Chapter 1

Finding molecular characteristics of disease

— molecular patient classification —

Synopsis: *In this chapter I review state of the art characterization of disease using high dimensional genomic data. It contains a description of statistical learning techniques, applied to the clinical setting of deriving a signature for patient diagnosis. I introduce basic concepts and present example algorithms, pointing out the crucial role of regularization techniques. I focus on microarray specific aspects, including gene selection and the evaluation of a signature's performance.*

1.1 Motivation

In clinical gene expression studies tissue samples are examined using microarray technology. Classification of patients is a powerful tool for diagnosis of disease, assessment of risk group and selection of treatment [130, 152]. From a statistical point of view, the major characteristic of microarray studies is that the number of genes is orders of magnitude larger than the number of patients. When blindly applying out-of-the-box classification algorithms, a model rather adapts to noise in the data than to the molecular characteristics of disease. It is a challenge to find a molecular characterization of disease that can be generalized from a study cohort to the entire disease population.

In the following we provide an overview of state of the art derivation of rules for molecular diagnosis. We describe statistical learning techniques, starting with fundamental concepts; we present example algorithms and highlight characteristics arising from the nature of microarray data. We also deal with the topic of assessing the predictive perfor-

mance of a classification rule, a subject of relevance in clinical studies but until recently often neglected in microarray analyses [6, 99, 129, 150].

1.2 Supervised classification of patients

Molecular diagnosis based on gene expression data is the most widely used approach in clinical microarray studies. The data consists of gene expression profiles of n patients. In addition, each patient has an attributed class label. The label reflects a clinical phenotype. Phenotypes can include previously defined disease entities as in the leukemia study of Yeoh et al. [157], risk groups like in the breast cancer studies of van't Veer et al. [147], or disease outcome as in the breast cancer study of West et al. [151]. The objective is to learn expression signatures that allow to predict the correct clinical phenotype for new patients.

It is important that the class labels must not be derived from the expression profiles themselves. This requirement embeds molecular diagnosis into the field of supervised machine learning and defines the difference to unsupervised class finding problems. There are many more genes on the arrays than patients in the study and gene-to-sample ratios typically are in the hundreds. This is the main difficulty faced in the supervised approaches. A large number of machine learning algorithms are available to overcome this problem and in the following we will summarize some basic ideas.

1.2.1 Notation

We measure p genes on n patients, with typically $p \gg n$. The data from each microarray is represented by a profile $\boldsymbol{x}^{(i)} \in \mathbb{R}^p$ and $i = 1,\ldots,n$. The corresponding label, which encodes one of K clinical phenotypes, is denoted by $\boldsymbol{y}_i \in \mathscr{K} = \{1,\ldots,K\}$, $\boldsymbol{y} \in \mathscr{K}^n$. The profiles are arranged as rows in a matrix $\mathbf{X} \in \mathbb{R}^{n\times p}$. The two quantities $(\mathbf{X}, \boldsymbol{y})$ are called a data set $\mathscr{D}$ that holds all data of a study in pairs of observations $\{(\boldsymbol{x}^{(i)}, \boldsymbol{y}_i)\}$.

It is often useful to assume a data-generating distribution $F(\boldsymbol{X}, Y)$ on $\mathbb{R}^p \times \mathscr{K}$. $F(\boldsymbol{X}, Y)$ is the joint distribution of expression profiles and associated clinical phenotypes. Patients might be be modeled as independent samples $\{(\boldsymbol{x}^{(i)}, \boldsymbol{y}_i)\}$ drawn from F. In the following capitalized quantities refer to random variables (e.g. $\boldsymbol{X}$ and Y), whereas realizations are set in lower case (as in $\boldsymbol{x}^{(i)}$ and $\boldsymbol{y}_i$).

1.2.2 Concepts for inferring a classification rule

In the supervised setting, one is concerned with inferring a *classification rule* from a given data set $\mathscr{D}$. As the classification problems we consider primarily are medical diagnoses, we use the terms *diagnostic signature* and *classification rule* interchangeably.

Such a rule or signature assigns one of a collection $\mathcal{K}$ of labels to an observation, or patient, $\boldsymbol{x}$. That is, it is a function $c : \mathbb{R}^p \to \mathcal{K}$. The objective is to derive a well generalizing classification rule, a rule that assigns the correct labels not only to the patients in $\mathcal{D}$, but also in future clinical practice.

Loss and risk To define a measure of generalizability for a classification rule, we need to distinguish between the performance on the samples in the study and the expected performance in clinical practice. First, we define a *loss function* l to quantify the loss of diagnosing profile $\boldsymbol{x}$ to have label $c(\boldsymbol{x})$, given the true label is y:

$$l(\boldsymbol{x}, c(\boldsymbol{x}), y) : \mathbb{R}^p \times \mathcal{K} \times \mathcal{K} \to [0, \infty) \quad . \tag{1.1}$$

A simple loss function is the 0/1 loss function, which assigns a loss of one to each misclassified sample. Let us now define the *risk* of a signature as

$$R[c] = \mathbb{E}\, l(\boldsymbol{X}, c(\boldsymbol{X}), Y) = \int l(\boldsymbol{x}, c(\boldsymbol{x}), y)\; dF(\boldsymbol{X}, Y) \quad , \tag{1.2}$$

which measures the expected loss of a diagnostic signature. It is the performance of the signature in clinical practice. Since we have no access to the population we do not know F and cannot calculate the risk explicitly. But we have access to the patients in the study. To approximate R, we define the *empirical risk* :

$$\hat{R}[c] = \int l(\boldsymbol{x}, c(\boldsymbol{x}), y)\; d\hat{F}(\boldsymbol{X}, Y) = \frac{1}{n} \sum_{i}^{n} l(\boldsymbol{x}^{(i)}, c(\boldsymbol{x}^{(i)}), \boldsymbol{y}_i) \quad , \tag{1.3}$$

where $\hat{F}$ is the empirical distribution function which puts weight $1/n$ on each observed data point. In the context of a microarray study, the empirical risk for the 0/1 loss function is the error rate of the signature on patients in the study. It can be calculated from the data.

