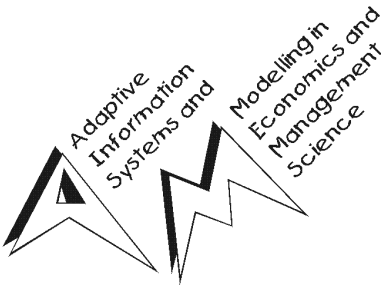


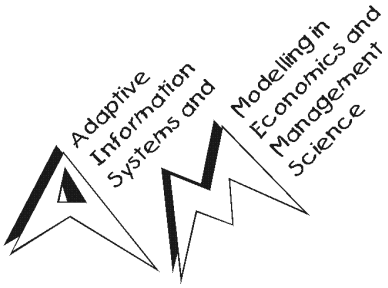
Report Series

The Design and Analysis of Benchmark Experiments

Torsten Hothorn, Friedrich Leisch,
Achim Zeileis, Kurt Hornik

Report No. 82
November 2003

Report Series



November 2003

SFB
'Adaptive Information Systems and Modelling in Economics and Management Science'

Vienna University of Economics
and Business Administration
Augasse 2–6, 1090 Wien, Austria

in cooperation with
University of Vienna
Vienna University of Technology

http://www.wu-wien.ac.at/am

Papers published in this report series
are preliminary versions of journal articles.

This piece of research was supported by the Austrian Science Foundation (FWF) under grant SFB#010 ('Adaptive Information Systems and Modelling in Economics and Management Science').

The Design and Analysis of Benchmark Experiments

TORSTEN HOTHORN
Institut für Medizininformatik, Biometrie und Epidemilogie,
Friedrich-Alexander-Universität Erlangen-Nürnberg
Waldstraße 6, D-91054 Erlangen, Germany
Torsten.Hothorn@rzmail.uni-erlangen.de

FRIEDRICH LEISCH
Institut für Statistik & Wahrscheinlichkeitstheorie, Technische Universität Wien,
Wiedner Hauptstr. 8–10, 1040 Wien, Austria
Friedrich.Leisch@ci.tuwien.ac.at

ACHIM ZEILEIS
Institut für Statistik & Wahrscheinlichkeitstheorie, Technische Universität Wien,
Wiedner Hauptstr. 8–10, 1040 Wien, Austria
Achim.Zeileis@ci.tuwien.ac.at

KURT HORNIK
Institut für Statistik, Wirtschaftsuniversität Wien,
Augasse 2–6, 1090 Wien, Austria
Kurt.Hornik@wu-wien.ac.at

Abstract

The assessment of the performance of learners by means of benchmark experiments is established exercise. In practice, benchmark studies are a tool to compare the performance of several competing algorithms for a certain learning problem. Cross-validation or resampling techniques are commonly used to derive point estimates of the performances which are compared to identify algorithms with good properties. For several benchmarking problems, test procedures taking the variability of those point estimates into account have been suggested. Most of the recently proposed inference procedures are based on special variance estimators for the cross-validated performance.

We introduce a theoretical framework for inference problems in benchmark experiments and show that standard statistical test procedures can be used to test for differences in the performances. The theory is based on well defined distributions of performance measures which can be compared with established tests. To demonstrate the usefulness in practice, the theoretical results are applied to benchmark studies in a supervised learning situation based on artificial and real-world data.

Keywords: Model Comparison; Performance; Hypothesis Testing; Cross-Validation; Bootstrap.

1 Introduction

In machine learning we refer to a benchmark study as to an empirical experiment with the aim of comparing learners or algorithms with respect to a certain performance measure. The quality of several candidate algorithms is usually assessed by point estimates of their performances on some data set or some data generating process of interest. Although nowadays commonly used in the above sense, the term "benchmarking" has its root in geology. Patterson (1992) describes the original meaning in land surveying as follows:

> A benchmark in this context is a mark, which was mounted on a rock, a building or a wall. It was a reference mark to define the position or the height in topographic surveying or to determine the time for dislocation.

In analogy to the original meaning, we measure performances in a landscape of learning algorithms while standing on a reference point, the data generating process of interest, in benchmark experiments. But in contrast to geological measurements of heights or distances the statistical measurements of performance are not sufficiently described by point estimates as they are influenced by various sources of variability. Hence, we have to take this stochastic nature of the measurements into account when making decisions about the shape of our algorithm landscape, that is, deciding which learner performs best on a given data generating process.

The assessment of the quality of an algorithm with respect to a certain performance measure, for example misclassification or mean squared error in supervised classification and regression, has been addressed in many research papers of the last three decades. The estimation of the generalisation error by means of some form of cross-validation started with the pioneering work of Stone (1974) and major improvements were published by Efron (1983, 1986) and Efron and Tibshirani (1997), for an overview we refer to Schiavo and Hand (2000). The topic is still a matter of current interest, as indicated by recent empirical (Wolpert and Macready 1999; Bylander 2002), algorithmic (Blockeel and Struyf 2002) and theoretical (Dudoit and van der Laan 2003) investigations.

However, the major goal of benchmark experiments is not only the performance assessment of different candidate algorithms but the identification of the best among them. The comparison of algorithms with respect to point estimates of performance measures, for example computed via cross-validation, is an established procedure in benchmark studies. Current examples are recent benchmark studies (as for example Meyer, Leisch, and Hornik 2003), or research papers illustrating the gains of refinements to the bagging procedure (Breiman 2001; Hothorn and Lausen 2003b). However, the problem of identifying a superior algorithm is structurally different from the performance assessment task, although we notice that asymptotic arguments indicate that cross-validation is able to select the best algorithm when provided with infinitively large learning samples (Dudoit and van der Laan 2003) because the variability tends to zero. Anyway, the comparison of raw point estimates in finite sample situations does not take their variability into account, thus leading to uncertain decisions without controlling any error probability.

While many solutions to the instability problem suggested in the last years are extremely successful in reducing the variance of algorithms by turning weak into strong learners, especially ensemble methods like boosting (Freund and Schapire 1996), bagging (Breiman 1996) or random forests (Breiman 2001), the variability of performance measures and associated test procedures has received less attention. The taxonomy of inference problems in the special case of supervised classification problems developed by Dietterich (1998) is helpful to distinguish between several problem classes and approaches. For a data generating process under study, we may either want to select the best out of a set of candidate algorithms or to choose one out of a set of predefined fitted models ("classifiers") when we are faced with large or small learning samples. Standard statistical test procedures are available for comparing the performance of fitted models when an independent test sample is available (questions 3 and 4 in Dietterich 1998) and some benchmark studies restrict themself to those applications (Bauer and Kohavi 1999). The problem whether some out of a set of candidate algorithms outperform all others (questions 7 and 8) is commonly addressed by the derivation of special variance estimators and associated tests. Estimates of the variability of the naive bootstrap estimator of misclassification error are given in Efron and Tibshirani (1997). Some procedures for solving problem 8 such as the 5×2 cv test are given by Dietterich (1998), further investigated by Alpaydin (1999)

and applied in a benchmark study on ensemble methods (Dietterich 2000). Pizarro, Guerrero, and Galindo (2002) suggest to use some classical multiple test procedures for solving this problem. Other approaches like mixed models are used in benchmark studies (for example Lim, Loh, and Shih 2000). A basic problem common to those approaches is that the correlation between internal performance estimates, such as those calculated for each fold in k-fold cross-validation, violates the assumption of independence. This fact is either ignored when the distribution of newly suggested test statistics under the null hypothesis of equal performances is investigated (for example in Dietterich 1998; Alpaydin 1999; Vehtari and Lampinen 2002) or special variance estimators taking this correlation into account are derived (Nadeau and Bengio 2003).

In this paper, we introduce a sound and flexible theoretical framework for the comparison of candidate algorithms and algorithm selection for arbitrary learning problems. The approach to the inference problem in benchmark studies presented here is fundamentally different from the procedures cited above: We show how one can sample from a well defined distribution of a certain performance measure, conditional on a data generating process, in an independent way. Consequently, standard statistical test procedures can be used to test many hypotheses of interest in benchmark studies and no special purpose procedures are necessary. The definition of appropriate sampling procedures makes special "a posteriori" adjustments to variance estimators unnecessary. Moreover, no restrictions or additional assumptions, neither to the candidate algorithms (like linearity in variable selection, see George 2000, for an overview) nor to the data generating process are required.

Throughout the paper we assume that the interest is in problems associated with a learning sample of n observations $\mathcal{L} = \{z_1, \ldots, z_n\}$ where a set of candidate algorithms as potential problem solvers is available. Each of those candidates is a two step algorithm a: In the first step a model is fitted based on a learning sample $\mathcal{L}$ yielding a function $a(\cdot \mid \mathcal{L})$ which, in a second step, can be used to compute certain objects of interest. For example, in a supervised learning problem, those objects of interest are predictions of the response based on input variables or, in density estimation, $a(\cdot \mid \mathcal{L})$ may return an estimated density.

When we search for the best solution, the candidates need to be compared by some problem specific performance measure. Such a measure depends on the algorithm and a learning sample: The function $p(a, \mathcal{L})$ assesses the goodness of the function $a(\cdot \mid \mathcal{L})$, that is the goodness of algorithm a based on learning sample $\mathcal{L}$. Since $\mathcal{L}$ is a random learning sample, $p(a, \mathcal{L})$ is a random variable whose variability is induced by the variability of learning samples following the same data generating process as $\mathcal{L}$.

It is therefore natural to compare the distribution functions of the performance measures when we need to decide whether any of the candidate algorithms performs superior to all the others. The idea is to draw independent random samples from the distribution of the performance measure for an algorithm a by evaluating $p(a, \mathcal{L})$, where the learning sample $\mathcal{L}$ follows a properly defined data generating process which reflects our knowledge about the world. By using appropriate and well investigated statistical test procedures we are able to test the hypothesis of the equality of a set of candidates with respect to the associated distribution of the performance measure and, consequently, we are in the position to control the error probability of falsely declaring any of the candidates as the winner.

