Computational Statistics and Data Analysis 54 (2010) 2976–2989

Contents lists available at ScienceDirect

Computational Statistics and Data Analysis

journal homepage: www.elsevier.com/locate/csda

Measuring the prediction error. A comparison of cross-validation, bootstrap and covariance penalty methods

Simone Borra[a], Agostino Di Ciaccio[b,*]

[a] *Department of Economics and Territory, University of Rome "Tor Vergata", via Columbia 2, 00133, Italy*

[b] *Department of Statistics, Probability and Appl. Statistics, University of Rome "La Sapienza", P.le Aldo Moro 5, 00185, Italy*

ARTICLE INFO

Article history:
Received 1 December 2008
Received in revised form 12 February 2010
Accepted 9 March 2010
Available online 20 March 2010

Keywords:
Prediction error
Extra-sample error
In-sample error
Optimism
Cross-validation
Leave-one-out
Bootstrap
Covariance penalty
Regression trees
Projection pursuit regression
Neural networks

ABSTRACT

The estimators most widely used to evaluate the *prediction error* of a non-linear regression model are examined. An extensive simulation approach allowed the comparison of the performance of these estimators for different non-parametric methods, and with varying signal-to-noise ratio and sample size. Estimators based on resampling methods such as Leave-one-out, parametric and non-parametric Bootstrap, as well as repeated Cross Validation methods and Hold-out, were considered. The methods used are Regression Trees, Projection Pursuit Regression and Neural Networks. The repeated-corrected 10-fold Cross-Validation estimator and the Parametric Bootstrap estimator obtained the best performance in the simulations.

1. Introduction

Given a predictive statistical model that has been estimated on the available sample data, a fundamental problem in statistics is that of obtaining an accurate estimate of its *prediction error*, i.e. the expected loss of the estimated model on future observations. This task is particularly relevant when a large sample is not available, the underlying distribution is not known and an evaluation of the *prediction error* of a non-parametric model, which could overfit data, must be carried out.

This condition is quite common in Data Mining problems where the objective of the analysis is often to predict correctly the values of a target variable on newly observed cases. For this purpose, the model with the best prediction performance within a class of heterogeneous models, having different parametrizations, should be selected. This requires evaluating each model and ensuring comparability with other competing models. Consequently, we need a reliable prediction-error estimator which does not assume a specific model and which does not require strong hypotheses on the data generation process.

In the relevant literature, most studies regarding model evaluation refer to the choice of the best model in a fixed class (e.g., linear regression models). In this case, a precise assessment of model *prediction error* is not necessary because it is sufficient to use a "relative measure" for comparing the (usually nested) models to select, for example, the most important explanatory variables. In this respect, the most famous proposals are Mallows' C_p, Akaike's information criterion (AIC) and the Bayesian information criterion (BIC) (Mallows, 1973; Akaike, 1973; Schwarz, 1978) or Adaptive Model Selection, based on a concept of generalized degrees of freedom proposed by Shen and Ye (2002).

[*] Corresponding author. Tel.: +39 0649910709; fax: +39 064949241, +39 064959241.
E-mail address: agostino.diciaccio@uniroma1.it (A. Di Ciaccio).

0167-9473/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.csda.2010.03.004

Alternative approaches try to estimate directly the model prediction error and are based on cross-validation or bootstrap techniques. In the literature, the most relevant proposals using these approaches focus mainly on linear and near-linear models (see Shao, 1993; Zhang, 1993; Efron, 2004) or on the asymptotic properties of the estimators (Stone, 1977; Dudoit and van der Laan, 2005).

Using the *optimism* theorem (Efron, 1986), the problem of estimating the *prediction error* can be seen as the problem of estimating covariance-penalty terms. Recently, Efron (2004) proposed a new Parametric Bootstrap approach and new developments in the covariance-penalty approach for nonparametric regression and classification have been provided also by Zhang (2008) and Daudin and Mary-Huard (2008).

The performance of prediction error estimators was investigated through a simulation approach, by Molinaro et al. (2005), Kohavi (1995), Jiang and Simon (2007), Wehberg and Schumacher (2004), Kim (2009) and Wisnowski et al. (2003), mainly in a classification context. Moreover, these papers consider only specific data-sets and it is difficult to evaluate the relevance of these results for other data. To the best of our knowledge, this is the first large-scale study on the performance of *prediction error* estimators taking into consideration non-linear regression problems and a large number of estimators.