Bayes classifier and Bayes error Before we explain how to build diagnostic signatures in practice we introduce a purely theoretical construct: The best thinkable signature, also called the *Bayes classifier*. While it cannot be built in practice it is helpful for the development of the theory. Assume we know what is called the *posterior densities* of F, i.e. $P(Y = k | \boldsymbol{X} = \boldsymbol{x})$. Then the Bayes classifier is defined as:

$$c^{\star}(\boldsymbol{x}) = \underset{k \in \mathcal{K}}{\operatorname{argmax}}\, P(Y = k | \boldsymbol{X} = \boldsymbol{x}) \quad . \tag{1.4}$$

Its error is called the *Bayes error* . The Bayes error can be different from zero, which means that it is impossible to construct a molecular signature that never fails. This is not necessarily a matter of insufficient bioinformatics expertise, but can be an intrinsic property of the disease population F. If the same expression profiles can occur

under different clinical phenotypes, it is obvious that false classifications cannot be entirely ruled out. The Bayes classifier can be derived theoretically from the risk defined in Equation (1.2): Take the 0/1 loss, i.e. $l(\boldsymbol{x}, c(\boldsymbol{x}), y) = \mathbb{I}(c(\boldsymbol{x}) \neq y)$, where $\mathbb{I}$ denotes the indicator function. That is, $\mathbb{I}(c(\boldsymbol{x}) \neq y) = 1$ if $c(\boldsymbol{x}) \neq y$ and zero for correct predictions. Minimizing the risk at some $\boldsymbol{x} \in \mathbb{R}^p$ is equivalent to minimizing the probability of future misclassifications at this point. This, in turn, is the same as always assigning the most probable class. That is precisely the statement of Equation (1.4). The Bayes classifier is a purely theoretical construct, but it might be approximated based on study data.

Minimal empirical risk and maximum likelihood In this paragraph we introduce a general principle for inferring a signature from a given data set. We start by assuming a specific structure of the data generating distribution F, modeling the dependency of Y on $\boldsymbol{X}$ via a *link function* g_β, indexed by parameters β. For binary y we only need to model $P(Y = 1|g_\beta(\boldsymbol{X}))$, as probabilities need to sum to one. An example for a link function is the *logistic function* discussed later on. In general we can define the *likelihood* of the observed y_i and $\boldsymbol{x}^{(i)}$ as:

$$P(\mathscr{D}|g_\beta) = \prod_i^n P(Y = y_i|\boldsymbol{X} = \boldsymbol{x}^{(i)}, g_\beta) P(\boldsymbol{X} = \boldsymbol{x}^{(i)}) \quad . \tag{1.5}$$

Maximization of the above quantity with respect to β is assumed to yield a suitable g_β. This is the same as taking the log, dropping g-independent terms and minimizing the negative log-likelihood:

$$\mathscr{L}_l[g] = -\sum_i^n \log(P(Y = y_i|\boldsymbol{X} = \boldsymbol{x}^{(i)}, g)) \quad . \tag{1.6}$$

This, in turn, is equivalent to minimizing the empirical risk defined in equation (1.3) over β choosing the loss function to be $l = -\log P(Y|g_\beta(\boldsymbol{X}))$. It is not possible to conversely interpret every loss function in terms of a probability $P(Y|g_\beta(\boldsymbol{X}))$.

Regularized risk and priors As mentioned before, the main challenge in microarray based diagnosis is the large number of genes on the array compared to the few patients in the study. In this section we demonstrate mathematical implications of this situation.

In the last paragraph we have seen that minimizing the empirical risk over a family of functions f indexed by a parameter β can be equivalent to the maximum likelihood approach. This suggests that minimizing the empirical risk over a class of candidate signatures $\mathscr{C}$, i.e. $\hat{c} := \text{argmin}_{c \in \mathscr{C}} \hat{R}[c]$ is a suitable approach to infer a classifier. This approach can lead to ill posed problems, where the optimum is not uniquely defined. For high dimensional microarray data, this is the case even for simple signature classes $\mathscr{C}$. An example is *linear discriminant analysis*, which will be discussed in Section 1.2.3.

As a simplified illustration imagine a study where two patients, each representative of a certain phenotype, are selected. Then the `mRNA` abundance of two genes is measured.

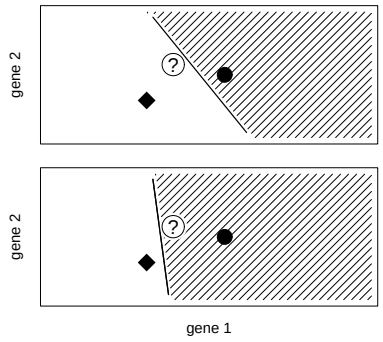

Figure 1.1: Linear separation with as many genes as patients. *In this toy example, the solid square and the circle represent patients from two different disease types. They can be separated linearly in several ways; two separating signatures are shown. The question mark represents a patient with unknown diagnosis. Both signatures yield conflicting predictions. The data alone does not suggest that one of the signatures is better than the other, it does not define a unique diagnosis. This situation occurs if the number of genes is larger than or equal to the number of patients. This is always the case in microarray studies.*

Further on, we want to construct a linear signature that can discriminate between the two classes. This is the same problem as finding a straight line between two points (each representing a patient) in a plane. The candidate signatures in $\mathscr{C}$ now correspond to all possible straight lines. There is no unique solution minimizing the empirical risk with the 0/1 loss function, see Figure 1.1. Next, think about a third point which incidentally does not lie on the line going through the first two points. Imagine it represents a new patient with unknown diagnosis. It is always possible to linearly separate the first two points such that the new one lies on the same side as either one of them. The two training patients do not contain sufficient information to diagnose the third patient uniquely. We are in this situation whenever the number of genes is larger than or equal to the number of patients. Due to the large number of genes on the chip this problem is inherent in microarray studies.

A way to deal with such ill posed problems are *regularization* approaches [46, 47, 142]. In our example above this can correspond to finding the straight line which separates the two points *and* which has maximal distance to both of them. This strategy is implemented in *support vector machines*, a classification algorithm discussed in the next section. Regularization results in the minimization of the *regularized risk functional*

$$\hat{R}_{\text{reg}}[c] := \hat{R}[c] + \lambda\Omega[c] \quad , \tag{1.7}$$

where Ω is called *regularization operator* and penalizes the complexity of signatures. The parameter λ determines a trade off between performance on the study data ($\hat{R}[c]$) and complexity of the classifier ($\Omega[c]$).