We derive the theoretical basis of our proposal in Section 2 and focus on the special case of supervised learning problems in Section 3. Once that appropriate random samples from the performance distribution have been drawn, a comprehensive toolbox of well established statistical test procedures can be applied and we shortly review the most interesting of them in Section 4. Especially we focus on tests for some inference problems which are addressed in the applications presented in Section 5.

2 Comparing Performance Measures

In this section we introduce a general framework for the comparison of candidate algorithms. Independent samples from the distributions of the performance measures are drawn conditionally on the data generating process of interest. We show how standard statistical test procedures can be used to test the hypothesis of equal performances in benchmark studies.

Suppose that B independent and identically distributed learning samples have been drawn from some data generating process DGP

$$\begin{aligned} \mathcal{L}^1 &= \left\{z_1^1, \ldots, z_n^1\right\} \sim DGP, \\ &\vdots \\ \mathcal{L}^B &= \left\{z_1^B, \ldots, z_n^B\right\} \sim DGP, \end{aligned}$$

where each of the learning samples $\mathcal{L}^b$ ($b = 1, \ldots, B$) consists of n observations. Furthermore we assume that there are $K > 1$ potential candidate algorithms a_k ($k = 1, \ldots, K$) available for the solution of the underlying problem. For each algorithm a_k the function $a_k(\cdot \mid \mathcal{L}^b)$ is based on the observations from the learning sample $\mathcal{L}^b$. Hence it is a random variable depending on $\mathcal{L}^b$ and has itself a distribution A_k which depends on the data generating process of the $\mathcal{L}^b$:

$$a_k(\cdot \mid \mathcal{L}^b) \sim A_k(DGP), \quad k = 1, \ldots, K.$$

For algorithms a_k with deterministic fitting procedure (for example histograms or linear models) the function $a_k(\cdot \mid \mathcal{L}^b)$ is fixed whereas for algorithms involving non-deterministic fitting or where the fitting is based on the choice of starting values or hyper parameters (for example neural networks or random forests) it is a random variable. Note that $a_k(\cdot \mid \mathcal{L}^b)$ is a prediction function that must not depend on hyper parameters anymore: The fitting procedure incorporates both tuning as well as the final model fitting itself.

As sketched in Section 1, the goodness or performance of the candidate algorithm a_k when provided with the learning sample $\mathcal{L}^b$ is measured by a scalar function p:

$$p_{kb} = p(a_k, \mathcal{L}^b) \sim P_k = P_k(DGP).$$

The random variable p_{kb} follows a distribution function P_k which again depends on the data generating process DGP. For algorithms with non-deterministic fitting procedure this implies that it may be appropriate to integrate with respect to its distribution A_k when evaluating its performance.

The K different random samples $\{p_{k1}, \ldots, p_{kB}\}$ with B independent and identically distributed observations are drawn from the distributions $P_k(DGP)$ for algorithms a_k ($k = 1, \ldots, K$). The null hypothesis of interest for most problems is the equality of the candidate algorithms with respect to the distribution of their performance measure and can be formulated by writing

$$H_0 : P_1 = \cdots = P_K.$$

In particular, this hypothesis implies the equality of location and variability of the performances. In order to specify an appropriate test procedure for the hypothesis above one needs to define an alternative to test against. The alternative depends on the optimality criterion of interest, which we assess using a scalar functional ϕ: An algorithm a_k is better than an algorithm $a_{k'}$ with respect to a performance measure p and a functional ϕ iff $\phi(P_k) < \phi(P_{k'})$. The optimality criterion most commonly used is based on some location parameter such as the expectation $\phi(P_k) = \mathsf{E}(P_k)$ or the median of the performance distribution, that is, the average expected loss. In this case we are interested in detecting differences in mean performances between the candidate algorithms and the test problem can be defined as

$$H_0 : P_k = P_{k'} \text{ vs. } H_1 : P_k(z) = P_{k'}(z - \Delta) \text{ with } \Delta \neq 0.$$

Other alternatives may be derived from optimality criteria focusing on the variability of the performance measures. Under any circumstances, the inference is conditional on the data generating process of interest. Examples for appropriate choices of sampling procedures for the special case of supervised learning problems are given in the next section.

3 Supervised Learning

In this section we show how the general framework for testing the equality of algorithms derived in the previous section can be applied to the special but important case of supervised learning problems. Moreover, we focus on applications that commonly occur in practical situations.

3.1 Comparing Predictors

In supervised learning problems, the observations z in the learning sample are of the form $z = (y, x)$ where y denotes the response variable and x describes a vector of input variables. The aim of the learning task is to construct predictors which, based on input variables only, provide us with information about the unknown response variable. Consequently, the function constructed by each of the K candidate algorithms is of the form $\hat{y} = a_k(x \mid \mathcal{L}^b)$. In classification problems $\hat{y}$ may be the predicted class for observations with input x or the vector of the estimated conditional class probabilities. In survival analysis the conditional survival curve for observations with input status x is of special interest. The discrepancy between the true response y and the predicted value $\hat{y}$ for one single observation is measured by a scalar loss function $L(y, \hat{y})$.

The performance measure p is defined by some functional μ of the distribution of the loss function which depends on the data generating process DGP only:

$$p_{kb} = p(a_k, \mathcal{L}^b) = \mu\left(L\left(y, a_k\left(x \mid \mathcal{L}^b\right)\right)\right) \sim P_k(DGP).$$

Consequently, the randomness of $z = (y, x)$ and the randomness induced by algorithms a_k with non-deterministic fitting are removed by appropriate integration with respect to the associated distribution functions.

Again, the expectation is a common choice for the functional μ under quadratic loss $L(y, \hat{y}) = (y - \hat{y})^2$ and the performance measure is given by the so called conditional risk

$$p_{kb} = \mathsf{E}_{a_k}\mathsf{E}_{z=(y,x)}L\left(y, a_k\left(x \mid \mathcal{L}^b\right)\right) = \mathsf{E}_{a_k}\mathsf{E}_{z=(y,x)}\left(y - a_k\left(x|\mathcal{L}^b\right)\right)^2, \tag{1}$$

where $z = (y, x)$ is drawn from the same distribution as the observations in a learning sample $\mathcal{L}$. Other conceivable choices of μ are the median, corresponding to absolute loss, or even the supremum or theoretical quantiles of the loss functions.

3.2 Special Problems

The distributions of the performance measure $P_k(DGP)$ for algorithms a_k ($k = 1, \ldots, K$) depend on the data generating process DGP. Consequently, the way we draw random samples from $P_k(DGP)$ is determined by the knowledge about the data generating process available to us. In supervised learning problems, one can distinguish two situations:

- Either the data generating process is known, which is the case in simulation experiments with artificially generated data or in cases where we are practically able to draw infinitively many samples (e.g., network data),
- or the information about the data generating process is determined by a finite learning sample $\mathcal{L}$. In this case the empirical distribution function of $\mathcal{L}$ typically represents the complete knowledge about the data generating process we are provided with.

In the following we show how random samples from the distribution of the performance measure $P_k(DGP)$ for algorithm a_k can be drawn in three basic problems: The data generating process is known (Simulation), a learning sample as well as a test sample are available (Competition) or one single learning sample is provided only (Real World).

Problem I: Simulation

Artificial data are generated from some distribution function Z, where each observation z_i $(i = 1, \ldots, n)$ in a learning sample is distributed according to Z. The learning sample $\mathcal{L}$ consists of n independent observations from Z which we denote by $\mathcal{L} \sim Z_n$. In this situation the data generating process $DGP = Z_n$ is used. Therefore we are able to draw a set of B independent learning samples from Z_n:

$$\mathcal{L}^1, \ldots, \mathcal{L}^B \sim Z_n.$$

We assess the performance of each algorithm a_k on all learning samples $\mathcal{L}^b (b = 1, \ldots, B)$ yielding a random sample of B observations from the performance distribution $P_k(Z_n)$ by calculating

$$p_{kb} = p(a_k, \mathcal{L}^b) = \mu \left(L \left(y, a_k \left(x \mid \mathcal{L}^b \right) \right) \right), \quad b = 1, \ldots, B.$$

The associated hypothesis under test is consequently

$$H_0 : P_1(Z_n) = \cdots = P_K(Z_n).$$

If we are not able to calculate μ analytically we can approximate it up to any desired accuracy by drawing a test sample $\mathcal{T} \sim Z_m$ of m independent observations from Z where m is large and calculating

$$\hat{p}_{kb} = \hat{p}(a_k, \mathcal{L}^b) = \mu_{\mathcal{T}} \left(L \left(y, a_k \left(x \mid \mathcal{L}^b \right) \right) \right).$$

Here $\mu_{\mathcal{T}}$ denotes the empirical analogon of μ for the test observations $z = (y, x) \in \mathcal{T}$. When μ is defined as the expectation with respect to test samples z as in (1) (we assume a deterministic a_k for the sake of simplicity here), this reduces to the mean of the loss function evaluated for each observation in the learning sample

$$\hat{p}_{kb} = \hat{p}(a_k, \mathcal{L}^b) = m^{-1} \sum_{z=(y,x) \in \mathcal{T}} L \left(y, a_k \left(x \mid \mathcal{L}^b \right) \right).$$

Analogously, the supremum would be replaced by the maximum and theoretical quantiles by their empirical counterpart.

Problem II: Competition

In real world applications no precise knowledge about the data generating process is available but instead we are provided with one learning sample $\mathcal{L} \sim Z_n$ of n observations from some distribution function Z. The empirical distribution function $\hat{Z}_n$ covers all knowledge that we have about the data generating process. Therefore, we mimic the data generating process by using the empirical distribution function of the learning sample: $DGP = \hat{Z}_n$. Now we are able to draw independent and identically distributed random samples from this, emulated, data generating process. In a completely non-parametric setting, the non-parametric bootstrap can be applied here or, if the restriction to certain parametric families is appropriate, the parametric bootstrap can be used to draw samples from the data generating process. For an overview of those issues we refer to Efron and Tibshirani (1993).

