LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

INSTITUT FÜR STATISTIK

Christoph Bernau, Thomas Augustin & Anne-Laure Boulesteix

Correcting the optimally selected resampling-based error rate: A smooth analytical alternative to nested cross-validation

Technical Report Number 105, 2011
Department of Statistics
University of Munich

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

Correcting the optimally selected resampling-based error rate: A smooth analytical alternative to nested cross-validation

Christoph Bernau[1], Thomas Augustin[2], and Anne-Laure Boulesteix[1]
[1] Department for Medical Informatics, Biometry and Epidemiology.
Marchioninistr. 15, D-81377 Munich, Germany
[2] Department of Statistics, University of Munich.
Ludwigstr 33, D-80539 Munich, Germany

High-dimensional binary classification tasks, e.g. the classification of microarray samples into normal and cancer tissues, usually involve a tuning parameter adjusting the complexity of the applied method to the examined data set. By reporting the performance of the best tuning parameter value only, over-optimistic prediction errors are published. The contribution of this paper is two-fold. Firstly, we develop a new method for tuning bias correction which can be motivated by decision theoretic considerations. The method is based on the decomposition of the unconditional error rate involving the tuning procedure. Our corrected error estimator can be written as a weighted mean of the errors obtained using the different tuning parameter values. It can be interpreted as a smooth version of nested cross-validation (NCV) which is the standard approach for avoiding tuning bias. In contrast to NCV, the weighting scheme of our method guarantees intuitive bounds for the corrected error. Secondly, we suggest to use bias correction methods also to address the bias resulting from the optimal choice of the classification method among several competitors. This method selection bias is particularly relevant to prediction problems in high-dimensional data. In the absence of standards, it is common practice to try several methods successively, which can lead to an optimistic bias similar to the tuning bias. We demonstrate the performance of our method to address both types of bias based on microarray data sets and compare it to existing methods. This study confirms that our approach yields estimates competitive to NCV at a much lower computational price.

1 Introduction

Resampling-based procedures are routinely applied in order to assess the performance of statistical learning methods by estimating their prediction error. If the available data set were large enough, it would be recommended to partition the data into learning and validation data, to fit a model using the learning data, and to estimate its error based on the validation data. In the common case of small sample high-dimensional data considered in the present paper, however, the available data set is usually too small for such a partitioning. Resampling-based procedures are thus particularly useful in the context of "$n \ll p$" data analysis, i.e. when the number of predictors exceeds the number of observations.

In practice, most common classification methods for high-dimensional data involve a tuning parameter, e.g. the cost parameter in linear Support Vector Machines (SVM) or the number of neighbors in k-nearest-neighbors (kNN). If the error of a classification method is estimated by a resampling method several times with different values of the tuning parameter successively, each parameter value possibly yields a different estimated error. The approach consisting in selecting the parameter value yielding the smallest resampling error estimate and only reporting this resampling estimate is biased [5]. That is because the minimal resampling error can be seen as the result of an optimal selection. As such, it is a biased estimate of the generalization error rate, i.e. of the error that would be obtained with this parameter value on independent data. This bias, that was quantitatively assessed by [16] in the "$n \ll p$" setting, is often denoted as *tuning bias*. Note that the term "tuning" may be ambiguous since researchers from different fields might have different understandings of tuning. In this paper, we consider a parameter as a tuning parameter if it is not optimized by an analytical method (like the least squares criterion for the coefficients in linear regression) but rather by trying several values successively and using the value yielding the best prediction performance on test data. When choosing the parameter value based on the performance yielded by different candidate values, one indirectly uses the test data for learning the decision function, leading to an optimistic bias.

The same type of bias occurs if a researcher tries out several classification methods successively and reports only the results of the method yielding the minimal error rate. For instance, suppose we compute the resampling error rate of Support Vector Machines (SVM), Random Forests (RF), k-nearest-neighbors (kNN), and L_1-penalized regression for a particular data set. Suppose further that kNN yields the smallest error rate in the resampling approach. This error rate is likely to be smaller than the error rate of kNN on independent data, because it was *optimally* selected across four error estimates that all show some variability. The resulting bias which we denote as *method selection bias* in this paper may be considerable, as illustrated by [3, 9]. In an empirical study based on real microarray data sets, [3] show that systematically selecting the method with the smallest cross-validation (CV) error rate can result in an error estimate as low as 30% [3] even after permutation of the class labels (i.e. with fully uninformative predictors).

In the context of microarray-based classification, [16] suggest to apply nested cross-validation (NCV) to correct for the tuning bias outlined above. NCV is based on an

additional *internal* CV loop performed for tuning purposes – in contrast to the external CV performed to estimate the error. In this approach, *internal CV* is performed within each learning set. The value yielding the smallest error in internal CV is then selected and used to predict test observations in external CV. In this way, for each external CV iteration the choice of the parameter value is performed without using information from the test set, thus addressing the tuning bias outlined above.

A similar procedure might also be used to address the method selection bias induced by the optimal choice of the classification method. However, the NCV technique is computationally expensive, since it requires an additional CV loop on each learning set. The computational burden might rapidly become intractable, especially if the considered classification methods themselves involve tuning parameters that also have to be optimized using internal CV. Moreover, three embedded CVs are not only a computational challenge. The size of the learning sets decreasing with each CV loop, one finally has to work with substantially smaller learning sets which are not representative of the larger total sample. Furthermore, NCV tends to yield highly variable results, sometimes leading to absurd "corrected" errors outside the range of the original errors of the considered methods. In the context of tuning bias correction, [14] suggest a computationally effective alternative to NCV that does not rely on internal CV. Like NCV, this method could also be generalized to the correction of the method selection bias considered here. However, it tends to strongly over-estimate the bias in some settings, as already acknowledged by the authors themselves.

In this paper, we suggest an alternative bias correction approach which also does not rely on an internal cross-validation loop and can be applied to address both the tuning and the method selection bias. We decompose the unconditional error rate in such a way that the corrected error estimate is given as a weighted mean of the resampling errors obtained using the different parameter values/methods. The weight of a particular parameter value/method is the unconditional probability that it yields the minimal error estimate. In a broad sense, NCV can also be seen as a weighted procedure, where the weights are empirically determined in internal CV. On the contrary, we estimate the weights using an analytical approach based on the results of the external CV only. Our method can be interpreted as a smooth version of NCV that builds the average of the global error estimates instead of averaging iteration-wise errors like NCV. This procedure guarantees intuitive bounds, increases stability compared to NCV, and reduces the computation time drastically since it does not rely on internal CV.

The rest of the paper is structured as follows. Section 2 introduces the settings and notations and recalls the different types of error rates in this framework. This section also revisits existing approaches in the perspective of bias correction. Section 3 presents our new correction method. In section 4, this method is illustrated and compared to existing approaches based on four cancer microarray data sets as well as modified versions of these data sets with completely randomly generated class labels. This comparison focuses on tuning bias correction as well as method selection bias. Finally, section 5 summarizes and discusses some characteristics of our method.

