoGAP domain (GTPase-activator protein).	
Dlg7	C	Potential oncogenic target of AURKA, may play a role in human carcinogenesis.	15987997, 12527899
Exo1	C	Possible cancer predisposing gene.	15328369
Fam64A	N	Unknown function.	
Fbxo5	CC	Function in ubiquitination, inhibits the anaphase promoting complex.	
Foxm1	C	Stimulates the proliferation of tumor cells.	16489016
Gpr116	N	Has two immunoglobulin-like repeats, may have a role in the immune system.	12435584
H2Afz	C	Chromatin remodeling at the c-myc oncogene involves the local exchange of this histone.	15878876
Hist1H1B	N	Histone protein.	
Hist1H1E	N	Histone protein.	
Hist1H4B	N	Histone protein.	
Kif20A	C	Required for cytokinesis, related to AURKB. Likely to be involved in pancreatic cancer.	15263015, 15665285
Kif23	CC	Interacts with CYK4, to form the central spindle complex. Essential for cell division.	11782313
Kif2C	CC	Important for anaphase chromosome segregation.	
Kifc1	CC	Involved in localization of PLK1, AURKB, and CDC14A during anaphase.	15263015
Mad2L1	C	Mitotic checkpoint gene, involved mainly in colorectal carcinogenesis.	12970887
Nek2	B	Significantly up-regulated in breast carcinomas.	15492258
Pitrm1	N	Novel member of the metalloendoprotease superfamily.	10360838
Prr11	N	Unknown function. Predicted interactions with E2F, which is involved in cancer.	16437386
Pttg2	C	Potent oncogene, expressed at high levels in various human tumors and tumor cell lines.	10806349
Racgap1	B	Implicated in in breast cancer cell proliferation	15863513
Rad54L	B	Candidate oncosupressor in breast or colon carcinomas, lymphomas and meningiomas.	12614485
Spbc25	CC	Essential kinetochore component, significant role in mitotic events	14699129
Stil	C	Involved in mitosis and in increased mitotic activity in tumor cells.	
Tk1	C	Marker for non-small cell lung cancer. May be important in epithelial ovarian cancer.	15809747, 11992400
Tpx2	C	May be important in both progression lung cancer, possible prognostic predictor.	16489064
Ttk	CC	Required for centrosome duplication and for the normal progression of mitosis.	15618221, 14657364
Ube2C	C	Required for destruction of mitotic cyclins. Highly expressed in human primary tumors.	12874022
Ube2S	N	Essential component of the ubiquitin-protein conjugation system.	15454246

Table 8.2: Genes with multivariate expression patterns discovered by the RIT algorithm for the breast cancer data. Status column: B, Breast cancer-specific; C, Cancer-related; CC, Cell cycle-related; N, novel.

8.2.7 Discussion

The RIT algorithm is a principled, general approach that increases the power of small-sample, genome-wide expression studies by considering not only univariate differential expression but also multivariate effects. RIT may thus be very useful in situations where little univariate differential expression is observed. However, the RIT algorithm itself does not address the conceptually different problem we encountered in the breast cancer data set: since cancer is associated with major transcriptional changes, a large fraction of the genes were found to be relevant. Indeed, for cancer data sets, even the fraction of differentially expressed genes has previously been estimated to be on the order of 50% of all genes [163], and the set S^A is presumably much larger, perhaps encompassing most of the known genome.

With small sample sizes this does not impose a practical problem, since typically only a small fraction of the relevant genes can be called significant; but as power increases, a principled approach for prioritizing among all relevant genes is urgently needed. For the cancer data, we had RIT prioritize the findings by considering stronger correlations to be more important. This is not entirely unreasonable, and we were able to confirm the end results in this case against the literature. However, the problem ultimately does not lie with the inference method; rather, the notion of relevance that defines S^A is simply not useful in this case. To obtain a more precise definition of relevance that admits reasonable number of candidate features, it seems necessary to integrate other kinds of information into the analysis [60]. A possible step towards a principled solution building upon the present work would be to combine the independence tests used here with other data sources and prior beliefs (perhaps in the form of Bayesian probabilities) to guide the RIT algorithm towards more "interesting" genes.

In the studies presented in this section have been limited to two-class data. However, it is straightforward to extend the RIT algorithm to find multivariate expression patterns with other types of target variables, such as multiple classes data or continuous target variables such as survival times. To accomplish this, only the independence tests used need to be replaced. This "modularity" is a useful property of RIT: to handle different situations, it is sufficient to "plug in" different independence tests. For example, a continuous target variable could be handled by using the Fisher z-transformation also for testing $X_j \perp Y$. More complex, non-linear independence relations may be handled using nonparametric tests such as the Kolmogorov-Smirnov test [138] or kernel-based tests

Function RMB(X, Y, V)
Input: target node Y, data X, visited nodes V
  Let $S = M(Z^{(1:l)})$, the estimated Markov boundary of Y in X;
  foreach $X_i, i \in S \setminus V$ **do**
    $S = S \cup \text{RMB}(Y, X_{\neg i}, V)$;
    $V = V \cup S$
  end
  return S
end

Figure 8.6: Recursive Markov Boundary (RMB)

[65]. See Section 4.1.1.

Dynamic (time-series) data could also be considered, although some additional assumptions may be necessary to ensure PCWT distributions in this case. For example, assuming a Markov condition, time-series data can be modelled using Dynamic Bayesian Networks (DBNs) [55]. The DBN methodology essentially transforms a dynamic model over n nodes into an ordinary BN over $2n$ nodes. Thus, DBNs also result in PCWT distributions as described herein (albeit of twice the dimensionality) and RIT is therefore applicable to detecting multivariate changes in dynamic as well as in static data.

8.3 The Recursive Markov Boundary algorithm

8.3.1 Outline

In this section I consider a second algorithm for discovering S^A called Recursive Markov Boundary (RMB). Instead of the pair-wise tests employed by RIT, this algorithm based on a given estimator of Markov boundaries of Y. The idea here is that existing feature selection methods may be re-used for Markov boundary estimation as a component of this algorithm. In this section, we need to consider Markov boundaries with respect to different feature sets. I denote a Markov boundary of Y with respect to a feature set S as $M^*(S) \in 2^S$.

Assume that for every $M^*(S)$ we have an estimate $M(Z_S^{(1:l)})$. Briefly, the

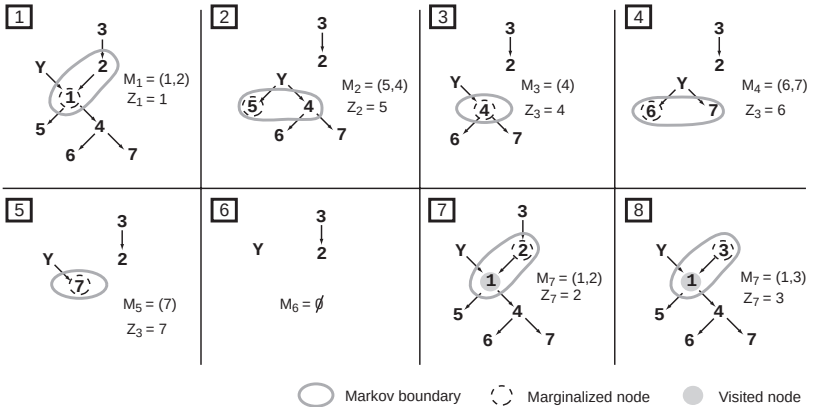

Figure 8.7: A contrieved example of the RMB algorithm for a distribution faithful to the DAG shown in black arrows. Numbers denote the relevant features, Y denotes the target variable. As in theorem 8.12, M_i denotes Markov boundaries and Z_i denotes marginalized nodes. Note that the marginal distributions from step 2 onwards may not be DAG-faithful, so absence of arrows should not be read as independencies.