Introducing a regularization term can also be motivated from the following perspective: In the paragraph about maximum likelihood we assumed a dependency g between the two random variables $\boldsymbol{X}$ and Y. Expressing prior beliefs about g in a *prior distribution* $P(g)$, an application of Bayes' rule leads to

$$\underbrace{P(g|\mathscr{D})}_{\text{posterior}} = \underbrace{P(\mathscr{D}|g)}_{\text{likelihood}} \underbrace{P(g)}_{\text{prior}} \frac{1}{P(\mathscr{D})} \quad . \tag{1.8}$$

The *posterior* represents the probability of a model g, given the data. Usually g corresponds to a certain diagnostic signature. The $g^\star$ that maximizes the posterior is a reasonable choice to yield a classifier, since it best explains the data at hand and matches prior beliefs. The classifier corresponding to $g^\star$ is called *maximum aposteriori* or MAP estimate. Finding the MAP estimate is equivalent to minimizing a regularized risk, as can be seen by the following argument. The posterior is proportional to the likelihood times the prior. The *likelihood* is, as discussed before, the probability of the data given the model. We have also seen that maximizing the likelihood can be viewed as minimizing a suitable empirical risk. The regularization now comes via the prior. It puts more weight onto models g that we deem more likely than others, without having seen the data. This can resolve ambiguities where the likelihood alone is undecided. We can see the equivalence as follows: Ignore the last factor in Equation (1.8), since it does not depend on g. Then recall Equation (1.5), set $l = -\log P(Y|g(\boldsymbol{X}))$ and choose $\lambda\Omega[g] = -\log P(g)$ in Equation (1.7).

1.2.3 Supervised classification algorithms

Having introduced basic concepts and terminology of supervised machine learning, we now present a collection of classification algorithms commonly used with microarray data. It is important to distinguish between a diagnostic signature c and a learning algorithm L. A signature takes expression profiles as an input and returns class labels as an output. A learning algorithm is used to build signatures. It takes training data as an input and returns signatures as an output. To review notation, L is a set of rules how to generate a signature $\hat{c}$, given data $\mathscr{D}$ consisting of n independent samples from $F(X,Y)$, i.e. $L : \mathscr{D} \mapsto \hat{c}$. We will discuss gene selection based methods, penalized logistic regression, and support vector machines as well as bagging and boosting. Several of these algorithms build linear signatures.

Discriminant analysis and feature selection Discriminant analysis can be motivated by modeling the data generating distribution F in a parametric way: $P(X,Y) = P(\boldsymbol{X} = \boldsymbol{x}|Y = k)P(Y = k) := f_k(\boldsymbol{x})\pi_k$. We further assume that the f_k are Gaussian densities:

$$f_k(\boldsymbol{x}) = |2\pi\Sigma_k|^{-\frac{1}{2}} \exp\big[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)^T\Sigma_k^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k)\big] \tag{1.9}$$

$$\begin{aligned}\log f_k\pi_k &= \log|2\pi\Sigma_k|^{-\frac{1}{2}} - \frac{1}{2}\boldsymbol{x}^T\Sigma_k^{-1}\boldsymbol{x} \\ &\quad + \boldsymbol{x}^T\Sigma_k^{-1}\boldsymbol{\mu}_k - \frac{1}{2}\boldsymbol{\mu}_k^T\Sigma_k^{-1}\boldsymbol{\mu}_k + \log\pi_k \quad .\end{aligned} \tag{1.10}$$

For the unknown parameters unbiased estimates for mean and covariance $(\hat{\boldsymbol{\mu}}_k, \hat{\Sigma}_k)$ can be employed [115]. The priors π_k can be estimated by the relative class sizes $\hat{\pi}_k := n_k/n$. This yields estimates $\hat{f}_k$ and the Bayes classifier can be approximated via $\hat{c}(\boldsymbol{x}) = \text{argmax}_{k\in\mathcal{K}}\ \hat{f}_k\hat{\pi}_k$. The classification boundary between any two classes i and j is defined as

$$\left\{\boldsymbol{x} : P(Y = i|\boldsymbol{X} = \boldsymbol{x}) = P(Y = j|\boldsymbol{X} = \boldsymbol{x})\right\} \quad .$$

That is, at the points on the classification boundary the posterior probability to belong to either class is the same. Assume equal covariance matrices in Equations (1.9) and (1.10). This case is called the *homoscedastic* case. Then the first two terms in (1.10) are the same for all classes k, and the classification boundaries are linear in $\boldsymbol{x}$. This method is termed *linear discriminant analysis* (LDA). If the Σ_k are assumed different between the phenotypes (*heteroscedastic case*), the decision boundaries are quadratic. Then the method is called *quadratic discriminant analysis* (QDA). In case the covariances are assumed to be diagonal, i.e. $\Sigma_k = \text{diag}(\sigma_{k1}, \ldots, \sigma_{kp}) =: \text{diag}(\boldsymbol{\sigma}_k)$, the corresponding name is *diagonal discriminant analysis* (DDA) . Consequently, the assumption of equal diagonal covariance matrices leads to *diagonal linear discriminant analysis* (DLDA).

Deriving the signature requires the inversion of the estimated covariance matrices. For QDA this leads to the constraints $n_i \geq p+1$, where the n_i denote the class sizes, and for LDA to $n \geq p+K$. In other words, QDA and LDA require more patients than genes on the chip, which in our setting is unrealistic. For DDA the estimates $\hat{\Sigma}_k$ are always invertible and it can be applied to expression data directly. Only with the help of *gene selection* LDA and QDA become applicable to microarray data. Gene selection has to be done prior to the estimation of model parameters. From the entire set of genes only a small number of genes is selected, and then discriminant analysis is applied using only the selected subset of genes. Also the performance of DLDA can be improved using gene selection [37]. A popular method equipping a variant of DLDA with gene selection was proposed by Tibshirani et al. [139]. We discuss it now in some detail, as it will be used to derive signatures in Chapter 2.