2 Error rate of the best method and nested cross validation

2.1 Settings and notations

From a statistical point of view, binary supervised classification can be described in the following way. On the one hand we have a response variable taking values in $\mathcal{Y} = \{1, 2\}$. On the other hand we have predictors taking values in $\mathcal{X} \subset \mathbf{R}^p$ that will be used for constructing a classification rule. Predictors and response follow an unknown joint distribution on $\mathcal{X} \times \mathcal{Y}$ denoted by $P(\mathbf{x}, y)$. The observed i.i.d. sample of size n is denoted by $s_0 = (\mathbf{x}_1, y_1)...(\mathbf{x}_n, y_n)$. The classification task consists in building a decision function $\hat{f}$ that maps elements of the predictor space $\mathcal{X}$ into the response space $\mathcal{Y}$:

$$\hat{f}^S : \quad \mathcal{X} \quad \mapsto \quad \mathcal{Y} \quad , \quad \mathbf{x} \quad \mapsto \quad \hat{f}^S(\mathbf{x}),$$

where the superscript S indicates that the decision function is built using the sample S. From now on, we denote by method k (with $k \in 1, \ldots, K$) the considered combination of method and tuning parameter values. As an example, method 1 may stand for SVM with cost= 1, method 2 for kNN with k= 10, and so on. As a special case, $1, \ldots, K$ might represent different parameter values of the same method. The decision function obtained by fitting the prediction method k to the sample s_0 is denoted as $\hat{f}_k^{s_0}$.

2.2 Estimating the error

Suppose we have estimated a decision function $\hat{f}^S$. The true prediction error that has to be estimated can be written as

$$\varepsilon[\hat{f}_k^{s_0}] = \mathbf{E}_P \left[L\left(\hat{f}^S(\mathbf{x}), y\right)\right] = \int_{\mathcal{X}\times\mathcal{Y}} L\left(\hat{f}^S(\mathbf{x}), y\right) dP(\mathbf{x}, y), \tag{1}$$

where $\mathbf{E}_P$ stands for the mean over the joint distribution P and $L(.,.)$ is an adequate loss function, e.g. the indicator loss yielding the error rate considered in this paper.

The true error $\varepsilon[\hat{f}_k^{s_0}]$ of method k constructed using sample s_0 is denoted by $\varepsilon[\hat{f}_k^{s_0}] = \varepsilon(k \parallel s_0)$. This error is commonly referred to as *conditional* error since it corresponds to the decision function constructed on the specific sample s_0. In this perspective, $\varepsilon(k \parallel S)$ should be seen as a random variable, where S stands for a random sample that follows the distribution P^n. The mean

$$\varepsilon^n(k) = \mathbf{E}_{P^n}\left[\varepsilon(k \parallel S)\right] \tag{2}$$

of the random variable $\varepsilon(k \parallel S)$ is usually referred to as the *unconditional* true error rate of method k. It depends only on the method k, on the size n of the sample S and on the joint distribution P, and can be seen as a fixed quantity.

Since the joint distribution $P(x, y)$ is unknown, the conditional errors $\varepsilon(1 \parallel s_0), \ldots, \varepsilon(K \parallel s_0)$ and the unconditional errors $\varepsilon^n(1), \ldots, \varepsilon^n(K)$ have to be estimated. Standard estimation approaches are based on CV or repeated subsampling, see e.g. [4] for an overview. We focus on the repeated subsampling method in this paper because our new correction method involves the estimation of the unconditional variance of the estimated error.

To our knowledge the estimator proposed by [11] – which is used here – works for repeated subsampling only, and we are not aware of any convincing alternative estimator applicable to CV.

In repeated subsampling the whole data set is randomly split into learning and test sets several times. Each learning set $L_b,\ b = \{1, \ldots, B\}$ of size n_L (with $n_L < n$) is used to estimate a decision function that is subsequently evaluated on the corresponding test set $S \setminus L_b$. For each iteration $b = \{1, \ldots, B\}$ and each method $k,\ k = \{1, \ldots, K\}$, one obtains an estimated error $e(k \parallel L_b, S \setminus L_b)$, where the term "$L_b, S \setminus L_b$" means that method k is fitted to the learning set L_b and evaluated on the test set $S \setminus L_b$. Note that we use the notation e for estimators and ε for true errors. In contrast to the conditional true error $\varepsilon(k \parallel S)$, the estimated error $e(k \parallel L_b, S \setminus L_b)$ is conditional on the considered sample not only with regard to the estimation of the decision function but also with regard to the estimation of the error. For each method k, the iteration-wise test errors are eventually combined into an error rate estimate by averaging over the iterations $b = 1, \ldots, B$, yielding

$$e(k \parallel S) = \frac{1}{B} \sum_{b=1}^{B} e(k \parallel L_b, S \setminus L_b), \tag{3}$$

which obviously may depend on the random choices of the partitions $\{L_b, T_b\},\quad b = 1, \ldots, B$, a fact that however is supressed in the notation.

2.3 The "best" method

Let us further denote the method yielding the smallest error rate based on S as $k^*(S)$, i.e. $k^*(S) = \arg\min_k e(k \parallel S)$. Note that the random variable $k^*(S)$ depends not only on the sample S but also on the considered learning sets $L_b,\ b = 1, \ldots, B$. In our notation we will again ignore this dependence on the specific learning sets.

For a given sample s_0, the error estimate $e(k^*(s_0) \parallel s_0)$ obtained by repeated resampling incorporates a source of a downward bias because $k^*(s_0)$ is chosen based on s_0, i.e. such that $e(k^*(s_0) \parallel s_0)$ is minimal. If one simply chooses the method yielding the minimal error rate $e(k^*(s_0) \parallel s_0)$, this minimal error rate underestimates the true conditional error rate $\varepsilon(k^*(s_0) \parallel s_0)$ of the chosen method. The problem is that the same sample s_0 is used both for error estimation and for the choice of the optimal classification method ($k^*(s_0)$). The corresponding classification rule $\hat{f}_k^{s_0}$ is expected to perform worse on an independent sample s_1 which was not used for choosing the method. This bias is related to the problem of multiple comparisons. The minimal error rate out of K methods decreases with increasing K. The resulting bias can also be seen as the result of the variability of the estimates $e(k||S)$.

In this paper, we aim at correcting the bias of $e(k^*(s_0) \parallel s_0)$ as an estimator of $\varepsilon(k^*(s_0) \parallel s_0)$, because, roughly speaking, we are interested in the expected performance of the "best method" $k^*(s_0)$ on independent data. Since the only available data are s_0, $\varepsilon(k^*(s_0) \parallel s_0)$ can obviously not be estimated directly. Thus, we reformulate the problem as the estimation of

$$Err = \mathbf{E}_{P_n}(\varepsilon(k^*(S) \parallel S)) = \varepsilon^n(k^*(S)). \tag{4}$$

The error $\varepsilon(k^*(s_0) \parallel s_0)$, that we want to estimate, is a realization of the random variable $\varepsilon(k^*(S) \parallel S)$ whose mean over P^n is Err. We show in Section 2.4 that the well-known NCV estimator can be reformulated as an estimator of Err, and we suggest an alternative estimator in Section 3.