Our simulations are based on samples drawn from several data generating distributions, considering different levels of signal-to-noise ratio, overfitting, non-linearity of data and sample size. The comparison between the estimated and the true *prediction error*, calculated on a very large number of samples, allows us to investigate the performance of the estimators better than in the previous literature. The choice of three non-linear methods, Regression Trees, Projection Pursuit Regression and Feedforward Neural Networks, casts new light upon estimator performance, as it is known that the stability of the method significantly influences the performance of the estimators (Elisseeff and Pontil, 2003).

Section 2 explains the concepts of *extra-sample error*, *in-sample error*, *expected extra-sample error, expected in-sample error* and their relationships. Definitions of *optimism* and *expected optimism* are presented in Section 3. All these concepts are considered together, with the same mathematical notation, for the first time, allowing one to clarify their relationship. Section 4 gives an overview of *prediction error* estimators based on cross-validation, including the fairly unknown corrected cross-validation (Burman, 1989). In Section 5 we present the estimators based on non-parametric bootstrap, including the improvements 0.632 and 0.632+. An adaptation of Parametric Bootstrap (Efron, 2004) for non-parametric models is proposed in Section 6. In Sections 7–9 we show the simulation results and the evaluation measures applied to compare the estimators. Some final comments on the results are shown in Section 10.

2. The prediction error

Let $Y = f\left(X_1, X_2, \ldots, X_J\right) + \varepsilon$, where f is a general unknown function of the random covariates $\mathbf{X} = \left(X_1, X_2, \ldots, X_J\right)$ and ε is a random noise with zero mean and variance σ^2, independent of the covariates.

Let

$$\mu = f\left(x_1, x_2, \ldots, x_J\right) = E\left(Y|\mathbf{x}\right) \tag{1}$$

be the conditional expected value of Y given the values of the covariates. We can write the conditional distribution of each Y_i given $\mathbf{x}_i$ as $(Y_i|\mathbf{x}_i) \sim G_{\mu_i}$. Suppose that $\varepsilon \sim N(0; \sigma_\varepsilon^2)$, then we obtain

$$(Y_i|\mathbf{x}_i) \sim N\left(\mu_i; \sigma_\varepsilon^2\right). \tag{2}$$

Given a sample $\mathbf{s}$ we can estimate the function $f\left(\mathbf{X}\right)$ using the entire sample or, more generally, using a subset, the *training-set* $\mathbf{c} = \{y_i, \mathbf{x}_i\}_1^n \subseteq \mathbf{s}$, thus obtaining the estimate $\hat{f}_\mathbf{c}$. We also denote the training set by $\mathbf{c} = (\mathbf{y_c}, \mathbf{X_c})$, where $\mathbf{y_c}$ is the vector of observed Y_i and $\mathbf{X_c}$ is the matrix of the corresponding covariate values.

Given $\mathbf{c}$ and the estimated function $\hat{f}_\mathbf{c}$, we want to know its prediction capability for all the possible values of the covariates $\mathbf{X}$.

The *prediction error* of a fixed $\hat{f}_\mathbf{c}$ is given by

$$\mathrm{Err}\left(\hat{f}_\mathbf{c}\right) = E_\mathbf{X} E_{Y|\mathbf{X}}[L(Y, \hat{f}_\mathbf{c}(\mathbf{X}))|\hat{f}_\mathbf{c}, \mathbf{X}], \tag{3}$$

where $L\left(\cdot\right)$ is an appropriate loss-function. In the following we will denote the prediction error simply as Err.

The *expected prediction error* is obtained by averaging the prediction error $\mathrm{Err}(\hat{f}_\mathrm{c})$ with respect to all possible training samples:

$$\overline{\mathrm{Err}} = E_\mathbf{c}\left\{\mathrm{Err}\left(\hat{f}_\mathbf{c}\right)\right\}. \tag{4}$$

Note that $E_\mathbf{c}$ averages over training samples, not fixing $\hat{f}_\mathbf{c}$, while $E_\mathbf{X}$ and $E_{Y|\mathbf{X}}$ in (3) consider all possible values of $\mathbf{X}$ and Y for a fixed $\hat{f}_\mathbf{c}$. Err is also called the *generalization error*, and $\overline{\mathrm{Err}}$ the *average generalization error*.

