LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

INSTITUT FÜR STATISTIK

A.-L. Boulesteix, C. Strobl, T. Augustin & M. Daumer

Evaluating microarray-based classifiers: an overview

Technical Report Number 005, 2007
Department of Statistics
University of Munich

http://www.stat.uni-muenchen.de

Evaluating microarray-based classifiers: an overview

A.-L. Boulesteix*, C. Strobl†, T. Augustin†, M. Daumer*

August 7th, 2007

*Sylvia Lawry Centre for MS Research (SLCMSR), Hohenlindenerstr. 1, D-81677 Munich, Germany

†Department of Statistics, University of Munich (LMU), Ludwigstr. 33, 80539 Munich, Germany

Abstract

For the last eight years, microarray-based class prediction has been the subject of numerous publications in medicine, bioinformatics and statistics journals. However, in many articles, the assessment of classification accuracy is carried out using suboptimal procedures and is not paid much attention. In this paper, we carefully review various statistical aspects of classifier evaluation and validation from a practical point of view. The main topics addressed are accuracy measures, error rate estimation procedures, variable selection, choice of classifiers and validation strategy.

Keywords: Accuracy measures, classification, conditional and unconditional error rate, error rate estimation, validation data, variable selection

1 Introduction

In the last few years, microarray-based class prediction has become a major topic in many medical fields. Cancer research is one of the most important fields of application of microarray-based prediction, although classifiers have also been proposed for other diseases such as multiple sclerosis (Bomprezzi et al., 2003). An interesting overview of studies published until April 2003 can be found in Ntzani and Ioannidis (2003). The two major applications of such methods are prediction of future events, e.g., response to treatment or disease course, and molecular diagnosis, though both of them are identical from a statistical point of view – as long as no measurement error occurs.

Let us consider a standard class prediction problem where expression data of p transcripts and the class information are available for a group of n patients. From a statistical point of view, patients are *observations* and transcripts are *variables*. Note that a particular gene might be represented several times. To avoid misunderstandings, we prefer the statistical term 'variable' to the ambiguous term 'gene'. In microarray studies, p is huge compared to n (typically, $5000 \leq p \leq 50000$ and $20 \leq n \leq 300$), which makes standard statistical prediction methods inapplicable. This dimensionality problem is also encountered in other fields such as proteomics or chemometrics.

Hence, the issues discussed in the present article are not specific to microarray data. The term *response class* refers to the categorical variable that has to be predicted based on gene expression data. It can be, e.g., the presence or absence of disease, a tumor subtype such as ALL/AML (Golub et al., 1999) or the responder status to a therapy (Ghadimi et al., 2005). The number of classes may be higher than two, though binary class prediction is by far the most frequent case in practice.

Note that gene expression data may also be used to predict survival times, ordinal scores or continuous parameters. However, class prediction is the most relevant prediction problem in practice. The interpretation of results is much more intuitive for class prediction than for other prediction problems for several reasons. From a medical point of view, it is often sensible to summarize more complex prediction problems such as, e.g. survival prediction or ordinal regression as binary class prediction. Moreover, we think that the model assumptions required by most survival analysis methods and methods for the prediction of continuous outcomes are certainly as questionable as the simplification into a classification problem. However, one has to be aware that transforming a general prediction problem into class prediction may lead to a loss of information, depending on the addressed medical question.

Beside some comparative studies briefly recalled in Section 2, several review articles on particular aspects of classification have been published in the last five years. For example, an extensive review on machine learning in bioinformatics including class prediction can be found in Larranaga et al. (2006), whereas Chen (2007) reviews both class comparison and class prediction with emphasis on univariate test statistics and model choice from the point of view of pharmacogenomics. Asyali et al. (2006) gives a wide overview of class prediction and related problems such as data preparation and clustering. User-friendly guidelines for good practice in microarray data analysis including class prediction can be found in Dupuy and Simon (2007), who also give a critical synthesis of cancer research articles published in 2004.

In contrast to all these, the present article focuses specifically on the statistical evaluation of microarray-based prediction methods. After a brief overview of existing classification methods in Section 2, measures of classification accuracy including error rate, sensitivity and specificity as well as ROC-curve analysis are addressed in Section 3. Section 4 reviews different evaluation strategies such as leaving-one-out cross-validation or bootstrap methods from a technical point of view, whereas Section

5 gives guidelines for practical studies. An overview of software for microarray-based class prediction in the R system for statistical computing (R Development Core Team, 2006) is given in the appendix.

2 Overview of existing Classifiers

Coping with high-dimensional data

There exist a variety of classification methods addressing exactly the same statistical problem. Several classifiers have been invented or adapted to address specifically prediction problems based on high-dimensional microarray data. Class prediction can also be addressed using machine learning approaches.

The aim of this section is to provide a concise overview of the most well-known classification approaches rather than an exhaustive enumeration. In contrast to other authors, we organize this overview with respect to the scheme used to handle high-dimensionality and not to the classifier itself. From this perspective, methods for handling high-dimensional data can basically be grouped into three categories: approaches based on (explicit) variable selection, procedures based on dimension reduction and methods performing intrinsic variable selection.

It should be noted that the three mentioned types of approaches for handling high dimensional data can also be combined with each other. For instance, variable selection may be performed prior to dimension reduction or before applying a method handling $n < p$.

Variable selection

The most intuitive approach consists of first selecting a small subset of variables and then applying a traditional classification method to the reduced data set. By traditional methods, we mean well-known statistical methods handling a rather limited number of variables, such as discriminant analysis methods reviewed and compared by Dudoit et al. (2002) including linear and quadratic discriminant analysis or Fisher's linear discriminant analysis, classical logistic regression or k-nearest-neighbors (e.g. Dudoit et al., 2002) which, in principle, could be applied to a high number of variables but performs poorly on noisy data.

Many variable selection approaches have been described in the bioinformatics literature. Overviews include the works by Stolovitzky (2003) and Jeffery et al. (2006). The methods applied can be classified as *univariate* and *multivariate* approaches. Univariate approaches consider each variable separately: they are based on the marginal utility of each variable for the classification task. Variables are ranked according to some criterion reflecting their association to the phenotype of interest. After ranking, the first variables of the list are selected for further analysis. Many criteria are conceivable, for instance usual test statistics like Student's t-statistic or nonparametric statistics such as Wilcoxon's rank sum statistic. Further non-parametric univariate criteria include more heuristic measures such as the TnoM score by Ben-Dor et al. (2000). Some of the nonparametric univariate approaches are reviewed by Troyanskaya et al. (2002). The t-statistic, the Mann-Whitney statistic and the heuristic signal-to-noise ratio suggested by Golub et al. (1999) are the most widely-used criteria in practice (Dupuy and Simon, 2007).

In the context of differential expression detection, several regularized variants of the standard t-statistic have been proposed in the last few years. They include, e.g., empirical Bayes methods (Smyth, 2004). An overview can be found in Opgen-Rhein and Strimmer (2007). Although these empirical Bayes methods are usually considered as univariate approaches, such methods involve a multivariate component in the sense described below, since the statistic of each variable is derived by borrowing information from other variables.

Univariate methods are fast and conceptually simple. However, they do not take correlations or interactions between variables into account, resulting in a subset of variables that may not be optimal for the considered classification task. This is obvious in the extreme case where, say, the 10 first variables correspond to the same transcript, yielding a strong correlation structure. It is then suboptimal to select these 10 redundant variables instead of variables with a worse univariate criterion value but giving non-redundant information.

Multivariate variable selection approaches for microarray data have been the subject of a few tens of rather theoretical articles. They take the preceding argument seriously that the subset of the variables with best univariate discrimination power is not necessarily the *best subset of variables*, due to interactions and correlations between variables. Therefore, multivariate variable selection methods do not score each

variable individually but rather try to determine which combinations of variables yield high prediction accuracy. A multivariate variable selection method is characterized by i) the criterion used to score the considered subsets of variables and ii) the algorithm employed to search the space of the possible subsets, an exhaustive enumeration of the 2^{p-1} possible subsets being computationally unfeasible. Scoring criteria can be categorized into *wrapper criteria*, i.e. criteria based on the classification accuracy or *filter criteria* that measure the discrimination power of the considered subset of variables without involving the classifier, for instance the Mahalanobis distance well-known from cluster analysis (which can roughly be seen as multivariate t-statistic).

There have also been various proposals regarding the search algorithms. Some methods, which could be denoted as "semi-multivariate" restrict the search to pairs of variables (Bo and Jonassen, 2002) or subsets of low-correlated and thus presumably non-redundant variables derived from the list of univariately best variables (Jäger et al., 2003). In contrast, other authors seek for globally optimal subsets of variables based on sophisticated search algorithms such as genetic algorithms (Goldberg, 1989) applied to microarray data by, e.g., Ooi and Tan (2003).

Note that most multivariate variable selection methods take only correlations between variables but not interactions into account, depending on the considered criterion used to score the variable subsets. The recent method suggested by Diaz-Uriarte and de Andrés (2006) based on random forests (Breiman, 2001) is one of the very few methods taking interactions into account explicitly. Potential pitfalls of multivariate methods are the computational expense, the sensitivity to small changes in the learning data and the tendency to overfitting. This is particularly true for methods looking globally for good performing subsets of variables, which makes semi-multivariate methods preferable in our view. Note that univariate variable selection methods, which select the top variables from a ranked list, may be seen as a special case of multivariate selection, where the candidate subsets are defined as the subsets formed by top variables.

Dimension reduction

A major shortcoming of variable selection when applied in combination with classification methods requiring the sample size n to be larger than the number p of variables is that only a small part of the available information is used. For example, if one applies logistic regression to a data set of size $n = 50$, the model should include at most

10 variables, which excludes possibly interesting candidates. Moreover, correlations between variables are not taken into account and can even pose a problem in model estimation, the more as gene expression data are known to be highly correlated. An option to circumvent these problems is dimension reduction, which aims at 'summarizing' the numerous predictors in form of a small number of new components (often linear combinations of the original predictors). Well-known examples are principal component analysis (PCA) or Partial Least Squares (PLS, Nguyen and Rocke, 2002; Boulesteix, 2004; Boulesteix and Strimmer, 2007) and its generalizations (Fort and Lambert-Lacroix, 2005; Ding and Gentleman, 2005). A concise overview of dimension reduction methods that have been used for classification with microarray data is given in Boulesteix (2006).