In the case of LDA diagnosis consists of classifying a new sample to the class k with the nearest group centroid $\boldsymbol{\mu}_k$ (modulo the influence of the priors π_k). Distance is measured terms of the *Mahalanobis distance* , i.e. $\hat{c}(\boldsymbol{x}) = \text{argmin}_{k\in\mathcal{K}}(\boldsymbol{x} - \hat{\boldsymbol{\mu}}_k)^T\hat{\Sigma}^{-1}(\boldsymbol{x} - \hat{\boldsymbol{\mu}}_k)$. Tibshirani et al. [139] restrict Σ to be diagonal. Additionally it is pointed out, that the estimate of the mean might be obscured by noise present in the data. Therefore Tibshirani et al. advocate the use of a denoised or *shrunken* centroid for each group k in the distance calculation. That is

$$\begin{aligned} \hat{c}(\boldsymbol{x}) &= \underset{k\in\mathcal{K}}{\text{argmin}}(\boldsymbol{x} - \tilde{\boldsymbol{\mu}}_k)^T\hat{\Sigma}^{-1}(\boldsymbol{x} - \tilde{\boldsymbol{\mu}}_k) + \log\hat{\pi}_k \\ \tilde{\boldsymbol{\mu}}_k &= \hat{\boldsymbol{\mu}} + \boldsymbol{\Delta}_k(\delta) \end{aligned} \tag{1.11}$$

where $\hat{\Sigma} = \text{diag}(\hat{\boldsymbol{\sigma}})$ and $\sqrt{\hat{\boldsymbol{\sigma}}_i}$ is the pooled within class standard deviation of class k for gene i plus a fudge factor s that reduces the effect of nearly constant genes. $\hat{\boldsymbol{\mu}}$ is the usual estimate for the overall mean. The vector $\boldsymbol{\Delta}_k$ extracts only genes in which the mean of group k strongly differs from the overall mean, namely:

$$(\boldsymbol{\Delta}_k)_i = \text{sgn}(\hat{\boldsymbol{\mu}}_k - \hat{\boldsymbol{\mu}})_i \left|\,|(\hat{\boldsymbol{\mu}}_k - \hat{\boldsymbol{\mu}})_i| - m_k(s_i + s)\,\delta\right|_+$$

where $m_k s_i$ estimates the standard error of $(\hat{\boldsymbol{\mu}}_k - \hat{\boldsymbol{\mu}})_i$ and $|x|_+ = x$ for $x > 0$ and zero otherwise. That is, each component of $\hat{\boldsymbol{\mu}}_k$ is shrunken towards the overall mean in units of the (fudged) standard error. Additionally to the classification function this approach can yield an estimate of the posterior class probability $P(Y = k|\boldsymbol{X} = \boldsymbol{x})$ by the $\hat{f}_k(\boldsymbol{x})$ and an application of Bayes' rule. The value for the gene selection parameter δ is obtained by *cross validation*, which we discuss in Section 1.2.4. A link between this approach and classical linear models is discussed in Huang et al. [66]. Other types of discriminant analysis have been applied to gene expression data as well. For an overview as well as a comparison with other methods see [37, 87].

Penalized Logistic regression In our discussion above we modeled the data generating distribution explicitly via the class conditional probabilities $f_k = P(X|Y = k)$. This is also called *generative modeling* [79]. Now we take a *discriminative approach*. That is, we explicitly specify the dependency structure g we discussed in the paragraph about maximum likelihood. Here we choose the *logistic link function*

$$P(Y = k|g_\beta(\boldsymbol{X})) = \frac{\exp[\boldsymbol{\beta}_k^T \boldsymbol{x}]}{\sum_i^K \exp[\boldsymbol{\beta}_i^T \boldsymbol{x}]} \tag{1.12}$$

with identifiability constraints. The classification rule corresponding to this model imitates the Bayes classifier: $\hat{c}(\boldsymbol{x}) = \text{argmax}_{k \in \mathcal{K}} P(Y = k|\boldsymbol{X} = \boldsymbol{x}, \hat{\boldsymbol{\beta}}_k)$. To identify the parameters $\boldsymbol{\beta}_k$ we use the maximum likelihood approach discussed before. If we focus on the two class problem with $y_i \in \{\pm 1\}$, Equation (1.12) reduces to $\mathsf{p} := P(Y = 1|\boldsymbol{X} = \boldsymbol{x}) = 1/(1 + \exp(-\boldsymbol{\beta}^T \boldsymbol{x}))$ and $P(Y = -1|\boldsymbol{X} = \boldsymbol{x}) = 1 - \mathsf{p}$. The likelihood of the class labels can be modeled via independent biased coin flips:

$$\mathscr{L} = \prod_{i=1}^n P(Y = y_i|X = \boldsymbol{x}^{(i)}, \boldsymbol{\beta}) = \prod_{i=1}^n \mathsf{p}_i^{\frac{1}{2}|y_i+1|}(1 - \mathsf{p}_i)^{\frac{1}{2}|y_i-1|} \quad , \tag{1.13}$$

where we have dropped $P(X = \boldsymbol{X})$ as it does not depend on $\boldsymbol{\beta}$. Now $\hat{\boldsymbol{\beta}}$ can be obtained as the minimizer of the negative log-likelihood, i.e. $\hat{\boldsymbol{\beta}} = \text{argmin}_{\boldsymbol{\beta}} -\log \mathscr{L}$. The dependence of $\mathscr{L}$ on $\boldsymbol{\beta}$ is through p and the minimization can be done directly or via an iterated least squares procedure. Note that this is equivalent to minimizing the empirical risk with a loss function chosen as $l = -\log[1 + \exp(y\boldsymbol{\beta}^T \boldsymbol{x})]$, the *logistic loss function*. Due to the high number of genes on the arrays the optimum of equation (1.13) is not unique. As a solution a regularized risk (see Equation (1.7)) can be minimized choosing

an appropriate regularization term. Common choices are the L_1- and the L_2-penalty, where $\Omega[c] = \|\boldsymbol{\beta}\|_1^2$ and $\Omega[c] = \|\boldsymbol{\beta}\|_2^2$, respectively. The L_2-penalty is usually combined with gene selection techniques [43, 161], while the L_1-penalty automatically produces sparse solutions. That is, only few genes contribute to the posterior [119, 126]. For the minimization Roth [119] utilizes an iterative least squares procedure extending an algorithm of Osborne et al. [105], while Shevade et al. [126] use a Gauss–Seidel method. Kim et al. [83] propose a gradient descent algorithm for problems of this form.