Under some circumstances, an additional test sample $\mathcal{T} \sim Z_m$ of m observations is given, for example in machine learning competitions. In this situation, the performance needs to be assessed with respect to $\mathcal{T}$ only. Again, we would like to draw a random sample of B observations from $\hat{P}_k(\hat{Z}_n)$, which in this setup is possible by bootstrapping:

$$\mathcal{L}^1, \ldots, \mathcal{L}^B \sim \hat{Z}_n,$$

where $\hat{P}$ denotes the distribution function of the performance measure evaluated using $\mathcal{T}$, that is, the performance measure is computed by

$$\hat{p}_{kb} = \hat{p}(a_k, \mathcal{L}^b) = \mu_{\mathcal{T}} \left(L \left(y, a_k \left(x \mid \mathcal{L}^b \right) \right) \right)$$

where $\mu_{\mathcal{T}}$ is again the empirical analogon of μ for all $z = (y, x) \in \mathcal{T}$. The hypothesis we are interested in is

$$H_0 : \hat{P}_1(\hat{Z}_n) = \cdots = \hat{P}_K(\hat{Z}_n),$$

where $\hat{P}_k$ corresponds to the performance measure $\mu_{\mathcal{T}}$. Since the performance measure is defined in terms of one single test sample $\mathcal{T}$, it should be noted that we may favour algorithms that perform well on that particular test sample $\mathcal{T}$ but worse on other test samples just by chance.

Problem III: Real World

The most common situation we are confronted with in daily routine is the existence of one single learning sample $\mathcal{L} \sim Z_n$ with no dedicated independent test sample being available. Again, we mimic the data generating process by the empirical distribution function of the learning sample: $DGP = \hat{Z}_n$. We redraw B independent learning samples from the empirical distribution function by bootstrapping

$$\mathcal{L}^1, \ldots, \mathcal{L}^B \sim \hat{Z}_n.$$

The corresponding performance measure is computed by

$$\hat{p}_{kb} = \hat{p}(a_k, \mathcal{L}^b) = \hat{\mu}\left(L\left(y, a_k\left(x \mid \mathcal{L}^b\right)\right)\right)$$

where $\hat{\mu}$ is an appropriate empirical version of μ. There are many possibilities of choosing $\hat{\mu}$ and the most obvious ones are given in the following.

If n is large, one can divide the learning sample into a smaller learning sample and a test sample $\mathcal{L} = \{\mathcal{L}', \mathcal{T}\}$ and proceed with $\mu_{\mathcal{T}}$ as in Problem II. If n is not large enough for this to be feasible, the following approach seems to be a first natural choice: In Problem I the models are fitted on samples from Z_n and their performance is evaluated on samples from Z. Here, the models are trained on samples from the empirical distribution function $\hat{Z}_n$ and so we could want to assess their performance on $\hat{Z}$ which corresponds to emulating $\mu_{\mathcal{T}}$ by using the learning sample $\mathcal{L}$ as test sample:

- Problem III-Learn. For each model fitted on a bootstrap sample, the original learning sample $\mathcal{L}$ itself is used as test sample $\mathcal{T}$.

Of course, this choice often leads to overfitting problems. Those can be addressed by well known cross-validation strategies. The test sample $\mathcal{T}$ can be defined in terms of the out–of–bootstrap observations when evaluating $\mu_{\mathcal{T}}$:

- Problem III-OOB. For each bootstrap sample $\mathcal{L}^b (b = 1, \ldots, B)$ the out–of–bootstrap observations $\mathcal{L} \setminus \mathcal{L}^b$ are used as test sample.

Note that using the out–of–bootstrap observations as test sample leads to non-independent observations of the performance measure, however, their correlation vanishes as n tends to infinity. Another way is to choose a cross-validation estimator of μ:

- Problem III-CV. Each bootstrap sample $\mathcal{L}^b$ is divided into k folds and the performance $\hat{p}_{kb}$ is defined as the average of the performance measure on each of those folds. Since it is possible that one observation from the original learning sample $\mathcal{L}$ is part of both the learning folds and the validation fold due to sampling n–out–of–n with replacement, those observations are removed from the validation fold in order to prevent any bias. Such bias may be induced for some algorithms that perform better on observations that are part of both learning and test sample.

Common to all choices in this setup is that one single learning sample provides all information. Therefore, we cannot compute the theoretical performance measures and hence cannot test hypotheses about these as this would require more knowledge about the data generating process. The standard approach is to compute some empirical performance measure, such as those suggested here, instead to approximate the theoretical performance. For any empirical performance measure, the hypothesis needs to be formulated by

$$H_0 : \hat{P}_1(\hat{Z}_n) = \cdots = \hat{P}_K(\hat{Z}_n),$$

meaning that the inference is conditional on the performance measure under consideration.

4 Test Procedures

As outlined in the previous sections, the problem of comparing K algorithms with respect to any performance measure reduces to the problem of comparing K numeric distribution functions or certain characteristics as their expectation, for example. A lot of attention has been payed to this and similar problems in the statistical literature and so a rich toolbox can be applied here. We will discuss appropriate test procedures for only the most important test problems commonly addressed in benchmark experiments. A complete overview is far beyond the scope of this paper and actually not even possible due to the overwhelming amount of literature in this field and we therefore refer to prominent text books such as Lehmann (1994) and Hájek, Šidák, and Sen (1999) or introductory texts (for example Rossman and Chance 2001) for further reading.

4.1 Experimental Designs

Basically, the choice of two experimental designs is possible. Either a parallel group (or independent K sample) design, where the redrawn learning samples are independent between algorithms or a matched pairs (or dependent K sample) design can be used. For the latter, the performance of all K algorithms is evaluated using the same random samples $\mathcal{L}^1, \ldots, \mathcal{L}^B$. The independent K sample design is more comfortable from a statistical point of view but the algorithms are never provided with the same learning samples. This is not intuitional when we would like to compare algorithms in benchmark studies, so we choose a matched pairs design in our experiments in Section 5 and compare the algorithms based on the same set of learning samples.

4.2 Comparing Locations

For the independent two samples case, well known standard test procedures like the two-sample t-test or non-parametric alternatives like the Wilcoxon-Mann-Whitney test are appropriate for the comparison of two location parameters. When we need to compare $K > 2$ algorithms, the classical F-test in the analysis of variance (ANOVA) framework can be used to test whether there is any difference between the performances of the K candidate algorithms. The Kruskal-Wallis test is a non-parametric alternative for comparing $K > 2$ independent performance distributions.

For the dependent K samples design, the paired t-test or the Wilcoxon-signed-rank test for comparing two algorithms or the Friedman test when $K > 2$ algorithms are under study can be applied. A sensible test statistic for comparing two performance distributions with respect to their locations in a matched pairs design is formulated in terms of the average $\bar{d}$ of the differences $d_b = p_{1b} - p_{2b} (b = 1, \ldots, B)$ for the observations p_{1b} and p_{2b} of algorithms a_1 and a_2. Under the null hypothesis of equality of the performance distributions, the studentized statistic

$$t = \sqrt{B} \frac{\bar{d}}{(B-1)^{-1} \sum_b \left(d_b - \bar{d}\right)^2} \tag{2}$$

follows a t-distribution with $B - 1$ degrees of freedom. However, the unconditional t-distribution of the test statistic t and similar statistics is derived under some parametric assumption, such as symmetry or normality, on the distributions under test. The question whether conditional or unconditional test procedures should be applied has some philosophical aspects and is one of the controversial questions in recent discussions (see Berger 2000; Berger, Lunneborg, Ernst, and Levine 2002, for example). However, we doubt if parametric assumptions to the performance distribution such as normality are ever appropriate and the inference is conditional on an observed learning sample at least in problems II and III anyway, thus conditional test procedures, where the null distribution is determined from the data actually seen, are natural to use for the test problems addressed here. Since we are able to draw as many random samples from the performance distributions under test as required, the application of the asymptotic distribution of the test statistics of the corresponding permutation tests is possible in cases where the determination of the exact conditional distribution is difficult.

Maybe the most prominent problem is to test whether K algorithms perform equally well against the alternative that at least one of them outperforms all other candidates. In a dependent K samples design, the test statistic of the associated permutation test is given by

$$T^\star = \frac{\sum_k \left(B^{-1} \sum_b \hat{p}_{kb} - (BK)^{-1} \sum_{k,b} \hat{p}_{kb} \right)^2}{\sum_{k,b} \left(\hat{p}_{kb} - K^{-1} \sum_k \hat{p}_{kb} - B^{-1} \sum_b \hat{p}_{kb} - (BK)^{-1} \sum_{k,b} \hat{p}_{kb} \right)^2} \tag{3}$$

whose distribution can be obtained by permuting the labels $1, \ldots, K$ of the algorithms for each sample $\mathcal{L}^b (b = 1, \ldots, B)$ independently, see Pesarin (2001) for the details.

4.3 Comparing Scales

The equality of $K > 2$ distributions against scale alternatives may be tested using the Fligner-Killeen test (Conover, Johnson, and Johnson 1981). When we need to compare $K = 2$ algorithms only, the non-parametric Ansari-Bradley test is a popular choice.

4.4 Multiple Comparison Procedures

Once the global hypothesis of equality of $K > 2$ performances could be rejected in a dependent K samples design, it is of special interest to identify the algorithms that caused the rejection. All partial hypotheses can be tested at level α following the closed testing principle, for a description see Hochberg and Tamhane (1987). In a parallel group design, simultaneous confidence intervals for all-pair comparisons (Tukey's honestly significant differences) or, if the interest is in showing that one predefined algorithm is superior to all others, many-to-one comparisons (Dunnett 1955) can be applied.

4.5 Tests for Equivalence

A problem still ignored in many studies is the fact that the lack of a significance result when testing the null hypothesis does not mean that the alternative is true. This is especially important when the aim is to show that two algorithms a_1 and a_2 perform equally well. For example one might want to show that two software implementations of algorithms with non-deterministic fitting procedure are equal or that a simple algorithm with intuitive meaning to practitioners performs as good as a more sophisticated one. In those situations the test problem may read

$$H_0 : \frac{\mathsf{E}(P_1)}{\mathsf{E}(P_2)} \notin [\rho_1, \rho_2] \text{ vs. } H_1 : \frac{\mathsf{E}(P_1)}{\mathsf{E}(P_2)} \in [\rho_1, \rho_2]$$

with equivalence range $0 < \rho_1 < 1 < \rho_2 < \infty$. An overview of some test procedures is, for example, given in Pflüger and Hothorn (2002).