After dimension reduction, one can basically apply any classification method to the constructed components, for instance logistic regression or discriminant analysis. However, as opposed to the original genetic or clinical variables, the components constructed with dimension reduction techniques themselves may not be interpretable any more.

Methods handling a high number of variables directly

Instead of reducing the data to a small number of (either constructed or selected) predictors, methods handling large numbers of variables may be used. Preliminary variable selection or dimension reduction are then unnecessary in theory, although often performing well in practice in the case of huge data sets including several tens of thousands of variables. Methods handling a high number of variables ($p \gg n$) directly can roughly be divided into two categories: statistical methods based on penalization or shrinkage on the one hand, and computationally intensive approaches borrowed from the machine learning community on the other hand. The first category includes, e.g., penalized logistic regression (Zhu, 2004), the Prediction Analysis of Microarrays (PAM) method based on shrunken centroids (Tibshirani et al., 2002) or the more recent regularized linear discriminant analysis (Guo et al., 2007).

Support Vector Machines (SVM) (Vapnik, 1995) or ensemble methods based on recursive partitioning belong to the second category. Ensemble methods include for example bagging procedures (Breiman, 1996) applied to microarray data by Dudoit et al. (2002), boosting (Freund and Schapire, 1997) used by Dettling and Bühlmann

(2003), BagBoosting (Dettling, 2004) or Breiman's (2001) random forests examined by Diaz-Uriarte and de Andrés (2006) in the context of variable selection for classification. These methods may be easily applied in the $n < p$ setting, especially SVM. However, most of them become untractable when the number of features reaches a few tens of thousands, as usual in recent data sets. They should then be employed in combination with variable selection or dimension reduction.

Methods handling a high number of variables can be seen as performing intrinsic variable selection. Shrinkage and penalization methods allow to distinguish irrelevant from relevant variables through modifying their coefficients. Tree-based ensemble methods also distinguish between irrelevant and relevant variables intrinsically, through variable selection at each split.

Comparison studies

Prediction methods have been compared in a number of articles published in statistics and bioinformatics journals. Some of the comparisons are so-to-say neutral, whereas others aim at demonstrating the superiority of a particular method. Neutral comparison studies include Dudoit et al. (2002); Romualdi et al. (2003); Man et al. (2004); Lee et al. (2005); Statnikov et al. (2005). Comparison of different classification methods can also be found in biological articles with strong methodological background (e.g., Natsoulis et al., 2005). Most of these studies include common "benchmark" data sets such as the well-known leukemia (Golub et al., 1999) and colon (Alon et al., 1999) data sets.Table 2 (Appendix B) summarizes the characteristics and results of six published comparison studies, which we took as neutral, because they satisfy the following criteria:

- The title includes explicitly words such as "comparison" or "evaluation", but no specific method is mentioned in the title, thus excluding articles whose main aim is to demonstrate the superiority of a particular (new) method.
- The article has a clear methodological orientation. In particular, the methods are described precisely (including, e.g., the chosen variant or the choice of parameters) and adequate statistical references are provided.
- The comparison is based on at least two data sets.

- The comparison is based on at least one of the following evaluation strategies: CV, MCCV, bootstrap methods (see Section 4).

However, even if those criteria are met, optimistically biased results are likely to be obtained with the method(s) from the authors' expertise area. For example, authors are aware of all available implementations of that method and will quite naturally choose the best one. They may also tend to choose the variable selection method (e.g., t-test or Mann-Whitney test) according to their previous experience of classification, which has been mostly gained with this particular method. Similarly, an unexperienced investigator might overestimate the achievable error rate of methods involving many tuning parameters by setting them to values that are known to the experts as suboptimal.

The connection between classifiers and variable selection

When performed as a preliminary step, e.g. for computational reasons, variable selection should be seen as a part of classifier construction. In particular, when a classifier is built using a *learning* data set and tested subsequently on an independent *test* data set, variable selection must be performed based on the learning set only. Otherwise, one should expect non-negligible positive bias in the estimation of prediction accuracy. In the context of microarray data this problem was first pointed out by Ambroise and McLachlan (2002). Although it is obvious that test observations should not be used for variable selection, variable selection is often (wrongly) carried out as a "preliminary" step, especially when classification accuracy is measured using leave-one-out cross-validation. Even if performing t-tests or Wilcoxon tests $n \times p$ times becomes a daunting task when p reaches several tens of thousands, preliminary variable selection using all n arrays and leaving no separate test set for validation should definitively be banished. Bad practice related to this aspect has probably contributed to much "noise discovery" (Ioannidis, 2005).

A further important connection between classifiers and variable selection is the use of classifiers to evaluate the influence of single variables on the response class *a posteriori*. Parametric models, such as the logistic regression model, provide parameter estimates for main effects and interactions of predictor variables that can be interpreted directly for this purpose. However, the modern nonparametric approaches from machine learning, e.g., random forests, also provide variable importance mea-

sures that can be used not only for the preselection of relevant variables (Diaz-Uriarte and de Andrés, 2006) but are also a means of evaluating the influence of a variable – both individually and in interactions – on the response. Random forest variable importance measures have thus become a popular and widely used tool in genetics and related fields. However, when the considered predictor variables vary in their scale of measurement or their number of categories, as, e.g., when both genetic and clinical covariates are considered, the computation of the variable importance can be biased and must be performed differently (Strobl et al., 2007).

3 Measures of Classification Accuracy

We have seen in the previous section that in large-scale association studies classification can either be conducted with previous variable selection, dimension reduction, or with special classification methods that can deal with small n large p problems by intrinsically performing variable selection. However, these methods are very diverse, both in their methodological approach and their statistical features. In the following, we review concepts that allow to evaluate and compare all these different strategies and models, and is adaptable to special needs of investigators, e.g., if asymmetric misclassification costs are supposed to be modelled.

Error rate

We consider the random vector $\mathbf{X} \in \mathbb{R}^p$ and the random variable $Y \in \{0, \ldots, K-1\}$ giving the 'class membership'. Let $\mathbf{F}$ denote the joint distribution function of $\mathbf{X}$ and Y. A *classifier* is a function from $\mathbb{R}^p$ to $\{0, \ldots, K-1\}$ that assigns a predicted class to a vector of gene expressions corresponding to a patient:

$$\begin{array}{rccl} C: & \mathbb{R}^p & \rightarrow & \{0, \ldots, K-1\} \\ & \mathbf{X} & \rightarrow & \hat{Y}, \end{array} \tag{1}$$

where $\mathbf{X}$ denotes the p-vector giving the gene expression levels of the p considered variables for one patient and $\hat{Y}$ is his or her predicted class. If the joint distribution $\mathbf{F}(X, Y)$ of the gene expressions $\mathbf{X}$ and the class membership $\mathbf{Y}$ were known, one could use it to construct the Bayes classifier

$$C_{Bayes}(\mathbf{X}) = \arg\max_k P(Y = k|\mathbf{X}) \tag{2}$$

by deriving the posterior distribution $P(Y|\mathbf{X})$ of the response class given the gene expressions $\mathbf{X}$. The Bayes classifier based on the true, but unfortunately unknown, distributions minimizes the theoretical error rate, i.e. the probability of classifying into the wrong class:

$$Err(C) = P_{\mathbf{F}}(C(\mathbf{X}) \neq Y) = E_{\mathbf{F}}(I(C(\mathbf{X}) \neq Y)). \tag{3}$$

Note that this and all following definitions of error rates are appropriate in the case of unordered response classes only. For ordinal response classes it may be desirable that misclassification in a more distant class affects the error term more severely than misclassification in a neighboring class, which could be modelled via pseudo-distances serving as weights in the computation of the error rate. For classifiers that return class probabilities instead of predicted class membership, such as Bayesian methods but also some versions of recursive partitioning, the difference between the predicted class probability and the true class membership can be computed, e.g., by the Brier Score (i.e. the quadratic distance, see, e.g., Spiegelhalter, 1986, for an introduction).

Since the theoretical joint distribution $\mathbf{F}$ is always unknown in real data analysis, the classifier has to be estimated from an available data set. Moreover, once a classifier is constructed, its error rate also has to be estimated from some available data. Hence, the estimation of the error rate of a classification method involves two estimation components. Suppose we have a data set including n patients whose class membership has been determined independently of gene expression data, e.g., by clinical examination. The available data set $\mathbf{D} = (\mathbf{d}_1, \ldots, \mathbf{d}_n)$ consists of n identically distributed independent observations $\mathbf{d}_i = (y_i, \mathbf{x}_i)$, where $y_i \in \{0, \ldots, K-1\}$ denotes the class membership and $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})^T$ the p-vector of gene expressions of the i-th patient.

The data set used to construct (i.e.'learn') a classifier is usually denoted as 'training' or 'learning' data set. In this article, we use the term '*learning data*'. Let $\mathbf{l} = (l_1, \ldots, l_L)$ denote the indices of patients included in the learning data set and $\mathbf{D_l} = (\mathbf{d}_{l_1}, \ldots, \mathbf{d}_{l_L})$ the corresponding data set, where L is the number of observations in $\mathbf{l}$. In practice, there are several ways to define $\mathbf{l}$ and $\mathbf{t}$, see Section 4. A *classification method* takes the learning data set $\mathbf{D_l}$ as input and learns a classifier function C as defined in Eq. 1. From now on, $C^M_{\mathbf{D_l}}$ denotes the classifier learnt from the data set $\mathbf{D_l}$ using the classification method M. Examples of classification methods are, e.g., 'SVM with

linear kernel without preliminary variable selection' or 'linear discriminant analysis with the 20 best variables according to the t-test'.