RMB algorithm first computes the estimate $M(Z^{(1:l)})$, then $M(Z^{(1:l)}_{\neg i})$ for each $i \in M(Z^{(1:l)})$, and so on recursively until no more nodes are found (Figure 8.6). For efficiency, we also keep track of previously visited nodes V to avoid visiting the same nodes several times. We start the algorithm with RMB$(X, Y, V = \emptyset)$. A contrived example of the RMB algorithm for a DAG-faithful distribution is given in Figure 8.7. Note that the recursive formulation allows the algorithm to "backtrack" as in step 7 in this example.

8.3.2 Asymptotic correctness

Next, I prove that also the RMB algorithm is consistent for any PCWT distribution, assuming that the estimator of Markov boundaries used is consistent. The proof relies on the existence of a minimal independence map (see Section 2.2) for the distribution. We need only require the graph over X to be a minimal I-map (not over $X \cup \{Y\}$), so that this minimal I-map is unique for any positive distribution satisfying $f(x) > 0$. [130]. The proof is similar in spirit to that for RIT; we will develop two

lemmas assuming that the estimators $M(Z_S^{(1:l)})$ are correct, so that we may substitute these for the true Markov boundaries $M^*(S)$. Consistency then follows for consistent estimators since the number of Markov boundaries estimated is finite.

Lemma 8.10. *For any PCWT distribution, and any shortest path* $Z_{1:m} = \{X_{i_1}, \ldots X_{i_m}\}$ *between* $Z_1 = X_{i_1}$ *and* $Z_m = X_{i_m}$ *in the undirected minimal I-map over* X*, it holds that* $Z_1 \not\perp Z_m \,|\, X \setminus Z_{1:m}$.

Proof. Since $f(x) > 0$, the undirected minimal I-map over X is unique. The proof proceeds by induction. For $m = 2$, the lemma follows immediately from the definition of the minimal I-map [130]. Also, it holds that

$$Z_i \not\perp Z_{i+1} \,|\, X \setminus Z_{i,i+1} \tag{8.8}$$

$$Z_i \perp Z_j \,|\, X \setminus Z_{i,j}, \quad j > i+1. \tag{8.9}$$

Take any distinct $Z_{i,i+1}, Z_k$ and assume that $Z_i \perp Z_{i+1} \,|\, X \setminus Z_{i,i+1,k}$. Then $Z_i \perp Z_{i+1,k} \,|\, X \setminus Z_{i,i+1,k}$ by contraction with (8.9), and therefore $Z_i \perp Z_{i+1} \,|\, X \setminus Z_{i,i+1}$ by weak union. This contradicts (8.8), so we conclude that

$$Z_i \not\perp Z_{i+1} \,|\, X \setminus Z_{i,i+1,k}. \tag{8.10}$$

Next, take any sequence $Z_{i:i+2}$. Applying (8.10), we obtain $Z_i \not\perp Z_{i+1} \,|\, X \setminus Z_{i:i+2}$ and $Z_{i+1} \not\perp Z_{i+2} \,|\, X \setminus Z_{i:i+2}$. Using weak transitivity implies either $Z_i \not\perp Z_{i+2} \,|\, X \setminus Z_{i:i+2}$ or $Z_i \not\perp Z_{i+2} \,|\, X \setminus Z_{i,i+2}$. The latter alternative contradicts (8.9), so we conclude

$$Z_i \not\perp Z_{i+2} \,|\, X \setminus Z_{i:i+2}. \tag{8.11}$$

Finally, take any Z_i, Z_j, Z_k such that neither Z_i, Z_j nor Z_j, Z_k are consecutive in the path $Z_{1:m}$. Using (8.9) with the intersection (Theorem 2.10) and and decomposition (Theorem 2.9), properties, we find

$$\left.\begin{array}{l} Z_i \perp Z_j \,|\, X \setminus Z_{i,j} \\ Z_j \perp Z_k \,|\, X \setminus Z_{j,k} \end{array}\right\} \implies Z_i \perp Z_j \,|\, X \setminus Z_{i,j,k}. \tag{8.12}$$

Equations (8.10),(8.11) and (8.12) show that the properties (8.8) and (8.9) hold also for the shortened path $Z'_1, \ldots, Z'_{m-1}$ given by $Z'_1 = Z_1$ and $Z'_i = Z_{i+1}, 2 \leq i < m$ (*i.e.*, removing Z_2). The lemma follows from (8.8) by induction. □

Lemma 8.11. *For any PCWT distribution, a feature* X_{i_m} *is relevant iff there exists a path* $Z_{1:m} = \{X_{i_1}, \ldots X_{i_m}\}$ *in the minimal I-map of* $f(x)$ *between* i_m *and some* $i_1 \in M^*$*. In particular, for such a path it holds that* $Y \not\perp Z_m \,|\, X \setminus Z_{1:m}$.

Proof. If $i_m \in M^*$ (that is, $m = 1$), the lemma is trivial. Consider any $i_m \notin M^*$. First, assume that there exists no path $Z_{1:m}$. Then $Z_m \perp X_{M^*} \,|\, X_S$ for any $S \subseteq V_n \setminus M^* \setminus \{i_m\}$. Fix such an S. Since $Z_m \perp Y \,|\, X_{M^* \cup S}$, contraction and weak union gives $Z_m \perp X_M \,|\, \{Y\} \cup X_S$. Again using $Z_m \perp X_{M^*} \,|\, X_S$, weak transitivity gives

$$Z_m \perp Y \,|\, X_S \;\vee\; Y \perp X_{M^*} \,|\, X_S.$$

The latter alternative is clearly false; we therefore conclude that $Z_m \perp Y \,|\, X_S$. Next, fix any $S' \subseteq M^*$. By decomposition, $Z_m \perp X_{M^*} \,|\, X_S \implies Z_m \perp X_{S'} \,|\, X_S$. Combining with the above result, by the composition property

$$\left.\begin{array}{c} Z_m \perp X_{S'} \,|\, X_S \\ Z_m \perp Y \,|\, X_S \end{array}\right\} \implies Z_m \perp \{Y\} \cup X_{S'} \,|\, X_S.$$

Finally, weak union gives $Z_m \perp Y \,|\, X_{S \cup S'}$. Since $S \cup S'$ is an arbitrary subset of $V_n \setminus \{i_m\}$, we conclude that Z_m is irrelevant.

For the converse, assume that there exists a path $Z_{1:m}$. By lemma 8.10, we have $Z_1 \not\perp Z_m \,|\, X \setminus Z_{1:m}$. Also, since $i_1 \in M^*$ and $\{i_2, \ldots, i_m\} \cap M^* = \emptyset$, it holds that $Y \not\perp Z_1 \,|\, X_S$ for any S that contains $M^* \setminus \{i_1\}$. In particular, take $S = V_n \setminus \{i_1, \ldots, i_m\}$. Weak transitivity then yields

$$\left.\begin{array}{c} Z_1 \not\perp Z_m \,|\, X \setminus Z_{1:m} \\ Y \not\perp Z_1 \,|\, X \setminus Z_{1:m} \end{array}\right\} \implies Z_m \not\perp Y \,|\, X \setminus Z_{1:m} \;\vee\; Z_m \not\perp Y \,|\, X \setminus Z_{2:m}.$$

But the latter alternative is false, since $X \setminus Z_{2:m}$ contains M^* by assumption. We conclude that $Z_m \not\perp Y \,|\, X \setminus Z_{1:m}$ so that Z_m is relevant. □

With these lemmas in place, we are now in a position to prove the main theorem of this section.