5 Illustrations and Applications

Although the theoretical framework presented in Sections 2 and 3 covers a wider range of applications, we restrict ourselves to a few examples from supervised classification and regression in order to illustrate the basic concepts. As outlined in Section 3.2, the degree of knowledge about the data generating process available to the investigator determines how well we can approximate the theoretical performance by using empirical performance measures. For a simple artificial data generating process from a univariate regression relationship we will study the power of tests based on the empirical performance measures for Problems I, II and III.

Maybe the most interesting question addressed in benchmarking experiments "Is there any difference between state–of–the–art algorithms with respect to a certain performance measure?" in the situation of Problem III is investigated for some established and recently suggested supervised learning algorithms by means of four real world learning samples from the UCI repository (Blake and Merz 1998).

5.1 Nested Linear Models

The data generating process follows a univariate regression equation

$$y = 2x + \beta x^2 + \varepsilon \tag{4}$$

where the input x is drawn from a uniform distribution on the interval $[0, 5]$ and the error terms are independent realisations from a standard normal distribution. We fix the number of observations in a learning sample to $n = 50$. Two predictive models are compared:

- a_1: a simple linear regression taking x as input and therefore not including a quadratic term and
- a_2: a simple quadratic regression taking both x and x^2 as inputs. Consequently, the regression coefficient β is estimated.

The discrepancy between a predicted value $\hat{y} = a_k(x|\mathcal{L}), k = 1, 2$ and the response y is measured by squared error loss $L(y, \hat{y}) = (y - \hat{y})^2$.

Basically, we are interested to check if algorithm a_1 performs better than algorithm a_2 for values of β varying in a range between 0 and 0.25. As described in detail in Section 3.2, both the performance measure and the sampling from the performance distribution depend on the degree of knowledge available and we therefore distinguish between three different problems.

Problem I: Simulation

The data generating process Z_n is known by equation (4) and we are able to draw as many learning samples of $n = 50$ observations as we would like. It is, in principle, possible to calculate the mean squared error of the predictive functions $a_1(\cdot \mid \mathcal{L}^b)$ and $a_2(\cdot \mid \mathcal{L}^b)$ when learning sample $\mathcal{L}^b$ was observed. Consequently, we are able to formulate the test problem in terms of the performance distribution depending on the data generating process in a one-sided way:

$$H_0 : \mathsf{E}(P_1(Z_n)) \leq \mathsf{E}(P_2(Z_n)) \text{ vs. } H_1 : \mathsf{E}(P_1(Z_n)) > \mathsf{E}(P_2(Z_n)). \tag{5}$$

However, analytical calculations are only of academic interest in benchmark experiments and we therefore approximate $P_k(Z_n)$ by $\hat{P}_k(Z_n)$ using a large test sample $\mathcal{T}$, in our case with $m = 2000$ observations. In order to study the goodness of the approximation we, in addition, choose a smaller test sample with $m = 50$. Note that in this setup the inference is conditional under the test sample.

Problem II: Competition

We are faced with a learning sample $\mathcal{L}$ with $n = 50$ observations. The performance of any algorithm is to be measured by an additional test sample $\mathcal{T}$ consisting of $m = 50$ observations. Again, the inference is

β	I $m = 2000$	I $m = 50$	II	III-Learn	III-OOB	III-OOB $n = 100$	III-CV
0.000	0.000	0.000	0.082	0.292	0.059	0.041	0.054
0.025	0.000	0.128	0.133	0.342	0.092	0.087	0.080
0.050	0.233	0.457	0.293	0.485	0.190	0.218	0.164
0.075	0.980	0.617	0.487	0.658	0.333	0.423	0.317
0.100	1.000	0.747	0.658	0.816	0.521	0.637	0.490
0.125	1.000	0.844	0.778	0.913	0.694	0.823	0.670
0.150	1.000	0.899	0.859	0.965	0.827	0.908	0.818
0.175	1.000	0.938	0.919	0.989	0.920	0.962	0.911
0.200	1.000	0.965	0.939	0.996	0.969	0.985	0.964
0.225	1.000	0.982	0.969	0.999	0.988	0.994	0.984
0.250	1.000	0.988	0.980	1.000	0.996	0.998	0.996

Table 1: Simulation experiments: Power of the tests in situations I, II and III for varying values of the regression coefficient β of the quadratic term.

conditional under the observed test sample and we may, just by chance, observe a test sample favouring a quadratic model. The data generating process is emulated by the empirical distribution function $DGP = \hat{Z}_n$ and we resample by using the non-parametric bootstrap.

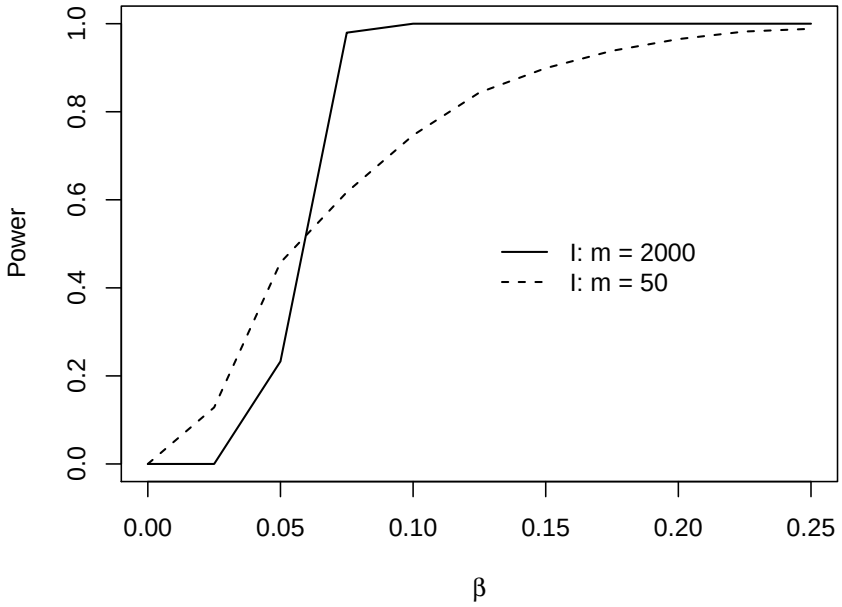
Problem III: Real World

Most interesting and most common is the situation where the knowledge about the data generating process is completely described by one single learning sample and the non-parametric bootstrap is used to redraw learning samples. Several performance measures are possible and we investigate those suggested in Section 3.2: Problem III-Learn, Problem III-OOB and Problem III-CV. For Problem III-CV, the performance measure is obtained from a 5-fold cross-validation estimator. Each bootstrap sample is divided in five folds and the mean squared error on each of these folds is averaged. Observations which are elements of training and validation fold are removed from the latter. In addition, we compare the out–of–bootstrap empirical performance measure in the real world problem with the empirical performance measure in the competition problem:

- Problem III-OOB, $n = 100$. A hypothetical learning sample and test sample of size $m = n = 50$ each are merged into one single learning sample with 100 observations and we proceed as in Problem III-OOB.

For our investigations here, we draw $B = 250$ learning samples either from the true data generating process Z_n (Problem I) or from the empirical distribution function $\hat{Z}_n$ by the non-parametric bootstrap (Problems II and III). The performance of both algorithms is evaluated on the same learning samples in a matched pairs design and the null hypothesis of equal performance distributions is tested by the corresponding one-sided permutation test where the asymptotic distribution of its test statistic (2) is used. The power curves are estimated by means of 5000 Monte-Carlo replications.

The numerical results of the power investigations are given in Table 1 and are depicted in Figures 1 and 2. Recall that our main interest is to test whether the simple linear model a_1 outperforms the quadratic model a_2 with respect to its theoretical mean squared error. For $\beta = 0$, the bias of the predictions $a_1(\cdot|\mathcal{L})$ and $a_2(\cdot|\mathcal{L})$ is zero but the variance of the predictions of the quadratic model are larger compared to the variance of the predictions of the simple model a_1. Therefore, the theoretical mean squared error of a_1 is smaller than the mean squared error of a_2 for $\beta = 0$ which reflects the situation under the null hypothesis in test problem (5). As β increases only a_2 remains unbiased. But as a_1 has still smaller variance there is a trade-off between bias and variance before a_2 eventually outperforms a_1 which corresponds to the alternative in test problem (5). This is also reflected in the second column of Table 1 (Problem I, $m = 2000$). The test problem is formulated in terms of the theoretical performance measures $P_k(Z_n), k = 1, 2$, but we are