In practical studies, investigators are often interested in the true error rate of a classifier built with all the available observations:

$$Err(C^M_{\mathbf{D}}), \tag{4}$$

where $\mathbf{D}$ is considered as fixed, hence the term *conditional error rate*. However, $\mathbf{D}$ can also be considered as random. The *unconditional* (or *expected*) true *error rate* is defined as

$$\epsilon^M_{\mathbf{F}^n} = E_{\mathbf{F}^n}(Err(C^M_{\mathbf{D}})), \tag{5}$$

where $\mathbf{F}^n$ describes the multivariate distribution of $\mathbf{D}$ based on $\mathbf{F}(\mathbf{X}, Y)$. The unconditional error rate $\epsilon^M_{\mathbf{F}^n}$ depends only on the classification method M, the size n of the used data set and the joint distribution $\mathbf{F}$ of $\mathbf{X}$ and $\mathbf{Y}$, but not on the specific data set $\mathbf{D}$. We use the notation ϵ instead of Err to outline the difference between conditional and unconditional error rate. Few articles distinguish between both of them. However, the relative performance of error estimation methods may depend on whether one considers the conditional or unconditional error rate. For instance when using the unconditional error rate results are somewhat in favor of the bootstrap (Efron and Tibshirani, 1997).

Estimating the error rate

Suppose we use a learning set $\mathbf{D_l}$ to construct the classifier $C^M_{\mathbf{D_l}}$. The joint distribution function $\mathbf{F}$ being unknown, the true conditional error rate

$$Err(C^M_{\mathbf{D_l}}) = E_{\mathbf{F}}(I(Y \neq C^M_{D_l}(\mathbf{X}))|\mathbf{D_l}) \tag{6}$$

of this classifier is also unknown and has to be estimated based on available test data. Similarly to $\mathbf{l}$ above, collecting the indices corresponding to the learning data set, we consider the T-vector $\mathbf{t} = (t_1, \ldots, t_T)$ giving the indices of test observations and $\mathbf{D_t} = (\mathbf{d}_{t_1}, \ldots, \mathbf{d}_{t_T})$ the corresponding data set. The estimator of the error rate of C based on $\mathbf{D_t}$ is then given as

$$\widehat{Err}(C^M_{\mathbf{D_l}}, \mathbf{D_t}) = \frac{1}{T}\sum_{i=1}^{T} I(y_{t_i} \neq C^M_{\mathbf{D_l}}(\mathbf{x}_{t_i})), \tag{7}$$

where $\mathbf{x}_{t_i} = (x_{t_i 1}, \ldots, x_{t_i p})^T$ is p-vector giving the gene expressions for the t_i-th observation. Note that, in simulations, the learning set $\mathbf{D_l}$ can be varied and the test data set $\mathbf{D_t}$ may be virtually as large as computationally feasible, thus providing an accurate estimation of $Err(C^M_{\mathbf{D_l}})$.

Sensitivity and specificity

Using the error rate as defined in Eq. (6), one implicitly considers all misclassifications as equally damaging. In practice, the proportion of misclassified observations might not be the most important feature of a classification method. This is particularly true if one wants to predict therapy response. If a non-responder is incorrectly classified as responder, possible inconveniences are the potentially severe side-effects of a useless therapy and - from an economic point of view - the cost of this therapy. On the other hand a responder who is incorrectly classified as non-responder may be refused an effective therapy, which might lead to impairment or even death.

In the medical literature, these two different aspects are often formulated in terms of sensitivity and specificity. If $Y = 1$ denotes the condition that has to be detected (for instance responder to a therapy), the *sensitivity* of the classifier is the probability $P(C^M_{\mathbf{D_l}}(\mathbf{X}) = 1 | Y = 1)$ of correctly identifying a responder. It can be estimated by the proportion of observations from the test data set with $Y = 1$ that are correctly predicted:

$$\widehat{Se}(C^M_{\mathbf{D_l}}, \mathbf{D_t}) = \frac{\sum_{i=1}^T I(y_{t_i} = 1) \cdot I(C^M_{\mathbf{D_l}}(\mathbf{x}_{t_i}) = 1)}{\sum_{i=1}^T I(y_{t_i} = 1)}, \tag{8}$$

whereas the *specificity* is the probability $P(C^M_{\mathbf{D_l}}(\mathbf{x}) = 0 | Y = 0)$ of correctly identifying a non-responder and can be estimated by the proportion of observations with $Y = 0$ that are correctly predicted:

$$\widehat{Sp}(C^M_{\mathbf{D_l}}, \mathbf{D_t}) = \frac{\sum_{i=1}^T I(y_{t_i} = 0) \cdot I(C^M_{\mathbf{D_l}}(\mathbf{x}_{t_i 1}) = 0)}{\sum_{i=1}^T I(y_{t_i} = 0)}. \tag{9}$$

Related useful concepts are the *positive predictive value* and the *negative predictive value*, which depend on the prevalence of the condition $Y = 1$ in the population. It does not make sense to calculate them if the class frequencies for the considered n patients are not representative for the population of interest, as is often the case in case-control studies.

Decision theoretic aspects

When considering sensitivity and specificity it can be interesting to incorporate the idea of *cost* or *loss* functions from decision theory to evaluate misclassification costs. Instead of the error rate defined in Eq. (7), where a neutral cost function is used implicitly, one could use other cost functions, where the costs, and thus the weights in the computation of the error rate, are defined depending of the relative seriousness of misclassifications.

More precisely, a neutral, often referred to as scientific cost function assigns unit costs whenever an observation is misclassified (regardless of the true and predicted class), and no costs when the observation is correctly classified. However, if, for instance, classifying an observation with $Y = 1$ as $Y = 0$ is more serious than vice-versa, such errors should have more weight, i.e. higher misclassification costs. Many classifiers allow to assign such asymmetric misclassification costs, either directly or via class priors. The following principle is obvious for Bayesian methods, where different prior weights may be assigned to the response classes, but also applies to, e.g., classification trees. Imagine that there are much more observations in class 0 than in class 1. Then, in order to reduce the number of misclassifications predicting class 1 for all observations - regardless of the values of the predictor variables - would be a pretty good strategy, because it would guarantee a high number of correctly classified observations.

This principle can be used to train a classifier to concentrate on one class, even if the proportions of class 0 and 1 observations in the actual population and data set are equal: one either has to "make the classifier believe" that there were more observations of class 0 by means of setting a high artificial prior probability for this class, or one has to "tell" the classifier directly that misclassifications of class 0 are more severe by means of specifying higher misclassification costs (cf, e.g., Ripley, 1996). Obviously, such changes in the prior probabilities and costs, that are internally handled as different weights for class 0 and 1 observations, affect sensitivity and specificity. For example, when misclassification of a responder as a non responder is punished more severely than vice-versa, the sensitivity (for correctly detecting a responder) will increase, while at the same time the specificity (for correctly identifying a non-responder) will decrease, because the classifier will categorize more observations

as responders than under a neutral cost scheme.

From a decision theoretic point of view, what we considered as costs so far were really "regrets" in the sense that the overall costs, e.g., for diagnosing a subject, were not included in our reasoning: only the particular costs induced by a wrong decision were considered, while the costs of correct decisions were considered to be zero. This approach is valid for the comparison of classifiers because the additional costs, e.g., for diagnosing a subject are equal for all classifiers.

ROC curves

To account for the fact that the sensitivity and specificity of a classifier are not fixed characteristics, but are influenced by the misclassification cost scheme, the *receiver operating characteristic* (ROC) approach (cf., e.g., Swets, 1988, for an introduction and application examples) could be borrowed from signal detection, and could be used for comparing classifier performance, incorporating the performance under different cost schemes. Then, for each classifier a complete ROC curve describes the sensitivity and specificity under different cost schemes. The curves of two classifiers are directly comparable when they do not intersect. In this case the curve that is further from the diagonal, which would correspond to random class assignment, represents the better classifier. Confidence bounds for ROC curves can be computed (e.g., Schäfer, 1994). The distance from the diagonal, the so called *area under curve* (AUC), is another useful diagnostic (Hanley and McNeil, 1982) and can be estimated via several approaches (e.g., DeLong et al., 1988). The AUC can also be used to compare intersecting ROC curves.

After this overview on accuracy measures for the comparison of classifiers, the next section describes possible sampling strategies for the evaluation of accuracy measures. Suggestions on the use of these sampling strategies, as well as a discussion of possible abuses, are given in Section 5.

Credal classification

So far we have considered only the case that the classifier gives a clear class prediction for each observation, say 0 or 1. In addition to this we noted that some classifiers may also return predicted class probabilities instead. Obviously, when the probability

for class 1 is, say, 99% we would predict that class without hesitation. However, tree classifiers or ensemble methods that perform majority voting would also predict class 1 when its predicted probability is only, say, 51% - as long as the probability for class 1 is higher than that for class 0, no matter how little the difference. In such a situation one might argue that there should be a third option, namely refusing to predict a class whenever the predicted probability is within a certain threshold or returning the extra value "in doubt" (cf. Ripley, 1996, p. 5; 17 f.), if further information would be needed to classify an observation.

Several authors have argued along a similar line, for instance the fuzzy set approach by Chianga and Hsub (2002), whose classifier returns the predicted degree of possibility for every class rather than a single predicted class, and Zaffalon (2002), who argues in favor of so called "credal classification", where a subset of possible classes for each configuration of predictor variables is returned when there is not enough information to predict one single class (see also Zaffalon et al., 2003, for an application to dementia diagnosis).

4 Evaluation Strategies

For simplicity, we assume in the following that the error rate is used as an accuracy measure, but the same principles hold for other measures such as the sensitivity or the specificity. The goal of classifier evaluation is the estimation of the conditional error rate $Err(C^M_{\mathbf{D}})$ from Eq. 4 or of the unconditional error rate $\epsilon^M_{\mathbf{F}^n}$ (cf. Eq. 5), where the focus on $Err(C^M_{\mathbf{D}})$ or $\epsilon^M_{\mathbf{F}^n}$ depends on the concrete context. For example, a study that aims at designing a classifier based on a particular data set for concrete use in medical practice will focus on $Err(C^M_{\mathbf{D}})$ rather than $\epsilon^M_{\mathbf{F}^n}$, whereas a statistical comparison study of classification methods should be as general as possible, and thus focus on the unconditional error rate $\epsilon^M_{\mathbf{F}^n}$. Readers interested in the difference between conditional and unconditional error rate may refer to Molinaro et al. (2005); Efron and Tibshirani (1997). In general, the question whether unconditional or conditional inference should be preferred is one of the central foundational issues in statistics, where in the frequentist-Bayesian debate the former usually advocate in favor of the unconditional point of view while Bayesian inference is eo ipso conditional (cf., e.g., Berger, 1980, Section 1.6). Also a view at the corresponding discussion in sampling theory on evalu-

ating post stratification is illuminating in this context (see, e.g., Hold and Smith, 1979, for a classical paper).