If we want to obtain a measure of the prediction capability of a function $\hat{f}_\mathbf{c}$ estimated on the observed training set $\mathbf{c}$, we are more interested in Err. On the other hand, expectation (4) could be useful in obtaining a general evaluation of the given model, independently of a particular sample.

Expression (3) is also called *extra-sample error* (Efron and Tibshirani, 1997) because the average $E_\mathbf{X}$ refers to all possible values of $\mathbf{X}$.

Define the *expected prediction error* at the generic point $\mathbf{x}_o$ of $\mathbf{X}$ as

$$\overline{\mathrm{Err}}_{(\mathbf{x}_o)} = E_{\mathbf{c}}\left\{E_{Y_o}\left[L(Y_o, \hat{f}_{\mathbf{c}}(\mathbf{x}_o))|\hat{f}_{\mathbf{c}}\right]\right\}, \tag{5}$$

where Y_o is the random variable $(Y_o|\mathbf{x}_o) \sim N(\mu_o; \sigma_\varepsilon^2)$.

If we choose the quadratic loss function $L(Y_o, \hat{f}_{\mathbf{c}}(\mathbf{x}_o)) = (Y_o - \hat{f}_{\mathbf{c}}(\mathbf{x}_o))^2$ and denote $\hat{\mu}_{0\mathbf{c}} = \hat{f}_{\mathbf{c}}(\mathbf{x}_o)$ then

$$\begin{aligned}\overline{\mathrm{Err}}_{(\mathbf{x}_o)} &= E_{\mathbf{c}}E_{Y_o}\left(Y_o - \hat{\mu}_{o\mathbf{c}}\right)^2 = \sigma_\varepsilon^2 + \left(\mu_o - E_{\mathbf{c}}\left(\hat{\mu}_{o\mathbf{c}}\right)\right)^2 + E_{\mathbf{c}}\left(E_{\mathbf{c}}\left(\hat{\mu}_{o\mathbf{c}}\right) - \hat{\mu}_{o\mathbf{c}}\right)^2 \\ &= \sigma_\varepsilon^2 + \text{prediction bias}^2 + \text{prediction variance},\end{aligned} \tag{6}$$

where σ_ε^2 is the variance of the noise, sometimes called the *irreducible error*.

A more restrictive definition of the *prediction error* is the *in-sample error* of $\hat{f}_{\mathbf{c}}$, where the values of the covariates are considered fixed at their observed sample values $\mathbf{x}_i \in \mathbf{X}_{\mathbf{c}}$ while Y_i are random variables as defined in (2), thus

$$\mathrm{Err}_{\mathrm{in}} = \frac{1}{n}\sum_{i=1}^{n} E_{Y_i}\left[L(Y_i, \hat{f}_{\mathbf{c}}(\mathbf{x}_i))|\hat{f}_{\mathbf{c}}, \mathbf{X}_{\mathbf{c}}\right]. \tag{7}$$

This expression has been considered by several authors, mainly in a model selection approach (see e.g., Efron, 1986, 2004; Ye, 1998; Shen and Ye, 2002).

When considering the *in-sample error* approach, we can define the *expected in-sample prediction error* at the sample point $\mathbf{x}_i \in \mathbf{X}_{\mathbf{c}}$ as:

$$\overline{\mathrm{Err}}_{\mathrm{in}}(\mathbf{x}_i) = E_{Y_{\mathbf{c}}}\left\{E_{Y_i}\left[L(Y_i, \hat{f}_{\mathbf{c}}(\mathbf{x}_i))|\hat{f}_{\mathbf{c}}, \mathbf{X}_{\mathbf{c}}\right]\right\}. \tag{8}$$

Note that $E_{Y_{\mathbf{c}}}$ averages with respect to training samples which have fixed $\mathbf{X}_{\mathbf{c}}$ (the values of the covariates in the training set) but different $\mathbf{y}_{\mathbf{c}}$, then $\hat{f}_{\mathbf{c}}$ depends on $Y_{\mathbf{c}}$.