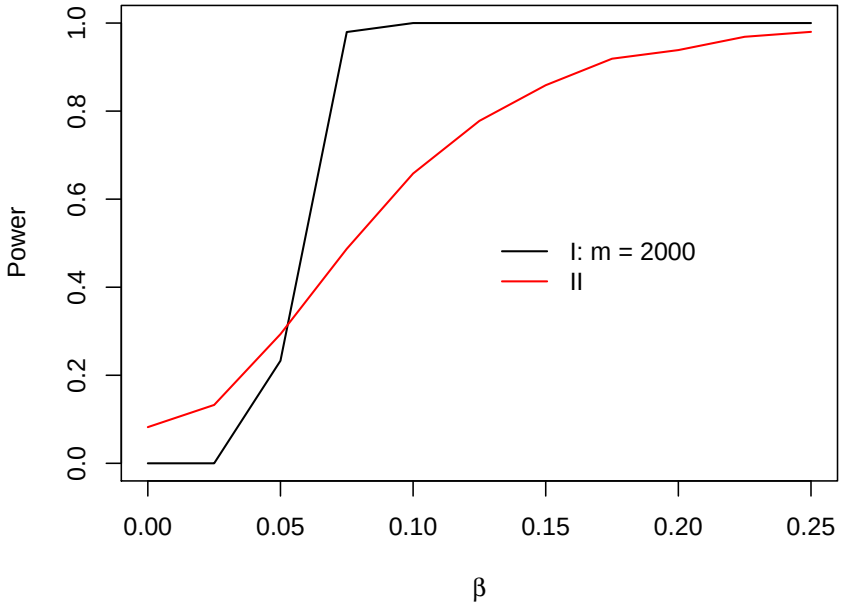
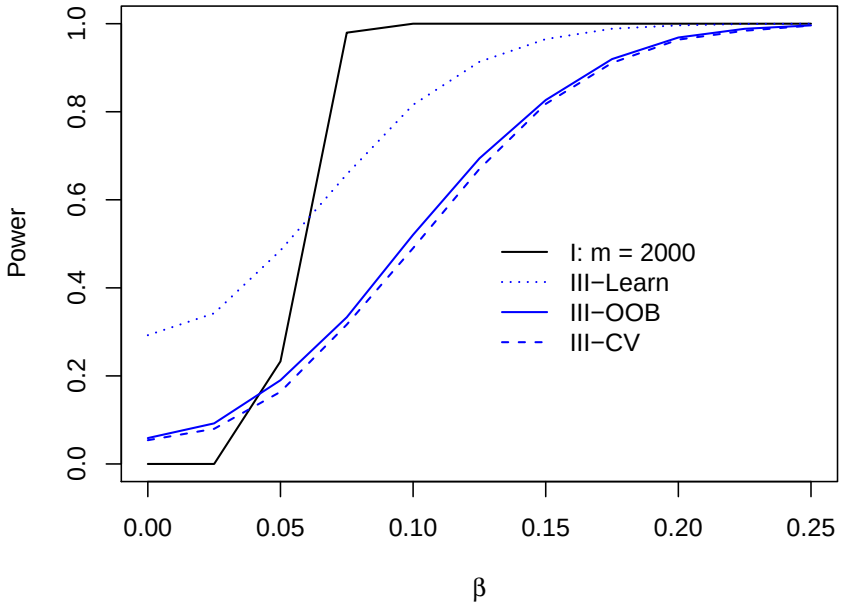


Figure 1: Simulation experiments: Power of the test in Problem I with large and small test sample (top) and for Problem II (bottom).



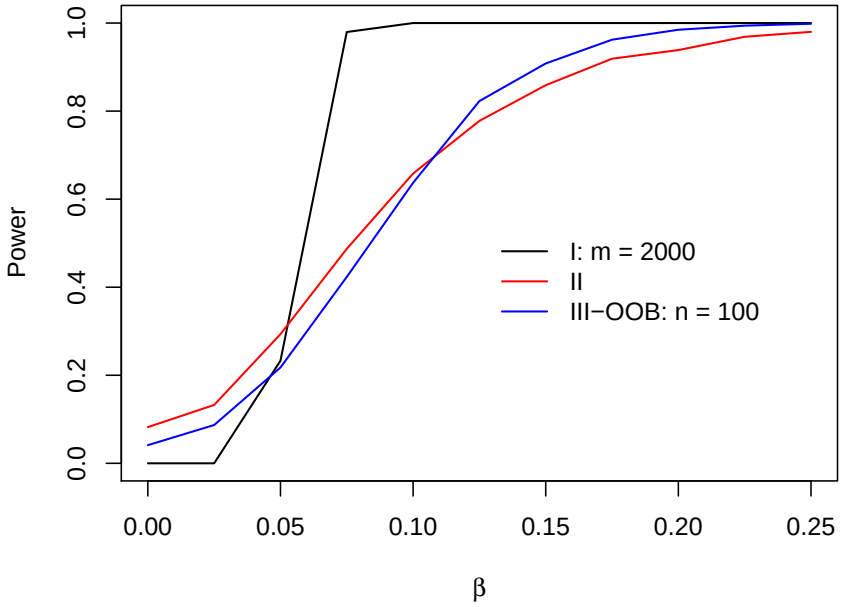


Figure 2: Simulation experiments: Power of different approaches to Problem III (top) and a comparison of Problem II and III (bottom).

never able to draw samples from these distributions in realistic setups. Instead, we approximate them in Problem I by $\hat{P}_k(Z_n)$ either very closely with $m = 2000$ or less accurately with $m = 50$ which we use for comparisons with Problem II and III where the empirical performance distributions $\hat{P}_k(\hat{Z}_n)$ are used.

Problem I ($m = 2000$) in the second column of Table 1 offers the closest approximation of the comparison of the theoretical performance measures: For $\beta = 0$ we are always able to detect that a_1 outperforms a_2. As β increases the performance of a_2 improves compared to a_1 and eventually outperforms a_1 which we are able to detect always for $\beta \geq 0.1$. As Problem I ($m = 2000$) gives the sharpest distinction between the two models this power curve is used as reference mark in all plots.

For the remaining problems the case of $\beta = 0$ is analysed first where it is known that the theoretical predictive performance of a_1 is better than that of a_2. One would expect that although not this theoretical performance measure but only its empirical counterpart is used only very few rejections occur reflecting the superiority of a_1. In particular, one would hope that the rejection probability does not exceed the nominal size $\alpha = 0.05$ of the test too clearly. This is true for the Problems I, III-OOB and III-CV but not for II and III-Learn due to the usage of a fixed test sample and overfitting respectively. It should be noted that this cannot be caused by size distortions of the test because under any circumstances the empirical size of the permutation test is, up to derivations induced by using the asymptotic distribution of the test statistic or by the discreteness of the test statistic (2), always equal to its nominal size α. The discrepancy between the nominal size of tests for (5) and the empirical rejection probability in the first row of Table 1 is caused for Problem II by the choice of a fixed test sample which may favour a quadratic model even for $\beta = 0$ and so the power is 0.08. Overfitting favours the quadratic model a_2 in Problem III when using the original learning sample as test sample as well. For the performance measures defined in terms of out–of–bootstrap observations or cross-validation estimates, the estimated power for $\beta = 0$ ranges between 0.041 and 0.059. This indicates a good correspondence between the test problem (5) formulated in terms of the theoretical performance and the test which compares the empirical performance distributions.

For $\beta > 0$, the power curves of all other problems are flatter than that for Problem I ($m = 2000$) reflecting that there are more rejections when the theoretical performance of a_1 is still better and fewer rejections when the theoretical performance of a_2 is better. Thus, the distinction is not as sharp as in the (almost) ideal situation in Problem I with $m = 2000$. However, the procedures based on out–of–bootstrap and cross-validation—which are virtually indistinguishable—are fairly close to the power curve for Problem I with $m = 50$ observations in the test sample: Hence, the test procedures based on those empirical performance measures have very high power compared with the situation where the complete knowledge about the data generating process is available (Problem I, $m = 50$).

It should be noted that, instead of relying on Problem II when a separate test sample is available, the conversion into Problem III seems appropriate: The power curve is higher for large values of β and the value 0.041 covers the nominal size $\alpha = 0.05$ of the test problem (5) better for $\beta = 0$. The definition of a separate test sample when only one single learning sample is available seems inappropriate in the light of this result.

5.2 Benchmarking Applications

The basic concepts are illustrated in the preceding paragraph by means of a univariate regression model and we now focus on the application of test procedures implied by the theoretical framework to four real world benchmarking applications from the UCI repository (Blake and Merz 1998). Naturally, we are provided with one learning sample consisting of a moderate number of observations for each of the applications:

- For the Boston Housing regression problem 13 inputs for $n = 506$ observations are available.
- The Breast Cancer dataset is a two-class classification problem with $n = 699$ observations and 9 input variables.
- The Ionosphere dataset consists of $n = 351$ observations with 34 inputs and a dichotomous response variable.
- The famous Pima Indian Diabetes problem is a two-class problem as well with $n = 768$ observations and 8 input variables.

Consequently, we are able to test hypotheses formulated in terms of the performance distributions implied by the procedures suggested for Problem III in Section 3.2. Both the 5-fold cross-validation estimator as well as the out–of–bootstrap observations are used to define performance measures based on misclassification or squared error loss. Again, observations that occur both in learning and validations folds in cross-validation are removed from the latter.

The algorithms under study are well established procedures or recently suggested solutions for supervised learning applications. The comparison is based on their corresponding implementations in the R system for statistical computing (Ihaka and Gentleman 1996; R Development Core Team 2003). An interface to support vector machines (SVM, Vapnik 1998) via the LIBSVM library (Chang and Lin 2001) is available in package `e1071` (Meyer 2001). Hyper parameters are tuned on each bootstrap sample by cross-validation, for the technical details we refer to Meyer et al. (2003). A stabilised linear discriminant analysis (sLDA, Läuter 1992) as implemented in the `ipred` package (Peters, Hothorn, and Lausen 2002) as well as the binary logistic regression model (GLM) are under study. Bagging (Breiman 1996), random forests (Breiman 2001) and bundling (Hothorn 2003; Hothorn and Lausen 2003a) as a combination of bagging, sLDA, nearest neighbours and GLM are included in this study as representatives of tree-based ensemble methods. Bagging and bundling are implemented in the `ipred` package while random forests are available in the `randomForest` package (Liaw and Wiener 2002). The ensemble methods average over 250 trees.

We draw independent samples from the performance distribution of the candidate algorithms based on $B = 250$ bootstrap samples in a dependent K samples design. The distribution of the test statistic $T^\star$ from (3) is determined via conditional Monte-Carlo (Pesarin 2001) when at least three algorithms are under test. Once the global null hypothesis has been rejected at nominal size $\alpha = 0.05$, we are interested in all pairwise comparisons in order to find the differences that lead to the rejection. This is possible by an appropriate closed testing procedure as indicated in Section 4.