In this article, we arbitrarily use the notation $\hat{\epsilon}$ for all the estimators, which refers to the unconditional error rate. However, the reviewed estimators can also be seen as estimators of $Err(C^M_{\mathbf{D}})$. For each method, we denote the estimator in a way that all the quantities influencing it are visible. These expressions, and the corresponding formulas, should be understood as pseudo-code to be used for implementing the procedure. In addition, all the methods reviewed in the present section are summarized in Table 1.

Resubstitution

The easiest – and from a statistical point of view by far the worst– evaluation strategy consists of building and evaluating a classifier based on the same data set $\mathbf{D_l}$. Usually, the data set $\mathbf{D_l}$ includes all the available data, i.e. $\mathbf{D_l} = \mathbf{D}$, yielding the estimator

$$\hat{\epsilon}^M_{RESUB}(\mathbf{D}) = \frac{1}{n}\sum_{i=1}^{n} I(y_i \neq C^M_{\mathbf{D}}(\mathbf{x}_i)) \quad (10)$$

$$= \widehat{Err}(C^M_{\mathbf{D}}, \mathbf{D}). \quad (11)$$

$\hat{\epsilon}^M_{RESUB}$ is a downwardly biased estimator of $\epsilon^M_{\mathbf{F}^n}$ and $Err(C^M_{\mathbf{D}})$, i.e. accuracy is overestimated. Since the constructed classifier $C^M_{\mathbf{D}}$ was especially designed to fit $\mathbf{D}$, it usually performs well on it. The problem of *overfitting*, i.e., that the classifier is too closely adapted to the learning sample, is not specific to microarray data, but it can be enhanced by their high dimensionality: with a huge number of predictor variables, a very subtle partition of the feature space can be achieved, yielding distinct predictions for very small groups of observations. In such a situation it is possible to find a prediction rule such that almost all observations from the learning data set are predicted correctly. However, this does not imply that the prediction rule that is highly adapted to the learning data set will also predict independent new observations correctly.

Test data set

To evaluate the performance of a classification method on independent observations, one should consider non-overlapping learning and test data sets. A classifier is built based on the learning data set only and subsequently applied to the test observations.

If, as above, $\mathbf{l}$ and $\mathbf{t}$ contain the indices of the observations included in the learning and test data sets, respectively, the error rate is estimated as

$$\hat{\epsilon}^M_{TEST}(\mathbf{D}, (\mathbf{l}, \mathbf{t})) = \frac{1}{T} \sum_{i=1}^{T} I(y_{t_i} \neq C^M_{\mathbf{D_l}}(\mathbf{x}_{\mathbf{t}_i})) \quad (12)$$

$$= \widehat{Err}(C^M_{\mathbf{D_l}}, \mathbf{D_t}), \quad (13)$$

where T denotes the size of $\mathbf{t}$. In practice, $\mathbf{l}$ and $\mathbf{t}$ most often form a partition of $\{1, \ldots, n\}$, i.e. $\mathbf{t} = \{1, \ldots, n\} \setminus \mathbf{l}$, and ϵ^M_{TEST} can be seen as a function of $\mathbf{D}$ and $\mathbf{l}$ only. However, we keep the notation as general as possible by including $\mathbf{t}$ in $\epsilon^M_{TEST}(\mathbf{D}, (\mathbf{l}, \mathbf{t}))$, in order to allow the specification of learning and test sets that do not form a partition of $\{1, \ldots, n\}$ (for instance, when there are two different test data sets). Note that, in contrast to resubstitution, this procedure may have a random component: it depends on the learning and test sets defined by $(\mathbf{l}, \mathbf{t})$. When $\mathbf{l}$ and $\mathbf{t}$ are not defined randomly but are chosen by the user (e.g., chronologically where the first recruited patients are assigned to $\mathbf{l}$ and the following patients to $\mathbf{t}$), $\hat{\epsilon}_{TEST}$ depends on the number of patients in $\mathbf{l}$, which is fixed by the user.

Note that, due to the fact that some of the observations from the learning data set are held back for the test set and thus the learning data set contains only $L < n$ observations, the estimation of the prediction rule from the learning data set is worse and the resulting prediction error increases. Therefore $\hat{\epsilon}^M_{TEST}$ has positive bias as an estimator of $\epsilon^M_{\mathbf{F}^n}$ and $Err(C^M_{\mathbf{D}})$, i.e. the obtained prediction accuracy is worse than if all n observations were used. This effect does not only occur here, where the original learning data set is split into one learning and test set, but also in the following sections whenever the number of observations in the learning data set is decreased. The .632 estimator introduced below addresses this problem.

For a discussion of potential changes in the data generating process over time see Section 5.

Cross-validation

Another option to evaluate prediction accuracy consists of considering all the available observations as test observations successively in a procedure denoted as *cross-validation* (see, e.g. Hastie et al., 2001). The available observations $\{1, \ldots, n\}$ are

divided into m non-overlapping subsets whose indices are given by $\mathbf{t}^{(1)}, \ldots, \mathbf{t}^{(m)}$. The cross-validation procedure consists of a succession of m iterations, hence the name m-fold cross-validation. In the j-th iteration, the observations defined by $\mathbf{t}^{(j)}$ are considered as test data and the remaining observations form the learning data set defined by $\mathbf{l}^{(j)} = \{1, \ldots, n\} \setminus \mathbf{t}^{(j)}$. The test observations from $\mathbf{D}_{\mathbf{t}^{(j)}}$ are then predicted with the classifier $C^M_{\mathbf{D}_{\mathbf{l}^{(j)}}}$ constructed using $\mathbf{D}_{\mathbf{l}^{(j)}}$.

A prediction is thus obtained for each of the n observations. The error rate is estimated as the mean proportion of misclassified observations over all cross-validation iterations:

$$\hat{\epsilon}_{CV}(\mathbf{D}, (\mathbf{t}^{(j)})_{j=1,\ldots,m}) = \sum_{j=1}^{m} \frac{n_{\mathbf{t}_j}}{n} \hat{\epsilon}^M_{TEST}(\mathbf{D}, (\mathbf{l}^{(j)}, \mathbf{t}^{(j)})). \tag{14}$$

This formula simplifies to

$$\hat{\epsilon}_{CV}(\mathbf{D}, (\mathbf{t}^{(j)})_{j=1,\ldots,m}) = \frac{1}{m} \sum_{j=1}^{m} \hat{\epsilon}^M_{TEST}(\mathbf{D}, (\mathbf{l}^{(j)}, \mathbf{t}^{(j)})). \tag{15}$$

if $\mathbf{t}^{(1)}, \ldots, \mathbf{t}^{(m)}$ are equally sized. Note that $\mathbf{l}^{(j)}$ does not appear as an argument of ϵ^M_{CV}, since $\mathbf{l}^{(j)}$ is derived deterministically from $\mathbf{t}^{(j)}$ as $\mathbf{l}^{(j)} = \{1, \ldots, n\} \setminus \mathbf{t}^{(j)}$.

In this setting again decision theoretic considerations could be very helpful, leading to criteria going beyond the mere averaging of misclassified observations. For instance, a more conservative approach inspired by the minimax-decision criterion would be to consider for each classifier the maximum, instead of the average, proportion of misclassified observations over all cross-validation samples, and finally choose the classifier with the minimal maximum proportion of misclassified observations over all classifiers. This approach could be called for in situations where not the average or expected performance is of interest but rather it is necessary to guarantee that a certain performance standard is held even in the worst case.

An important special case of cross-validation is $m = n$, where $\mathbf{t}^{(j)} = j$, i.e. the n observations are considered successively as singleton test data sets. This special case is usually denoted as *leave-one-out cross-validation* (LOOCV), since at each iteration one observation is left out of the learning data set. The corresponding error rate estimator can be expressed as

$$\hat{\epsilon}^M_{LOOCV}(\mathbf{D}) = \frac{1}{n} \sum_{i=1}^{n} \hat{\epsilon}^M_{TEST}(\mathbf{D}, i). \tag{16}$$

LOOCV is deterministic, in contrast to cross-validation with $m < n$ which possibly yields different results depending on the (randomly) chosen partition $\mathbf{t}^{(1)}, \ldots, \mathbf{t}^{(m)}$. As an estimator of $\epsilon^M_{\mathbf{F}^n}$ and $Err(C^M_{\mathbf{D}})$, $\hat{\epsilon}^M_{LOOCV}(\mathbf{D})$ is almost unbiased, since classifiers are built based on $n-1$ observations. However, as an estimator of $\epsilon^M_{\mathbf{F}^n}$, it can have high variance because the learning sets are very similar to each other (Hastie et al., 2001).

In order to reduce the variability of cross-validation results due to the choice of the partition $\mathbf{t}^{(1)}, \ldots, \mathbf{t}^{(m)}$, it has been proposed to average the results of cross-validation obtained for several different partitions. As an example, Braga-Neto and Dougherty (2004) examine what they denote as $CV10$:

$$\hat{\epsilon}^M_{CV10}(\mathbf{D}, (\mathbf{t}^{(j)k})_{k=1,\ldots,10,\ j=1,\ldots,m}) = \frac{1}{10}\sum_{k=1}^{10} \hat{\epsilon}_{CV}(\mathbf{D}, (\mathbf{t}^{(j)k})_{j=1,\ldots,m}), \qquad (17)$$

where $(\mathbf{t}^{(1)k}, \ldots, \mathbf{t}^{(m)k})$ is the partition corresponding to the k-th cross-validation. Note that, like $\hat{\epsilon}^M_{CV}$, the estimator $\hat{\epsilon}^M_{CV10}$ has a random component. However, its variance is reduced by averaging over several partitions.

In stratified cross-validation, each subset $\mathbf{t}^{(j)}$ contains the same proportion of observations of each class as the whole data set. It is well-established that stratified cross-validation improves the estimation of the error rate.