Theorem 8.12. *For any PCWT distribution such that a given estimator $M(Z_S^{(1:l)})$ of the Markov boundary of Y with respect to a feature set S is consistent for every $S \subseteq V_n$, the RMB algorithm is consistent.*

Proof. For every $S \subseteq V_n$, the marginal distribution over S is strictly positive, and therefore every Markov boundary $M^*(S)$ is unique by Theorem 3.2. Let G be the minimal I-map over the features X, and let $M_1 = M^*$. Fix any k in S^A. If $k \in M_1$, we know that X_k is found by RMB. Otherwise, by Lemma 8.11, there exists a shortest path $Z_{1:m}$ in G between some $i_1 \in M_1$ and $i_m = k$. We prove by induction over m that RMB visits every such path. The case $m = 1$ is trivial. Let

the induction hypothesis be that Z_p is visited. For Z_{p+1}, Lemma 8.11 implies $Y \not\perp Z_{p+1} \,|\, X \setminus Z_{1:p+1}$. Since Z_p is visited, RMB will also visit all nodes in $M_{p+1} = M^*(V_n \setminus \{i_1, \ldots, i_p\})$. However, M_{p+1} contains i_{p+1}, because it contains all i satisfying $Y \not\perp X_i \,|\, X \setminus Z_{1:p} \setminus \{X_i\}$ by theorem 6.3. □

It is easy to see that the RMB algorithm (Figure 8.6) requires computing $|S_A|$ Markov boundaries. We might attempt to speed it up by marginalizing out several nodes at once, but in that case we cannot guarantee consistency. A general algorithm for estimating Markov boundaries is given by Peña et al. [131]. This estimator is consistent assuming that $f(x) > 0$ and the composition property holds, so it follows that RMB is consistent in PCWT with this choice of $M(Z_S^{(1:l)})$.

At first sight, RMB may seem to be more computationally intensive that RIT. However, since the Markov boundary is closely related to S^* (Chapter 3), an approximate version of RMB may be implemented using existing methods for estimating S^* in place of $M(Z^{(1:l)})$. This approximation would of course be exact for distribution classes where the Markov boundary coincides with the set S^*. For example, this holds for two-class Gaussian distributions discussed previously (Section 3.1.2). From a computational point of view, SVM-based methods (Section 4.3) are an attractive option, as there exist efficient optimization methods for re-computation of the SVM solution (and thus the estimate $M(Z_S^{(1:l)})$) after marginalization [92].

8.4 Related work

I have not been able to find any previous work directly treating inference of the set S^A. The problem is somewhat related to inference of graphical probability models: for the class of DAG-faithful distributions, one can discover S^A by inferring a Bayesian network and then taking S_A to be the connected component of Y in that network. However, this is clearly less efficient than our direct approach, since Bayesian network inference is asymptotically NP-hard even in the rather restricted DAG-faithful class [26]. Certainly, such a strategy seems inefficient as it attempts to "solve a harder problem as an intermediate step" (by inferring a detailed model of the data distribution merely to find the set S^A), thus violating Vapnik's famous principle [180, pp. 39].

Some features selection methods originally intended for optimization of

predictor performance do in fact attempt to find all relevant features, since they do not rule out the weakly relevant ones. These include Focus [6], which considers the special case of binary $\mathcal{X}$ and noise-free labels; Relief [94], a well-known approximate procedure based on nearest-neighbors; and Markov blanket filtering [101, 190], which considers a particular case based on marginal dependencies (and is therefore fundamentally different from RMB, despite the similar name). All known methods are either approximate or have exponential time-complexity. None of these methods is known to be consistent. See Chapter 4 for more details on these algorithms.

8.5 Summary

In this chapter I have analyzed the problem of discovering the set S^A of all features relevant to a target variable Y. This problem has hitherto received little attention in the machine learning field. With the advent of new, major applications in genome-wide biology, where identifying features *per se* is often a more important goal than building accurate predictors, we anticipate that the set S^A may constitute an important inference problem, since many biologically important features may not be predictive (Section 3.3). I have herein provided a first analysis, showing that discovering S^A is harder than discovering the predictive features S^* — the problem is intractable even for strictly positive distributions (*cf.* Chapter 6). I have therefore proposed two consistent, polynomial-time algorithms for a more restricted distribution class, which I argue to be realistic for biological data.

Importantly, for the RIT algorithm I have also provided a method of error control, which enables a principled application to real data. This algorithm is sound, reasonably computationally efficient, and can utilize a wide array of hypothesis test to accommodate different data types or data distributions. However, application of RIT to real biological data also shows that the set S^A in some cases (*e.g.* cancer gene expression) is much too large to be useful as for hypothesis generation. In such cases, prior knowledge must be used to reduce the number of candidates. Currently, RIT is primarily useful in situations where standard differential expression tests yield few candidates. In comparison with gene set testing methods (Section 4.4), RIT is useful in that it still gives evidence for individual genes, not sets of genes. It may thus be viewed as a direct multivariate analogue of differential expression tests.

9

Conclusions

9.1 Model-based feature selection

In this thesis, I have tried to promote a *statistical* view of feature selection, where algorithms are based on a statistical data model. Historically, feature selection has to a large extent focused on algorithmic and computational issues. In my opinion, the definition of underlying models and precise, operational measures of feature relevance has often been treated casually or even been entirely neglected. A multitude of algorithms have been proposed, but since most are heuristic, it is often difficult to understand how they relate to each other, or what their exact purpose is. A typical example is the Relief algorithm [6], which has been largely successful on practical problems, while a precise understanding of its working principles is still lacking [102, 147].

One reason why feature selection has been effective in practise — in spite of little theoretical understanding — may be that the final "quality" of feature selection has always been easy to assess: the most important measure of the "success" of feature selection has been the final accuracy of the predictor, and possibly the dimension of the reduced data. Both of these criteria are straightforward to evaluate in practise for experimental data; all that is required is a separate data set for testing. Granted, such data sets may be expensive to obtain, but at least there is no major conceptual problem as to how to proceed.

This no longer holds true when feature selection is applied in life science. Previously, the selected features *per se* have not truly been important, as long as they are few and the resulting predictor is accurate. In solving biological and medical problems, however, the identity of the relevant features are of great importance, since they provide information about the underlying biological mechanisms. Therefore, it here becomes crucial to avoid false positives (and false negatives), so that the "candidates" generated by feature selection do not lead the researcher astray.

But what is a false positive? To answer this question, we must first ask "what is a positive?", that is, what is the precise meaning of "relevance". To precisely define "relevance", we require a data model. In this thesis, the data model is a statistical distribution over data (features) and the target variable. Thus, the requirements of biology inevitably lead to the problem of defining a data model and specifying the precise meaning of relevance. This issue was treated in Chapter 3, where different types of relevance were defined. Without a data model, the result of inference methods is hard to interpret.

Only with this foundation in place is it possible to construct sound algorithms which are able to discover the relevant features with statistical control over false positives. Thus, I conclude that feature selection algorithms for life science must necessarily be *model-based*. Without a data model, we cannot define relevance, the goal of the analysis; without a clearly defined goal, we cannot measure success. This insight, although it might seem trivial to the reader by now, has only recently surfaced in the feature selection field. Guyon et al. [69] lists among topics for future research "*to understand what problem [feature selection algorithms] seek to solve*". Hopefully, this thesis constitutes a step towards that goal.

A fundamental difference between optimizing predictive accuracy and controlling false positives is that the true features are always unknown in biology. Therefore, one can never *test* feature selection algorithms in this respect — there are no "test sets". Instead, one must *prove* — using model assumptions and mathematical arguments — that a particular algorithm cannot select excessive amounts of false positives. This cannot be done unless inference is model-based. This is a quite general conclusion that applies to many important inference problems in modern biology. An important example which has received much attention recently is gene network reconstruction [59, 167, 168]. Also here, current algorithms are to a large extent heuristic, their results are difficult to interpret, the relations between different algorithms are not clear, and there are no statistical methods for controlling error rates.