The general expression of the *expected in-sample prediction error* is obtained by averaging with respect to the training units:

$$\overline{\mathrm{Err}}_{\mathrm{in}} = \frac{1}{n}\sum_{i=1}^{n}\left\{\overline{\mathrm{Err}}_{\mathrm{in}}(\mathbf{x}_i)\right\}. \tag{9}$$

If we choose the quadratic loss function, expression (8) can be written as

$$\overline{\mathrm{Err}}_{\mathrm{in}}(\mathbf{x}_i) = \sigma_\varepsilon^2 + (\text{prediction bias}^2|\mathbf{X}_{\mathbf{c}}) + (\text{prediction variance}|\mathbf{X}_{\mathbf{c}}), \tag{10}$$

where the covariate values in the new training sets are fixed at their observed values $\mathbf{X}_{\mathbf{c}}$.

We expect that by increasing the size of the training sample, the difference between (3) and (7) will decrease because $\mathbf{X}_{\mathbf{c}}$ includes more $\mathbf{x}_i$ of the generating distribution.

If we fit by least-squares a linear function with p parameters, expression (10) can be written as (Hastie et al., 2001, page 197):

$$\overline{\mathrm{Err}}_{\mathrm{in}}(\mathbf{x}_i) = \sigma_\varepsilon^2 + (\mu_i - E_{Y_i}(\hat{\mu}_{i\mathbf{c}}))^2 + \|h(\mathbf{x}_i)\|^2\,\sigma_\varepsilon^2, \tag{11}$$

where $\mathbf{x}_i \in \mathbf{X}_{\mathbf{c}}$, $h(\mathbf{x}_i)$ is the vector of linear weights that produce the fit $\hat{\mu}_{i\mathbf{c}} = \hat{f}_{\mathbf{c}}(\mathbf{x}_i) = \mathbf{x}_i'\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{y} = h(\mathbf{x}_i)\,\mathbf{y}$, so $\mathrm{var}(\hat{f}_{\mathbf{c}}(\mathbf{x}_i)) = \|h(\mathbf{x}_i)\|^2\,\sigma_\varepsilon^2$. The average of this quantity with respect to the training data is $\frac{p}{n}\sigma_\varepsilon^2$. Finally we obtain:

$$\frac{1}{n}\sum_i \overline{\mathrm{Err}}_{\mathrm{in}}(\mathbf{x}_i) = \sigma_\varepsilon^2 + \frac{1}{n}\sum_i(\mu_i - E_{Y_i}(\hat{\mu}_{i\mathbf{c}}))^2 + \frac{p}{n}\sigma_\varepsilon^2. \tag{12}$$

From expressions (6), (10) and (12) we can observe the well-known bias-variance trade-off (Breiman, 1992): minimizing the bias may increase the variance and vice-versa.

In general, the use of overly complex models (overfitting) may produce low bias and high variability while the use of overly simple models (underfitting) may produce high bias and low variability. Specifically in (12), if the signal-to-noise ratio is high, overfitting should not be a matter of concern because the *expected in-sample prediction error* will be at most double of the error variance while underfitting could result in a much higher *prediction error*.

3. Apparent error and optimism

When the data distribution is unknown, we cannot calculate (3), (4), (7) or (8), so it is necessary to use an estimator.

The simplest estimator is the *apparent error* (AE) or *resubstitution error*, defined as

$$\mathrm{err} = \frac{1}{n}\sum_{i=1}^{n} L(y_i, \hat{f}_{\mathbf{c}}(\mathbf{x}_i)), \quad \mathbf{x}_i \in \mathbf{X}_{\mathbf{c}}\; y_i \in \mathbf{y}_{\mathbf{c}} \tag{13}$$

i.e. the average of the loss function on the training data-set $\mathbf{c}$.

The expression (13) can be calculated easily but it is well known that err is an optimistic estimator of both the *extra-sample error* Err and *in-sample error* $\mathrm{Err}_{\mathrm{in}}$, because it uses the same data for both training and evaluating of the model. Moreover, by using powerful non-linear methods such as Neural Networks, it is easy to obtain a very low value of the *apparent error* just by including more parameters in the model.

