The Design and Analysis of Benchmark Experiments

Torsten Hothorn
Friedrich-Alexander-Universität
Erlangen-Nürnberg

Friedrich Leisch
Technische Universität Wien

Achim Zeileis
Wirtschaftsuniversität Wien

Kurt Hornik
Wirtschaftsuniversität Wien

Abstract

The assessment of the performance of learners by means of benchmark experiments is an 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 regression and classification benchmark studies based on artificial and real world data.

Keywords: model comparison, performance, hypothesis testing, cross-validation, bootstrap.

1. Introduction

In statistical 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.

This is a preprint of an article published in *Journal of Computational and Graphical Statistics*, Volume 14, Number 3, Pages 675–699.

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 2005) 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 exercise, at least among statisticians influenced by the "algorithmic modelling culture" (Breiman 2001b). In fact, many of the popular benchmark problems first came up in the statistical literature, such as the Ozone and Boston Housing problems (by Breiman and Friedman 1985). Friedman (1991) contributed the standard artificial regression problems. Other well known datasets like the Pima indian diabetes data or the forensic glass problem play a major role in text books in this field (e.g., Ripley 1996). Further 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 2001a; Hothorn and Lausen 2003).

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 2005) 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 1996a) or random forests (Breiman 2001a), 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"). Dietterich (1998) distinguishes between situations where 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 themselves to those applications (Bauer and Kohavi 1999). The problem whether some out of a set of candidate algorithms outperform all others in a large (question 7) and small sample situation (question 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 question 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. Mixed models are applied for the comparison of algorithms across benchmark problems (for example Lim, Loh, and Shih 2000; Kim and Loh 2003). A basic problem common to these 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 a learning sample of n observations $\mathfrak{L} = \{z_1, \dots, z_n\}$ is given and 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 $\mathfrak{L}$ yielding a function $a(\cdot \mid \mathfrak{L})$ 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 unsupervised situations like density estimation, $a(\cdot \mid \mathfrak{L})$ 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 the learning sample drawn from some data generating process: The function $p(a, \mathfrak{L})$ assesses the performance of the function $a(\cdot \mid \mathfrak{L})$, that is the performance of algorithm a based on learning sample $\mathfrak{L}$. Since $\mathfrak{L}$ is a random learning sample, $p(a, \mathfrak{L})$ is a random variable whose variability is induced by the variability of learning samples following the same data generating process as $\mathfrak{L}$.

It is therefore natural to compare the distribution 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 $p(a, \mathfrak{L})$, where the learning sample $\mathfrak{L}$ 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 hypotheses about the distributions of the performance measures of a set of candidates 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 regression and classification problems in Section 3. Once appropriate random samples from the performance distribution have been drawn, the established statistical test and analysis 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 in benchmark studies, for example in order to test the hypothesis of equal performances.

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

$$\begin{aligned} \mathfrak{L}^1 &= \{z_1^1, \dots, z_n^1\} \sim DGP, \\ &\vdots \\ \mathfrak{L}^B &= \{z_1^B, \dots, z_n^B\} \sim DGP, \end{aligned}$$

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

$$a_k(\cdot \mid \mathfrak{L}^b) \sim \mathcal{A}_k(DGP), \quad k = 1, \dots, K.$$

For algorithms a_k with deterministic fitting procedure (for example histograms or linear models) the function $a_k(\cdot \mid \mathfrak{L}^b)$ 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 $a_k(\cdot \mid \mathfrak{L}^b)$ 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 performance of the candidate algorithm a_k when provided with the learning sample $\mathfrak{L}^b$ is measured by a scalar function p:

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

The random variable p_{kb} follows a distribution function $\mathcal{P}_k$ 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 $\mathcal{A}_k$ when evaluating its performance.

The K different random samples $\{p_{k1}, \dots, p_{kB}\}$ with B independent and identically distributed observations are drawn from the distributions $\mathcal{P}_k(DGP)$ for algorithms a_k $(k = 1, \dots, K)$. These performance distributions can be compared by both exploratory data analysis tools as well as formal inference procedures. 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

$$H_0 : \mathcal{P}_1 = \cdots = \mathcal{P}_K.$$

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 $\phi(\mathcal{P}_k) < \phi(\mathcal{P}_{k'})$. The optimality criterion most commonly used is based on some location parameter such as the expectation $\phi(\mathcal{P}_k) = \mathsf{E}(\mathcal{P}_k)$ 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:

$$H_0 : \mathsf{E}(\mathcal{P}_1) = \cdots = \mathsf{E}(\mathcal{P}_K) \text{ vs. } H_1 : \exists\, i, j \in \{1, \dots, K\} : \mathsf{E}(\mathcal{P}_i) \neq \mathsf{E}(\mathcal{P}_j).$$

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. Regression and classification

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 statistical 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 $\hat{y} = a_k(x \mid \mathfrak{L}^b)$. 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 and the distribution of p_{kb} depends on the data generating process DGP only:

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

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

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

where $z = (y, x)$ is drawn from the same distribution as the observations in a learning sample $\mathfrak{L}$. 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 $\mathcal{P}_k(DGP)$ for algorithms a_k $(k = 1, \dots, K)$ depend on the data generating process DGP. Consequently, the way we draw random samples from $\mathcal{P}_k(DGP)$ 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 $\mathfrak{L}$. In this case the empirical distribution function of $\mathfrak{L}$ 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 $\mathcal{P}_k(DGP)$ 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). Special choices of the functional μ appropriate in each of the three problems will be discussed.