Another benefit of model-based feature selection is that the inference problems may be simplified. In machine learning, researchers seem reluctant to admit *any* assumptions on the data distribution. Certainly, one should be restrictive in making assumptions, and those made should be carefully justified; but in general, *some* assumptions are required to obtain tractable problems. A striking example of this is found in Chapter 6, where a mild restriction to strictly positive distributions is sufficient to turn an intractable problem into one which can be solved in polynomial time. Without assumptions, one must resort to heuristics; but heuristics are in a sense assumptions "in disguise". An example of this is the correspondence between loss/regularizer and conditional/prior distribution discussed in Section 2.7.3. In my opinion, this is more problematic, since it may imply assumptions that are not even understood (and may therefore be quite severe).

9.2 Recommendations for practitioners

The literature on feature selection is extensive and quite bewildering to the newcomer or experimentalist. Probably the most pressing question from a practical data analysis perspective is "which method should I choose for my problem?". Of course, the answer to this question depends on many aspects of the problem at hand. Below, I list ten points that may serve as a first guide to the methods available.

1. First, are you at all interested in the selected features *per se*, or is your goal an accurate predictive model? These are two different problems. If your goal is prediction, then probably there is no reason to perform feature selection in the first place. Any modern, regularized inducer such as the support vector machine [18] can be used directly. If it is suspected that the great majority of features are irrelevant, try a sparse (L_1) regularizer, *e.g.*, the Lasso [170] or the LP-SVM [57].

2. Is your data high-dimensional, that is, do you have many more features than samples? If not, then again the techniques described herein are of less interest. For low-dimensional data, handling nonlinearities is typically more important, and the best bet is probably a regularized kernel method [27, 153].

3. Are you particularly interested in predictive features? If so, you might consider the bootstrap method described in Chapter 7. Care-

fully choose a predictive model with reasonable assumptions. Try simple methods first, *e.g.*, the Naive-Bayes classifier (Example 2.6).

4. Are you mainly interested in marginal dependencies? Do you have a particular dependency in mind (*e.g.*, differential expression)? If so, choose an appropriate univariate statistical test (Section 4.1.1). Carefully examine whether the assumptions of your test is reasonable in your situation.

5. Is your target variable continuous? If so, use a regression method. Try to avoid discretization if possible, since discretizing a continuous variable will inevitably waste information.

6. Carefully consider your sample size. If you have very small samples, it is likely that simple, univariate methods will work best.

7. Can you safely reduce dimension before applying feature selection? For example, if you know the detection limit of your measurement device, try to remove "silent" features which cannot possibly contain information. This is important prior knowledge which should be utilized.

8. If you end up with too many features to be interpretable, consider any external information that could be used to prioritize among the results, such as functional annotations [9]. Simply taking a fix number of "most significant" features may be misleading. Also consider higher-level interpretations if possible, *e.g.*, gene set testing [61, 165, 169].

9. Be aware that feature selection in high dimensions may not be possible to replicate between independent data sets (see Chapter 7). Rather, take care to ensure acceptable false discovery rates through sound statistical methodology.

10. Keep in mind that not all statistically relevant features are likely to be *biologically* relevant. Statistical significance should be viewed as a necessary but not sufficient condition. Domain knowledge is also essential to select high-quality candidates.

9.3 FUTURE RESEARCH

In the cancer gene expression example in Chapter 8 we encountered a fundamental limitation of feature selection by statistical dependence

measures: in some cases, there are *too many* significant findings, even with stringent error control. This somewhat ironical situation of having "too much power" clearly cannot be remedied by developing better algorithms. Here, statistical dependence simply is not a sufficiently fine-grained criteria for prioritizing between findings; other measures of relevance must be added. In the cancer gene expression case, we circumvented the difficulty by heuristically prioritizing the strongest correlations. However, this is not a satisfactory long-term solution. A more natural approach to this problem is to integrate additional data sources into the analysis [60, 150]. Kernel methods [152] provide an attractive solution for such data integration. While such methods have been used successfully for prediction purposes [3, 20], employing data integration in feature selection is still largely unexplored. An important question in this context is how to weigh the importance of the different data sources against each other [173].

On the theoretical side, there are many unanswered questions. As mentioned, many popular feature selection methods are still poorly understood. Further analysis is needed to establish consistency and error control results. For example, Relief [6] is an interesting candidate for such analysis in the context of finding all relevant features. Moreover, the set of features optimal for prediction $S^\dagger$ (the small-sample case) warrants a closer analysis. An interesting fact observed empirically in Chapter 5 is that $S^\dagger$ tends to coincide with the asymptotically optimal S^* for the support vector machine. An interesting conjecture then is that, because the support vector machine (and many other kernel methods) bases all inference on the gram matrix $K(x_i, x_j)$, which is independent of input space dimension, the size of S is not important, so that indeed $S^* = S^\dagger$. If, and to what extent, this is true is an open problem.

Many "NP-hard"-results in machine learning concern asymptotic cases [8, 26, 36]. In contrast, in Chapter 6 I explore the computational complexity of computing consistent *estimates*. I believe that this is a more relevant problem in practise. An interesting question is whether problems which are intractable in the asymptotic case may still permit consistent estimates which are computable in polynomial time. If so, one might find good approximations to many problems which are currently considered intractable. For example, such a consistency result was recently found by Kalisch and Bühlmann [91] for the case of Bayesian network inference, even though this problem is NP-hard in the asymptotic case [26]. A similar study could be of interest for the problem of finding all relevant features.

Bibliography

[1] Hervé Abdi. *Encyclopedia of Measurement and Statistics*. Thousand Oaks, 2007.

[2] Ruedi Aebersold and Matthias Mann. Mass spectrometry-based proteomics. *Nature*, 422:198–207, March 2003.

[3] Stein Aerts, Diether Lambrechts, Sunit Maity, Peter Van Loo, Bert Coessens, *et al.* Gene prioritization through genomic data fusion. *Nature Biotechnology*, 24(5):537–544, May 2006.

[4] John Aldrich. R.A. Fisher and the making of maximum likelihood 1912-1922. *Statistical Science*, 12(3):162–176, 1997.

[5] David B. Allison, Xiangqin Cui, Grier P. Page, and Mahyar Sabripour. Microarray data analysis: from disarray to consolidation and consensus. *Nature Reviews Genetics*, 7:55–65, January 2006.

[6] Hussein Almuallim and Thomas G. Dietterich. Learning with many irrelevant features. In *Proceedings of the Ninth National Conference on Artificial Intelligence*, volume 2, pages 547–552, Anaheim, California, 1991.

[7] U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences*, 96(12):6745–6750, 1999.

[8] Edoardo Amaldi and Viggo Kann. On the approximability of minimizing non-zero variables or unsatisfied relations in linear systems. *Theoretical Computer Science*, 209:237–260, 1998.

[9] Michael Ashburner, Catherine A. Ball, Judith A. Blake, David Botstein, Heather Butler, J. Michael Cherry, *et al.* Gene ontology: tool for the unification of biology. *Nature Genetics*, 25:25–29, 2000.

[10] Pierre Bald and Anthony D. Long. A bayesian framework for the analysis of microarray expression data: regularized t-test and statistical inferences of gene changes. *Bioinformatics*, 17:509–519, 2001.

[11] Yoshua Bengio, Olivier Delalleau, Nicolas le Roux, Jean-François Paiement, Pascal Vincent, *et al. Feature Extraction; Foundations and Applications*, chapter Spectral Dimensionality Reduction, pages 519–550. Springer, 2005.

[12] Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society, Series B*, 57(1):289–300, 1995.

[13] James O. Berger. *Statistical decision theory and Bayesian analysis*. Springer series in statistics. Springer-Verlag, 2nd edition, 1985.

[14] Roger L. Berger. Multiparameter hypothesis testing and acceptance sampling. *Technometrics*, 24(4):295–300, November 1982.