2.4 Revisiting Nested Cross-Validation

In this section, we propose an interpretation of the well-known NCV method as a natural estimator of $Err = \mathbf{E}_{P_n}(\varepsilon(k^*(S) \parallel S)$ from Eq. (4). Note that, strictly speaking, the term "NCV" implies that CV is used both in internal CV (to choose the tuning parameter/method) and in external CV (to estimate the error rate). However, the idea of NCV can be directly generalized to other evaluation schemes such as repeated subsampling. In this paper, we stick to the standard terminology "NCV" but use repeated subsampling to estimate the error in the external loop. In our notation, the NCV error estimate can be written as

$$\widehat{Err}_{NCV} = \frac{1}{B}\sum_{b=1}^{B} e(k^{*b}(L_b) \parallel L_b, S \setminus L_b). \tag{5}$$

This formula, which is at first view very similar to formula (3), can be interpreted as follows. For each iteration b ($b = 1, \ldots, B$), the following procedure is repeated. Firstly, the "L_b-best method" $k^{*b}(L_b)$ is determined in internal CV within L_b. Secondly, the classification rule fitted on L_b using the best method $k^{*b}(L_b)$ is evaluated on $S \setminus L_b$, yielding $e(k^{*b}(L_b) \parallel L_b, S \setminus L_b)$.

The difference to Eq. (3) is that NCV builds the average error of the best methods $k^{*b}(L_b)$ (as assessed in internal CV) instead of averaging the error rates of a specific method k. Note that these L_b-best methods again vary with the choice of the internal learning sets which means that one may not obtain the same final results when repeating the same procedure twice – even if the outer learning sets L_b are fixed. Roughly speaking, in NCV the parameter $Err = \mathbf{E}_{P_n}(\varepsilon(k^*(S) \parallel S))$ is estimated through averaging over B subsets of s_0. Each term $e(k^{*b}(L_b) \parallel L_b, S \setminus L_b)$ can be seen as an estimator of $\varepsilon(k^{*b}(L_b) \parallel L_b)$, which roughly plays the role of a realization of $\varepsilon(k^*(S) \parallel S)$. Note, however, that the L_b subsets are smaller than s_0, which implies over-estimation of the error rate.

Most importantly, the determination of $k^{*b}(L_b)$ within each iteration is computationally expensive, which makes NCV very difficult to apply in practice when the prediction methods are time consuming, especially when they involve a tuning step that itself has to be performed through internal CV. As a consequence, the determination of $k^{*b}(L_b)$ is in practice often based on a fast procedure such as 3-fold-CV, yielding even more variable results than other resampling approaches. In extreme cases, this high variability may lead to estimates $\widehat{Err}_{NCV}$ larger than $\max_k e(k \parallel s_0)$ or lower than $\min_k e(k \parallel s_0)$, which is very unintuitive. Motivated by these inconveniences, we suggest an alternative computational effective estimator of Err in the Section 3.

2.5 Method proposed by [14]

[14] recognize the inconveniences of NCV in this context and suggest an alternative fast and simple method. Their basic idea is to estimate the tuning bias in each resampling iteration and then build the average over the B iterations. Although the method is originally presented in a CV framework, it can also directly be applied to repeated subsampling. The corrected error rate estimate $\widehat{Err}_{TT}$ suggested by [14] is obtained as the sum of the minimal estimated error rate $e(k^*(S) \parallel S)$ and the average of the differences $e(k^*(S) \parallel L_b, S \setminus L_b) - e(k^{\sharp b}(S \setminus L_b) \parallel L_b, S \setminus L_b)$ which can be interpreted as the counterpart of the tuning bias on the repective folds:

$$\begin{aligned} \widehat{Err}_{TT} &= \frac{1}{B}\sum_{1=b}^{B} e(k^*(S) \parallel L_b, S \setminus L_b) \\ &+ \frac{1}{B}\sum_{1=b}^{B} \left[e(k^*(S) \parallel L_b, S \setminus L_b) - e(k^{\sharp b}(S \setminus L_b) \parallel L_b, S \setminus L_b) \right], \end{aligned} \tag{6}$$

where $k^{\sharp b}(S \setminus L_b)$ denotes the method/tuning parameter performing best on the bth test fold $S \setminus L_b$ corresponding to the bth resampling iteration. This approach is computationally efficient, since it is based on the $K \times B$ estimated error rates $e(k \parallel L_b, S \setminus L_b)$. It does not require any additional computations. However, a major problem is that the error rates $e(k \parallel L_b, S \setminus L_b)$ obtained on the single test sets are much more variable than the error rates $e(k \parallel S)$ that are averaged over $B \gg 1$ test sets. Therefore the method proposed by [14] often overestimates the bias and the corrected error, as already noticed in their simulation study with non-informative data.

3 A smooth analytical alternative to NCV

3.1 Principle

The rationale behind NCV is that the construction of the decision function *and* the tuning/ method selection process, which are normally applied to the whole sample $S = s_0$, are mimicked on each learning set L_b of the external CV. In this way the tuning/method selection process is empirically incorporated into the estimation procedure. In practice, the best method $k^{*b}(L_b)$ is typically not the same for all iterations $b = 1, \ldots, B$. Hence, NCV builds a hard-weighted average of error estimates obtained with different methods or tuning parameters. By hard-weighted, we mean that for each resampling iteration b only one of the $e(k \parallel L_b, S \setminus L_b)$ $(k = 1, \ldots, K)$ is chosen (by internal CV) to be included in the average. The weight of $e(k^{*b}(L_b) \parallel L_b, S \setminus L_b)$ is 1, while the weight of all other $e(k \parallel L_b, S \setminus L_b)$ (for $k \neq k^{*b}(L_b)$) is 0. It is important to note that this way results from different tuning parameters are eventually combined.

Our new method is also based on a combination of error estimates of different parameter values/methods, though in a completely different and more direct way. While NCV combines errors of different parameter values/methods $e(k \parallel L(b), S \setminus L_b)$ computed for

different test sets, the new procedure combines the global error estimates $e(k \parallel s_0)$ of the different parameter value/methods k. Furthermore, the way these average errors are combined does not depend on an empirical experiment as performed in the internal CV. Our main idea is to decompose the unconditional error rate $\mathbf{E}_{P_n}\left[\varepsilon(k^*(S) \parallel S)\right]$ with regard to the random variable $k^*(S)$, i.e. the index of the best method:

$$\mathbf{E}_{P_n}\left[\varepsilon(k^*(S) \parallel S\right] = \sum_{k=1}^{K} P\left(k^*(S) = k\right) \cdot \mathbf{E}\left[\varepsilon(k \parallel S) | k^*(S) = k\right]. \tag{7}$$

As argued below, in most cases, it is reasonable to assume that, for a fixed method k,

$$\varepsilon(k \parallel S) \perp k^*(S), \tag{8}$$

i.e. the conditional error rate of method k constructed on S is independent from $k^*(S)$. It follows from Eq. (8) that the conditional expectations in Eq. (7), which cannot be estimated easily, can be replaced by the respective unconditional expectations:

$$\mathbf{E}_{P_n}(\varepsilon(k^*(S) \parallel S) \approx \sum_{k=1}^{K} P\left(k^*(S) = k\right) \cdot \mathbf{E}_{P_n}\left[\varepsilon(k \parallel S)\right]. \tag{9}$$

In order to perform this approximation of $\mathbf{E}_{P_n}(\varepsilon(k^*(S) \parallel S)$ we only have to estimate the quantities in Eq. (9). The terms $\mathbf{E}_{P_n}\left(\varepsilon(k \parallel S)\right)$ can be estimated by $e(k \parallel s_0)$. The probabilities $\mathrm{P}(k^*(S) = k)$ are more difficult to estimate, see Section 3.2 for details.