The simulation problem

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

on all learning samples $\mathfrak{L}^b (b = 1, \dots, B)$ yielding a random sample of B observations from the performance distribution $\mathcal{P}_k(\mathcal{Z}_n)$ by calculating

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

The associated hypothesis under test is consequently

$$H_0 : \mathcal{P}_1(\mathcal{Z}_n) = \cdots = \mathcal{P}_K(\mathcal{Z}_n).$$

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

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

Here $\mu_{\mathfrak{T}}$ denotes the empirical analogue of μ for the test observations $z = (y, x) \in \mathfrak{T}$. 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

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

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

The competition problem

In most practical applications no precise knowledge about the data generating process is available but instead we are provided with one learning sample $\mathfrak{L} \sim \mathcal{Z}_n$ of n observations from some distribution function $\mathcal{Z}$. The empirical distribution function $\hat{\mathcal{Z}}_n$ 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: $DGP = \hat{\mathcal{Z}}_n$. 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 or Bayesian 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 $\mathfrak{T} \sim \mathcal{Z}_m$ of m observations is given, for example in machine learning competitions. In this situation, the performance needs to be assessed with respect to $\mathfrak{T}$ only. Again, we would like to draw a random sample of B observations from $\hat{\mathcal{P}}_k(\hat{\mathcal{Z}}_n)$, which in this setup is possible by bootstrapping $\mathfrak{L}^1, \dots, \mathfrak{L}^B \sim \hat{\mathcal{Z}}_n$, where $\hat{\mathcal{P}}$ denotes the distribution function of the performance measure evaluated using $\mathfrak{T}$, that is, the performance measure is computed by

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

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

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

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

The real world problem

The most common situation we are confronted with in daily routine is the existence of one single learning sample $\mathfrak{L} \sim \mathcal{Z}_n$ with no dedicated independent test sample being available. Again, we mimic the data generating process by the empirical distribution function of the learning sample: $DGP = \hat{\mathcal{Z}}_n$. We redraw B independent learning samples from the empirical distribution function by bootstrapping: $\mathfrak{L}^1, \dots, \mathfrak{L}^B \sim \hat{\mathcal{Z}}_n$. The corresponding performance measure is computed by

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

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 $\mathfrak{L} = \{\mathfrak{L}', \mathfrak{T}\}$ and proceed with $\mu_{\mathfrak{T}}$ as in the competition problem. If n is not large enough for this to be feasible, the following approach is a first naive choice: In the simulation problem, the models are fitted on samples from $\mathcal{Z}_n$ and their performance is evaluated on samples from $\mathcal{Z}$. Here, the models are trained on samples from the empirical distribution function $\hat{\mathcal{Z}}_n$ and so we could want to assess their performance on $\hat{\mathcal{Z}}$ which corresponds to emulating $\mu_{\mathfrak{T}}$ by using the learning sample $\mathfrak{L}$ as test sample, i.e., for each model fitted on a bootstrap sample, the original learning sample $\mathfrak{L}$ itself is used as test sample $\mathfrak{T}$.

Except for algorithms able to compute 'honest' predictions for the observations in the learning sample (for example bagging's out-of-bag predictions, Breiman 1996b), this choice leads to overfitting problems. Those can be addressed by well known cross-validation strategies. The test sample $\mathfrak{T}$ can be defined in terms of the out-of-bootstrap observations when evaluating $\mu_{\mathfrak{T}}$:

- RW-OOB. For each bootstrap sample $\mathfrak{L}^b (b = 1, \dots, B)$ the out-of-bootstrap observations $\mathfrak{L} \setminus \mathfrak{L}^b$ 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:

- RW-CV. Each bootstrap sample $\mathfrak{L}^b$ 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 $\mathfrak{L}$ 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 sample 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

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

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, such as their expectation. A lot of attention has been paid to this and similar problems in the statistical literature and so a rich toolbox can be applied here. We will comment on appropriate test procedures for only the most important test problems commonly addressed in benchmark experiments and refer to the standard literature otherwise.

4.1. Experimental designs

A matched pairs or dependent K samples design is the most natural choice for the comparison of algorithms since the performance of all K algorithms is evaluated using the same random samples $\mathfrak{L}^1, \dots, \mathfrak{L}^B$. We therefore use this design for the derivations in the previous sections and the experiments in Section 5 and compare the algorithms based on the same set of learning samples. The application of an independent K sample design may be more comfortable from a statistical point of view. Especially the derivation of confidence intervals for parameters like the difference of the misclassification errors of two algorithms or the visualisation of the performance distributions is straightforward in the independent K samples setup.

4.2. Analysis

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 $d_b = p_{1b} - p_{2b}$ $(b = 1, \dots, B)$ 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

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

is asymptotically normal and follows a t-distribution with $B-1$ degrees of freedom when the differences are drawn from a normal distribution. The unconditional distribution of this and other similar test statistics is derived under some parametric assumption, such as symmetry or normality, about the distributions of the underlying observations. However, we doubt that such parametric assumptions to performance distributions are ever appropriate. 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). In the competition and real world problems however, the inference is conditional on an observed learning sample 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 $K > 2$ 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

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

can be used to construct a permutation test, where the distribution of $t^\star$ is obtained by permuting the labels $1, \dots, K$ of the algorithms for each sample $\mathfrak{L}^b (b = 1, \dots, B)$ independently (Pesarin 2001).

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, where the hierarchical hypotheses formulate subsets of the global hypothesis and can be tested at level α, for a description see Hochberg and Tamhane (1987). However, closed testing procedures are computationally expensive for more than $K = 4$ algorithms. In this case, one can apply simultaneous test procedures or confidence intervals designed for the independent K samples case to the aligned performance measures (Hájek, Šidák, and Sen 1999).

It is important to note that one is able to detect very small performance differences with very high power when the number of learning samples B is large. Therefore, practical relevance instead of statistial significance needs to be assessed, for example by showing relevant superiority by means of confidence intervals. Further comments on those issues can be found in Section 6.

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 regression and classification 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 simple artificial data generating processes from a univariate regression relationship and a two-class classification problem we will study the power of tests based on the empirical performance measures for the simulation, competition and real world problems.

Maybe the most interesting question addressed in benchmarking experiments is "Are there any differences between state–of–the–art algorithms with respect to a certain performance measure?". For the real world problem we investigate this for some established and recently suggested supervised learning algorithms by means of three real world learning samples from the UCI repository (Blake and Merz 1998). All computations were performed within the R system for statistical computing (Ihaka and Gentleman 1996; R Development Core Team 2004), version 1.9.1.