Boston Housing Bagging and random forests in their regression version are compared via their mean squared error, measured by cross-validation and out–of–bootstrap performance, for the Boston Housing data. The means of the cross-validated performance distributions are 13.64 (bagging) and 13.00 (random forests). For the out–of–bootstrap performance measure, the means are 12.88 (bagging) and 11.91 (random forests). The two random samples of size $B = 250$ each are graphically summarised by boxplots and a kernel density estimator in Figure 3: The performance distributions are heavily skewed. The hypothesis of equal performance distributions is tested using the test statistic (2). For the performance measure based on cross-validation, the value of the test statistic is $t = -4.554$ and the conditional P-value is less than 0.001, thus the null hypothesis can be rejected at level $\alpha = 0.05$. For the performance measure defined in terms of the out–of–bootstrap observations, the value of the test statistic is $t = -5.684$, which corresponds to the elevated difference in the means compared to cross-validation. Again, the P-value is less than 0.001 and a significant decrease with respect to prediction error by injection of additional randomness via random forests can be postulated.

Breast Cancer The performance of sLDA, SVM, random forests and bundling for the Breast Cancer classification problem is investigated. Figure 4 depicts the empirical performance distributions. Especially for the cross-validated performance measure many outliers are visible for SVM, the most extreme of them not displayed in the plots. An inspection of the graphical representation leads to the presumption that for both performance measures the random samples for random forests have the smallest variability and expectation. The global hypothesis of equality of all four algorithms with respect to their cross-validated performance can be rejected (P-value ≤ 0.001) and the multiple test procedure suggests that this is due to the superiority of random forests compared to all other candidates while no significant differences between SVM and sLDA (P-value $= 0.106$) can be found. For the performance measure based on the out–of–bootstrap observations the global hypothesis can be rejected as well. For this performance measure, this is caused by significant differences between the ensemble methods on the one hand and SVM and sLDA on the other hand while the elementary hypotheses of the equality of the out–of-bootstrap performances of random forests and bundling (P-value $= 0.071$) and between SVM and sLDA (P-value $= 0.776$) can not be rejected at level $\alpha = 0.05$.

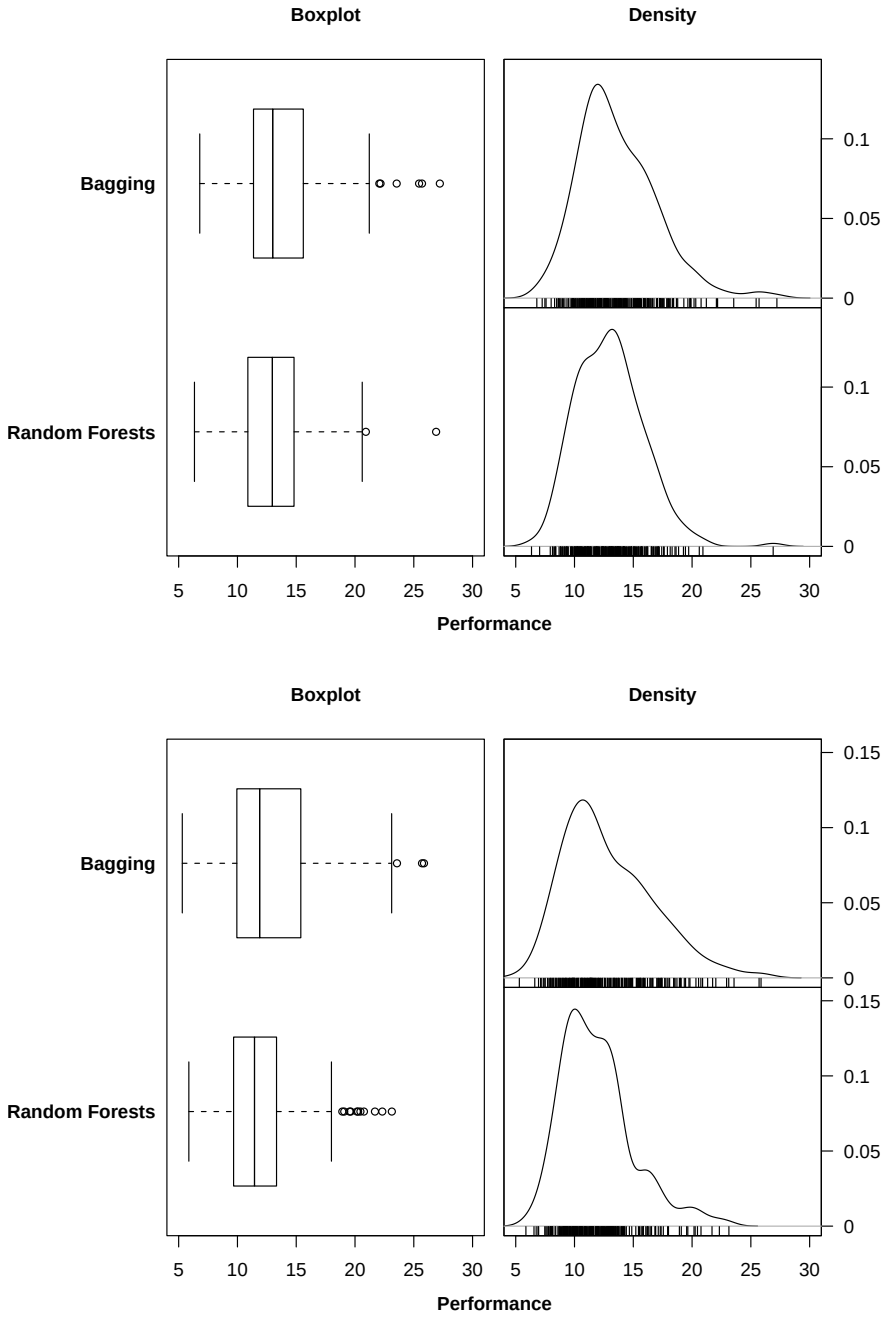


Figure 3: The distribution of the cross-validation (top) and out–of–bootstrap (bottom) performance measure for the Boston Housing data visualised via boxplots and a density estimator.

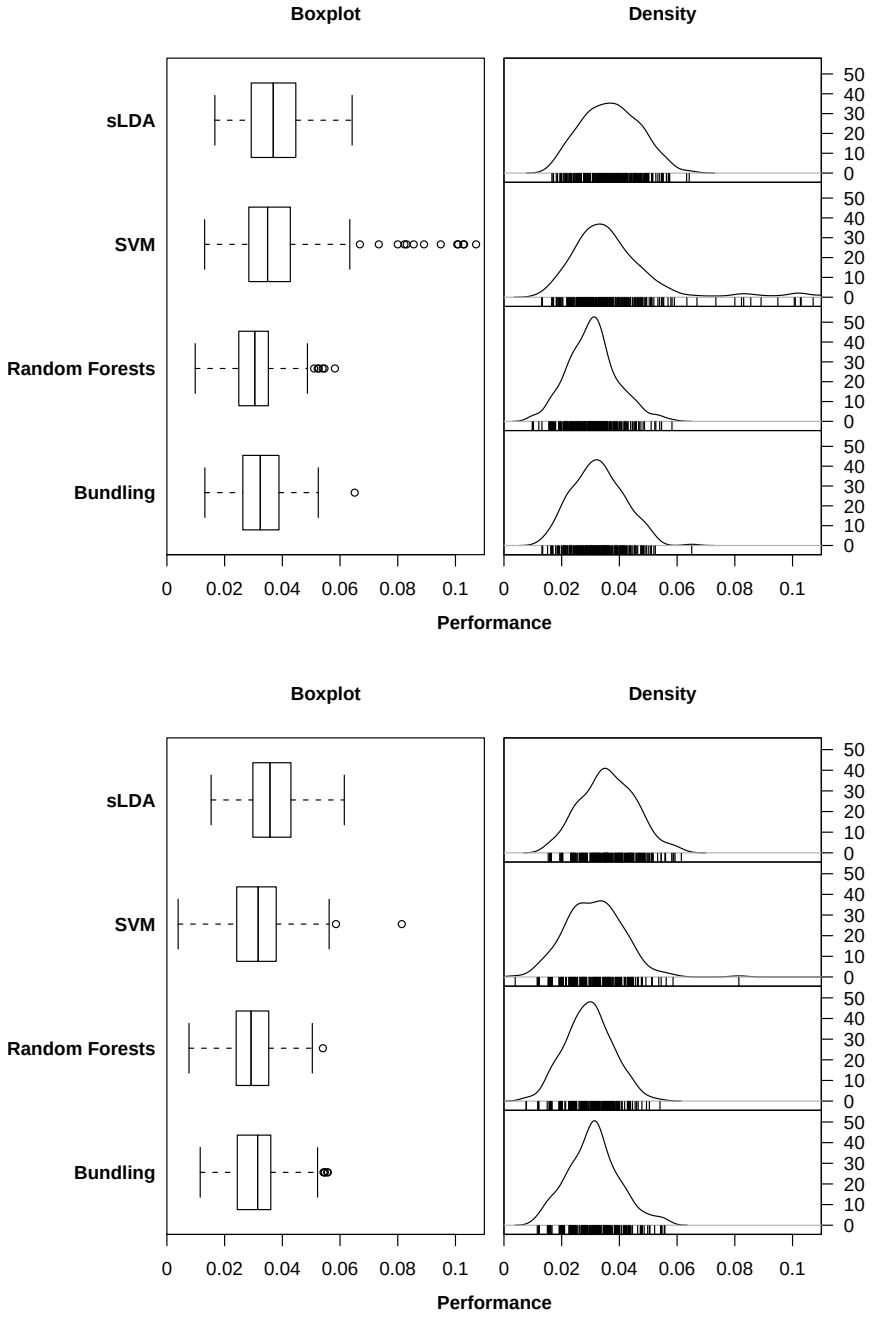


Figure 4: The distribution of the cross-validation (top) and out–of–bootstrap (bottom) performance measure for the Breast Cancer data.

Ionosphere For the Ionosphere data the supervised learners SVM, random forests and bundling are compared. The graphical representation of the estimated densities in Figure 5 indicate some degree of skewness for all methods. Note that this is not visible in the boxplot representations. The global hypothesis can be rejected at level $\alpha = 0.05$ (P-value ≤ 0.001) and the closed testing procedure indicates that this is due to a significant difference between the distributions of the performance measures for SVM and the tree based ensemble methods while no significant difference between bundling and random forests (P-value $= 0.063$) can be found. In this sense, the ensemble methods perform indistinguishably and both are outperformed by SVM. For the out–of–bootstrap performance measure, significant differences between all three algorithms can be stated: Bundling performs slightly better than random forests for the Ionosphere data (P-value $= 0.008$).