[15] Jimbo Bi, Kristin P. Bennett, Mark Embrechts, Curt M. Breneman, and Mnghu Song. Dimensionality reduction via sparse support vector machines. *Journal of Machine Learning Research*, 3:1229–1243, 2003.

[16] Avril L. Blum and Pat Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97:245–271, 1997.

[17] Karsten M. Borgwardt, Arthur Gretton, Malte J. Rasch, Hans-Peter Kriegel, Bernhard Scholkopf, *et al.* Integrating structured biological data by Kernel Maximum Mean Discrepancy. *Bioinformatics*, 22(14): e49–57, 2006.

[18] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. In D Haussler, editor, *Proceedings of the 5th annual ACM workshop on computational learning theory*, pages 144–152, Pittsburgh, PA, July 1992.

[19] P. S. Bradley and O. L. Mangasarian. Feature selection via concave minimization and support vector machines. In *Proceedings of the 15th International Conference on Machine Learning*, pages 82–90, San Francisco, CA, 1998.

[20] Sarah Calvo, Mohit Jain, Xiaohui Xie, Sunil A Sheth, Betty Chang, *et al.* Systematic identification of human mitochondrial disease genes through integrative genomics. *Nature Genetics*, 38(5):576–582, May 2006.

[21] George Casella and Roger L. Berger. *Statistical Inference*. Duxbury advanced series. Duxbury, 2nd edition, 2002.

[22] Howard Y. Chang, Dimitry S. A. Nuyten, Julie B. Sneddon, Trevor Hastie, Robert Tibshirani, *et al.* Robustness, scalability, and integration of a wound-response gene expression signature in predicting breast cancer survival. *Proceedings of the National Academy of Sciences*, 102 (10):3738–3743, 2005.

[23] Olivier Chapelle, Vladimir Vapnik, Olivier Bousquet, and Sayan Muhkherjee. Choosing multiple parameters for support vector machines. *Machine Learning*, 46:131–159, 2002.

[24] Rama Chellappa and Anil Jain, editors. *Markov random fields : theory and application*. Academic Press, Boston, 1993.

[25] David Chickering and Christopher Meek. Finding optimal bayesian networks. In *Proceedings of the 18th Annual Conference on Uncertainty in Artificial Intelligence*, pages 94–102, San Francisco, CA, 2002.

[26] David Maxwell Chickering, David Heckerman, and Christopher Meek. Large-sample learning of bayesian networks is NP-hard. *Journal of Machine Learning Research*, 5:1287–1330, 2004.

[27] Nello Christianini and John Shawe-Taylor. *An introduction to support vector machines*. Cambridge University Press, 2000.

[28] Robert T. Collins, Yanxi Liu, and Marius Leordeanu. Online selection of discriminative tracking features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1631–1643, 2005.

[29] Rat Genome Sequencing Project Consortium. Genome sequence of the brown norway rat yields insights into mammalian evolution. *Nature*, 428 (6982):493–521, April 2004.

[30] The International Human Genome Mapping Consortium. A physical map of the human genome. *Nature*, 409:934–941, February 2001.

[31] The Mouse Genome Sequencing Consortium. Initial sequencing and comparative analysis of the mouse genome. *Nature*, 420:520–562, 2002.

[32] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[33] Thomas M. Cover and Jan M. van Campenhout. On the possible orderings of the measurement selection problem. *IEEE Transactions on Systems, Man, and Cybernetics*, 7(9):657–661, September 1977.

[34] Thomas M. Cover and J.A. Thomas. *Elements of information theory*. John Wiley and Sons, New York, 1991.

[35] James E. Darnell. Transcription factors as targets for cancer therapy. *Nature Reviews Cancer*, 2:740–749, 2002.

[36] Scott Davies and Stuart Russel. NP-completeness of searches for smallest possible feature sets. In *Proceedings of the 1994 AAAI fall symposium on relevance*, pages 37–39. AAAI Press, 1994.

[37] Luc Devroye, Lázló Györfi, and Gábor Lugosi. *A probabilistic theory of pattern recognition*. Applications of mathematics. Springer-Verlag, New York, 1996.

[38] Thomas G. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10: 1895–1923, 1998.

[39] Chris H. Ding and Xiaofeng He. k-means clustering via principal component analysis. In *Proceedings of the 21st International Conference of Machine Learning, 2004*.

[40] Edward R. Dougherty. The fundamental role of pattern recognition for gene-expression/microarray data in bioinformatics. *Pattern Recognition*, 38:2226–2228, 2005.

[41] E.E. Ntzani and J.P. Ioannidis. Predictive ability of DNA microarrays for cancer outcomes and correlates: an empirical assessment. *The Lancet*, 362:1434–1444, 2003.

[42] Bradley Efron. Boostrap methods: another look at the jackknife. *The Annals of Statistics*, 7(1):1–26, 1977.

[43] Bradley Efron. The efficiency of logistic regression compared to normal discriminant analysis. *Journal of the American Statistical Association*, 70(352):892–898, December 1979.

[44] Bradley Efron and Robert J. Tibshirani. *An introduction to the bootstrap*. Chapman & Hall, Inc., 1993.

[45] Bradley Efron, Robert Tibshirani, John D. Storey, and Virginia Tusher. Empirical bayes analysis of a microarray experiment. *Journal of the American Statistical Association*, 96(456):1151–1160, December 2001.

[46] Liat Ein-Dor, Itai Kela, Gad Getz, David Givol, and Eytan Domany. Outcome signature genes in breast cancer: is there a unique set? *Bioinformatics*, 21(2):171–178, 2005.

[47] Liat Ein-Dor, Or Zuk, and Eytan Domany. Thousands of samples are needed to generate a robust gene list for predicting outcome in cancer. *Proceedings of the National Academy of Sciences*, 103(15):5923–5928, 2006.

[48] J. Craig Venter *et al.* The sequence of the human genome. *Science*, 291 (5507):1304–1351, February 2001.

[49] Ronald A. Fisher. Frequency distribution of the values of the correlation coefficient in samples of an indefinitely large population. *Biometrika*, 10: 507–521, 1915.

[50] Ronald A. Fisher. The statistical utilization of multiple measurements. *Annals of eugenics*, 8:376–386, 1938.

[51] Evelyn Fix and J.L. Hodges, Jr. Discriminatory analysis — nonparametric discrimination: consistency properties. Technical Report Report Number 4, Project Number 21-49-004, USAF School of Aviation Medicine, Randolph Field, Texas, USA, 1951. Reprinted in *International Statistical Review*, vol 57, 1989.

[52] D François, V Werz, and M Verleysen. The permutation test for feature selection by mutual information. In *Proceedings of the 2006 European Symposium on Artificial Neural Networks*, pages 239–244, 2006.

[53] Yoav Freund. An adaptive version of the boost by majority algorithm. *Machine Learning*, 43(3):293–318, 2001.

[54] Yoav Freund. Boosting a weak learning algorithm by majority. In *Proceedings of the Third Annual Workshop on Computational Learning Theory*, pages 202–216, 1990.

[55] Nir Friedman. Inferring cellular networks using probabilistic graphical models. *Science*, 303(5659):799–805, 2004.

[56] G. Fritz and B. Kaina. Rho-GTPases: promising cellular targets for novel anticancer drugs. *Current Cancer Drug Targets*, 6(1):1–14, 2006.

[57] Glenn Fung and O. L. Mangasarian. A feature selection newton method for support vector machine classification. *Computational Optimization and Applications*, 28:185–202, 2004.

[58] T.S. Furey, N. Christianini, N. Duffy, D.W. Bednarski, M. Schummer, etal Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, 16(10):906–914, 2000.

[59] Timothy S. Gardner, Diego di Bernardo, David Lorenz, and James J. Collins. Inferring genetic networks and identifying compound mode of action via expression profiling. *Science*, 301: 102-105, 2003.