Before that, let us come back to the crucial assumption (8). It means that the true error rate $\varepsilon(k \parallel s_0)$ of method k fitted on s_0 does not depend on which method performed best in repeated subsampling based on s_0 – the unconditional error rates $\varepsilon^n(1), \ldots, \varepsilon^n(K)$ being fixed. Note that this assumption, of course, should not be misinterpreted in the sense that parameter tuning with CV is useless. Even if assumption (8) holds, tuning is useful to identify which method may have the smallest unconditional error rate $\varepsilon^n(k)$. A counter-example for which assumption (8) does not completely hold is support vector machines (SVM) – denoted as k_1 here – in the case of a sample with a mislabeled observation. The error rate $\varepsilon(1 \parallel s_0)$ of SVM is likely to be large, because SVM classifiers are strongly affected by mislabeled observations that often take the role of support vectors. Hence, $k^*(s_0) = 1$ is not likely. Thus, in this case, we obviously do not have $\varepsilon(k \parallel S) \perp k^*(S)$. However, especially in the presence of variable selection, assumption (8) holds in most cases, as illustrated by [8] based on an extensive empirical study.

3.2 Estimating $P(k^*(S) = k)$

The most complicated task in our approximation is to derive appropriate estimates for $P(k^*(S) = k)$, $k = 1, \ldots, K$. A pragmatic solution to this problem is suggested in the rest of this section. We assume continuous distributions of the errors and $|\mathrm{Cor}(e(k_1 \parallel S), e(k_2 \parallel S))| \neq 1$, $k_1 \neq k_2$. The latter includes the restriction that the methods

$1, \ldots, K$ have to be truly different. Together these assumptions imply $\mathrm{P}(e(k_1 \parallel S) = e(k_2 \parallel S)) = 0$, $k_1 \neq k_2$. Now $\mathrm{P}(k^*(S) = k)$ can be reformulated as

$$\mathrm{P}(k^*(S) = k) \;=\; \mathrm{P}(e(k \parallel S) < e(k' \parallel S)\ \forall k' \neq k). \tag{10}$$

We suggest a procedure based on a parametric modeling of this distribution. More precisely, we assume a multivariate normal distribution and estimate its mean and covariance. The mean μ of vector $\mathbf{e}$ is simply estimated as $\hat{\mu} = (e(1\|s_0), \ldots, e(K \parallel s_0))^\top$. The correlation coefficients $\mathrm{Cor}(e(k_1 \parallel S), e(k_2 \parallel S))$ (for $k_1, k_2 = 1, \ldots, K$, $k_1 \neq k_2$) are estimated as the corresponding sample correlation between $e(k_1 \parallel L_b, S \setminus L_b)$ and $e(k_2 \parallel L_b, S \setminus L_b)$, $b = 1, \ldots, B$. The most delicate task is the estimation of the variances of the estimators $e(k \parallel S)$, $k = 1, \ldots, K$. [11] suggest two estimators in this context. One of them involves repeated random splitting of the data sets into two equally sized subsets, which is impossible in our case both for computational reasons and because the resulting sample size would be too small. The other estimator proposed by [11] is given through

$$\left(\frac{1}{B} + \frac{\widehat{\rho}}{1-\widehat{\rho}}\right) \cdot \frac{1}{B-1} \sum_{b=1}^{B} \left(e(k \parallel L_b, s_0 \setminus L_b) - e(k \parallel s_0)\right)^2, \tag{11}$$

where ρ is the correlation between the errors $e(k \parallel L_b, s_0 \setminus L_b)$ $(b = 1, \ldots, B)$ obtained in different iterations from the same sample s_0, and $\widehat{\rho}$ is its estimator. [11] suggest to use the simple estimator $\widehat{\rho} = \frac{n - n_L}{n}$, which we also adopt here. This estimator works for repeated subsampling only, which is also the reason why we use repeated subsampling in our simulations.

Using these estimates of the mean, the correlation matrix and the respective variances of the random vector $\mathbf{e} = (\mathbf{e}(\mathbf{1} \parallel \mathbf{S}), \ldots, \mathbf{e}(\mathbf{K} \parallel \mathbf{S}))^\top$ and assuming a multivariate normal distribution, we can easily estimate the probability at which each method k performs best on independent test data, i.e. $\mathrm{P}(k^*(S) = k)$. For each k this probability can be reformulated in the following way:

$$\begin{aligned} \mathrm{P}(k^*(S) = k) &= \mathrm{P}(e(k \parallel S) \leq e(j \parallel S), \forall j : j \neq k) \\ &= \mathrm{P}(e(k \parallel S) - e(j \parallel S) \leq 0, \forall j : j \neq k). \end{aligned}$$

Consequently, if we consider all the $K-1$ differences $\delta_j = e(k \parallel S) - e(j \parallel S)$ for $j \neq k$, which are simple linear combinations of the original random vector $e(1 \parallel S), \ldots, e(K \parallel S)$, the probability $\mathrm{P}(k^*(S) = k)$ can be estimated from the density of the multivariate normal distribution of the random vector of differences δ as the integral

$$\mathrm{P}(e(k \parallel S) - e(j \parallel S) \leq 0, \forall j : j \neq k) \approx$$

$$\int_{-\infty}^{0} \cdots \int_{-\infty}^{0} \frac{1}{(2\pi)^{\frac{K-1}{2}} \sqrt{\mathbf{T}\hat{\mathbf{\Sigma}}\mathbf{T}^\top}} \exp\left((\delta - \mathbf{T}\hat{\mu})^\top \left(\mathbf{T}\hat{\mathbf{\Sigma}}\mathbf{T}^\top\right)^{-\mathbf{1}} (\delta - \mathbf{T}\hat{\mu}) \right) \Pi_j d\delta_j,$$

where the $(K-1) \times K$ matrix $\mathbf{T}$ contains the linear combinations yielding the corresponding differences, i.e. such that $\delta = \mathbf{Te}$. These integrals can be approximated very

precisely by usual statistical software like the function *pmvnorm* from the **R**-package *mvtnorm* [6]. Computation times of this function are marginally small in comparison with computation times of other steps of the analysis. Of course, the normality assumption can not hold exactly since the considered errors are averages of binary variables. In order to assess the deviation from the normal distribution we provide normal quantile plots for the distribution of the average errors for all classifiers and all simulation setups in the web-based supplementary materials (Web Appendix E). These distributions depend on the respective data set and method. In many cases the assumption seems to hold whereas some plots indicate that these distributions tend to more extreme values than expected under normality assumptions.