Pima Indian Diabetes A logistic regression model (GLM) is compared with random forests and SVM on the Pima Indian Diabetes dataset. The distributions of the performance measures are skewed to some extent as can be seen in Figure 6. The global hypothesis as well as all elementary hypotheses can be rejected (P-value ≤ 0.001) showing a significant difference between all three distributions with respect to the cross-validation performance measure as well as the out–of–bootstrap error and the superiority of GLM can be stated.

The kernel density estimates for all four benchmarking problems indicate that the performance distributions are skewed in most situations, especially for support vector machines, and the variability differs between algorithms. Therefore, assumptions like normality or homoskedasticity are hardly appropriate and test procedures relying on those assumptions should not be used. The conclusions drawn when using the out–of–bootstrap performance measure agree with those obtained when using a performance measure defined in terms of cross-validation both quantitatively and qualitatively.

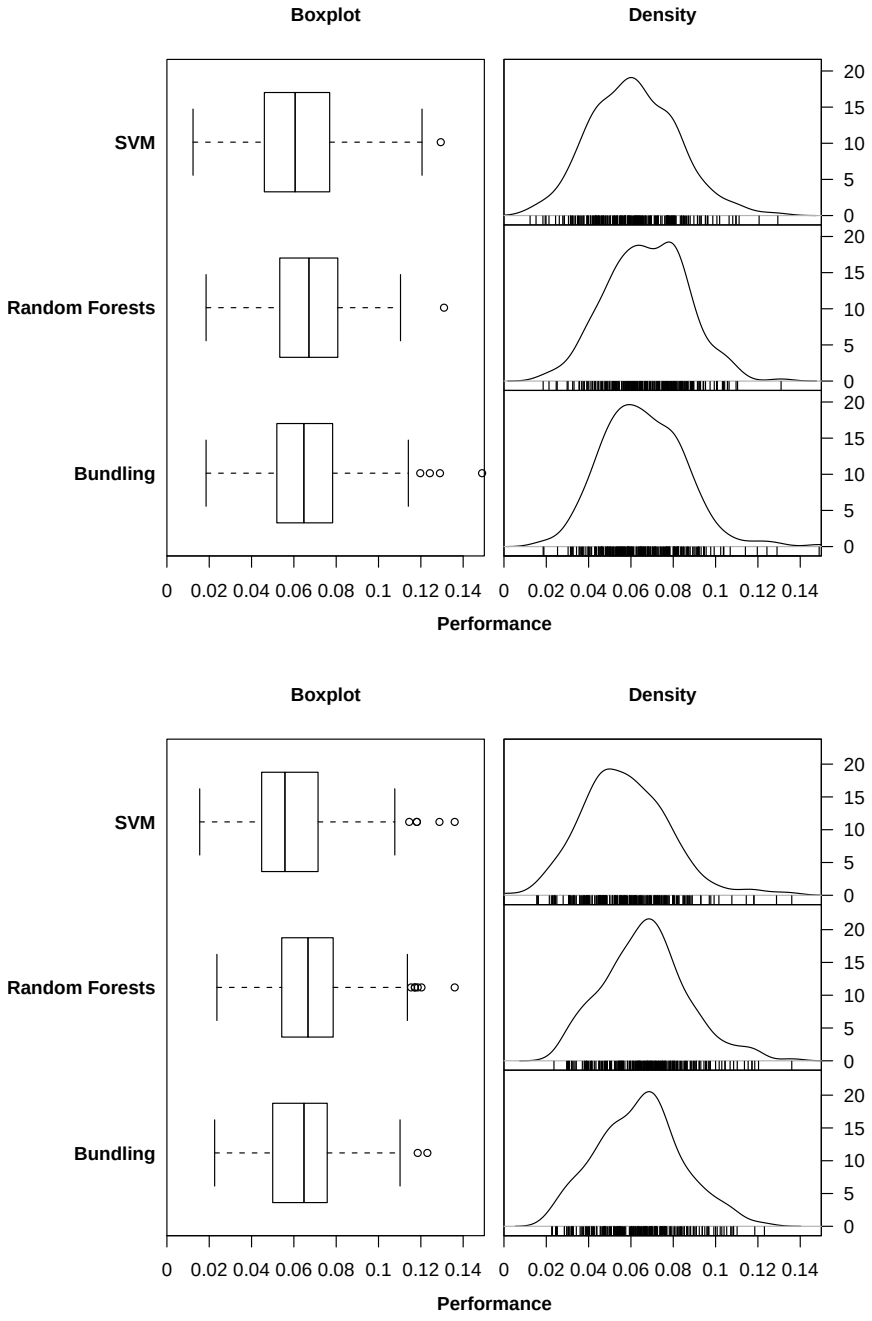


Figure 5: The distribution of the cross-validation (top) and out–of–bootstrap (bottom) performance measure for the Ionosphere data.

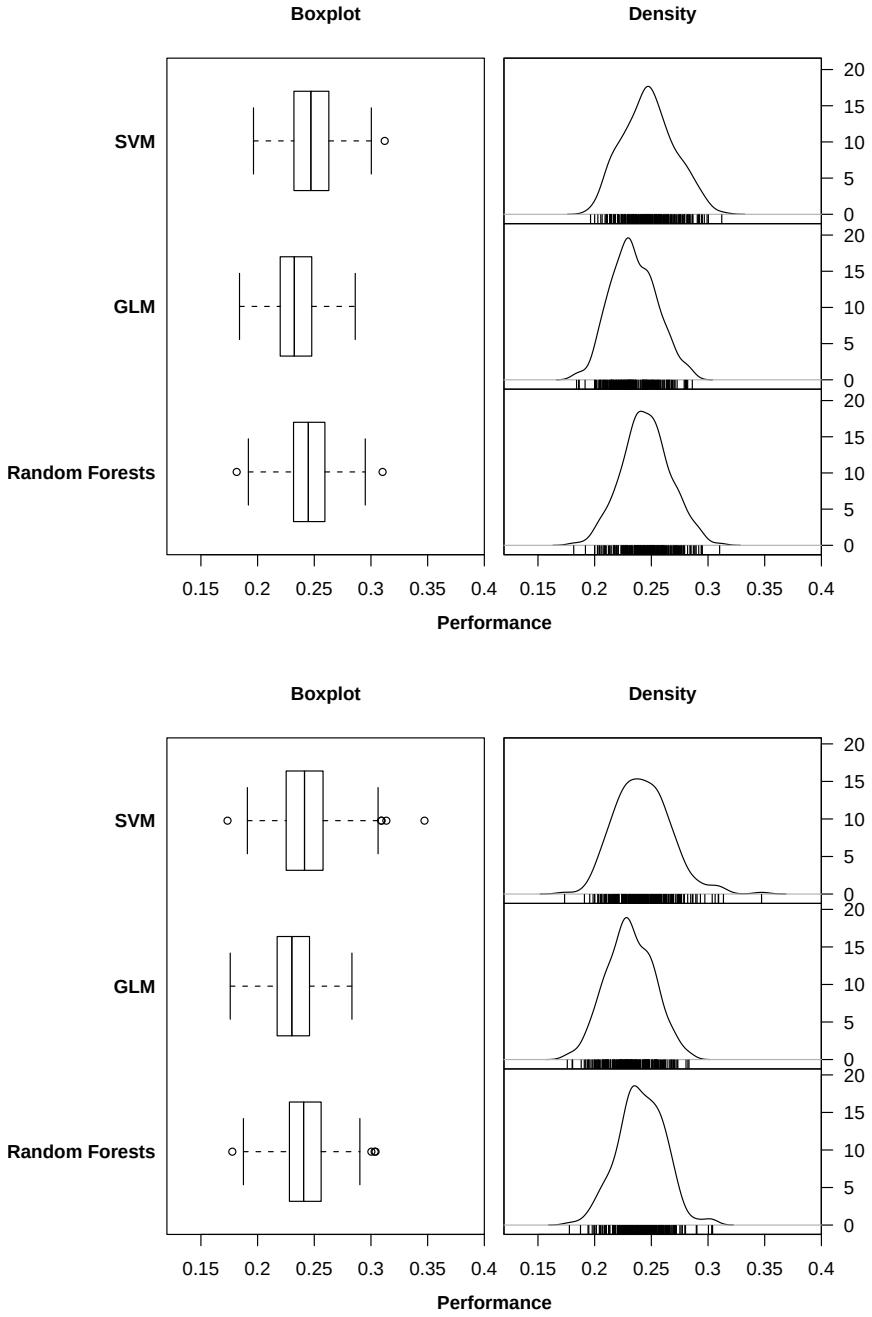


Figure 6: The distribution of the cross-validation (top) and out–of–bootstrap (bottom) performance measure for the Pima Indian Diabetes data.

6 Discussion and Future Work

Within the theoretical framework presented in this paper, the problem of comparing the performance of a set of algorithms is reduced to the problem of comparing random samples from K numeric distributions. This test problem has been one of the major research fields in statistics in the last century and still remains a matter of debate. Therefore, we now enjoy the fruits of 100 years of research in experimental design and test theory in benchmark experiments as well.

Apart from mapping the original problems into a well known one, the theory presented here clarifies which hypotheses we ideally would like to test and which kind of inference is actually possible given the data. It turns out that in real world applications all inference is conditional on the empirical performance measure and we cannot test hypotheses about the theoretical performance distributions. The discrepancy between those two issues is best illustrated by the power simulations for Problem II in Section 5.1. The empirical performance measure is defined by the average loss on a prespecified test sample which may very well, just by chance, favour overfitting instead of the algorithm fitting the true regression relationship. Consequently, it is unwise to set a test sample aside for performance evaluation. Instead, the performance measure should be defined in terms of cross-validation or out–of–bootstrap estimates for the whole learning sample.