The *optimism* is usually defined as the difference between $\mathrm{Err}_{\mathrm{in}}$ and err on the training data, while the *expected optimism* is defined as

$$\overline{\mathrm{op}} = E_{Y_{\mathbf{c}}}\left[\mathrm{Err}_{\mathrm{in}} - \mathrm{err}|\mathbf{X}_{\mathbf{c}}\right], \tag{14}$$

that is, the difference between $\mathrm{Err}_{\mathrm{in}}$ and err, for the estimated $\hat{f}_{\mathbf{c}}$, on new training data, having random Y but fixing $\mathbf{X}$ at the observed sample values (Efron, 1986; Tibshirani and Knight, 1999).

Note that $\hat{f}_{\mathbf{c}}$ in (14) changes for each training sample, as in expression (8).

Expected optimism is typically positive, since the *apparent error* is usually biased downward as an estimate of the (*in-sample*) *prediction error*.

For a quadratic loss and a general function $f(\mathbf{X})$, from (14) we obtain (Efron, 1986):

$$E_{Y_{\mathbf{c}}}\left[\mathrm{Err}_{\mathrm{in}}\right] = E_{Y_{\mathbf{c}}}\left[\mathrm{err}\right] + \overline{\mathrm{op}} = E_{Y_{\mathbf{c}}}\left[\mathrm{err}\right] + \frac{2}{n}\sum_{i}\mathrm{cov}_{Y_{\mathbf{c}}}\left(\hat{\mu}_{i\mathbf{c}}, Y_i\right). \tag{15}$$

Note that, to simplify the notation, we dropped the conditioning on $\mathbf{X}_{\mathbf{c}}$.

This expression can be obtained by noting that:

$$E_{Y_{\mathbf{c}}}\left[\mathrm{Err}_{\mathrm{in}}\right] = E_{Y_{\mathbf{c}}}\left\{\frac{1}{n}\sum_{i=1}^{n}E_{Y_i}\left[Y_i - \hat{\mu}_{i\mathbf{c}}\right]^2\right\} = \sigma_{\varepsilon}^2 + \frac{1}{n}\sum_{i=1}^{n}E_{Y_{\mathbf{c}}}\left(\hat{\mu}_{i\mathbf{c}} - \mu_i\right)^2,$$

and

$$E_{Y_{\mathbf{c}}}\left[\mathrm{err}\right] = E_{Y_{\mathbf{c}}}\left[\frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{\mu}_{i\mathbf{c}}\right)^2\right] = \sigma_{\varepsilon}^2 + \frac{1}{n}\sum_{i=1}^{n}E_{Y_{\mathbf{c}}}\left(\hat{\mu}_{i\mathbf{c}} - \mu_i\right)^2 - \frac{2}{n}\sum_{i=1}^{n}E_{Y_{\mathbf{c}}}\left[(Y_i - \mu_i)\cdot\left(\hat{\mu}_{i\mathbf{c}} - \mu_i\right)\right],$$

so finally

$$\overline{\mathrm{op}} = \frac{2}{n}\sum_{i=1}^{n}E_{Y_{\mathbf{c}}}\left[(Y_i - \mu_i)\cdot\left(\hat{\mu}_{i\mathbf{c}} - \mu_i\right)\right] = \frac{2}{n}\sum_{i=1}^{n}\mathrm{cov}_{Y_{\mathbf{c}}}\left(Y_i, \hat{\mu}_{i\mathbf{c}}\right). \tag{16}$$

The covariance penalty term that is added to the *apparent error* in (15) represents the influence of the Y_i values on their estimates $\hat{\mu}_{i\mathbf{c}}$ in the sample $\mathbf{c}$.

Note that the covariance penalty theory can be generalized to a broad and important class of loss functions $L(\cdot)$, the Bregman q-class divergence. This class accounts for different types of dependent variables and includes the quadratic loss, misclassification loss and other popular loss functions (see Efron, 2004; Zhang, 2008).

The covariance term in (15) and (16) is also used by Ye (1998) to define an extension of the degrees of freedom as follow:

$$\mathrm{df} = \sum_{i}\mathrm{cov}_{\mathbf{Y_c}}\left(Y_i, \hat{\mu}_{i\mathbf{c}}\right)/\sigma_{\varepsilon}^2. \tag{17}$$

This index can be used to evaluate and compare general model structures such as decision trees.