A related approach is that of West et al. [151] and Spang et al. [133], who model $P(Y = 1|X = \boldsymbol{x}) = \Phi(\boldsymbol{x}^T\boldsymbol{\beta})$ via the *probit regression model* . Here Φ denotes the cumulative density function of the standard normal distribution. But in contrast to what corresponds to a maximum aposteriori estimate in the approaches before, a full Bayesian analysis is employed and the posterior distribution of all model parameters is sampled. Regularization is achieved via the introduction of hierarchical normal priors.

Support vector classification Assume a classification problem with only two possible clinical phenotypes, i.e $y_i \in \{\pm 1\}$. Then support vector machines [123, 148, 149] fit a maximal (soft) margin hyperplane between the two classes. The margin of a hyperplane refers to the distance of the point closest to it. In high dimensional problems there are always several perfectly separating hyperplanes (the maximum likelihood approach leads to an ill posed problem). But there is only one separating hyperplane with maximal distance to the nearest training points of either class (regularization).

This concept is typically combined with the *kernel trick* to allow for flexible non-linear classification boundaries. The kernel trick is applicable to classification algorithms that can be expressed in terms of inner products of the inputs $\boldsymbol{x}^{(i)}$ who reside in what is called the *input space* . This is the case for the maximum margin separating hyperplane. The inner products $\boldsymbol{x}^T\boldsymbol{x}$ are then substituted by a *kernel function* $k(\boldsymbol{x}, \boldsymbol{x}')$, which corresponds to a *feature map* φ that maps the profiles from the input space into a *feature space* $\mathscr{H}$:

$$\begin{array}{rccc} \varphi: & \mathbb{R}^p & \longrightarrow & \mathscr{H} \\ & \boldsymbol{x} & \mapsto & \varphi(\boldsymbol{x}) \quad . \end{array} \tag{1.14}$$

This results in the original algorithm being now carried out in $\mathscr{H}$ and leads to non-linear decision boundaries in the input space.

After applying the kernel trick, a class of linear functions in feature space is given by $\mathscr{F} := \{f(\boldsymbol{x}) = \sum_i^n \alpha_i k(\boldsymbol{x}, \boldsymbol{x}^{(i)}) + b \,|\, \alpha_i, b \in \mathbb{R}\}$. The associated diagnostic signatures read $c(\boldsymbol{x}) = \operatorname{sgn} f(\boldsymbol{x})$. Finding the maximum margin hyperplane is the same as minimizing a regularized risk. The loss function employed is called *soft margin loss* and the regularization term is $\|f\|_{\mathscr{H}}^2 := \boldsymbol{\alpha}^T\mathbf{K}\boldsymbol{\alpha}$. Here $\mathbf{K}$ is the *kernel matrix* and $\mathbf{K}_{ij} = k(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)})$. Regularization is essential to counter the additional flexibility acquired by the kernel trick. In summary:

$$\hat{f}(\boldsymbol{x}) = \underset{f \in \mathscr{F}}{\operatorname{argmin}} \Big\{ \sum_i^n \max(0, 1 - y_i f(\boldsymbol{x}^{(i)})) + \frac{\lambda}{2} \boldsymbol{\alpha}^T\mathbf{K}\boldsymbol{\alpha} \Big\} \quad . \tag{1.15}$$

In this formulation separability of the two classes is not required and margin violations are allowed. The trade-off between margin violations and margin size (measured as $\|f\|_{\mathcal{H}}^{-2}$) is reflected by the regularization parameter λ. Support vector classification, in combination with various gene selection methods, has been applied to microarray data [51, 111] and compared favorably [22, 87, 94]. A SVM-specific *wrapper method* (see below) for feature selection is presented in Guyon et al. [59]. The generalization to more than two classes is not straight-forward and different methods are compared in Hsu et al. [64].

Bagging Bagging [19] is a method of aggregating weak classifiers via bootstrapping the data at hand. A weak classifier is a classifier of limited complexity and weak performance. Bagging stands for **b**ootstrap **agg**regat**ing** and works as follows: M bootstrap samples are drawn from the data $\mathcal{D}$. A *bootstrap sample* from $\mathcal{D}$ is an iid sample $\{(\boldsymbol{x}^{(i)}, \boldsymbol{y}_i)\}_{i=1}^{n_B}$ of size n_B from the empirical distribution function of the data, which puts weight $1/n$ onto each observation. Then a simple learning algorithm is trained on each bootstrap sample minimizing the empirical risk. This results in a set of weak classifiers $\{\hat{c}_m\}_{m=1}^{M}$. In the end all weak classifiers $\hat{c}_m$ are averaged to obtain a final strong signature: $\hat{c}(\boldsymbol{x}) = \text{argmax}_k \sum_{m=1}^{M} \mathbb{I}_{\hat{c}_m(\boldsymbol{x})=k}$. The class k which most of the $\hat{c}_m$ agree on gets chosen. If the weak classifiers produce estimates of the class conditional probabilities, these can be averaged instead [62].

Random forests [20] constitute an application of this concept. There weak classifiers are classification trees [21], grown using only a random subset of genes. The forest of trees (weak signatures) is then averaged over the bootstrap samples. Random forests are applied to gene expression data e.g. by Gunther et al. [57] and Diaz-Uriarte et al. [40].

Boosting Another method falling into the category of aggregated classifiers is *boosting*. The algorithm *Adaboost* was introduced by Freund et al. [48]. Several weak signatures c_m are combined to form an aggregate classifier. Hastie et al. [62] present Adaboost as an forward stage-wise additive modeling approach. The empirical risk is minimized choosing the *exponential loss function* $l(\boldsymbol{x}, c(\boldsymbol{x}), y) = \exp(-yc(\boldsymbol{x}))$ for $y \in \{\pm 1\}$. Basically, the classifier $c(\boldsymbol{x})$ is viewed as an expansion in the weak c_m, namely $c(\boldsymbol{x}) = \text{sgn}\left(\sum_i^M \beta_m c_m(\boldsymbol{x})\right)$. But in contrast to bagging, the different c_m are not independently fit to bootstrap samples. Rather, given coefficients $\{\hat{\beta}_i\}_{i=1}^{m-1}$ and classifiers $\{\hat{c}_i\}_{i=1}^{m-1}$, the next pair $(\hat{\beta}_m, \hat{c}_m)$ is determined to optimally supplement the previous ones in terms of minimizing the empirical risk. This results in an iterative optimization strategy. The complexity of the classifiers can be regulated by the number of iterations allowed.