5.1. Nested linear models

In order to compare the mean squared error of two nested linear models consider the data generating process following a univariate regression equation

$$y = \beta_1 x + \beta_2 x^2 + \varepsilon \tag{4}$$

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 regression coefficient $\beta_1 = 2$ and the number of observations in a learning sample to $n = 150$. 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_2 is estimated.

The discrepancy between a predicted value $\hat{y} = a_k(x|\mathfrak{L}), k = 1, 2$ 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_2 varying in a range between 0 and 0.16. As described in detail in Section 3.2, both the performance measure and the sampling from the performance distribution depend on the degree

	Simulation		Competition	Real World		
β_2	$m = 2000$	$m = 150$		OOB	OOB-$2n$	CV
0.000	0.000	0.000	0.072	0.054	0.059	0.054
0.020	0.029	0.287	0.186	0.114	0.174	0.109
0.040	0.835	0.609	0.451	0.297	0.499	0.279
0.060	0.997	0.764	0.683	0.554	0.840	0.523
0.080	1.000	0.875	0.833	0.778	0.973	0.777
0.100	1.000	0.933	0.912	0.925	0.997	0.926
0.120	1.000	0.971	0.953	0.984	1.000	0.978
0.140	1.000	0.988	0.981	0.996	1.000	0.996
0.160	1.000	0.997	0.990	1.000	1.000	1.000

Table 1: Regression experiments: Power of the tests for the simulation, competition and real world problems for varying values of the regression coefficient β_2 of the quadratic term. Learning samples are of size $n = 150$.

of knowledge available and we therefore distinguish between three different problems discussed above.

Simulation

The data generating process Z_n is known by equation (4) and we are able to draw as many learning samples of $n = 150$ observations as we would like. It is, in principle, possible to calculate the mean squared error of the predictive functions $a_1(\cdot \mid \mathfrak{L}^b)$ and $a_2(\cdot \mid \mathfrak{L}^b)$ when learning sample $\mathfrak{L}^b$ 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:

$$H_0 : \mathsf{E}(\mathcal{P}_1(\mathcal{Z}_n)) \leq \mathsf{E}(\mathcal{P}_2(\mathcal{Z}_n)) \text{ vs. } H_1 : \mathsf{E}(\mathcal{P}_1(\mathcal{Z}_n)) > \mathsf{E}(\mathcal{P}_2(\mathcal{Z}_n)). \tag{5}$$

However, closed form solutions are only possible in very simple cases and we therefore approximate $\mathcal{P}_k(\mathcal{Z}_n)$ by $\hat{\mathcal{P}}_k(\mathcal{Z}_n)$ using a large test sample $\mathfrak{T}$, in our case with $m = 2000$ observations. Some algebra shows that, for $\beta_2 = 0$ and model a_1, the variance of the performance approximated by a test sample of size m is $\mathsf{V}(p_{b1}) = m^{-1}(350/3 \cdot (2 - \hat{\beta}_1)^4 + 25/2 \cdot (2 - \hat{\beta}_1)^2 + 2)$. In order to study the goodness of the approximation we, in addition, choose a smaller test sample with $m = 150$. Note that in this setup the inference is conditional under the test sample.

Competition

We are faced with a learning sample $\mathfrak{L}$ with $n = 150$ observations. The performance of any algorithm is to be measured by an additional test sample $\mathfrak{T}$ consisting of $m = 150$ observations. Again, the inference is conditional under the observed test sample and we may, just by chance, observe a test sample favouring a quadratic model even if $\beta_2 = 0$. The data generating process is emulated by the empirical distribution function $DGP = \hat{\mathcal{Z}}_n$ and we resample by using the non-parametric bootstrap.

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 based on the out-of-bootstrap observations (RW-OOB) and cross-validation (RW-CV) suggested in Section 3.2. For cross-validation, the performance measure is obtained from a 5-fold cross-validation estimator. Each bootstrap sample is divided into 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

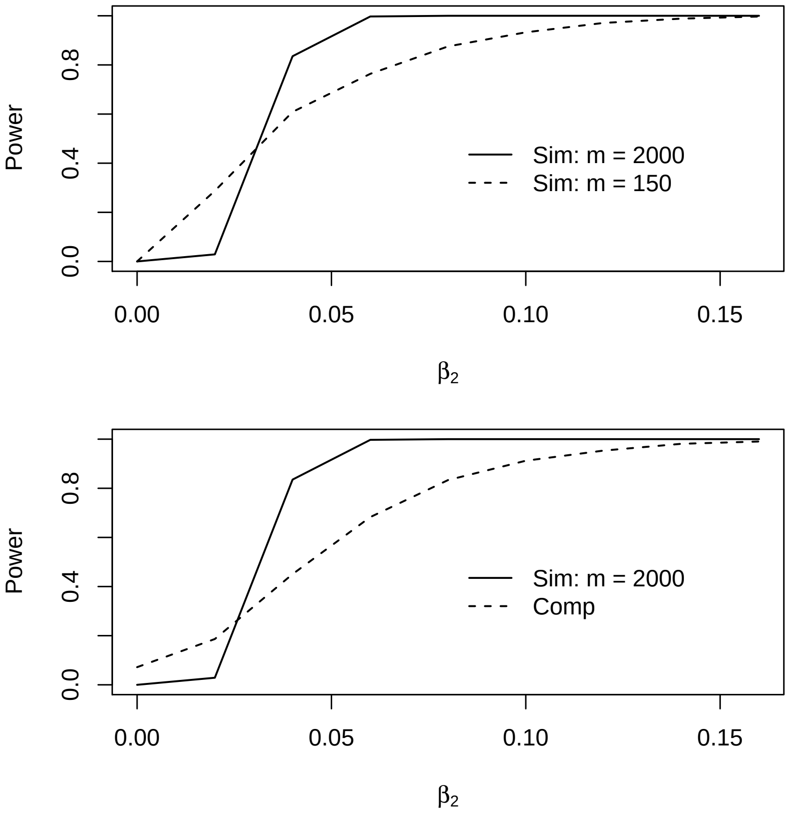

Figure 1: Regression experiments: Power curves depending on the regression coefficient β_2 of the quadratic term for the tests in the simulation problem (Sim) with large ($m = 2000$) and small ($m = 150$) test sample (top) and the power curve of the test associated with the competition problem (Comp, bottom).

performance measure in the real world problem with the empirical performance measure in the competition problem:

- RW-OOB-2n. A hypothetical learning sample and test sample of size $m = n = 150$ each are merged into one single learning sample with 300 observations and we proceed as with the out-of-bootstrap approach.