4. Estimators based on cross-validation

In K-fold cross-validation (CV), we split the sample into K disjoint subsets $\boldsymbol{t}_h$ $(h = 1, 2, \ldots, K)$ of (approximately) equal size. We train the model K times, each time leaving out one of the subsets from the training, but using only the omitted subset to compute the *prediction error*. The mean of these K values is the CV-estimate of the *extra-sample error*.

Denote by $\mathbf{c}^h$ $(h = 1, 2, \ldots, K)$ the training set obtained by removing the h-th subset $\boldsymbol{t}_h$ and let $m = n/K$ be the number of units in each subset (assuming that n is a multiple of K). The CV-estimator is defined as the average error on the K analyses:

$$\mathrm{err}^{\mathrm{CV}} = \frac{1}{K}\sum_{h=1}^{K}\frac{1}{m}\sum_{j\in\boldsymbol{t}_h}L(y_j, \hat{f}_{\mathbf{c}^h}(\mathbf{x}_j)). \tag{18}$$

To simplify the notation, the expression $(y_j; \mathbf{x}_j) \in \boldsymbol{t}_h$ is denoted by $j \in \boldsymbol{t}_h$.

If K equals the training sample size n, we obtain the "Leave-One-Out" cross-validation (LOO). Therefore, LOO can be considered as a special case of K-fold cross-validation.

It is known that K-fold CV is a biased estimate of Err and that by increasing the number of folds we can reduce the bias (Kohavi, 1995). This is due to the fact that $\mathrm{err}^{\mathrm{CV}}$ is based on functions $\hat{f}_{\mathbf{c}^h}$ estimated on samples of size $(n - m)$, so it tends to overestimate Err.

We can interpret expression (18) also from another point of view. If we see the training sets $\mathbf{c}^1, \mathbf{c}^2, \ldots, \mathbf{c}^K$ as samples of size $(n - m)$, we are estimating $\hat{f}_{\mathbf{c}^h}$ on different samples, so $\mathrm{err}^{\mathrm{CV}}$ is an unbiased estimator of the expectation of Err for sample-size $(n - m)$. We can thus obtain an approximate estimate of $E_{\mathbf{c}}\left[\mathrm{Err}\left(\hat{f}_{\mathbf{c}}\right)\right] \approx E_{\mathbf{c}^{(n-m)}}\left[\mathrm{Err}\left(\hat{f}_{\mathbf{c}^{(n-m)}}\right)\right]$.

The main problem with K-fold CV is that the training-sets $\mathbf{c}^1, \mathbf{c}^2, \ldots, \mathbf{c}^K$ are not independent samples, i.e. they have $(n-2m)$ cases in common, and also the test sets $\boldsymbol{t}_h$ come from the same data. This implies that the variance of $\mathrm{err}^{\mathrm{CV}}$ may be very large (Breiman, 1996) and several authors (Dietterich, 1998; Bengio and Grandvalet, 2003) have considered the difficulties of estimating this variance by showing that no unbiased estimator of $\mathrm{Var}(\mathrm{err}^{\mathrm{CV}})$ can be obtained.

LOO is an almost unbiased estimator of Err, but it has high variability, producing non-reliable estimates (Efron, 1983; Stone, 1977). Furthermore, it has been shown that when using LOO for model selection, we obtain an inconsistent procedure in the sense that the probability of selecting the model with the best predictive ability does not converge to 1 as the size of the dataset tends to infinity. On the contrary, K-fold CV shows lower variability than LOO but may have a large bias. Moreover K-fold CV (or Repeated HO, see below), differently from LOO, is consistent in linear model selection under the condition that $\frac{K}{n} \to 1$ as $n \to \infty$ (Shao, 1993; Zhang, 1993).

Some authors have defined bounds which guarantee that the performance of the LOO estimator will never be significantly worse than the *apparent error* estimator under weak assumptions of error stability (Kearns and Ron, 1999).

Many simulation and empirical studies have verified that a reliable estimate of Err can be obtained with $K = 10$ for $n > 100$ as recommended by Davison and Hinkley (1997). For choosing subsets of inputs in linear regression, Breiman and Spector (1992) found 10-fold and 5-fold cross-validation to work better than Leave-one-out. The variance of LOO is generally larger than 10-fold CV and this could be due to the instability of the model that is used, as in the case of decision trees (Kohavi, 1995; Elisseeff and Pontil, 2003).