3.3 A weighted mean approach

Eventually, our novel estimator Err_{WMC} of $\mathbf{E}_{P_n}\left[\varepsilon(k^*(S) \parallel S)\right]$ is given as

$$\widehat{Err}_{WMC} = \sum_{k=1}^{K} \hat{P}\left(k^*(S) = k\right) \cdot e(k \parallel s_0). \tag{12}$$

The estimator in Eq. (12) can be interpreted as a weighted mean of the average resampling errors of the different tuning parameters/methods. The terms $\hat{P}\left(k^*(S) = k\right)$ represent the weights. This is the reason why we refer to our approach as Weighted Mean Correction (WMC).

A natural question is whether different weights may also be appropriate. A naive approach consists in giving equal weights to all parameters/methods, i.e. in replacing $\hat{P}(k^*(S) = k)$ by $1/K$ in Eq. (12). Note that this equal weight approach can be considered as a sensible upper bound for the corrected error because it corresponds to a random choice of the parameter/method. By definition, a random choice cannot lead to a tuning or method selection bias. That is why we do not expect any corrected error to be higher than

$$Err_{RawMean} = \sum_{k=1}^{K} \frac{1}{K} \cdot e(k \parallel s_0). \tag{13}$$

Regardless of the way these weights are chosen, there is a strong relationship between our approach and NCV. Our new estimator can be paralleled to the NCV estimator through a reformulation as

$$\widehat{Err}_{WMC} \;=\; \frac{1}{B}\sum_{k=1}^{K}\sum_{b=1}^{B} \hat{\mathrm{P}}\left(k^*(S) = k\right) \cdot e(k \parallel L_b, S \setminus L_b). \tag{14}$$

Similarly, the estimator $\widehat{Err}_{NCV}$ can also be reformulated as

$$\widehat{Err}_{NCV} \;=\; \frac{1}{B}\sum_{k=1}^{K}\sum_{b=1}^{B} I(k^{*b}(L_b) = k) \cdot e(k \parallel L_b, S \setminus L_b). \tag{15}$$

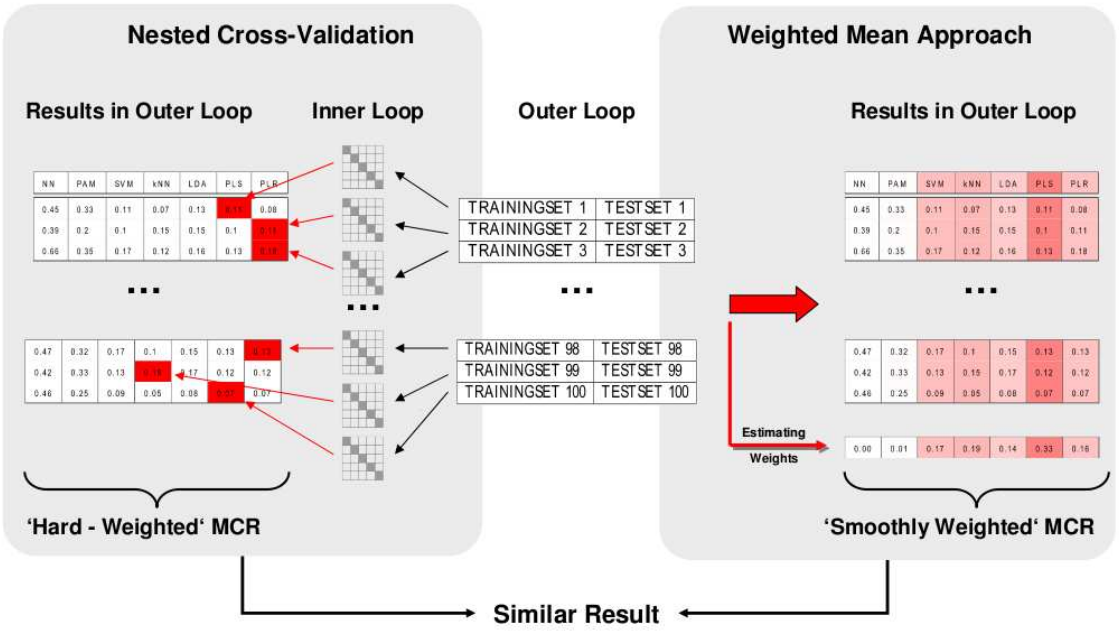

Figure 1: Comparison of the weighted approach and nested cross-validation. The misclassification rate (MCR) obtained by the weighted approach (compare to Eq. (14)) can be interpreted as a smoothed variant of the one obtained by nested cross-validation (compare to Eq. (15)).

Finally, the two estimators $\widehat{Err}_{NCV}$ and $\widehat{Err}_{WMC}$ have a similar form. The crucial difference is that $\widehat{Err}_{WMC}$ smoothly weights the errors $e(k \parallel L_b, S \setminus L_b)$ with the probabilities $\hat{P}(k^*(S) = k)$ estimated from an analytical parametric model (whose parameters are estimated from the quantities $e(k \parallel L_b, S \setminus L_b)$ only). On the contrary, in $\widehat{Err}_{NCV}$ the weights are empirical, discrete and depend on the results of a computationally intensive internal CV. Figure 1 illustrates the similarities and differences of the two methods.

3.4 Detour: Decision theoretic motivation

In this context we would like to provide a decision theoretic motivation of the tuning or method selection procedure. In decision theory one has to choose among different actions (the tuning parameters or methods) in dependence of certain states of nature. The goal is to minimize the global risk which is the unconditional error rate in our tuning setup. There are two options for the definition of the states of nature. Principally, the states of nature are specific characteristics of the data which indicate the use of a specific method/tuning parameter. These characteristics may refer either to the specific sample drawn from a data generating process (DGP) or to the DGP itself. Assuming the first definition, one can introduce an experiment in order to obtain information on the actual characteristics of the particular sample. This experiment corresponds to the resampling procedure in which we try to estimate the risks of the different actions. Using the

experiment one can define a strategy, e.g. that one always uses the action with minimal estimated risk. Looking at NCV, we see that it actually estimates the risk of such a sample-based strategy ($k^*(S)$) because it reselects the tuning parameter/method on each training fold of the outer loop according to the inner CV loop. As far as the estimation of a strategy is concerned, an inconsistency occurs with NCV at this point. NCV performs only one experiment although we know that the inner CV strongly depends on the specific partitioning. Due to this partitioning variability one could rather speak of a probabilistic strategy, since even for fixed states of nature one has certain probabilities at which each tuning parameter/method gets chosen. This is exactly the point where our method tries to improve NCV. We take into account that the tuning procedure in the inner loop could have decided differently depending on the specific partitioning and we estimate the probabilities $\mathrm{P}(k^*(S) = k)$ in order to mimic the probabilistic strategy. Please note that NCV is introduced as an intuitive approach for tuning bias correction and [16] do not exactly define the quantity they want to estimate.