Monte-Carlo cross-validation (or subsampling)

Like cross-validation, *Monte-Carlo cross-validation* (MCCV) strategies consist of a succession of iterations and evaluate classification based on test data sets that are not used for classifier construction. It may be seen as an averaging of the test set procedure over several splits into learning and test data sets. In contrast to cross-validation, the test sets are not chosen to form a partition of $\{1, \ldots, n\}$. In Monte-Carlo cross-validation (also called random splitting or subsampling), the learning sets $\mathbf{l}^{(b)}$ ($b = 1, \ldots, B$) are drawn out of $\{1, \ldots, n\}$ randomly and without replacement. The test sets consist of the remaining observations $\mathbf{t}^{(b)} = \{1, \ldots, n\} \setminus \mathbf{l}^{(b)}$. The common size ratio $n_{\mathbf{l}^{(b)}} : n_{\mathbf{t}^{(b)}}$ is fixed by the user. Usual choices are, e.g., $2:1$, $4:1$ or $9:1$. Each test set contains the observations that are not in the corresponding learning set. The MCCV error rate is given as

$$\hat{\epsilon}^M_{MCCV}(\mathbf{D}, (\mathbf{l}^{(b)})_{b=1,\ldots,B}) = \frac{1}{B}\sum_{b=1}^{B} \hat{\epsilon}^M_{TEST}(\mathbf{D}, (\mathbf{l}^{(b)}, \mathbf{t}^{(b)})). \qquad (18)$$

This formula is identical to the formula of $\hat{\epsilon}^M_{CV}$ for regular cross-validation, except that the summation is done with respect to the B random subsamples and that $\hat{\epsilon}^M_{MCCV}$ is considered as a function of the learning sets instead of the test sets here for consistency with the bootstrap sampling procedure reviewed in the next section. As an estimator of $\epsilon^M_{\mathbf{F}^n}$, $\hat{\epsilon}^M_{MCCV}$ has a smaller variance than, e.g., $\hat{\epsilon}^M_{LOOCV}$, since it is based on learning sets that are not as highly correlated as those of LOOCV. However, $\hat{\epsilon}^M_{MCCV}$ is again upwardly biased as an estimator of both $\epsilon^M_{\mathbf{F}^n}$ and $Err(C^M_{\mathbf{D}})$, i.e., accuracy is underestimated, since the prediction rules are constructed based on less than n observations.

Bootstrap sampling

In bootstrap sampling, the learning sets $\mathbf{l}^{*(b)}$ are drawn out of $\{1, \ldots, n\}$ randomly and with replacement. The $*$ symbol indicates that each observation may be represented several times in $\mathbf{l}^{*(b)}$. The (common) size of the learning sets is set to n. Each $\mathbf{l}^{*(b)}$ includes an average of $1 - (1 - 1/n)^n \approx_{n\to\infty} 63.2\%$ of the n observations at least once. The test sets $\mathbf{t}^{(b)}$ are again formed by the observations which are not in the corresponding learning set $\mathbf{l}^{*(b)}$. Note that each test may have a different number of observations. In each of the B bootstrap iterations, the learning data set is used to construct a classifier $C^M_{\mathbf{D}_{\mathbf{l}^{*(b)}}}$ that is subsequently applied to the test set $\mathbf{D}_{\mathbf{t}^{(b)}}$. There are several variants for estimating the error rate based on these results. The first variant consists of considering all the predictions simultaneously and computing the global error as

$$\hat{\epsilon}^M_{BOOT1}(\mathbf{D}, (\mathbf{l}^{*(b)})_{b=1,\ldots,B}) = \frac{\sum_{i=1}^n \sum_{b=1}^B I_i^{(b)} \cdot I(y_i \neq C^M_{\mathbf{D}_{\mathbf{l}^{*(b)}}}(\mathbf{x}_i))}{\sum_{i=1}^n \sum_{b=1}^B I_i^{(b)}}, \tag{19}$$

with

$$\begin{aligned} I_i^{(b)} &= 0 \quad \text{if observation } i \text{ is included in the test set } \mathbf{l}^{*(b)} \text{ at least once,} \\ &= 1 \quad \text{else.} \end{aligned}$$

Note that the MCCV error estimator presented in the previous section may also be expressed in this way. In contrast, the second bootstrap variant considers each observation individually and estimates the error rate as

$$\hat{\epsilon}^M_{BOOT2}(\mathbf{D}, (\mathbf{l}^{*(b)})_{b=1,\ldots,B}) = \frac{1}{n} \sum_{i=1}^n \hat{E}_i, \tag{20}$$

where $\hat{E}_i$ is the averaged individual error rate of observation i over the iterations:

$$\hat{E}_i = \frac{\sum_{b=1}^{B} I_i^{(b)} \cdot I(y_i \neq C^M_{\mathbf{D}_{\mathbf{l}^{*(b)}}}(\mathbf{x}_i))}{\sum_{b=1}^{B} I_i^{(b)}}.$$

These two variants agree when $B \to 0$ and usually produce nearly identical results (Efron and Tibshirani, 1997).

Note that the principle of bootstrap learning samples that determine their own test samples (the observations not included in the current bootstrap sample, also called *"out-of-bag" observations*) is also incorporated in the recent ensemble methods *bagging* (Breiman, 1996) and *random forests* (Breiman, 2001). Here the prediction accuracy of ensembles of classifiers learned on bootstrap samples is evaluated internally on the out-of-bag observations. Therefore these methods have a built-in control against overoptimistic estimations of the error rate.

The $.632$ and $.632+$ estimators

Bootstrap estimators of the error rate are upwardly biased, since classifiers are built using in average only 63.2% of the available observations. That is why Efron and Gong (1983) suggest an estimation procedure that combines the bootstrap sampling error rate and the resubstitution error rate. They define the *.632 estimator* as

$$\hat{\epsilon}^M_{.632}(\mathbf{D}, (\mathbf{l}^{*(b)})_{b=1,\ldots,B}) = 0.368\hat{\epsilon}^M_{RESUB}(\mathbf{D}) + .632\hat{\epsilon}_{BOOT1}(\mathbf{D}, (\mathbf{l}^{*(b)})_{b=1,\ldots,B}) \quad (21)$$

which is designed to correct the upward bias in $\hat{\epsilon}_{BOOT1}$ by averaging it with the downwardly biased resubstitution error rate $\hat{\epsilon}^M_{RESUB}$. The $.632+$ estimator is suggested by Efron and Tibshirani (1997) as a less biased compromise between resubstitution and bootstrap errors designed for the case of strongly overfitting classifiers. These estimates have lower bias than MCCV or simple bootstrap sampling estimates. Their principle is generalized to survival prediction by Gerds and Schumacher (2007).

Bootstrap cross-validation

Fu et al. (2005) suggest an approach denoted as bootstrap cross-validation combining bootstrap estimation and LOOCV. The resulting error rate estimator can be seen as a bagging predictor, in the sense that the final error rate estimate results from the

	Iterations	Bias	Principle
Resubstitution	1	$\downarrow$	$\mathbf{l} = \mathbf{t} = \{1, \ldots, n\}$
Test	1	$\uparrow$	$\{\mathbf{l}, \mathbf{t}\}$ form a partition of $\{1, \ldots, n\}$
LOOCV	n	–	$\mathbf{t}^{(j)} = \{j\}, \mathbf{l}^{(j)} = \{1, \ldots, n\} \setminus \{j\}$, for $j = 1, \ldots, n$
m-fold-CV	m	$\uparrow$	$\mathbf{t}^{(1)}, \ldots, \mathbf{t}^{(m)}$ form a partition of $\{1, \ldots, n\}$ $\mathbf{l}^{(j)} = \{1, \ldots, n\} \setminus \mathbf{t}^{(j)}$, for $j = 1, \ldots, m$
MCCV	B (u.d.)	$\uparrow$	$\{\mathbf{l}^{(b)}, \mathbf{t}^{(b)}\}$ form a partition of $\{1, \ldots, n\}$, for $b = 1, \ldots, B$
Bootstrap	B (u.d.)	$\uparrow$	$\mathbf{l}^{*(b)}$ is a bootstrap sample drawn out of $\{1, \ldots, n\}$ $\mathbf{t}^{*(b)} = \{1, \ldots, n\} \setminus \mathbf{l}^{*(b)}$, for $b = 1, \ldots, B$
0.632,0.632+	B (u.d.)	–	Weighted sum of resubstitution and bootstrap error rates.
Bootstrap-CV	nB (u.d.)	–	LOOCV within B bootstrap samples.

Table 1: Summary of the reviewed evaluation strategies. **Iterations:** number of iterations, i.e. number of times a classifier is constructed and applied to data; u.d.= user-defined. **Bias:** Bias of the error estimation; $\uparrow$ means positive bias, i.e. underestimation of prediction accuracy and vice-versa. **Principle:** Gives the definition of the learning and test sets or the used combination of methods.

combination of several (LOOCV) estimates based on bootstrap samples. For each of the B bootstrap samples, LOOCV is carried out. Error estimation is then obtained by averaging the LOOCV result over the B bootstrap iterations.

Since bootstrap samples have duplicates, learning and test sets may overlap for the corresponding CV iterations. Fu et al. (2005) claim that such an overlapping should be seen as an advantage rather than a disadvantage for small samples, since correcting the upward bias of bootstrap error estimation. Bootstrap cross-validation is reported to perform better than bootstrap and the $.632$ and $.632+$ estimators (Fu et al., 2005).

5 Which evaluation scheme in which situation?

The evaluation of classification methods may have various goals. One goal may be to compare several classification methods from a methodological point of view and explain observed differences (for instance, Dudoit et al., 2002; Romualdi et al., 2003;

Statnikov et al., 2005). Medical or biological articles on the other hand are concerned with the performance *on future independent data* of the best classifier, which should be selected following a strict procedure (typically one of those used in the comparison studies mentioned above).

For that selection procedure, resubstitution should never be employed, since yielding far too optimistic estimates of accuracy. Even if the goal is to compare different methods rather than to estimate the absolute prediction accuracy, resubstitution turns out to be inappropriate, since artificially favoring those methods that overfit the learning data. Hence, an inescapable rule is that classifiers should not be evaluated only on the same data set they were trained on.