Dettling et al. [32] apply this procedure to microarray data. As weak signatures *decision stumps* are used, i.e. classification trees with only two terminal nodes. They use the *LogitBoost* algorithm of Friedman et al. [49]. There the exponential loss function of Adaboost is exchanged for the logistic loss function introduced earlier. As feature selection a non parametric filter method [106] is employed. A combination of bagging and boosting is presented in Dettling et al. [31] in the context of gene expression analysis.

Both methods, bagging and boosting, are also discussed in Tan et al. [136]. Zhang et al. [158, 159] also apply classification trees in the context of gene expression.

Gene selection

Combining classification algorithms with a gene selection procedure is common practice in microarray based diagnosis. It is done for two reasons. First and most importantly, gene selection reduces model complexity and in many cases impacts the predictive performance of the signature [37]. Here model complexity refers to the flexibility of the decision boundaries. Methods like linear discriminant analysis are not applicable at all without gene selection. Other methods, like nearest shrunken centroids [138] or L_1–penalized regression techniques, implicitly reduce the number of genes involved in the signatures. Secondly, the reduction of genes leads to a smaller and hence cheaper design of diagnostic chips or marker panels [77].

Feature selection has a strong impact on the predictive performance of a signature. For this reason it can not be considered to be a preprocessing step independent of the construction of the signature. It is an essential part of the signature building algorithm. There is an important difference between building a signature based on 10 genes and building a signature that depends on only 10 genes that, however, were chosen from a pool of 30,000 genes. The first is a low dimensional model and can be specified by 10 parameters. The second (in principle) still involves 30,000 parameters, although 29,990 of them were constrained to be zero. The important point is, that it was not agreed which of them should be zero *before* the data was considered. The importance of not separating gene selection from the signature building process becomes more apparent in Section 1.2.4.

Filter approaches When feature selection is performed independently of the learning algorithm, this is called a *filter approach* [78]. Straight-forward implementations univariately screen for genes, optimizing a score reflecting correlation with the class labels. Popular choices are the t–statistic, the (non parametric) Wilcoxon rank sum statistic, the absolute difference of the group means divided by the sum of the estimated standard deviations [55] or the F–statistic in the multiclass case. More sophisticated are filter approaches (heuristically) searching for an optimal *subset* of informative genes [15, 60]. Jäger et al. [76] and Ding et al. [35] additionally minimize the redundancy of the selected gene set.

Wrapper approaches If the learning algorithm is taken into account while looking for informative genes, this is called a *wrapper method* [78]; the feature selection procedure is "wrapped around" the learning algorithm. Either single features or subsets of features are sought that maximize a performance score of the learning algorithm, e.g. the estimated misclassification error. This is a generic procedure and does not depend on the specific learning algorithm employed. Looking for optimal feature subsets is a combinatorial problem and heuristics like forward selection and backward elimination can be

employed [78].

A prototypical example is *recursive feature elimination* (RFE) [59], taking a generalization bound as a performance score [110]. This concept, in turn, has been applied to gene expression data by Cho et al. [28]. Another approach is to use the penalty term in the regularized risk to ensure sparse solutions (few genes contribute), as it is the case for L_1–penalized logistic regression. Shevade et al. [126] use this property together with cross validation (see next section) to simultaneously select and assess the relevance of genes. Such methods are also called *embedded methods* [58].

1.2.4 Adaptive model selection and validation

This section covers two important steps in microarray based diagnosis: Adaptive model selection and the validation of the predictive performance of a molecular signature. While these are two different tasks, the methodologies in use are similar.

Adaptive model selection

The algorithms discussed in the previous section can all be linked to minimizing the empirical or the regularized risk (with an appropriate choice of the loss function) over a class $\mathscr{C}$ of candidate signatures. For instance, in the case of penalized logistic regression the signatures depend on the parametric form of $P(Y|X)$ (the right hand side of Equation (1.12)) and are parametrized by the $\boldsymbol{\beta}_k$. For kernel classifiers, $\mathscr{C}$ corresponds to signatures that can be written in the form $f(\boldsymbol{x}) = \operatorname{sgn}\left(\sum_i^n \alpha_i k(\boldsymbol{x}, \boldsymbol{x}^{(i)}) + b\right)$ with α_i and $b \in \mathbb{R}$. Depending on how rich this class of signatures is, the learning algorithm is able to implement more or less flexible boundaries between the K phenotypes. For microarray data, the typical situation is that even simple signature classes, such as hyperplanes, are extremely rich due to the high dimensionality of the profiles $\boldsymbol{x}^{(i)}$. For simplicity, we will refer to the richness of the set of candidate signatures as the complexity of a *diagnostic model*.

Bias variance trade-off When dealing with complex diagnostic models not the empirical risk needs to be optimized but the regularized risk in Equation (1.7). This allows for controlling complexity. The regularization term introduces a complexity penalty, effectively restricting the complexity of the derived diagnostic signature $\hat{c}$. The regularization parameter λ quantifies the trade-off between model fit and model complexity. See also Figure 1.2. With little regularization the algorithm can fit very flexible decision boundaries to the data. This results in few misclassifications on the data from patients in the study. Nevertheless, it can have poor predictive performance in clinical practice. The reason is, that the algorithm not only fits population properties (as desired), but also reflects noise resulting from patient sampling. This is referred to as *overfitting*. When the regularization term dominates Equation (1.7), the resulting signatures might be too restricted. Then we have poor performance on both, the study patients and in

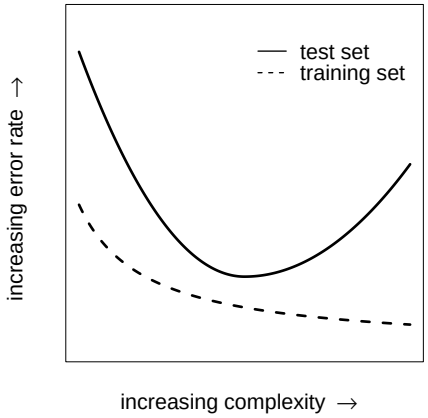

Figure 1.2: The bias variance trade-off. *Schematic picture to illustrate the trade-off between bias and variance (see text). The x-axis codes for model complexity and the y-axis for error rates. The dashed line displays the training error, the solid line the test error. Low complexity models produce high test errors (underfitting, low variance, high bias) and so do highly complex models (overfitting, high variance, low bias).*

future clinical practice. This is called *underfitting*. The problem described above is also known as the *bias variance trade-off*. Regularization introduces a bias into the estimation of model parameters, while at the same time reducing the sample variance. Sample variance here refers to $\hat{c}$ varying with different cohorts drawn from the same disease population.