For our investigations here, we draw $B = 250$ learning samples either from the true data generating process $\mathcal{Z}_n$ (simulation) or from the empirical distribution function $\hat{\mathcal{Z}}_n$ by the non-parametric bootstrap (competition or real world). 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, that are the proportions of rejections of the null hypothesis (5) for varying values of β_2, 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 quadratic model a_2 outperforms the simple linear model a_1 with respect to its theoretical mean squared error. For $\beta_2 = 0$, the bias of the predictions $a_1(\cdot|\mathfrak{L})$ and $a_2(\cdot|\mathfrak{L})$ 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,

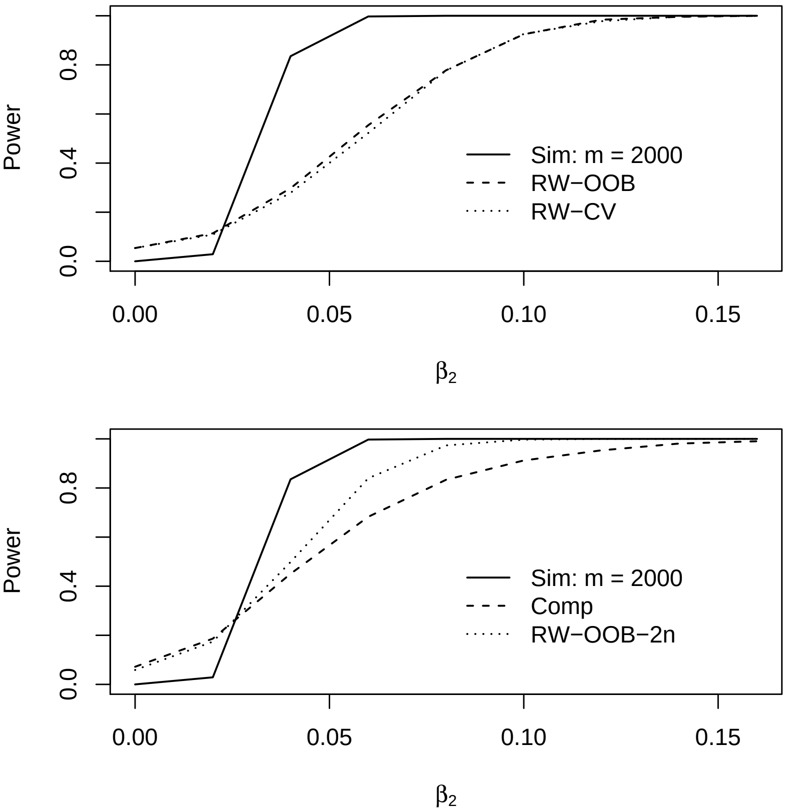

Figure 2: Regression experiments: Power of the out-of-bootstrap (RW-OOB) and cross-validation (RW-CV) approaches depending on the regression coefficient β_2 in the real world problem (top) and a comparison of the competition (Comp) and real world problem (RW-OOB) (bottom).

the theoretical mean squared error of a_1 is smaller than the mean squared error of a_2 for $\beta_2 = 0$ which reflects the situation under the null hypothesis in test problem (5). As β_2 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 (simulation, $m = 2000$). The test problem is formulated in terms of the theoretical performance measures $\mathcal{P}_k(\mathcal{Z}_n), k = 1, 2$, but we are never able to draw samples from these distributions in realistic setups. Instead, we approximate them in the simulation problem by $\hat{\mathcal{P}}_k(\mathcal{Z}_n)$ either very closely with $m = 2000$ or less accurately with $m = 150$ which we use for comparisons with the competition and real world problems where the empirical performance distributions $\hat{\mathcal{P}}_k(\hat{\mathcal{Z}}_n)$ are used.

The simulation problem with large test samples ($m = 2000$) in the second column of Table 1 offers the closest approximation of the comparison of the theoretical performance measures: For $\beta_2 = 0$ we are always able to detect that a_1 outperforms a_2. As β_2 increases the performance of a_2 improves compared to a_1 and eventually outperforms a_1 which we are able to detect always for $\beta_2 \geq 0.08$. As this setup 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 $\beta_2 = 0$ 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 simulation and real world problems but not for the competition problem due to the usage of a fixed test sample. 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 the competition problem, by the choice of a fixed test sample which may favour a quadratic model even for $\beta_2 = 0$ and so the power is 0.072. For the performance measures defined in terms of out-of-bootstrap observations or cross-validation estimates, the estimated power for $\beta_2 = 0$ is 0.054. 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 $\beta_2 > 0$, the power curves of all other problems are flatter than that for the simulation problem with large test samples ($m = 2000$) 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. However, the procedures based on out-of-bootstrap and cross-validation—which are virtually indistinguishable—are fairly close to the power curve for the simulation problem $m = 150$ 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 (simulation, $m = 150$).

It should be noted that, instead of relying on the competition setup when a separate test sample is available, the conversion into a real world problem seems appropriate: The power curve is higher for large values of β_2 and the value 0.059 covers the nominal size $\alpha = 0.05$ of the test problem (5) better for $\beta_2 = 0$. 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. Recursive partitioning and linear discriminant analysis

We now consider a data generating process for a two-class classification problem with equal class priors following a bivariate normal distribution with covariance matrix $\Sigma = \text{diag}\left((0.2, 0.2)\right)$. For the observations of class 1, the mean is fixed at $(0, 0)$, and for 50% of the observations of class 2 the means is fixed at $(0, 1)$. The mean of the remaining 50% of the observations of class 2 depends on a parameter γ via $(\cos(\gamma\pi/180), \sin(\gamma\pi/180))$. For angles of $\gamma = 0, \dots, 90$ degrees, this group of observations moves on a quarter circle line from $(1, 0)$ to $(0, 1)$ with distance 1 around the origin. In the following, the performance measured by average misclassification loss of recursive partitioning (package **rpart**, Therneau and Atkinson 1997) and linear discriminant analysis as implemented in package **MASS** (Venables and Ripley 2002) is compared.