[60] Comsmas Giallourakis, Charlotte Henson, Michael Reich, Xiaohui Xie, and Vamsi K. Mootha. Disease gene discovery through integrative genomics. *Annual Review of Genomics and Human Genetics*, 6:381–406, 2005.

[61] Jelle J. Goeman and Peter Buhlmann. Analyzing gene expression data in terms of gene sets: methodological issues. *Bioinformatics*, Advance Access published on February 15, 2007.

[62] Todd R. Golub, Donna K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, *et al.* Molecular classifiation of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537, 1999.

[63] PI Good. *Permutation Tests: A Practical Guide to Resampling Methods for Testing Hypotheses.* Springer-Verlag, 2nd edition, 2000.

[64] Arthur Gretton, Olivier Bousquet, Alexander Smola, and Bernhard Schölkopf. Measuring statistical dependence with hilbert-schmidt norms. In *Proceedings of the 16th International Conference on Algorithmic Learning Theory*, pages 63–78, 2005.

[65] Arthur Gretton, Ralf Herbrich, Alexander Smola, Olivier Bousquet, and Bernhard Schölkopf. Kernel methods for measuring independence. *Journal of Machine Learning Research*, 6:2075–2129, December 2005.

[66] Jenny E. Gunton, Rohit N. Kulkarni, SunHee Yim, Terumasa Okada, Wayne J. Hawthorne, *et al.* Loss of ARNT/HIF1β mediates altered gene expression and pancreatic-islet dysfunction in human type 2 diabetes. *Cell*, 122:337–349, August 2005.

[67] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[68] Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46:389–422, 2002.

[69] Isabelle Guyon, Steve Gunn, Masoud Nikravesh, and Lotfi A. Zadeh. *Feature Extraction; Foundations and Applications.* Springer, 2005.

[70] Sara Hägg, Josefin Skogsberg, Jesper Lundström, Peri Noori, Roland Nilsson, *et al.* Multi-organ gene expression profiling uncovers LIM domain binding 2 as a novel candidate gene in coronary artery disease. Manuscript, April 2007.

[71] Mark A. Hall and Geoffrey Holmes. Benchmarking attribute selection techniques for discrete class data mining. *IEEE Transactions on Knowledge and Data Engineering*, 15(6):1437–1447, November/December 2003.

[72] Douglas P. Hardin, Constantin Aliferis, and Ioannis Tsamardinos. A theoretical characterization of SVM-based feature selection. In *Proceedings of the 21st International Conference on Machine Learning*, 2004.

[73] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning.* Springer, 2001.

[74] Trevor Hastie, Saharon Rosset, Robert Tibshirani, and Ji Zhu. The entire regularization path for the support vector machine. *Journal of Machine Learning Research*, 5:1391–1415, 2004.

[75] S. Haykin. *Neural networks, a comprehensive foundation*. Prentice-Hall, 1999.

[76] A.E. Heorl and R.W. Kennard. Ridge regression: biased estimation of nonorthogonal problems. *Technometrics*, 12:69–82, February 1970.

[77] Yosef Hochberg. A sharper bonferroni procedure for multiple tests of significance. *Biometrika*, 75(4):800–802, 1988.

[78] Michael J. Holland. Transcript abundance in yeast varies over six orders of magnitude. *Journal of Biological Chemistry*, 277:14363–66, 2002.

[79] S. Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6:65–70, 1979.

[80] Andrew L. Hopkins and Colin R. Groom. The druggable genome. *Nature Reviews Drug Discovery*, 1:727–730, 2002.

[81] Harald Hotelling. The generalization of Student's ratio. *Annals of Mathematical Statistics*, 2:360–378, 1931.

[82] Yifan Huang, Haiyan Xu, Violeta Calian, and Jason C. Hsu. To permute or not to permute. *Bioinformatics*, 22(18):2244–2248, 2006.

[83] Tommi Jaakkola, Mark Diekhans, and David Haussler. Using the Fisher kernel method to detect remote protein homologies. In *Proceedings of the 7th International Conference on Intelligent Systems for Molecular Biology*, pages 149–158. AAAI Press, 1999.

[84] Anil K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Computing Surveys*, 31(3):264–323, 1999.

[85] Anil K. Jain and Douglas Zongker. Feature selection: evaluation, application and small sample performance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(2):153–158, 1997.

[86] Anil K. Jain and William G. Waller. On the optimal number of features in the classification of multivariate gaussian data. *Pattern Recognition*, 10:365–374, 1978.

[87] Anil K. Jain, Robert P.W. Duin, and Jianchang Mao. Statistical pattern recognition: a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–36, 2000.

[88] T.K. Jenssen and E. Hovig. Gene-expression profiling in breast cancer. *The Lancet*, 365:634–635, 2005.

[89] Thorsten Joachims. Estimating the generalization performance of an SVM efficiently. In *Proceedings of the Seventeenth International Conference on Machine Learning*, 2000.

[90] Markus Kalisch and Peter Bühlmann. Estimating high-dimensional directed acyclic graphs with the PC-algorithm. Technical report, Seminar für Statistik, ETH Zürich, Switzerland, 2005. Available at `http://stat.ethz.ch/~buhlmann/bibliog.html`.

[91] Markus Kalisch and Peter Bühlmann. Estimating high-dimensional directed acyclic graphs with the PC-algorithm. *Journal of Machine Learning Research*, 8:613–636, 2007.

[92] S. Sathiya Keerthi. Efficient tuning of SVM hyperparameters using radius/margin bound and iterative algorithms. *IEEE Transactions on Neural Networks*, 13(5):1225–1229, September 2002.

[93] S. Sathiya Keerthi and E.G. Gilbert. Convergence of a generalized SMO algorithm for SVM classifier design. *Machine Learning*, 46(1-3):351–360, 2002.

[94] K. Kira and L.A. Rendell. The feature selection problem: Traditional methods and a new algorithm. In *Proceedings of the Ninth National Conference on Artificial Intelligence*, pages 129–134. AAAI Press, 1992.

[95] Hiroaki Kitano. Computational systems biology. *Nature*, 420:206–210, 2002.

[96] Lev Klebanov, Craig Jordan, and Andrei Yakovlev. A new type of stochastic dependence revealed in gene expression data. *Statistical Applications in Genetics and Molecular Biology*, 5(1):Article 7, 2006.

[97] Ron Kohavi and George H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97:273–324, 1997.

[98] Ron Kohavi and Dan Sommerfield. Feature subset selection using the wrapper method: Overfitting and dynamic search space topology. In *Proceedings of the First International Conference on Knowledge Discovery and Data Mining*, pages 192–197, 1995.

[99] Ron Kohavi and George John. *Feature Extraction, Construction and Selection: A Data Mining Perspective*, chapter The wrapper approach. Kluwer Academic Publishers, 1998.

[100] Bryan Kolaczkowski and Joseph W. Thornton. Performance of maximum parsimony and likelihood phylogenetics when evolution is heterogeneous. *Nature*, 431:980–984, October 2004.

[101] Daphne Koller and Mehran Sahami. Towards optimal feature selection. In *Proceedings of the 13th International Conference of Machine Learning*, pages 248–292, 1996.

[102] Igor Kononenko. Estimating attributes: Analysis and extensions of RELIEF. In *European Conference on Machine Learning*, pages 171–182, 1994.

[103] Mineichi Kudo and Jack Sklansky. Comparison of algorithms that select features for pattern classifiers. *Pattern Recognition*, 33:25–41, 2000.

[104] S. Kullback and R.A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[105] John Langford. Tutorial on practical prediction theory for classification. *Journal of machine learning research*, 6:273–306, 2005.