In this context, the above warning should be repeated: A classical flaw encountered in the literature consists of selecting variables based on the whole data set and building classifiers based on this reduced set of variables. This approach should be banned, see Ambroise and McLachlan (2002) for a study on this topic. Even (and especially) when the number of variables reaches several tens of thousands, variable selection must carried out for each splitting into learning and test data sets successively.

Cross-validation, Monte-Carlo cross-validation and bootstrap for classifiers comparison

In a purely statistical study with focus on the comparison of classification methods in high dimensional settings, it is not recommended to estimate prediction accuracy based on a single learning data set and test data set, because for limited sample sizes the results depend highly on the chosen partition (cf., e.g., Hothorn et al., 2005). From a statistical point of view, when the original learning data set is split into one learning and one test set, increasing the size of the test set decreases the variance of the prediction accuracy estimation. However, it also decreases the size of the leftover learning data set and thus increases the bias, since using less observations than available for learning the prediction rule yields an artificially high and variable error rate. In the case of a very small n, this might even lead to the too pessimistic conclusion that gene expression does not contribute to prediction. Procedures like cross-validation, Monte-Carlo cross-validation or bootstrap sampling may be seen as an attempt to decrease the estimation bias by considering larger learning sets, while limiting the variability

through averaging over several partitions into learning and test data sets.

Contradicting studies have been published on the comparison of CV, MCCV and bootstrap strategies for error rate estimation. The use of CV (Eq. 14) in small sample settings is controversial (Braga-Neto and Dougherty, 2004) because of its high variability compared to MCCV (Eq. 18) or bootstrap sampling (Eq. 19,20). For instance, in the case of $n = 30$, each observation accounts for more than 3% in the error rate estimation. For a data set in which, say, at most three patients are difficult to classify, CV does not allow a fair comparison of classification methods. Braga-Neto and Dougherty (2004) discourage from using LOOCV for estimation purposes in small sample settings and recommend bootstrap strategies or repeated CV (denoted as CV10 in the present article, see Eq. 17) as more robust alternatives. In contrast, another study by Molinaro et al. (2005) taking small sample size and high-dimensionality into account reports low mean square error for LOOCV estimation, as well as for 5- and 10-fold CV and the $.632+$ estimator. The low bias of LOOCV, its conceptual simplicity as well as the fact that it does not have any random component make it popular in the context of microarray data. Meanwhile, it has become a standard measure of accuracy used for comparing results from different studies. However, if one wants to use CV, a more recommendable approach consists of repeating cross-validation several times, i.e. with different partitions $\mathbf{t}^{(1)}, \ldots, \mathbf{t}^{(m)}$, when m can take the values, e.g. $m = 5$ or $m = 10$. Averaging over several partitions reduces the variance associated with cross-validation (Braga-Neto and Dougherty, 2004).

Stable estimates of prediction accuracy can also be obtained via MCCV or bootstrap sampling. In MCCV, the choice of the ratio $n_{\mathbf{l}} : n_{\mathbf{t}}$ might depend on the goal of the study. If the goal is comparison only, a ratio like $2 : 1$ may be appropriate. If one is not only interested in the relative performance of the methods but also in the value of the prediction accuracy itself, larger learning sets are conceivable. However, for both CV and MCCV/bootstrap, it must be recalled that the estimate of prediction accuracy *always* tends to be pessimistic compared to the prediction accuracy that would be obtained based on the n observations, since less than n observations are used for classifier construction. Less biased estimators such as $.632+$ are recommended if the absolute value of the error rate is of importance.

When on the other hand the aim of a benchmark study is a complete ranking of all considered classifiers with respect to any performance measure the Bradley-Terry(-

Luce) model for paired comparisons (Bradley and Terry, 1952) or the recent approach of Hornik and Meyer (2007) for consensus rankings are attractive. In addition to the purely descriptive ranking of these approaches statistical inference on the performance differences between classifiers can be conducted when the test samples are drawn appropriately, e.g., when several CV- or bootstrap-samples are available (Hothorn et al., 2005).

Validation in medical studies

In medical studies, the problem is different. Investigators are not interested in the methods themselves but in their practical relevance and validity for future independent patient data. The addressed questions are

1. Can reliable prediction be performed for new patients?

2. Which classification method should be used on these new data?

Whereas the second question is basically the same as in statistical studies, the first question is most often ignored in statistical papers, whose goal is rather to compare methods from a theoretical point of view than to produce 'ready-to-use' classifiers to be used in medical practice.

Question 1 can be answered reliably only based on several, or one large, validation data set that has been made available to the statistician after construction and selection of an appropriate classifier. A validation set that remains unopened until the end of the analysis is necessary, in the vein of the validation policy developed by the Sylvia Lawry Centre for Multiple Sclerosis Research (Daumer et al., 2007).

Choice of the validation data set

The impact of the reported classifier accuracy in the medical community increases with the differences between validation data set and open data set. For example, it is much more difficult to find similar results (and thus much more impressing when such results are found) on a validation data set collected in a different lab at a different time and for patients with different ethnical, social or geographical background than in a validation set drawn at random from an homogenous data set at the beginning of the analysis. An important special case is when the learning and validation sets are defined

chronologically. In this scheme, the first recruited patients are considered as learning data and used for classifier construction and selection *before* the validation data set is collected, hence warranting that the validation data remain unopened until the end of the learning phase. Obviously, evaluating a classifier on a validation data set does not provide an estimate of the error rate which would be obtained if both learning and validation data set were used for learning the classifier. However, having an untouched validation data set is the only way to simulate prediction of new data.

Furthermore, if the learning and test sets are essentially different (e.g., from an ethnical or technical point of view), bad performance may be obtained even with a classifier that is optimal with respect to the learning data. The error rate on the validation set increases with i) the level of independence between Y and $\mathbf{X}$, ii) the difference between the joint distribution $\mathbf{F}$ of Y and $\mathbf{X}$ in the learning and validation sets, iii) the discrepancy between the optimal Bayes classifier and the constructed classifier. Whereas the components i) and iii) are common to all methods of accuracy estimation, component ii) is specific to validation schemes in which "validation patients" are different from "learning patients".

In this setting, however, it does make a difference here, if the learning and test set(s) are (random) samples from the same original data set, or if the test set is sampled, e.g., in a different center in a multi-center clinical trial or at a different point in time in a long-term study. The first case – ideally with random sampling of the learning and test set(s) – corresponds to the most general assumption for all kinds of statistical models, namely the "i.i.d." assumption that all data in the learning and test set(s) are randomly drawn independent samples from the same distribution, and that the samples only vary randomly from this distribution due to their limited sample size. This common distribution is often called the data generating process (DGP). A classifier that was trained on a learning sample is supposed to perform well on a test sample from the same DGP, as long as it does not overfit.

A different story is the performance of a classifier learned on one data set and tested on another one from a different place or time. If the classifier performs bad on this kind of test sample this can have different reasons: Either important confounder variables were not accounted for in the original classifier, e.g. an effect of climate when the classifier is supposed to be generalized over different continents (cf. Altman and Royston, 2000, who state that models may not be "transportable"), or – even more

severe for the scientist – the DGP has actually changed, e.g., over time, which is an issue discussed as "data drift", "concept drift" or "structural change" in the literature. In this latter case, rather than discarding the classifier, the change in the data stream should be detected (Kifer et al., 2004) and modelled accordingly – or in restricted situations it is even possible to formalize conditions under which some performance guarantees can be proven for the test set (Ben-David et al., 2007).

When on the other hand the ultimate goal is to find a classifier that is generalizable to all kinds of test sets, including those from different places or points in time, as a consequence we would have to follow the reasoning of "Occam's razor" for our statistical models: the most sparse model is always the best choice other things being equal. Such arguments can be found in Altman and Royston (2000) and, more drastically, Hand (2006), who uses this argument not only with respect to avoiding overfitting and the inclusion of too many covariates, but also, e.g., in favor of linear models as opposed to recursive partitioning, where it is, however, at least questionable from our point of view, if the strictly linear, parametric and additive approach of linear models is really more "sparse" than, e.g., simple binary partitioning.

Recommendations

With respect to the first question posed at the beginning of this subsection we therefore have to conclude that there are at least one clinical and one – if not a dozen – statistical answers, while for the second question we have a clear recommendation. Question 2 should be addressed based on the open learning data set only via cross-validation, repeated cross-validation, Monte-Carlo cross-validation or bootstrap approaches. The procedure is as follows:

1. Define N_{iter} pairs of learning and test sets $(\mathbf{l}^{(j)}, \mathbf{t}^{(j)})$, $j = 1, \ldots, N_{iter}$, following one of the evaluation strategies described in Section 4 (LOOCV, CV, repeated CV, MCCV, bootstrap, etc). For example, in LOOCV, we have $N_{iter} = n$.

2. For each iteration ($j = 1, \ldots, N_{iter}$), repeat the following steps:

   - Construct classifiers based on $\mathbf{l}^{(j)}$ using different methods M_1,M_2,...,M_q successively, where M_r ($r = 1, \ldots, q$) is defined as the combination of the variable selection method (e.g., univariate Wilcoxon-based variable se-

lection), the number of selected variables (e.g., $\tilde{p} = 50, 100, 500$) and the classification method itself (e.g., linear discriminant analysis).

- Predict the observations from the test set $\mathbf{t}^{(j)}$ using the constructed classifiers $C^{M_1}_{\mathbf{D}_{\mathbf{l}(j)}}, \ldots, C^{M_q}_{\mathbf{D}_{\mathbf{l}(j)}}$ successively.

3. Estimate the error rate based on the chosen procedure for all methods $M_1, \ldots, M_q$ successively.

4. Select the method yielding the smallest error rate.

5. Apply it to predict the observations from the independent validation set.

A critical aspect of this procedure is the choice of the "candidate" methods $M_1, \ldots, M_q$. On the one side, trying many methods increases the probability to find a method performing better than the other methods "by chance". On the other side, obviously, increasing the number of methods also increases the chance of finding the right method, i.e. the method that best reflects to the true data structure and is thus expected to show good performance on independent new data as well.

CV, MCCV or bootstrap procedures might also be useful in medical studies for accuracy estimation, but their results should not be over-interpreted. They give a valuable preview of classifier accuracy when the collected data set is still not large enough for putting aside a large enough validation set. In this case a systematic and comprehensive optimization of the method parameters is not feasible. Then statisticians should not rely on suboptimal choices based on local optima found by trial and error but rather adopt one the following approaches:

- *Using the default parameters*.