For $\gamma = 0$, two rectangular axis parallel splits separate the classes best, and recursive partitioning will outperform any linear method. As γ grows, the classes become separable by a single hyper plane which favours the linear method in our case. For $\gamma = 90$ degrees, a single axis parallel split through $(0, 0.5)$ is the optimal decision line. The linear method is optimal in this situation, however, recursive partitioning is able to estimate this cutpoint. For learning samples of size $n = 200$ we estimate the power for testing the null hypothesis 'recursive partitioning outperforms linear discriminant analysis' against the alternative of superiority of the linear method. Again, the power curves are estimated by means of 5000 Monte-Carlo replications and $B = 250$ learning samples are drawn.

The numerical results are given in Table 2. The theoretical performance measure is again best approximated by the second column of Table 2 (simulation, $m = 2000$). For angles between 0 and 15 degrees, we never reject the null hypothesis of superiority of recursive partitioning and the linear method starts outperforming the trees for γ between 20 and 30 degrees. Note that although linear discriminant analysis is the optimal solution for $\gamma = 90$ degrees, the null hypothesis is not rejected in a small number of cases. When a smaller test sample is used ($m = 200$), more rejections

	Simulation		Competition	Real World		
γ	$m = 2000$	$m = 200$		OOB	OOB-$2n$	CV
0	0.000	0.006	0.011	0.016	0.001	0.016
5	0.000	0.022	0.039	0.050	0.006	0.049
10	0.000	0.064	0.113	0.123	0.024	0.120
15	0.000	0.152	0.236	0.242	0.088	0.262
20	0.045	0.285	0.409	0.385	0.202	0.451
30	0.884	0.632	0.730	0.695	0.514	0.770
50	1.000	0.965	0.958	0.938	0.942	0.949
70	1.000	0.837	0.827	0.781	0.867	0.799
90	0.999	0.823	0.712	0.721	0.803	0.733

Table 2: Classification experiments: Power of the tests for the simulation, competition and real world problems for varying means of 50% of the observations in class 2 defined by angles γ. Learning samples are of size $n = 200$.

occur for angles between 0 and 20 degrees and fewer rejections for larger values of γ. The same can be observed for the competition and real world setups. The out-of-bootstrap and the cross-validation approach appear to be rather similar again. The two most important conclusions from the regression experiments can be stated for this simple classification example as well. At first, the test procedures based on the empirical performance measures have very high power compared with the situation where the complete knowledge about the data generating process is available but a small test sample is used (simulation, $m = 200$). And at second, the out-of-bootstrap approach with $2n$ observations is more appropriate compared to the definition of a dedicated test sample in the competition setup: For angles γ reflecting the null hypothesis, the number of rejections is smaller and the power is higher under the alternative, especially for $\gamma = 90$ degrees.

5.3. Benchmarking applications

The basic concepts are illustrated in the preceding paragraph by means of simple simulation models and we now focus on the application of test procedures implied by the theoretical framework to three 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 following applications:

Boston Housing: a regression problem with 13 input variables and $n = 506$ observations,

Breast Cancer: a two-class classification problem with 9 input variables and $n = 699$ observations,

Ionosphere: a two-class classification problem with 34 input variables and $n = 351$ observations.

Consequently, we are able to test hypotheses formulated in terms of the performance distributions implied by the procedures suggested for the real world problem 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. Again, observations that occur both in learning and validation 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. Meyer (2001) provides an interface to support vector machines (SVM, Vapnik 1998) via the **LIBSVM** library (Chang and Lin 2001) available in package **e1071** (Dimitriadou, Hornik, Leisch, Meyer, and Weingessel 2004). 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

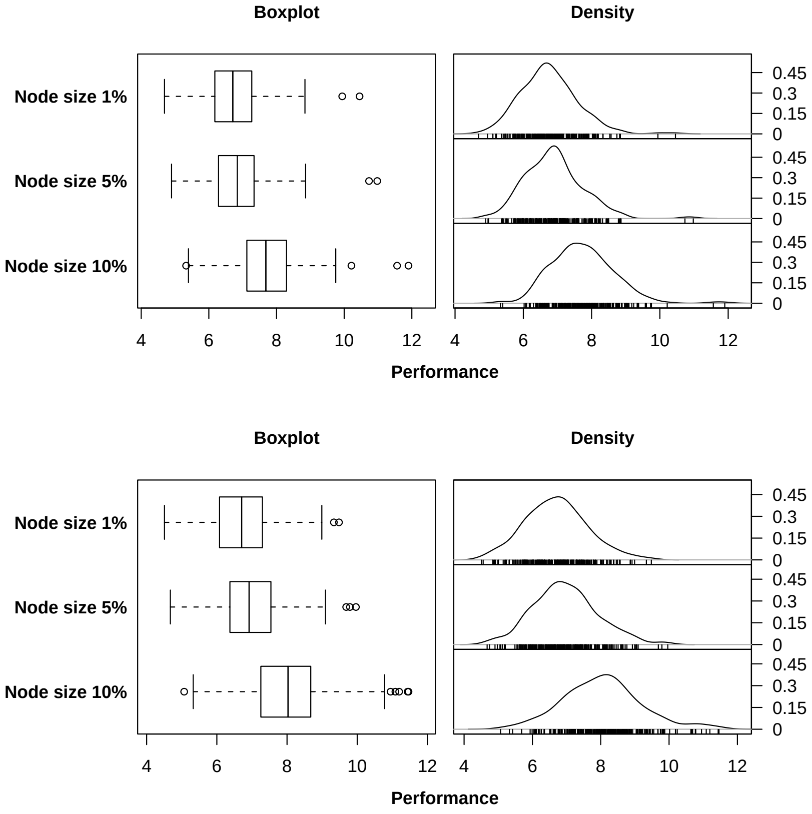

Figure 3: The distribution of the cross-validation (top) and out-of-bootstrap (bottom) performance measure for the Boston Housing data visualised via boxplots and a density estimator.