Choosing a trade-off via the hold out The regularized risk is a general example of how learning algorithms can implement a tuning parameter (in this case λ) that allows for balancing over- and underfitting. Further instances are the number of genes included after the variable selection process, or the amount of shrinkage in the nearest shrunken centroid method.

Before we start with explaining how these parameters can be tuned, we need some more notations: Recall that we need to build signatures from a finite data set $\mathscr{D}$, drawn from F. Let us define the *empirical error rate* of a signature $\hat{c}$ built on $\mathscr{D}$ as:

$$\hat{e}_{\lambda}^{emp}[\,\hat{c}\,] := \frac{1}{n}\sum_{i}^{n} \mathbb{I}(\hat{c}_{\lambda}(\boldsymbol{x}^{(i)}) \neq y_i) \quad , \tag{1.16}$$

where we have made the dependency of the classification rule on the regularization parameter λ explicit and $\mathbb{I}$ denotes the indicator function. The empirical error rate is equivalent to the empirical risk $\hat{R}$ when using the 0/1 loss function. The empirical error rate is a random variable since it depends on the random sampling of patients that were included into a study. When repeating the study with a second cohort of patients, one obtains a different empirical error rate.

We now split the data set into a *training* or *learning set* $\mathscr{D}_l$ and a *test set* $\mathscr{D}_t$. The training set constitutes a (smaller) study cohort, while the test set can be used like novel patients with unknown diagnosis. In this respect, we want little errors on the test set. To reduce test errors, we are even willing to pay a price in terms of some more errors on the training set. For a fixed value of the regularization parameter, one can learn a signature $\hat{c}_{\lambda} := L(\mathscr{D}_l; \lambda)$ by applying a learning algorithm to the training data only. Subsequently

its predictive performance can be evaluated by applying the generated signature to the independent test data and calculating the error on the test set. The value of the regularization parameter can be varied, identifying a value such that the above estimate is minimal. That is, the learning algorithm then consists of minimizing $\hat{R}_{\text{reg}}$ and choosing the regularization parameter via assessment of generalization performance. This procedure is referred to as *adaptive model selection*, since by determining the regularization parameter one chooses a model with approximate optimal complexity. However, different clinical classification problems need different amounts of regularization. By using a hold out set, model selection is adapted to the data at hand.

Using data more efficiently via cross validation Since microarray data is expensive and scarce one can make use of the following procedure: The data set $\mathscr{D}$ is randomly partitioned into Q bins $\{\mathscr{D}_q\}_{q=1}^{Q}$. Each one of the $\mathscr{D}_q$ is then used as hold out set in turn. More formally: Let $\kappa : \{1,\ldots,n\} \to \{1,\ldots,Q\}$ be a partitioning, i.e. $\kappa(i) = q$ for all $i \in \mathscr{D}_q$, $i \in \{1,\ldots,n\}$ and $q \in \{1,\ldots,Q\}$. Further let $\hat{c}_\lambda^{-\kappa(i)} := L(\mathscr{D} \setminus \mathscr{D}_{\kappa(i)}; \lambda)$ be a classifier trained on $\mathscr{D} \setminus \mathscr{D}_{\kappa(i)}$ for a fixed value for λ. Then we estimate the misclassification rate via

$$\hat{e}_\lambda^{cvQ}[\,\hat{c}\,] := \frac{1}{n}\sum_{i}^{n} \mathbb{I}(\hat{c}_\lambda^{-\kappa(i)}(\boldsymbol{x}^{(i)}) \neq y_i) \tag{1.17}$$

[53, 135]. This quantity estimates the expected error rate on future data. Again, one can do this for a grid of values of the regularization parameter and an approximately optimal value can be identified. Adaptive model selection is a part of the learning algorithm, as it was the case with gene selection. This needs to be kept in mind when assessing the performance of a signature.

Validation of the predictive performance of a molecular signature

After having derived a diagnostic signature one needs to estimate its expected performance in future clinical practice. This validation step constitutes one of the most critical steps in the whole process of molecular diagnosis and several pitfalls are involved. Estimators can be overly optimistic (biased), or they might have high sample variances. It also makes a difference, whether one is interested in estimating the performance of a fixed signature $\hat{c}$ (which is usually the case in clinical studies), or if one is interested in estimating the performance of the learning algorithm L that builds the signatures (which is usually the case in methodological projects). The performance of the fixed signature $\hat{c}$ varies due to the random sampling of the test set, while the performance of the algorithm L varies due to sampling of both, training and test set.

The two different situations correspond to two different theoretical error rates. The performance of a fixed signature $\hat{c}(\boldsymbol{x}) =: L(\boldsymbol{x};\mathscr{D})$, derived from a training set $\mathscr{D}$, is measured by the *conditional error rate(s)* or the *true error* :

$$\begin{aligned} E_{ij}[\hat{c}] &= P(L(\boldsymbol{X};\mathscr{D}) = j | Y = i, \mathscr{D}) \quad i \neq j \in \mathscr{K} \quad \text{and} \\ E[\hat{c}] &= P(L(\boldsymbol{X};\mathscr{D}) \neq Y | \mathscr{D}) \quad . \end{aligned} \tag{1.18}$$

The first quantity, E_{ij}, is the probability that the signature $\hat{c}$ will classify a patient from the disease population to belong to class j even though he actually belongs to the phenotypical class i. The second quantity only asks for wrong classifications of $\hat{c}$, no matter which group is mistaken for what other group. These quantities are not obtainable in practice, since the probabilities need to be calculated with respect to the unknown population distribution $F(\boldsymbol{X}, Y)$.