[106] Mei-Ling Ting Lee, Frank C. Kuo, G. A. Whitmore, and Jeffrey Sklar. Importance of replication in microarray gene expression studies: Statistical methods and evidence from repetitive cDNA hybridizations. *Proceedings of the National Academy of Sciences*, 97(18):9834–9839, 2000.

[107] Erich L. Lehmann. *Testing statistical hypotheses*. Wiley, New York, 1986.

[108] Erich L. Lehmann and Howard J.M. D'Abrera. *Nonparametrics : statistical methods based on ranks*. Holden-Day series in probability and statistics. Holden-Day, New York, 1975.

[109] Christina Leslie, Eleazar Eskin, and William Stafford Noble. The spectrum kernel: a string kernel for SVM protein classification. In *Pacific Symposium on Biocomputing*, 2003.

[110] Fan Li, Yiming Yang, and Eric P. Xing. From lasso regression to feature vector machine. In *Advances in Neural Information Processing Systems*, 2005.

[111] Huan Liu and Hiroshi Motoda. *Feature Extraction, Construction and Selection: A Data Mining Perspective*. Kluwer Academic Publishers, 1998.

[112] Yan Lu, Peng-Yuan Liu, Peng Xiao, and Hong-Wen Deng. Hotelling's T^2 multivariate profiling for detecting differential expression in microarrays. *Bioinformatics*, 21(14):3105–3113, 2005.

[113] Aldons J. Lusis. Atherosclerosis. *Nature*, 407:233–241, September 2000.

[114] Donna Maglott, Jim Ostell, Kim D. Pruitt, and Tatiana Tatusova. Entrez Gene: gene-centered information at NCBI. *Nucleic Acids Research*, 33:D54–D58, January 2005.

[115] Rui Mei, Patricia C. Galipeau, Cynthia Prass, Anthony Berno, Ghassan Ghandour, *et al.* Genome-wide detection of allelic imbalance using human SNPs and high-density DNA arrays. *Genome Research*, 10(8): 1126–1137, 2000.

[116] Stefan Michiels, Serge Koscielny, and Catherine Hill. Prediction of cancer outcome with microarrays: a multiple random validation strategy. *The Lancet*, 365:488–492, 2005.

[117] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K. Muller. Fisher discriminant analysis with kernels. In *Proceedings of IEEE Neural Networks for Signal Processing Workshop*, 1999.

[118] Sebastian Mika, Gunnar Rätsch, and Klaus-Robert Müller. A mathematical programming approach to the kernel fisher algorithm. In *Advances in Neural Information Processing Systems*, pages 591–597, 2000.

[119] Vamsi K. Mootha, C.M. Lindgren, K.F. Eriksson, A. Subramanian, S. Sihag, *et al.* PGC-1α-responsive genes involved in oxidative phosphorylation are coordinately downregulated in human diabetes. *Nature Genetics*, 34(3):267–273, July 2003.

[120] Sreerama K. Murthy. Automatic construction of decision trees from data: A multi-disciplinary survey. *Data Mining and Knowledge Discovery*, 2(4):345–389, 2004.

[121] Patrenahalli Narendra and Keinosuke Fukunaga. A branch and bound algorithm for feature subset selection. *IEEE Transactions on computers*, C-26(9):917–922, September 1977.

[122] Richard E. Neapolitan. *Learning Bayesian Networks*. Prentice Hall, 1st edition, 2003.

[123] David J. Newman, S. Hettich, C.L. Blake, and C.J. Merz. UCI repository of machine learning databases, 1998. URL `http://www.ics.uci.edu/~mlearn/MLRepository.html`.

[124] Michael A. Newton, Amine Noueiry, Deepayan Sarkar, and Paul Ahlquist. Detecting differential gene expression with a semiparametric hierarchical mixture method. *Biostatistics*, 5(2):155–176, 2004.

[125] Roland Nilsson. Feature selection for classification and network discovery from gene expression data. Master's thesis LiTH-IFM-Ex-1148, Linköping University, Institute of Technology, 2003.

[126] Roland Nilsson, Jose M. Peña, Johan Björkegren, and Jesper Tegnér. Consistent feature selection for pattern recognition in polyomial time. *Journal of Machine Learning Reseach*, 8:589–612, 2007.

[127] Roland Nilsson, José M. Peña, Johan Björkegren, and Jesper Tegnér. Evaluating feature selection for SVMs in high dimensions. In *Proceedings of the 17th european conference on machine learning*, pages 719–726, 2006.

[128] Emma Niméus-Malmström, Cecilia Ritz, Patrik Edén, Anders Johnsson, Mattias Ohlsson, *et al.* Gene expression profiles and conventional clinical markers to predict distant recurrences for premenopausal breast cancer patients after adjuvant chemotherapy. *European Journal of Cancer*, 42: 2729–2737, 2006.

[129] Judea Pearl. *Causality: models, reasoning, and inference.* Cambridge University Press, 2000.

[130] Judea Pearl. *Probabilistic reasoning in intelligent systems.* Morgan Kauffman Publishers, Inc., San Fransisco, California, 1988.

[131] José M. Peña, Johan Björkegren, and Jesper Tegnér. Scalable, efficient and correct learning of markov boundaries under the faithfulness assumption. In *Proceedings of the Eighth European Conference on Symbolic and Quantitative Approaches to Reasoning under Uncertainty*, pages 136–147, 2005.

[132] Jose M. Peña, Roland Nilsson, Johan Björkegren, and Jesper Tegnér. Identifying the relevant nodes before learning the structure. In *Proceedings of the 22nd Conference on Uncertainty in Artificial Intelligence*, pages 367–374, 2006.

[133] Suraj Peri, J. Daniel Navarro, Ramars Amanchy, Troels Z. Kristiansen, Chandra Kiran Jonnalagadda, *et al.* Development of human protein reference database as an initial platform for approaching systems biology in humans. *Genome Research*, 13(10):2363–2371, 2003.

[134] Simon Perkins, Kevin Lacker, and James Theiler. Grafting: Fast, incremental feature selection by gradient descent in function space. *Journal of Machine Learning Research*, 3:1333–1356, March 2003.

[135] E.J.G. Pitman. Significance tests which may be applied to samples from any populations. *Supplement to the Journal of the Royal Statistical Society*, 4(1):119–130, 1937.

[136] Natalie Pochet, Frank de Smet, Johan A.K. Suykens, and Bart L.R. de Moor. Systematic benchmarking of microarray data classification: assessing the role of non-linearity and dimensionality reduction. *Bioinformatics*, 20(17):3185–3195, 2004.

[137] S. James Press and Sandra Wilson. Choosing between logistic regression and discriminant analysis. *Journal of the American Statistical Association*, 73(364):699–705, December 1978.

[138] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. *Numerical recipes in C.* Cambridge University Press, 2nd edition, 1992.

[139] P. Proks and J.D. Lippiat. Membrane ion channels and diabetes. *Current Pharmaceutical Design*, 12(4):485–501, 2006.

[140] Pavel Pudil and Jana Novovičová. Novel methods for subset selection with respect to problem knowledge. *IEEE Intelligent Systems*, pages 66–74, March/April 1998.

[141] Laura Elena Raileanu and Kilian Stoffel. Theoretical comparison between the gini index and information gain criteria. *Annals of Mathematics and Artificial Intelligence*, 41(1):77–93, 2004.

[142] Sridhar Ramaswamy, Pablo Tamayo, Ryan Rifkin, Sayan Mukherjee, Chen-Hsiang Yeang, Mi*et al.* Multiclass cancer diagnosis using tumor gene expression signatures. *Proceedings of the National Acadademy of Sciences*, 98(26):15149–54, December 2002.

[143] Gunnar Rätsch and Manfred K. Warmuth. Efficient margin maximizing with boosting. *Journal of Machine Learning Research*, 6:2131–2152, December 2005.