ipred package (Peters, Hothorn, and Lausen 2002) as well as the binary logistic regression model (GLM) are under study. Random forests (Breiman 2001a) and bundling (Hothorn 2003; Hothorn and Lausen 2005) as a combination of bagging (Breiman 1996a), sLDA, nearest neighbours and GLM are included in this study as representatives of tree-based ensemble methods. Bundling is 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 and compare the distributions both graphically and by means for formal inference procedures. The distribution of the test statistic $t^\star$ from (3) is determined via conditional Monte-Carlo (Pesarin 2001). 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.

Boston Housing

Instead of comparing different algorithms we now investigate the impact of a hyper parameter on the performance of the random forest algorithm, namely the size of the single trees ensembled into a random forest. One possibility to control the size of a regression tree is to define the minimum number of observations in each terminal node. Here, we investigate the influence of the tree size on the performance of random forests with respect to a performance measure defined by the 95% quantile of the absolute difference of response and predictions, i.e., the $\lceil 0.95 \cdot m \rceil$th value of the m ordered differences $|y - a(x, \mathfrak{L})|$ of all m observations (y, x) from some test sample

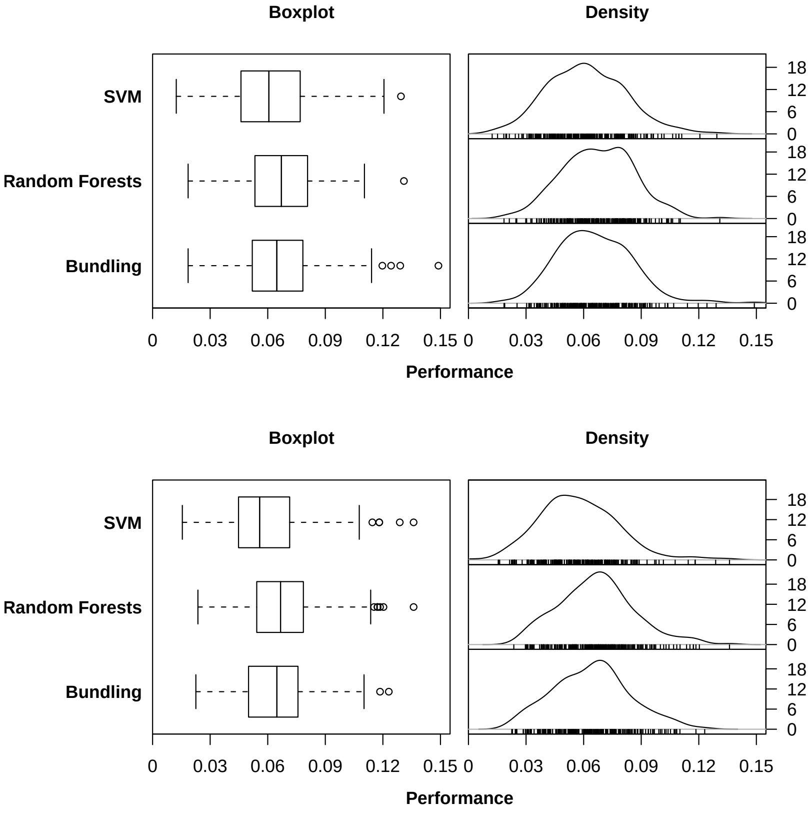

Figure 4: The distribution of the cross-validation (top) and out-of-bootstrap (bottom) misclassification error for the Ionosphere data.

$\mathfrak{T}$. This choice favours algorithms with low probability of large absolute errors rather than low average performance. We fix the minimal terminal node size to 1%, 5% and 10% of the number of observations n in the learning sample $\mathfrak{L}$.

The three random samples of size $B = 250$ each are graphically summarised by boxplots and a kernel density estimator in Figure 3. This representation leads to the impression that small terminal node sizes and thus large trees lead to random forest ensembles with smaller fraction of large prediction errors. The global hypothesis of equal performance distributions is tested using the permutation test based on the statistic (3). For the performance measure based on cross-validation, the value of the test statistic is $t^\star = 0.0018$ 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 $t^\star = 0.0016$, which corresponds to the elevated differences in the means compared to cross-validation. Again, the P-value is less than 0.001. One can expect an error of not more than US$ 6707,– for 95% of the house prices predicted with a forest of large trees while the error increases to US$ 8029,– for forests of small trees. It should be noted that out-of-bootstrap and cross-validation performance distributions lead to the same conclusions.

Ionosphere

For the Ionosphere data the supervised learners SVM, random forests and bundling are compared. The graphical representation of the estimated densities of the distribution of misclassification error in Figure 4 indicate some degree of skewness for all methods. Note that this is not visible in the