The difference between both definitions for the states of nature manifests itself in the crucial assumption $\varepsilon(k \parallel S) \perp k^*(S)$. Basically, this assumption means that resampling can be a useful tool for finding suitable tuning parameters/methods for a certain DGP but not for the particular samples inside a DGP, i.e. now we are dealing with DGP-based strategies in contrast to the sample-based strategies mentioned before. If we assume that the $\mathrm{P}(k^*(S) = k)$ are equal for all possible samples of the DGP, $k^*(S)$ actually becomes a combined action once the DGP is fixed. In decision theory a combined action is a random strategy that does not use an additional experiment to collect further information but simply chooses certain actions according to specific weights. Please note that in this broader context the unconditional error rate in Eq. (2), that we are estimating, can actually be interpreted as a conditional error rate, conditioned on the fixed DGP. Of course, any resampling method can also define such a DGP based strategy if one takes into account that it is an experiment which can choose actions on any DGP that might exist. In this sense the independence assumption formulated above can be interpreted as the presumption that usually resampling is not such a precise tool that it is really able to detect "bad" samples produced by a fixed DGP.

3.5 Implementation

The weighted mean correction method is implemented in the **R**-function *weighted.mcr* included in a new version of the Bioconductor package *CMA* [13] that can be downloaded from the companion website (`http://www.ibe.med.uni-muenchen.de/organisation/mitarbeiter/070_drittmittel/bernau/cvbias/index.html`). The codes implementing our analyses are also provided there.

4 Empirical results and comparison of the three estimators

The goal of the study is to compare our estimator $\widehat{Err}_{WMC}$ (Eq. (12)) to the existing estimators $\widehat{Err}_{NCV}$ (Eq. (5)) and $\widehat{Err}_{TT}$ (Eq. (6)) and to the naive raw mean estimator

$\widehat{Err}_{RawMean}$ (Eq. (13)). Additionally, we also compare these estimates to the minimal error $\min_k e(k \parallel s_0)$ and the maximal error $\max e(k \parallel s_0)$, which can be considered as natural bounds for corrected errors.

4.1 Study design

This study is based on four microarray data sets: a colon cancer data set [1] included in Bioconductor package *colonCA* with $n = 62$ diseased or healthy tissues and $p = 1991$ variables, a prostate cancer data set [12] with $n = 102$ diseased or healthy patients and $p = 12625$ variables, a leukemia data set [7] included in Bioconductor package *CMA* with $n = 38$ patients with two different leukemia subtypes and $p = 3051$ variables, and an ALL-leukemia data set included in Bioconductor package *ALL* [10] with $n = 100$ patients with and without relapse and $p = 12625$ variables. Additionally, we consider modified versions of these four data sets obtained by replacing the response Y by a randomly generated Bernoulli distributed variable $Y' \sim \mathcal{B}(1, 0.5)$. These modified data sets are denoted as "non-informative" setup, in contrast to the original version of the data sets including "informative" predictors.

In the whole study, error rate estimation is performed through repeated subsampling into learning and test sets with $B = 100$ subsampling iterations. The proportion of observations included in the learning sets is set to 80% and 63.2% successively. In contrast to the three other estimators, $\widehat{Err}_{NCV}$ involves another parameter, the number of folds in internal CV. There are no commonly accepted guidelines to choose the number of folds in internal CV, which can be seen as a further inconvenience of NCV. In this study, it is chosen such that each internal test set contains approximately 5 observations. In each setup, the whole procedure is repeated $T = 50$ times in order to analyze the variability of the results. By "repeated", we mean that $T = 50$ different sets of partitions $(L_b, T_b)_{b=1,\ldots,B}$ are considered successively for the original data sets, and that $T = 50$ different randomly generated responses Y' are considered successively for the modified data sets.

As outlined in the introduction and in Section 2, our methodology can both be applied to the correction of the tuning bias or to the correction of the method selection bias. To illustrate these two powerful features, we successively consider two setups. In the first setup (illustrating the correction of the tuning bias and denoted as "tuning setup", methods $1, \ldots, K$ stand for different parameter values of a unique classification method. Two classifiers are considered successively. The first classifier is k-nearest-neighbors (kNN), where methods $1, \ldots, K$ correspond to different values $(1, \ldots, 15)$ of the parameter "number of neighbors". The second classifier is Partial Least Squares dimension reduction followed by Linear Discriminant Analysis (PLS-LDA) as described in [2] where methods $1, \ldots, K$ correspond to different numbers $(1, \ldots, 10)$ of PLS components. In both cases, a preliminary variable selection is performed by selecting the variables yielding the lowest p-values with the two-sample t-test (50 variables for kNN, 250 variables for PLS-LDA). Note that, in all resampling iterations, variable selection is performed using the learning set only. With NCV, this holds for the outer as well as for the inner loop.

Table 1: Average corrected errors (over 50 replications) for informative pls and selection (sel) setups with training set portion 80%.

Setup	NCV	WMC	TT	Raw	Min	Max
pls-alon	0.176	0.168	0.194	0.186	0.149	0.21
pls-singh	0.087	0.08	0.099	0.091	0.075	0.18
pls-golub	0.048	0.03	0.045	0.035	0.024	0.041
pls-allrel	0.431	0.417	0.461	0.441	0.398	0.459
sel-alon	0.164	0.163	0.182	0.19	0.142	0.257
sel-singh	0.097	0.092	0.111	0.133	0.083	0.319
sel-golub	0.026	0.018	0.008	0.061	0.004	0.226
sel-allrel	0.398	0.383	0.414	0.42	0.365	0.452

In the second setup (illustrating the correction of method selection bias and denoted as "selection setup" (sel)), methods $1, \ldots, K$ correspond to different combinations of classification methods and parameter values. The parameters are fixed, because tuning them with internal CV would imply three embedded CVs for $\widehat{Err}_{NCV}$, which is computationally intractable. The following classification methods are considered: nearest shrunken centroids [15] with $\Delta = 0.5$, linear SVM with $cost = 50$, kNN with $k = 1$ neighbor based on the 20 top-variables, kNN with $k = 18$ neighbors based on the 50 top-variables, Diagonal Linear Discriminant Analysis (DLDA) based on the 20 top-variables, PLS-LDA with 3 PLS components based on the 100 top-variables) and L_2-penalized logistic regression with penalty $\lambda = 0.01$.

4.2 Study results

Table 1 gives a representative overview of the results for the setups using the original response whereas results for the non-informative setups can be found in Table 2. More results on each specific setup are given in the web-based supplementary materials (Web Appendices A–D, Web Tables 1–4). Tables 1 and 2 provide the averages over the 50 replications for the three error rates $\widehat{Err}_{WMC}$, $\widehat{Err}_{NCV}$ and $\widehat{Err}_{TT}$ as well as for the raw mean $\widehat{Err}_{RawMean}$, and the minimal and the maximal error rates $\min_k e(k \parallel s_0)$ and $\max e(k \parallel s_0)$.

As can be seen from Table 1, in most informative setups the new weighted approach yields corrected errors that do not differ from the NCV approach by more than 1.5% in average. When differences are observed, the corrected error estimated by the new approach is most often lower than its NCV counterpart. However, these differences can be considered as negligible considering the variability of the estimates across the 50 replications, and we also find setups, especially with the Alon data, where the new weighted mean approach produces slightly higher estimates.