- *Selecting parameter values by cross-validation* (or a related approach) within each iteration. The computational complexity of the last option is in n^2, which makes it prohibitive if the chosen classification method is not very fast, especially when it involves variable selection.

- *Selecting parameter values based on solid previous publications* analyzing other data sets.

In all cases, it should be mentioned that such an analysis does not replace an independent validation data set.

6 Summary and outlook

For fair evaluation of classifiers, the following rules should be taken into account.

- The constructed classifier should ideally be tested on a independent validation data set. If impossible (e.g., because the sample is too small), the error rate should be estimated with a procedure which tests the classifier based on data that were not used for its construction, such as cross-validation, Monte-Carlo cross-validation or bootstrap sampling.

- Variable selection should be considered as a step of classifier construction. As such, it should be carried out using the learning data only.

- Whenever appropriate, sensitivity and specificity of classifiers should be estimated. If the goal of the study is, e.g., to reach high sensitivity, it is important to design the classifier correspondingly.

Note that both the construction and the evaluation of prediction rules have to be modified if the outcome is not, as assumed in this paper, nominal, but ordinal, continuous or censored. While ordinal variables are very difficult to handle in the small sample setting and thus often dichotomized, censored survival variables can be handled using specific methods coping with the $n \ll p$ setting, see van Wieringen et al. (2007) for a neutral comparison study. Since censoring makes the use of usual criteria like the mean square error impossible, sophisticated evaluation procedures have to be used, such as the Brier score (see van Wieringen et al. (2007) for a review of several criteria).

Another aspect that has not been treated in the present paper because it would have gone beyond its scope is the stability of classifiers and classifier assessment. For instance, would the same classifier be obtained if an observation were removed from the data set? How does an incorrect response specification affect the classification rule and the estimation of its error rate? Further research is needed to answer these most relevant questions, which affect all microarray studies.

Further research should also consider the fact that due to the many steps involved in the experimental process, from hybridization to image analysis, even in high quality experimental data severe measurement error may be present (see, e.g., Rocke and Durbin, 2001; Tadesse et al., 2005; Purdom and Holmes, 2005). As a consequence,

prediction and diagnosis do not longer coincide, since prediction is usually still based on the mismeasured variables, while diagnosis tries to understand the material relations between the true variables. While several powerful procedures to correct for measurement error are available for regression models (see, e.g., Wansbeek and Meijer, 2000; Cheng and Ness, 1999; Carroll et al., 2006; Schneeweiß and Augustin, 2006, for surveys considering linear and nonlinear models, respectively), in the classification context well-founded treatment of measurement error is still in its infancy.

A further problem which is largely ignored by many statistical articles is the incorporation of clinical parameters into the classifier and the underlying question of the additional predictive value of gene expression data compared to clinical parameters alone. Although "adjustment for other classic predictors of the disease outcome [is] essential" (Ntzani and Ioannidis, 2003), this problem is largely ignored by most methodological articles. Specific evaluation and comparison strategies have to be developed to answer this question.

Acknowledgement

ALB was partly supported by the Porticus Foundation in the context of the International School for Clinical Bioinformatics and Technical Medicine.

References

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

Altman, D. G. and P. Royston (2000). What do we mean by validating a prognostic model? *Statistics in Medicine 19*, 453–73.

Ambroise, C. and G. J. McLachlan (2002). Selection bias in gene extraction in tumour classification on basis of microarray gene expression data. *Proceedings of the National Academy of Science 99*, 6562–6566.

Asyali, M. H., D. Colak, O. Demirkaya, and M. S. Inan (2006). Gene expression profile classification: A review. *Current Bioinformatics 1*, 55–73.

Ben-David, S., J. Blitzer, K. Crammer, and F. Pereira (2007). Analysis of representations for domain adaptation. In *Advances in Neural Information Processing Systems 20*, Cambridge, MA. MIT Press.

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

Berger, J. O. (1980). *Statistical Decision Theory and Bayesian Analysis (2nd edition)*. New York: Springer.

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

Bomprezzi, R., M. Ringnér, S. Kim, M. L. Bittner, J. Khan, Y. Chen, A. Elkahloun, A. Yu, B. Bielekova, P. S. Meltzer, R. Martin, H. F. McFarland, and J. M. Trent (2003). Gene expression profile in multiple sclerosis patients and healthy controls: identifying pathways relevant to disease. *Human Molecular Genetics 12*, 2191–2199.

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

Boulesteix, A. L. (2006). Reader's reaction to 'dimension reduction for classification with gene expression microarray data' by Dai et al (2006)'. *Statistical Applications in Genetics and Molecular Biology 5*, Issue 1, Article 16.

Boulesteix, A.-L. (2007). Wilcoxcv: An R package for fast variable selection in cross-validation. *Bioinformatics 1702-1704*, 23.

Boulesteix, A. L. and K. Strimmer (2007). Partial least squares: A versatile tool for the analysis of high-dimensional genomic data. *Briefings in Bioinformatics 8*, 32–44.

Bradley, R. and M. E. Terry (1952). Rank analysis of incomplete block designs, I. The method of paired comparisons. *Biometrika 39*, 324–345.

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

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

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

Carroll, R. J., D. Ruppert, L. A. Stefanski, and C. M. Crainiceanu (2006). *Measurement Error in Nonlinear Models: A Modern Perspective (2nd edition)*. New York: Chapman and Hall/CRC.

Chen, J. J. (2007). Key aspects of analyzing microarray gene-expression data. *Pharmacogenomics 8*, 473–482.

Cheng, C. L. and J. W. V. Ness (1999). *Statistical Regression with Measurement Error*. London: Arnold.

Chianga, I.-J. and J. Y.-J. Hsub (2002). Fuzzy classification trees for data analysis. *Fuzzy Sets and Systems 130*, 87–99.

Culhane, A., J. Thioulouse, G. Perriere, and D. G. Higgins (2005). MADE4: an R package for multivariate analysis of gene expression data. *Bioinformatics 21*, 2789–2790.

Daumer, M., U. Held, K. Ickstadt, M. Heinz, S. Schach, and G. Ebers (2007). Reducing the probability of false positive research findings by pre-publication validation. *Nature Precedings*.

DeLong, E. R., D. DeLong, and D. Clarke-Pearson (1988). Comparing the area under two or more correlated receiver operating characteristic curves: a nonparametric approach. *Biometrics 44*, 837–845.

Dettling, M. (2004). Bagboosting for tumor classification with gene expression data. *Bioinformatics 20*, 3583–3593.

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

Diaz-Uriarte, R. and S. A. de Andrés (2006). Gene selection and classification of microarray data using random forests. *BMC Bioinformatics 7*, 3.

Ding, B. and R. Gentleman (2005). Classification using generalized partial least squares. *Journal of Computational and Graphical Statistics 14*, 280–298.

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

Dupuy, A. and R. Simon (2007). Critical review of published microarray studies for cancer outcome and guidelines on statistical analysis and reporting. *Journal of the National Cancer Institute 99*, 147–157.

Efron, B. and G. Gong (1983). A leisurely look at the bootstrap, the Jackknife and cross-validation. *The American Statistician 37*, 36–48.

Efron, B. and R. Tibshirani (1997). Improvements on cross-validation: The .632+ bootstrap method. *Journal of the American Statistical Association 92*, 548–560.

Fort, G. and S. Lambert-Lacroix (2005). Classification using partial least squares with penalized logistic regression. *Bioinformatics 21*, 1104–1111.

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

Fu, W. J., R. J. Carroll, and S. Wang (2005). Estimating misclassification error with small samples via bootstrap cross-validation. *Bioinformatics 21*, 1979–1986.

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

Gerds, T. A. and M. Schumacher (2007). Efron-type measures of prediction error for survival analysis. *Biometrics*, doi:10.1111/j.1541–0420.2007.00832.x.

Ghadimi, B. M., M. Grade, M. J. D. an S. Varma, R. Simon, C. Montagna, and L. Fuzesi (2005). Effectiveness of gene expression profiling for response prediction of rectal adenocarcinomas to preoperative chemoradiotherapy. *Journal of Clinical Oncology 23*, 1826–1838.

Ghosh, D. (2003). Penalized discriminant methods for the classification of tumors from gene expression data. *Biometrics 59*, 992–1000.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. New York: Addison-Wesley.

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

Guo, Y., T. Hastie, and R. Tibshirani (2007). Regularized linear discriminant analysis and its application in microarrays. *Biostatistics 8*, 86–100.

Hand, D. J. (2006). Classifier technology and the illusion of progress. *Statistical Science 21*, 1–14.

Hanley, J. A. and B. McNeil (1982). The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology 143*, 29–36.

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

Hold, D. and T. M. F. Smith (1979). Post stratification. *Journal of the Royal Statistical Society 142*, 33–46.

Hornik, K. and D. Meyer (2007). Deriving consensus rankings from benchmarking experiments. In R. Decker and H.-J. Lenz (Eds.), *Advances in Data Analysis (Proceedings of the 30th Annual Conference of the Gesellschaft für Klassifikation e.V., Freie Universität Berlin, March 8-10, 2006.) Studies in Classification, Data Analysis, and Knowledge Organization*, pp. 163–170. Springer.

Hothorn, T., F. Leisch, A. Zeileis, and K. Hornik (2005). The design and analysis of benchmark experiments. *Journal of Computational and Graphical Statistics 14*, 675–699.

Huang, X., W. Pan, S. Grindle, X. Han, Y. Chen, S. J. Park, L. W. Miller, and J. Hall (2005). A comparative study of discriminating human heart failure etiology using gene expression profiles. *BMC Bioinformatics 6*, 205.

Ioannidis, J. P. (2005). Microarrays and molecular research: noise discovery. *The Lancet 365*, 488–492.

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

Jeffery, I. B., D. G. Higgins, and A. C. Culhane (2006). Comparison and evaluation of methods for generating differentially expressed gene lists from microarray data. *BMC Bioinformatics 7*, 359.