It should be noted that the framework can be used to compare a set of algorithms but does not offer a model selection or input variable selection procedure in the sense of Chapelle, Vapnik, and Bengio (2002); Bartlett, Boucheron, and Lugosi (2002) or Bengio and Chapados (2003). Those papers address the problem of identifying a model with good generalisation error from a rich class of flexible models which is beyond the scope of our investigations. The comparison of the performance of algorithms across applications (question 9 in Dietterich 1998), such as for all classification problems in the UCI repository, is not addressed here either.

It is good practice to visualise the data of interest. Since we mainly focus on performance distributions, a graphical representation highlighting the most important properties such as variability, skewness or outliers should always accompany tables reporting raw results. Kernel density estimators and simple boxplots as shown in Section 5.2 are appropriate tools for visualising the performance distributions and offer much more information than tables do.

The results for the artificial regression and the real world examples suggest that we may detect performance differences with fairly high power. One should always keep in mind that statistical significance does not imply a practically relevant discrepancy and therefore the amount of the difference should be inspected by confidence intervals and judged in the light of analytic expertise.

This paper leaves several questions open. Although virtually all statistical methods dealing with numeric distributions are in principle applicable to the problems arising in benchmark experiments, not all of them may be appropriate. From our point of view, procedures which require strong parametric assumptions should be ruled out in favour of inference procedures which condition on the data actually seen. The gains and losses of test procedures of different origin in benchmarking studies need to be investigated. The application of the theoretical framework to time series is easily possible when the data generating process is known (Problem I: Simulation). Drawing random samples from observed time series in a non-parametric way is much harder than redrawing from standard independent and identically distributed samples (see Bühlmann 2002, for a survey) and the application within our framework needs to be investigated. The amount of information presented in reports on benchmarking experiments is enormous. A numerical or graphical display of all performance distributions may be infeasible and therefore improved graphical representations need to be developed. In principle, all computational tasks necessary to draw random samples from the performance distributions are easy to implement or even already packaged in popular software systems for data analysis but a detailed description easy to follow by the practitioner is of main importance.

Summarising, the theory of inference for benchmark experiments suggested here can not offer a fixed reference mark such as for measurements in land surveying, however, the problems are embedded into the well known framework of statistical test procedures allowing for reasonable decisions in an uncertain environment.

7 Conclusion

The main contribution of this paper is a sound and general theoretical framework for inference about the performance distributions of a set of K algorithms. The most interesting question addressed in benchmark studies "Is there any difference between the candidates with respect to their performance?" can be answered by standard statistical test procedures for the comparison of several numeric distributions applied to random samples from the performance distributions of interest.

References

Alpaydin, E. (1999), "Combined 5×2 cv F Test for Comparing Supervised Classification Learning Algorithms," *Neural Computation*, 11, 1885–1892.

Bartlett, P. L., Boucheron, S., and Lugosi, G. (2002), "Model Selection and Error Estimation," *Machine Learning*, 48, 85–113.

Bauer, E. and Kohavi, R. (1999), "An empirical comparison of voting classification algorithms: Bagging, boosting, and variants," *Machine Learning*, 36, 105–139.

Bengio, Y. and Chapados, N. (2003), "Extensions to Metric-Based Model Selection," *Journal of Machine Learning Research*, 3, 1209–1227.

Berger, V. W. (2000), "Pros and cons of permutation tests in clinical trials," *Statistics in Medicine*, 19, 1319–1328.

Berger, V. W., Lunneborg, C., Ernst, M. D., and Levine, J. G. (2002), "Parametric Analyses in Randomized Clinical Trials," *Journal of Modern Applied Statistical Methods*, 1, 74–82.

Blake, C. L. and Merz, C. J. (1998), "UCI Repository of machine learning databases," .

Blockeel, H. and Struyf, J. (2002), "Efficient Algorithms for Decision Tree Cross-validation," *Journal of Machine Learning Research*, 3, 621–650.

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

— (2001), "Random Forests," *Machine Learning*, 45, 5–32.

Bylander, T. (2002), "Estimating Generalization Error on Two-Class Datasets Using Out-of-Bag Estimates," *Machine Learning*, 48, 287–297.

Bühlmann, P. (2002), "Bootstraps for time series," *Statistical Science*, 17, 52–72.

Chang, C.-C. and Lin, C.-J. (2001), *LIBSVM: A library for support vector machines*, Department of Computer Science and Information Engineering, National Taiwan University.

Chapelle, O., Vapnik, V., and Bengio, Y. (2002), "Model Selection for Small Sample Regression," *Machine Learning*, 48, 9–23.

Conover, W. J., Johnson, M. E., and Johnson, M. M. (1981), "A comparative study of tests for homogeneity of variances, with applications to the outer continental shelf bidding data." *Technometrics*, 23, 351–361, with correction (V26 p302).

Dietterich, T. G. (1998), "Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms," *Neural Computation*, 10, 1895–1923.

— (2000), "An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization," *Machine Learning*, 40, 139–157.

Dudoit, S. and van der Laan, M. J. (2003), "Asymptotics of Cross-Validated Risk Estimation in Model Selection and Performance Assessment," Tech. Rep. 126, Devision of Biostatistics, University of California, Berkeley, California.

Dunnett, C. W. (1955), "A multiple comparison procedure for comparing several treatments with a control," *Journal of the American Statistical Association*, 50, 1096–1121.

Efron, B. (1983), "Estimating the Error Rate of a Prediction Rule: Improvements on Cross-Validation," *Journal of the American Statistical Association*, 78, 316–331.

— (1986), "How Biased is the Apparent Error Rate of a Prediction Rule?" *Journal of the American Statistical Association*, 81, 461–470.

Efron, B. and Tibshirani, R. (1997), "Improvements on Cross-Validation: The .632+ Bootstrap Method," *Journal of the American Statistical Association*, 92, 548–560.

Efron, B. and Tibshirani, R. J. (1993), *An Introduction to the Bootstrap*, New York: Chapman & Hall.

Freund, Y. and Schapire, R. E. (1996), "Experiments with a new boosting algorithm," in *Machine Learning: Proceedings of the Thirteenth International Conference*, ed. Saitta, L., San Francisco: Morgan Kaufmann, pp. 148–156.

George, E. I. (2000), "The variable selection problem," *Journal of the American Statistical Association*, 95, 1304–1308.

Hájek, J., Šidák, Z., and Sen, P. K. (1999), *Theory of Rank Tests*, London: Academic Press, 2nd ed.

Hochberg, Y. and Tamhane, A. C. (1987), *Multiple Comparison Procedures*, New York: John Wiley & Sons.

Hothorn, T. (2003), "Bundling classifiers with an application to glaucoma diagnosis," Ph.D. thesis, Department of Statistics, University of Dortmund, Germany, `http://eldorado.uni-dortmund.de:8080/FB5/ls7/forschung/2003/Hothorn`.

Hothorn, T. and Lausen, B. (2003a), "Bundling Classifiers by Bagging Trees," *Preprint, Friedrich-Alexander-Universität Erlangen-Nürnberg*.

— (2003b), "Double-Bagging: Combining classifiers by bootstrap aggregation," *Pattern Recognition*, 36, 1303–1309.

Ihaka, R. and Gentleman, R. (1996), "R: A Language for Data Analysis and Graphics," *Journal of Computational and Graphical Statistics*, 5, 299–314.

Läuter, J. (1992), *Stabile multivariate Verfahren: Diskriminanzanalyse - Regressionsanalyse - Faktoranalyse*, Berlin: Akademie Verlag.

Lehmann, E. L. (1994), *Testing Statistical Hypothesis*, New York: Chapman & Hall, 2nd ed.

Liaw, A. and Wiener, M. (2002), "Classification and Regression by randomForest," *R News*, 2, 18–22.

Lim, T.-S., Loh, W.-Y., and Shih, Y.-S. (2000), "A Comparison of Prediction Accuracy, Complexity, and Training Time of Thirty-Three Old and New Classification Algorithms," *Machine Learning*, 40, 203–228.

Meyer, D. (2001), "Support Vector Machines," *R News*, 1, 23–26.

Meyer, D., Leisch, F., and Hornik, K. (2003), "The Support Vector Machine under Test," *Neurocomputing*, 55, 169–186.

Nadeau, C. and Bengio, Y. (2003), "Inference for the Generalization Error," *Machine Learning*, 52, 239–281.

Patterson, J. G. (1992), *Benchmarking Basics*, Menlo Park, California: Crisp Publications Inc.

Pesarin, F. (2001), *Multivariate Permutation Tests: With Applications to Biostatistics*, Chichester: John Wiley & Sons.

Peters, A., Hothorn, T., and Lausen, B. (2002), "ipred: Improved Predictors," *R News*, 2, 33–36.

Pflüger, R. and Hothorn, T. (2002), "Assessing Equivalence Tests with Respect to their Expected p-Value," *Biometrical Journal*, 44, 1002–1027.

Pizarro, J., Guerrero, E., and Galindo, P. L. (2002), "Multiple comparison procedures applied to model selection," *Neurocomputing*, 48, 155–173.

R Development Core Team (2003), *R: A language and environment for statistical computing*, R Foundation for Statistical Computing, Vienna, Austria, iSBN 3-900051-00-3.

Rossman, A. J. and Chance, B. L. (2001), *Workshop statistics: Discovery with data.*, Emeryville: Key College Publishing, 2nd ed.

Schiavo, R. A. and Hand, D. J. (2000), "Ten More Years of Error Rate Research," *International Statistical Review*, 68, 295–310.

Stone, M. (1974), "Cross-Validatory Choice and Assessment of Statistical Predictions," *Journal of the Royal Statistical Society, Series B*, 36, 111–147.

Vapnik, V. (1998), *Statistical Learning Theory*, New York: John Wiley & Sons.

Vehtari, A. and Lampinen, J. (2002), "Bayesian Model Assessment and Comparison Using Cross-Validation Predictive Densities," *Neural Computation*, 14, 2439–2468.

Wolpert, D. H. and Macready, W. G. (1999), "An efficient method to estimate bagging's generalization error," *Machine Learning*, 35, 41–51.