In most setups the NCV errors range between the new weighted mean errors and the raw mean errors. As pointed out before, the raw mean error is another sensible

upper bound for the corrected error because it corresponds to a random choice of the parameter/method which obviously cannot lead to a tuning or method selection bias. We do not expect a good correction method to produce estimates higher than the raw mean approach. Corrected errors estimated by NCV or the method suggested by [14], however, fall beyond this upper bound in some of the investigated setups, which makes poor sense in most situations and may be considered as an important disadvantage. Note, however, that in some hypothetical scenarios our method may also yield estimates higher than the raw mean. If several parameter values/methods perform similarly and another one yields a large but highly variable error, the latter may have a weight larger than $\frac{1}{K}$ in $\widehat{Err}_{WMC}$. Its unconditional probability to yield the minimal error may exceed $\frac{1}{K}$ even though it yields the maximal error in the present sample. Consequently, in particular situations we might have $\widehat{Err}_{WMC} > \widehat{Err}_{RawMean}$. However, such scenarios are quite unlikely hypothetical constructs. In our setups, our method indeed never produces estimates substantially higher than the raw mean.

In comparison with the approach by [14], the weighted mean approach provides less variable results which remain within intuitive bounds. On the one hand, the approach by [14] yields estimates close to those of NCV in many cases. On the other hand, it sometimes produces overly pessimistic corrected error estimates, for example on the Alon or ALL data set, where they sometimes noticeably exceed the raw mean error and even the worst error rate $\max_k e(k \parallel S)$ (results not shown here). In some non-informative setups Tibshirani's method yields corrected error estimates of 65% or 70%. The average corrected error is higher than the average maximal error rate in almost all non-informative setups. Corrected error rates exceeding the error rate of the worst classifier can be considered as obvious failures of this correction method. In this context, it is worth mentioning that NCV-corrected errors are also not upper-bounded by $\max_k e(k \parallel S)$. However, NCV-corrected errors higher than $\max_k e(k \parallel S)$ occur rarely and only with the Golub data – which is characterized by extremely small errors.

While our new weighted mean approach for error correction yields similar results to NCV in most informative setups, it tends to produce slightly over-optimistic results, i.e. to underestimate the error, in non-informative setups where the corrected error should approximately equal 50%. A representative example is displayed in Figure 2 representing the corrected errors obtained for 50 different sets of learning and test sets $(L_b, T_b)_{b=1,\ldots,B}$ with the four considered approaches using the Alon data set (with 80% observations in the training sets, non-informative selection setup). According to Table 2, the average corrected error over the 50 replications differs from NCV by at most 3.8% in the non-informative setups. This problem primarily occurs with the selection setups which are the most challenging setups for our new method due to the heterogeneity of the candidate methods. In our normal quantile plots (Web Figures 49–96) we can also observe the largest deviations from the normal distribution in these setups, especially for the error rates obtained by support vector machines. Quite generally, the selection setup seems to be more problematic for all correction methods, because the differences between the lowest and the highest errors are usually larger than in the tuning setups. Nonetheless, even in this difficult setup the new weighted mean approach produces results

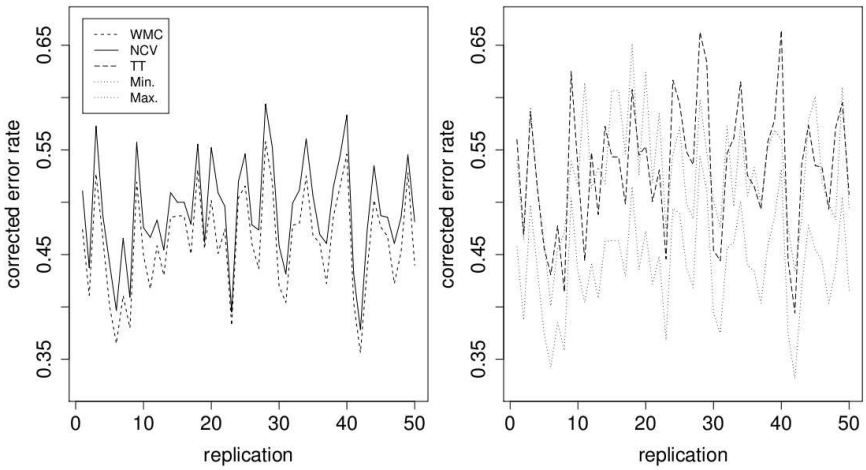

Figure 2: Comparison of NCV, the new weighting based method and the approach proposed by [14] for the non-informative selection setup on the Alon data set (proportion of observations in the training data: 80%).

Table 2: Average corrected errors (over 50 replications) for non-informative pls and selection (sel) setups with training set portion 80%.

Setup	NCV	WMC	TT	Raw	Min	Max
pls-alon	0.502	0.483	0.558	0.504	0.465	0.538
pls-singh	0.494	0.482	0.546	0.492	0.468	0.524
pls-golub	0.5	0.479	0.534	0.495	0.463	0.533
pls-allrel	0.495	0.482	0.54	0.498	0.469	0.523
sel-alon	0.492	0.462	0.532	0.488	0.44	0.531
sel-singh	0.504	0.479	0.535	0.503	0.463	0.541
sel-golub	0.498	0.466	0.543	0.493	0.443	0.536
sel-allrel	0.505	0.484	0.542	0.501	0.468	0.532

close to those of NCV, although there was room for large deviations as can be seen from an example in Figure 3.

5 Discussion and concluding remarks

We have proposed a new weighting-based method for tuning bias correction which avoids the additional computational costs of nested cross-validation while producing comparable results. In addition our method cannot only be applied in the well-known context of parameter tuning but also to address the method selection bias resulting from the optimal

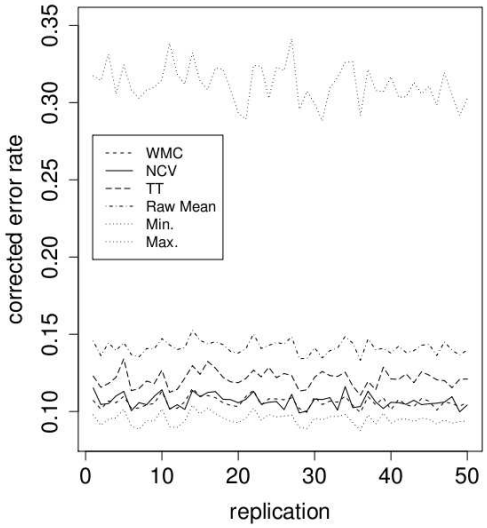

Figure 3: Comparison of NCV, the new weighting based method and the approach proposed by [14] for the informative selection setup on the Singh data set (proportion of observations in the training data: 63%).

choice of the method. To our knowledge, correction of the latter bias has never been addressed explicitly in the literature, neither with NCV nor with any other approach. In such selection setups our method performs slightly worse than in the tuning setups in the sense that it tends to under-correction. This is probably due to the high heterogeneity of the considered methods which complicates the estimation of the unconditional mean and variance of the estimated errors $e(k \parallel S)$. The estimation of these quantities in resampling setups is still subject of current research in the fields of statistics and machine learning. Our method might benefit from these researches, including a possible extension to cross-validation and bootstrap resampling.