Kifer, D., S. Ben-David, and J. Gehrke (2004). Detecting change in data streams. In M. A. Nascimento, M. T. Özsu, D. Kossmann, R. J. Miller, J. A. Blakeley, and K. B. Schiefer (Eds.), *Proceedings of the Thirtieth International Conference on Very Large Data Bases, Toronto, Canada, August 31 - September 3, 2004*, pp. 180–191. Morgan Kaufmann.

Larranaga, P., B. Calvo, R. Santana, C. Bielza, J. Galdiano, I. Inza, J. A. Lozano, R. Armananzas, G. Santafe, A. Perez, and V. Robles (2006). Machine learning in bioinformatics. *Briefings in Bioinformatics 7*, 86–112.

Lee, J., J. Lee, M. Park, and S. SongS (2005). An extensive comparison of recent classification tools applied to microarray data. *Computational Statistics and Data Analysis 48*, 869–885.

Man, M. Z., G. Dyson, K. Johnson, and B. Liao (2004). Evaluating methods for classifying expression data. *Journal of Biopharmaceutical Statistics 14*, 1065–1084.

Molinaro, A., R. Simon, and R. M. Pfeiffer (2005). Prediction error estimation: a comparison of resampling methods. *Bioinformatics 21*, 3301–3307.

Natsoulis, G., L. E. Ghaoui, G. R. G. Lanckriet, A. M. Tolley, F. Leroy, S. Dunleo, B. P. Eynon, C. I. Pearson, S. Tugendreich, and K. Jarnagin (2005). Classification of a large microarray data set: Algorithm comparison and analysis of drug signatures. *Genome Research 15*, 724–736.

Nguyen, D. V. and D. Rocke (2002). Tumor classification by partial least squares using microarray gene expression data. *Bioinformatics 18*, 39–50.

Ntzani, E. E. and J. P. A. Ioannidis (2003). Predictive ability of DNA microarrays for cancer outcomes and correlates: an empirical assessment. *The Lancet 362*, 1439–1444.

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

Opgen-Rhein, R. and K. Strimmer (2007). Accurate ranking of differentially expressed genes by a distribution-free shrinkage approach. *Statistical Applications in Genetics and Molecular Biology 6*, 9.

Purdom, E. and S. P. Holmes (2005). Error distribution for gene expression data. *Statistical Applications in Genetics and Molecular Biology 4*, Article 16.

R Development Core Team (2006). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing.

Ripley, B. D. (1996). *Pattern Recognition and Neural Networks*. Cambridge, UK: Cambridge University Press.

Rocke, D. and B. Durbin (2001). A model for measurement error for gene expression arrays. *Journal of Computational Biology 8(6)*, 557–569.

Romualdi, C., S. Campanaro, D. Campagna, B. Celegato, N. Cannata, S. Toppo, G. Valle, and G. Lanfranchi (2003). Pattern recognition in gene expression profiling using DNA array: a comparison study of different statistical methods applied to cancer classification. *Human Molecular Genetics 823–836*, 12.

Schäfer, H. (1994). Efficient confidence bounds for ROC curves. *Statistics in Medicine 13*, 1551–1561.

Schneeweiß H. and T. Augustin (2006). Some recent advances in measurement error models and methods. *Allgemeines Statistisches Archiv - Journal of the German Statistical Association 90*, 183–197; also printed in: Hübler, O. and Frohn, J. (Eds.) (2006): Modern Econometric Analysis – Survey on Recent Developments, 183–198.

Smyth, G. (2004). Linear models and empirical Bayes methods for assessing differential expression in microarray experiments. *Statistical Applications in Genetics and Molecular Biology 3*, 3.

Soukup, M., H. Cho, and J. K. Lee (2005). Robust classification modeling on microarray data using misclassification penalized posterior. *Bioinformatics 21*, i423–i430.

Soukup, M. and J. K. Lee (2004). Developing optimal prediction models for cancer classification using gene expression data. *Journal of Bioinformatics and Computational Biology 1*, 681–694.

Spiegelhalter, D. J. (1986). Probabilistic prediction in patient management and clinical trials. *Statistics in Medicine 5*, 421–433.

Statnikov, A., C. F. Aliferis, I. Tsamardinos, D. Hardin, and S. Levy (2005). A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis. *Bioinformatics 21*, 631–643.

Statnikov, A., I. Tsamardinos, Y. Dosbayev, and C. F. Aliferis (2005). GEMS: A system for automated cancer diagnosis and biomarker discovery from microarray gene expression data. *International Journal of Medical Informatics 47*, 491–503.

Stolovitzky, G. (2003). Gene selection in microarray data: the elephant, the blind man and our algorithms. *Current Opinion in Structural Biology 13*, 370–376.

Strobl, C., A.-L. Boulesteix, A. Zeileis, and T. Hothorn (2007). Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC Bioinformatics 8:25*.

Swets, J. A. (1988). Measuring the accuracy of diagnostic systems. *Science 240*, 1285–1293.

Tadesse, M. G., J. G. Ibrahim, R. Gentleman, S. Chiaretti, J. Ritz, and R. Foa (2005). Bayesian error-in-variable survival model for the analysis of genechip arrays. *Biometrics 61*, 488–497.

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

Trevino, V. and F. Falciani (2006). GALGO: an R package for multivariate variable selection using genetic algorithms. *Bioinformatics 22*, 1154–1156.

Troyanskaya, O. G., M. E. Garber, P. O. Brown, D. Botstein, and R. B. Altman (2002). Non-parametric methods for identifying differentially expressed genes in microarray data. *Bioinformatics 18*, 1454–1461.

van Wieringen, W., D. Kun, R. Hampel, and A.-L. Boulesteix (2007). Survival prediction using gene expression data: a review and comparison. *submitted*.

Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory*. New York: Springer.

Wansbeek, T. and E. Meijer (2000). *Measurement Error and Latent Variables in Econometrics*. Amsterdam: Elsevier.

Zaffalon, M. (2002). The naive credal classifier. *Journal of Statistical Planning and Inference 105*, 5–21.

Zaffalon, M., K. Wesnes, and O. Petrini (2003). Reliable diagnoses of dementia by the naive credal classifier inferred from incomplete cognitive data. *Artificial Intelligence in Medicine 29(1-2)*, 61–79.

Zhu, J. (2004). Classification of gene expression microarrays by penalized logistic regression. *Biostatistics 5*, 427–443.

Appendix A: Overview of software implementing classification methods in R

Most methods for microarray-based classification are implemented in R (www.R-project.org) which has become the standard statistical tool for handling high-dimensional genomic data. Simple univariate variable selection might be performed, e.g., based on the t-test (t.test) or the Mann-Whitney test (wilcox.test). Usual classifiers like logistic regression (R function glm), linear discriminant analysis (R function lda), quadratic discriminant analysis (R function qda) are also accessible in R without loading any particular package. The same holds for PCA dimension reduction (R function prcomp). Here is a list of specific R packages that are of particular interest for microarray-based classification and freely available without registration.

- pamr package for PAM (Tibshirani et al., 2002)
- rda package for shrunken centroids regularized discriminant analysis (Guo et al., 2007)
- plsgenomics package for PLS-based classification (Boulesteix, 2004; Fort and Lambert-Lacroix, 2005)
- gpls package for generalized partial least squares classification (Ding and Gentleman, 2005)
- e1071 package for SVM (Furey et al., 2000)
- randomForest for random forests classification (Diaz-Uriarte and de Andrés, 2006)
- logitBoost package for logitBoost (Dettling and Bühlmann, 2003)
- BagBoosting package for bagboosting (Dettling, 2004)
- MADE4 package for classification by the 'between-group analysis' (BGA) dimension reduction method (Culhane et al., 2005)
- pdmclass package for classification using penalized discriminant methods (Ghosh, 2003)

- `MLInterfaces` package including unifying functions for cross-validation and validation on test data in combination with various classifiers

Packages including functions for gene selection are

- `genefilter` package including a function that computes t-tests quickly
- `WilcoxCV` package for fast Wilcoxon based variable selection in cross-validation (Boulesteix, 2007)
- `varSelRF R` package for variable selections with random forests (Diaz-Uriarte and de Andrés, 2006)
- `GALGO R` package for variable selection with genetic algorithms (Trevino and Falciani, 2006) (`http://www.bip.bham.ac.uk/vivo/galgo/AppNotesPaper.htm`).
- `MiPP` package to find optimal sets of variables that separate samples into two or more classes (Soukup and Lee, 2004; Soukup et al., 2005)

Other software tools not based on `R` are reviewed in Statnikov et al. (2005).

Appendix B: Summary of six comparison studies of classification methods

Dudoit et al. (2002) 3 data sets MCCV 2:1	• Included: LDA, DLDA, DQDA, Fisher, kNN, trees,tree-based ensembles • Variable selection: F-statistic *Conclusion:* DLDA and kNN perform best
Romualdi et al. (2003) 2 data sets CV	• Included: DLDA, trees, neural networks SVM, kNN, PAM combined with: • Variable selection/dimension reduction: PLS, PCA, soft thresholding, GA/kNN *Conclusion*: PLS transformation is recommendable, No classifier uniformly better than the other
Man et al. (2004) 6 data sets LOOCV, bootstrap	• Included: kNN, PCA+LDA, PLS-DA, neural networks, random forests, SVM • Variable selection: F-statistic *Conclusion:* PLS-DA and SVM perform best
Lee et al. (2005) 7 data sets LOOCV, MCCV 2:1	• Included: 21 methods (e.g., tree ensembles, SVM, LDA, DLDA, QDA, Fisher, PAM) • Variable selection: F-statistic, rank-based score, soft thresholding *Conclusion:* No classifier uniformly better than the other, Rank-based variable selection performs best
Statnikov et al. (2005) 11 data sets LOOCV, 10-fold CV	• Included: SVM, kNN, probabilistic neural networks, backpropagation neural networks • Variable selection: BSS/WSS, Golub et al. (1999), Kruskal-Wallis test *Conclusion:* SVM performs best
Huang et al. (2005) 2 data sets LOOCV	• Included: PLS, penalized PLS, LASSO, PAM, random forests • Variable selection: F-statistic • Random forests perform slightly better *Conclusion:* No classifier uniformly better than the other

Table 2: Summary of six comparison studies of classification methods. This summary should be considered with caution, since not detailing the used variants of the considered methods.