If in contrast one is interested in the performance of the learning algorithm L, the sampling variability of the training set has to be taken into account. This makes the conditional error rates random variables. Keeping the size of the training set fixed and taking expectations leads to the *(unconditional) error rate(s)* or the *expected error* :

$$\begin{aligned} \bar{E}_{ij}[L] &= \mathbb{E}_{\mathscr{D}} E_{ij} = P(L(\boldsymbol{X};\mathscr{D}) = i | Y = j) \quad i \neq j \in \mathscr{K} \quad \text{and} \\ \bar{E}[L] &= \mathbb{E}_{\mathscr{D}} E = P(L(\boldsymbol{X};\mathscr{D}) \neq Y) \quad . \end{aligned} \tag{1.19}$$

As it was the case for the conditional error rates, these quantities depend on F and are not accessible. Hence both rates need to be estimated using the data at hand.

Estimating error rates One might assume the empirical error (equation (1.16)) can be employed to estimate the conditional error rate. The main problem with this approach is, that it uses the *same* data in $\mathscr{D}$ to train the classifier and to evaluate it later on. This can result in highly biased error rates grossly underestimating the true error. This is practically relevant in gene expression data analysis, since the high dimensionality of the data makes algorithms without complexity control prone to overfitting [6, 129].

A better approach is to use an independent test set. Only training data is used for gene selection, classifier learning and adaptive model selection. The final signature $\hat{c}$ is then evaluated on an independent test set. Unfortunately this estimator can have a substantial sample variance, due to the random selection of patients in the test set. This is especially the case if the test set is small. It falls in this line of thought that good performance in small studies can be a chance artifact [103].

More effective use of the data at hand can be made via the cross validation (CV) procedure introduced earlier. The leave one out version produces an estimator of the unconditional error rate with almost no bias. It is computationally more expensive than Q fold CV and suffers from a very high sample variance. The latter is reduced for moderate Q such as "somewhere between five and ten" [62, 84, 98]. Braga-Neto et al. [18] advise to average over many different partitionings. No unbiased estimator of the variance of the cross validation estimate exists, that is valid for all distributions F [11]. Cross validation error rates naturally refer to the classification algorithm L. In each iteration a different classifier is learned, based on (somewhat) different training data. The CV-performance is the average of the performance of different signatures. Nevertheless, CV-performance

can be also used as a bias corrected estimate of the conditional error rate. In fact, in applied work it is often used to validate fixed signatures that were derived by the evaluated algorithm.

Efron et al. [42] apply *bootstrap smoothing* to the leave one out cross validation estimate. The basic idea is to generate different *bootstrap replicates* $\{\mathscr{D}_b^*\}_{b=1}^{n_B}$, apply leave one out cross validation to each and then to average the results. A result of this approach is the so called "0.632+ estimator". It takes into account the possibility of overfitting and reduces the variance compared to the regular CV estimates. Ambroise et al. [6] have found it to work well with gene expression data.

Selection bias and nested loop cross validation As we have discussed in the previous section, feature selection techniques are a central element in the analysis of microarray data. In filter approaches, special care has to be taken when using cross validation: *The feature selection is part of the learning algorithm L*. For this reason feature selection has to be repeated again on each $\mathscr{D} \setminus \mathscr{D}_q$, that is Q times. Global gene selection before the cross validation (which is also called *incomplete cross validation* or *information leak*) can result in grossly over-optimistic (biased) estimates of the error rates [6]. For example, Simon et al. [129] describe a case, where the incomplete cross validation method and the fully cross validated method result in estimated error rates of 27% and 41%, respectively. Similarly, assume the algorithm L contains an adaptive model selection procedure. To get reasonable error rate estimates via CV, the selection procedure has to be applied to every $\mathscr{D} \setminus \mathscr{D}_q$ separately. This leads to a cross validation step inside a cross validation, i.e. to a *nested loop* cross validation [53, 122]. Applying the selection procedure to the complete data can lead to biased estimation of the error and over optimistic results. Ruschaupt et al. [122] and Wessels et al. [150] realize such a *complete validation procedure* and compare various methods.

Ntzani et al. [103] and Michiels et al. [99] report that, at least in studies up to 2003, most of the 84 considered studies lacked appropriate validation of derived signatures.

1.3 Discussion and chapter summary

We introduced concepts and methods used to address classification problems and pointed out characteristics relating to the classification of microarray data, such as gene selection, regularization and the selection bias. Books about machine learning include [34, 36, 62, 115, 123], where a more thorough treatment of the concepts and methodology can be found; focused on the analysis of microarray data are [98, 134].

The methodology above was presented in the classification context. It is tempting to interpret the genes driving the models, but this is dangerous. First, it is unclear how the regularization term biases the selection of signature genes. While a bias is a blessing from the diagnostic perspective, this is not necessarily the case from the biological perspective: In situations where the data is ambiguous, any generic regularization

scheme “decides” on a signature without regarding any constraints of biological plausibility. Second, signatures are generally not unique. While outcome prediction for breast cancer patients has been successful in various studies, e.g. [113, 132, 146], the respective signatures do not overlap at all. Further on, Ein-Dor et al. [44] derived a large number of almost equally performing signatures from a single data set. This is not too surprising considering the following. The molecular cause of a clinical phenotype might involve only a small set of genes. This primary event has secondary influences on other genes, which in turn deregulate more genes and so on. In clinical microarray analysis we typically observe an avalanche of secondary or later effects, often involving thousands of differentially expressed genes. While complicating biological interpretation of signatures, such an effect does not compromise the clinical usefulness of predictors. On the contrary, it is well conceivable that only signals enhanced through propagation lead to a well generalizing signature.

A crucial point in clinical microarray studies is the evaluation of the predictive performance of a signature. While nested loop cross validation can be used to derive a reasonable error estimate, it is confined to data collected in a homogeneous study context. Especially with possibly heterogeneous disease populations, it would be of value to apply the signatures to data *external* to the signature-deriving study and assess performance. Such a type of external validation requires the communication of signatures between scientists in different health care centers. This will be the topic of the next chapter.