[144] Jason D.M. Rennie. Regularized Logistic Regression is Strictly Convex. Unpublished manuscript.
URL `people.csail.mit.edu/jrennie/writing/convexLR.pdf`.

[145] Juha Reunanen. Overfitting in making comparisons between variable selection methods. *Journal of Machine Learning Research*, 3:1371–1382, 2003.

[146] Jorma Rissanen. Modeling by the shortest data description. *Automatica*, 14:465–471, 1978.

[147] Marko Robnik-Sikonja and Igor Kononenko. Theoretical and empirical analysis of ReliefF and RReliefF. *Machine Learning*, 53:23–69, 2003.

[148] F. Rosenblatt. *Principles of neurodynamics*. Spartam Books, 1962.

[149] S.N. Roy. On a heuristic method of test construction and its use in multivariate analysis. *Annals of Mathematical Statistics*, 24:220–38, 1953.

[150] Eric E Schadt, John Lamb, Xia Yanh, Jun Zhu, Steve Edwards, *et al.* An integrative genomics approach to infer causal associations between gene expression and disease. *Nature Genetics*, 37(7):710–717, 2005.

[151] Mark Schena, Dari Shalon, Ronald W. Davis, and Patrick O. Brown. Quantitative monitoring of gene expression patterns with a complementary dna microarray. *Science*, 270:467–470, October 1995.

[152] Bernhard Schölkopf, K. Tsuda, and J.-P. Vert. *Kernel Methods in Computational Biology*. MIT Press, 2004.

[153] Bernhard Schölkopf and Alexander J. Smola. *Learning with kernels*. Adaptive computation and machine learning. MIT Press, 2002.

[154] Eran Segal, Nir Friedman, Daphne Koller, and Aviv Regev. A module map showing conditional activity of expression modules in cancer. *Nature Genetics*, 36:1090–1098, 2004.

[155] Z. Šidàk. Rectangular confidence region for the means of multivariate normal distributions. *Journal of the American Statistical Association*, 62:626–633, 1967.

[156] Dinesh Singh, Phillip G. Febbo, Kenneth Ross, Donald G. Jackson, Judith Manola, *et al.* Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell*, 1:203–209, March 2002.

[157] Donna K. Slonim. From patterns to pathways: gene expression comes of age. *Nature Genetics*, 32:502–508, 2002. Supplement.

[158] Alex J. Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, 2004.

[159] Terry Speed, editor. *Statistical Analysis of Gene Expression Microarray Data*. Chapman & Hall, 2003.

[160] Ingo Steinwart. On the influence of the kernel on the consistency of support vector machines. *Journal of Machine Learning Research*, 2:67–93, March 2002.

[161] John D. Storey. A direct approach to false discovery rates. *Journal of the Royal Statistical Society, Series B*, 64(3):479–498, 2002.

[162] John D. Storey. The positive false discovery rate: a bayesian interpretation and the q-value. *Annals of Statistics*, 31:2013–2035, 2003.

[163] John D. Storey and Robert Tibshirani. Statistical significance for genomewide studies. *Proceedings of the National Academy of Sciencies*, 100(16):9440–45, August 2003.

[164] Student. The probable error of a mean. *Biometrika*, 6(1):1–25, March 1908.

[165] Aravind Subramanian, Pablo Tamayo, Vamsi K. Mootha, Sayan Mukherjee, Benjamin L. Ebert, *et al.* Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciencies*, 102(43): 15545–15550, 2005.

[166] J.A.K. Suykens and J. Vandewalle. Least squares support vector machine classifiers. *Neural processing letters*, 9:293–300, 1999.

[167] Jesper Tegnér and Johan Björkegren. Perturbations to uncover gene networks. *Trends in Genetics*, 23(1):34–41, 2007.

[168] Jesper Tegner, M. K. Stephen Yeung, Jeff Hasty, and James J. Collins. Reverse engineering gene networks: Integrating genetic perturbations with dynamical modeling. *Proceedings of the National Academy of Sciencies*, 100(10):5944–5949, 2003.

[169] Lu Tian, Steven A. Greenberg, Sek Won Kong, Josiah Altschuler, Isaac S. Kohane, and Peter J. Park. Discovering statistically significant pathways in expression profiling studies. *Proceedings of the National Academy of Sciencies*, 102(38):13544–13549, 2005.

[170] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58(1):267–288, 1996.

[171] Andrei N. Tikhonov and V.A. Arsenin. *Solution of Ill-posed Problems*. Winston & Sons, 1977.

[172] Michael E Tipping. Sparse bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, 1:211–244, 2001.

[173] Olga G. Troyanskaya, Kara Dolinski, Art B. Owen, Russ B. Altman, and David Botstein. A Bayesian framework for combining heterogeneous data sources for gene function prediction (in saccharomyces cerevisiae). *Proceedings of the National Academy of Sciencies*, 100(14):8348–8353, 2003.

[174] G.V. Trunk. A problem of dimensionality: a simple example. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(3):306–307, July 1979.

[175] Ioannis Tsamardinos, C. Aliferis, and A. Statnikov. Time and sample efficient discovery of Markov blankets and direct causal relations. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 673–678. 2003.

[176] Ioannis Tsamardinos and Constantin Aliferis. Towards principled feature selection: relevancy, filters and wrappers. In *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, 2003.

[177] Leslie G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, November 1984.

[178] Jan M. van Campenhout. *On the problem of measurement selection*. PhD thesis, Stanford University, 1978.

[179] Laura J. van't Veer, Hongyue Dai, Marc J. Van De Vijver, Yudong D. He, Augustinus A. M. Hart, *et al.* Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, 415:530–536, January 2002.

[180] Vladimir N. Vapnik. *The nature of statistical learning theory.* Springer, New York, 2nd edition, 2000.

[181] Vladimir N. Vapnik. *Statistical Learning Theory.* John Wiley and Sons, Inc., 1998.

[182] Michael Waddell, David Page, and Jr. John Shaughnessy. Predicting cancer susceptibility from single-nucleotide polymorphism data: a case study in multiple myeloma. In *Proceedings of the 5th international workshop on Bioinformatics*, pages 21–28, 2005.

[183] Yixin Wang, Jan GM Klijn, Yi Zhang, Anieta M Sieuwerts, Maxime P Look, *et al.* Gene-expression profiles to predict distant metastasis of lymph-node-negative primary breast cancer. *The Lancet*, 365:671–679, February 2005.

[184] BL Welch. The significance of the difference between two means when the population variances are unequal. *Biometrika*, 29(3/4):350–362, February 1938.

[185] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, *et al.* Feature selection for SVMs. In *Advances in Neural Information Processing Systems*, 2000.

[186] Jason Weston, André Elisseeff, Bernhard Schölkopf, and Mike Tipping. Use of the zero-norm with linear models and kernel methods. *Journal of Machine Learning Research*, 3:1439–1461, March 2003.

[187] Sandra L. Wong. The role of sentinel lymph node biopsy in the management of thin melanoma. *The American Journal of Surgery*, 190(2): 196–199, August 2005.

[188] Ying Xu and Edward C. Uberbacher. Automated gene identification in large-scale genomic sequences. *Journal of Computational Biology*, 4(3): 325–338, 1997.

[189] M.K. Stephen Yeung, Jesper Tegnér, and James J. Collins. Reverse engineering of genetic networks using singular value decomposition and robust regression. *Proceedings of the National Academy of Sciences*, 99 (9):6163–68, April 2002.

[190] Lei Yu and Huan Liu. Efficient feature selection via analysis of relevance and redundancy. *Journal of Machine Learning Research*, 5:1205–1224, December 2004.

[191] Hongyuan Zha, Xiaofeng He, Chris H. Q. Ding, Ming Gu, and Horst D. Simon. Spectral relaxation for k-means clustering. In *Advances in Neural Information Processing Systems 14*, pages 1057–1064, 2001.