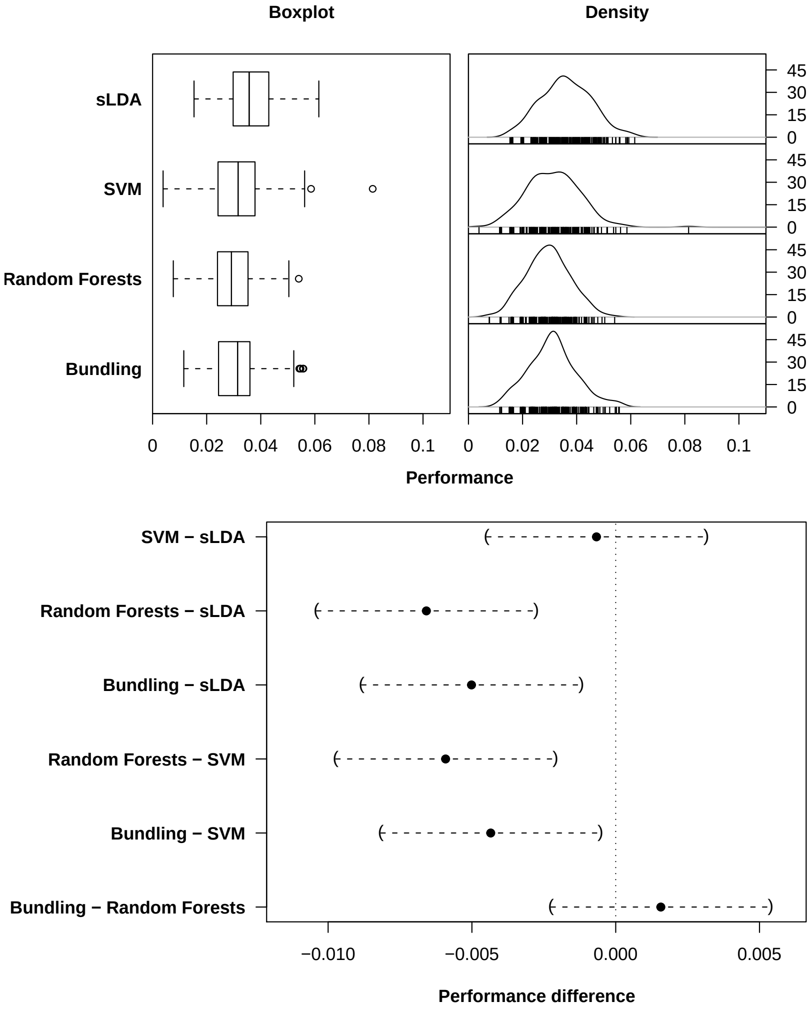

Figure 5: The distribution of the out-of-bootstrap performance measure for the Breast Cancer data (top) and asymptotic simultaneous confidence sets for Tukey all-pair comparisons of the misclassification errors after alignment (bottom).

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 $= 0.008$).

Breast Cancer

The performance of sLDA, SVM, random forests and bundling for the Breast Cancer classification problem is investigated under misclassification loss. Figure 5 depicts the empirical out-of-bootstrap performance distributions. An inspection of the graphical representation leads to the presumption that 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 out-of-bootstrap performance can be rejected (P-value ≤ 0.001). Asymptotic simultaneous confidence sets for Tukey all-pair comparisons after alignment indicate that this is due to the superiority of the ensemble methods compared to sLDA and SVM while no significant differences between SVM and sLDA on the one hand and random forests and bundling on the other hand can be found.

The kernel density estimates for all three 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.

6. Discussion and future work

The popularity of books such as 'Elements of Statistical Learning' (Hastie, Tibshirani, and Friedman 2001) shows that learning procedures with no or only limited asymptotic results for model evaluation are increasingly used in mainstream statistics. 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 received a lot of interest in the last 100 years and benchmarking experiments can now be analysed using this body of literature.

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 the competition problem in Sections 5.1 and 5.2. 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. Organizers of machine learning competitions could define a sequence of bootstrap or cross-validation samples as the benchmark without relying on a dedicated test 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 Bartlett, Boucheron, and Lugosi (2002), Pittman (2002) or Gu and Xiang (2001). These 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.

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. In some applications it is more appropriate to show either the relevant superiority of a new algorithm or the non-relevant inferiority of a well-established procedure, i.e., one is interested in testing one-sided hypotheses of the form

$$H_0 : \phi(\mathcal{P}_1) \leq \phi(\mathcal{P}_2) - \Delta,$$

where Δ defines a pre-specified practically relevant difference.

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 (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. Details of the framework to unsupervised problems have to be worked out. The amount of information presented in reports on benchmarking experiments is enormous. A numerical or graphical display of all performance distributions is difficult and therefore graphical representations extending the ones presented here need to be investigated and applied. Point estimates need to be accomplished by some assessement of their variability, for example by means of confidence intervals. 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.

To sum up, the theory of inference for benchmark experiments suggested here cannot 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.

Acknowledgements

This research was supported by the Austrian Science Foundation (FWF) under grant SFB#010 (‘Adaptive Information Systems and Modeling in Economics and Management Science’) and the Austrian Association for Statistical Computing. In addition, the work of Torsten Hothorn was supported by the Deutsche Forschungsgemeinschaft (DFG) under grant HO 3242/1-1. The authors would like to thank two anonymous referees and an anonymous associate editor for their helpful suggestions.

References

Alpaydin E (1999). “Combined 5 × 2 cv F Test for Comparing Supervised Classification Learning Algorithms.” *Neural Computation*, **11**(8), 1885–1892.

Bartlett PL, Boucheron S, Lugosi G (2002). “Model Selection and Error Estimation.” *Machine Learning*, **48**(1–3), 85–113.

Bauer E, Kohavi R (1999). “An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants.” *Machine Learning*, **36**(1–2), 105–139.

Berger VW (2000). “Pros and Cons of Permutation Tests in Clinical Trials.” *Statistics in Medicine*, **19**(10), 1319–1328.

Berger VW, Lunneborg C, Ernst MD, Levine JG (2002). “Parametric Analyses in Randomized Clinical Trials.” *Journal of Modern Applied Statistical Methods*, **1**(1), 74–82.

Blake CL, Merz CJ (1998). “UCI Repository of Machine Learning Databases.” http://www.ics.uci.edu/~mlearn/MLRepository.html.

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

Breiman L (1996a). "Bagging Predictors." *Machine Learning*, **24**(2), 123–140.

Breiman L (1996b). "Out-of-Bag Estimation." *Technical report*, Statistics Department, University of California Berkeley, Berkeley CA 94708. ftp://ftp.stat.berkeley.edu/pub/users/breiman/.

Breiman L (2001a). "Random Forests." *Machine Learning*, **45**(1), 5–32.