Besides the lower computational effort, an important advantage of our method over NCV is that the obtained corrected error remains within sensible bounds defined by the minimal and maximal errors. As shown in the results section, NCV may produce estimates outside this interval. Regardless of whether the NCV-estimates fall above the highest error or below the lowest error, such a "correction" makes poor sense. Our correction method is clearly superior in such cases. Another extreme situation where our method yields more plausible results is when all tuning parameter values/methods lead to the approximately the same error estimate ($e(1 \parallel s) \approx \cdots \approx e(K \parallel s)$). In this case our correction method does not perform any correction, which is intuitively reasonable. On the contrary, NCV may produce a different corrected error. This is essentially due to the additional variability component induced by NCV. Whereas the results of the new weighted mean correction are deterministic (once the outer learning sets are fixed), NCV depends on the specific choice of the internal learning sets when selecting the L_b-best

method $k^*(L_b)$. This aspect of NCV is consistent with its main idea of mimicking the selection or tuning process on each learning set of the resampling approach. Nevertheless, by this dependence, NCV suffers from another source of variability which is difficult to correct within a reasonable time. This variability may be addressed by increasing the number of internal CV iterations. However, a huge number of iterations would be necessary to reach a stable estimate, i.e. an estimate that does not noticeably depend on the partitions used in internal CV. In this context it is also worth mentioning that the resampling technique and the number of iterations or folds used in internal CV are additional parameters for which no precise guidelines exist.

The natural bounds of our correction method are certainly also a crucial advantage over the competing method proposed by [14]. Their method performs reasonably well in many cases but fails in a non-negligible proportion of (realistic) setups producing overly pessimistic results. As pointed out before, there exist extreme situations in which our method can also yield corrected errors estimates higher than the raw mean. We mentioned the case with many highly correlated classifiers and an additional uncorrelated one with large but highly variable error. This is in agreement with our estimation task since we aim at estimating $\mathbf{E}_{P_n}(\varepsilon(k^*(S) \parallel S))$, the expectation of the error of strategy $k^*(S)$ when fitted on S. If a classifier is highly instable, it might often get selected as $k^*(S)$ although its average performance is bad. In this sense, selecting $k^*(S)$ may in this extreme case be worse than a random choice as far as the expected error is concerned.

This example also highlights an important feature of our new correction method. In contrast to any other correction method it directly uses the information on the correlation between the errors of different parameter values/methods, which allows an assessment of the "effective cardinality" of the pool of parameter values/methods. Obviously, the potential for tuning or method selection bias increases with the number K of parameter values/methods that are tried out. However, if they are all very similar the bias is not expected to increase dramatically. Our method automatically takes into account correlation between errors including such highly correlated "blocks" of similar parameter values/methods.

Another practical advantage over NCV is that our approach can be applied "a posteriori" as long as one has used the same training sets for all classifiers and saved all fold errors $(e(k||L_b, S \setminus L_b), \forall b, k)$. With NCV the whole procedure has to be performed again if the classifier pool or the tuning grid is changed or enlarged.

Finally, let us discuss the small optimistic bias of our method observed in the non-informative setups. Our new method is based on a number of assumptions and possibly biased estimation steps. On the one hand, if assumption (8) is violated we would expect our method to be conservative i.e. to over-correct the error, because a method chosen by internal CV based on a specific data set is expected to perform better rather than worse when applied to this data set, yielding $\mathbf{E}_{P_n}\left[\varepsilon(k \parallel S) | k^*(S) = k\right] \leq \mathbf{E}_{P_n}\left[\varepsilon(k \parallel S)\right]$. On the other hand, the estimation of μ needed for the estimation of the weights $\hat{P}(k^*(S) = k)$ may introduce a bias in the opposite direction, i.e. lead to under-estimation. That is because the estimator $(e(1 \parallel S), \ldots, e(K \parallel S))^\top$ of the vector of means μ is directly obtained from the sample at hand and thus biased. Recursive estimation of the weights or shrinkage-based procedures might be useful to address the slight bias of our new

method in the non-informative setups.

Supplementary Materials

Web Figures and Tables, referenced in Section 3 and 4 are available under `http://www.ibe.med.uni-muenchen.de/organisation/mitarbeiter/070_drittmittel/bernau/cvbias/index.html`**.**

Acknowledgements

CB and ALB are supported by the LMU-innovativ Project BioMed-S. CB is supported by grant BO3139/2-1 from the German Research Foundation (Deutsche Forschungsgemeinschaft) headed by ALB. We would like to thank Vincent Guillemot for helpful remarks on our manuscript and on the analytical computation of the weights in our correction method.

References

[1] U. Alon, N. Barkai, D.A. Notterman, K. Gish, S. Ybarra, D. Mack, and A.J. Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissue probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences of the United States of America*, 15(24):6745–6750, 1999.

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

[3] A. L. Boulesteix and C. Strobl. Optimal classifier selection and negative bias in error rate estimation: An empirical study on high-dimensional prediction. *BMC Medical Research Methodology*, 9:85, 2009.

[4] A.-L. Boulesteix, C. Strobl, T. Augustin, and M. Daumer. Evaluating microarray-based classifiers: an overview. *Cancer Informatics*, 6:77–97, 2008.

[5] A. Dupuy and R. Simon. 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, 2007.

[6] Alan Genz, Frank Bretz, Tetsuhisa Miwa, Xuefei Mi, Friedrich Leisch, Fabian Scheipl, and Torsten Hothorn. *mvtnorm: Multivariate Normal and t Distributions*, 2011. R package version 0.9-96.

[7] T.R Golub, G.K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J.P. Mesirov, H. Coller, M.L. Loh, M.A. Downing, J.R. Caligiuri, C.D. Bloomfield, and E.S. Lander. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 15(23):531–537, 1999.

[8] B. Hanczar, J. Hua, and E. R. Dougherty. Decorrelation of the true and estimated classifier errors in high-dimensional settings. *EURASIP Journal of Bioinformatics and Systems Biology*, 2007:38473, 2007.

[9] M. Jelizarow, V. Guillemot, A. Tenenhaus, K. Strimmer, and A. L. Boulesteix. Over-optimism in bioinformatics: an illustration. *Bioinformatics*, 26:1990–1998, 2010.

[10] Xiaochun Li. *ALL: A data package*, 2009. R package version 1.4.7.

[11] Claude Nadeau and Yoshua Bengio. Inference for the generalization error. *Machine Learning*, 52(3):239–281, 2003.

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

[13] Martin Slawski, Anne-Laure Boulesteix, and Christoph Bernau. *CMA: Synthesis of microarray-based classification*, 2009. R package version 1.5.5.

[14] B. Tibshirani and R. Tibshirani. A bias correction for the minimum error rate in cross-validation. *Annals of Applied Statistics*, 3:822–829, 2009.

[15] R. Tibshirani, T. Hastie, B. Narasimhan, and G. Chu. Class prediction by nearest shrunken centroids, with applications to DNA microarrays. *Statistical Science*, 18(2):104–117, 2003.

[16] S. Varma and R. Simon. Bias in error estimation when using cross-validation for model selection. *BMC Bioinformatics*, 7:91, 2006.