Breiman L (2001b). "Statistical Modeling: The Two Cultures." *Statistical Science*, **16**(3), 199–231. With discussion.

Breiman L, Friedman JH (1985). "Estimating Optimal Transformations for Multiple Regression and Correlation." *Journal of the American Statistical Association*, **80**(391), 580–598.

Bylander T (2002). "Estimating Generalization Error on Two-Class Datasets Using Out-of-Bag Estimates." *Machine Learning*, **48**(1–3), 287–297.

Bühlmann P (2002). "Bootstraps for Time Series." *Statistical Science*, **17**(1), 52–72.

Chang CC, Lin CJ (2001). *LIBSVM: A Library for Support Vector Machines.* Department of Computer Science and Information Engineering, National Taiwan University. http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Dietterich TG (1998). "Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms." *Neural Computation*, **10**(7), 1895–1923.

Dietterich TG (2000). "An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization." *Machine Learning*, **40**(2), 139–157.

Dimitriadou E, Hornik K, Leisch F, Meyer D, Weingessel A (2004). **e1071***: Misc Functions of the Department of Statistics (e1071), TU Wien.* R package version 1.5-1, http://CRAN.R-project.org.

Dudoit S, van der Laan MJ (2005). "Asymptotics of Cross-Validated Risk Estimation in Estimator Selection and Performance Assessment." *Statistical Methodology*, **2**(2), 131–154.

Efron B (1983). "Estimating the Error Rate of a Prediction Rule: Improvements on Cross-Validation." *Journal of the American Statistical Association*, **78**(382), 316–331.

Efron B (1986). "How Biased is the Apparent Error Rate of a Prediction Rule?" *Journal of the American Statistical Association*, **81**(394), 461–470.

Efron B, Tibshirani R (1997). "Improvements on Cross-Validation: The .632+ Bootstrap Method." *Journal of the American Statistical Association*, **92**(438), 548–560.

Efron B, Tibshirani RJ (1993). *An Introduction to the Bootstrap.* Chapman & Hall, New York.

Freund Y, Schapire RE (1996). "Experiments with a New Boosting Algorithm." In L Saitta (ed.), "Machine Learning: Proceedings of the Thirteenth International Conference," pp. 148–156. Morgan Kaufmann, San Francisco.

Friedman JH (1991). "Multivariate Adaptive Regression Splines." *The Annals of Statistics*, **19**(1), 1–67.

George EI (2000). "The Variable Selection Problem." *Journal of the American Statistical Association*, **95**(452), 1304–1308.

Gu C, Xiang D (2001). "Cross-Validating Non-Gaussian Data: Generalized Approximate Cross-Validation Revisited." *Journal of Computational and Graphical Statistics*, **10**(3), 581–591.

Hájek J, Šidák Z, Sen PK (1999). *Theory of Rank Tests*. Academic Press, London, 2nd edition.

Hastie T, Tibshirani R, Friedman J (2001). *The Elements of Statistical Learning (Data Mining, Inference and Prediction)*. Springer Verlag, New York.

Hochberg Y, Tamhane AC (1987). *Multiple Comparison Procedures*. John Wiley & Sons, New York.

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, Lausen B (2003). "Double-Bagging: Combining Classifiers by Bootstrap Aggregation." *Pattern Recognition*, **36**(6), 1303–1309.

Hothorn T, Lausen B (2005). "Bundling Classifiers by Bagging Trees." *Computational Statistics & Data Analysis*, **49**, 1068–1078.

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

Kim H, Loh WY (2003). "Classification Trees with Bivariate Linear Discriminant Node Models." *Journal of Computational and Graphical Statistics*, **12**(3), 512–530.

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

Liaw A, Wiener M (2002). "Classification and Regression by **randomForest**." *R News*, **2**(3), 18–22.

Lim TS, Loh WY, Shih YS (2000). "A Comparison of Prediction Accuracy, Complexity, and Training Time of Thirty-Three Old and New Classification Algorithms." *Machine Learning*, **40**(3), 203–228.

Meyer D (2001). "Support Vector Machines." *R News*, **1**(3), 23–26.

Meyer D, Leisch F, Hornik K (2003). "The Support Vector Machine under Test." *Neurocomputing*, **55**(1-2), 169–186.

Nadeau C, Bengio Y (2003). "Inference for the Generalization Error." *Machine Learning*, **52**(3), 239–281.

Patterson JG (1992). *Benchmarking Basics*. Crisp Publications Inc., Menlo Park, California.

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

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

Pittman J (2002). "Adaptive Splines and Genetic Algorithms." *Journal of Computational and Graphical Statistics*, **11**(3), 615–638.

Pizarro J, Guerrero E, Galindo PL (2002). "Multiple Comparison Procedures Applied to Model Selection." *Neurocomputing*, **48**, 155–173.

R Development Core Team (2004). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-00-3, URL http://www.R-project.org.

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

Schiavo RA, Hand DJ (2000). "Ten More Years of Error Rate Research." *International Statistical Review*, **68**(3), 295–310.

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

Therneau TM, Atkinson EJ (1997). "An Introduction to Recursive Partitioning using the `rpart` Routine." *Technical Report 61*, Section of Biostatistics, Mayo Clinic, Rochester. `http://www.mayo.edu/hsr/techrpt/61.pdf`.

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

Vehtari A, Lampinen J (2002). "Bayesian Model Assessment and Comparison Using Cross-Validation Predictive Densities." *Neural Computation*, **14**(10), 2439–2468.

Venables WN, Ripley BD (2002). *Modern Applied Statistics with S*. Springer, New York, 4th edition. `http://www.stats.ox.ac.uk/pub/MASS4/`.

Wolpert DH, Macready WG (1999). "An Efficient Method to Estimate Bagging's Generalization Error." *Machine Learning*, **35**(1), 41–51.

Corresponding author:

Torsten Hothorn
Institut für Medizininformatik, Biometrie und Epidemiologie
Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany
E-mail: `Torsten.Hothorn@rzmail.uni-erlangen.de`
URL: `http://www.imbe.med.uni-erlangen.de/~hothorn/`