Burman (1989) showed that $E(\mathrm{err}^{\mathrm{CV}} - \mathrm{Err}) \approx s_0\,(K-1)^{-1}\,n^{-1}$. For $K = n$ the right-side term of the expression is $O(n^{-2})$, but when K is small this term is not necessarily very small. Moreover, the constant term s_0 is of the same order as the number of parameters being estimated and so CV may give a poor estimate of Err if the number of parameters is large. Therefore, Burman introduced a corrected version of cross-validation CV*:

$$\mathrm{err}^{\mathrm{CV}^*} = \mathrm{err}^{\mathrm{CV}} + \mathrm{err} - \bar{e}^{+}, \tag{19}$$

where $\bar{e}^{+} = \frac{1}{K}\sum_{h=1}^{K} e_h^{+}$ and the values e_h^{+} are obtained by considering $K-1$ folds for model estimation and, differently from CV, the whole sample for testing: $e_h^{+} = \frac{1}{n}\sum_{i=1}^{n} L\left(y_i, \hat{f}_{\mathbf{c}^h}(\mathbf{x}_i)\right)$.

We can write expression (19) as

$$\begin{aligned}\mathrm{err}^{\mathrm{CV}^*} &= \frac{1}{K}\sum_{h=1}^{K}\frac{1}{m}\sum_{j\in\boldsymbol{t}_h} L(y_j, \hat{f}_{\mathbf{c}^h}(\mathbf{x}_j)) + \frac{1}{n}\sum_{i=1}^{n} L(y_i, \hat{f}_{\mathbf{c}}(\mathbf{x}_i)) - \frac{1}{K}\sum_{h=1}^{K}\frac{1}{n}\sum_{i=1}^{n} L(y_i, \hat{f}_{\mathbf{c}^h}(\mathbf{x}_i)) \\ &= \frac{1}{K}\sum_{h=1}^{K}\frac{1}{m}\sum_{j\in\boldsymbol{t}_h} L(y_j, \hat{f}_{\mathbf{c}^h}(\mathbf{x}_j)) + \frac{1}{n}\sum_{i=1}^{n}\left[L(y_i, \hat{f}_{\mathbf{c}}(\mathbf{x}_i)) - \frac{1}{K}\sum_{h=1}^{K} L(y_i, \hat{f}_{\mathbf{c}^h}(\mathbf{x}_i))\right].\end{aligned} \tag{20}$$

The second term of (20) will usually be negative to compensate for the bias of $\mathrm{err}^{\mathrm{CV}}$.

The author showed that $E\left(\mathrm{err}^{\mathrm{CV}^*} - \mathrm{Err}\right) \approx s_1\,(K-1)^{-1}\,n^{-2}$ where the constant s_1 depends on the loss function $L(\cdot)$ and on the density distribution of the variables.

In short, err uses a function $\hat{f}_{\mathbf{c}}$ estimated on all data, while e_j^{+} uses functions estimated on $(n-m)$ data. Consequently, usually, $\mathrm{err} < e_j^{+}$, and the difference increases for larger amounts of overfitting, and thus, larger values of $\mathrm{err}^{\mathrm{CV}}$.

We can also write expression (19) as:

$$\mathrm{err}^{\mathrm{CV}^*} = \mathrm{err} + (\mathrm{err}^{\mathrm{CV}} - \bar{e}^{+}), \tag{21}$$

where $(\mathrm{err}^{\mathrm{CV}} - \bar{e}^{+})$ can be seen as an estimate of *optimism* referring to the *extra sample error*.

Let $\delta_{jh} = L(y_j, \hat{f}_{\mathbf{c}^h}(\mathbf{x}_j))$. We can write this estimate of *optimism* as:

$$\begin{aligned}\widehat{op}_E &= \frac{1}{K}\sum_{h=1}^{K}\frac{1}{m}\sum_{j\in\boldsymbol{t}_h}\delta_{jh} - \frac{1}{K}\sum_{h=1}^{K}\frac{1}{n}\sum_{j=1}^{n}\delta_{jh} = \frac{1}{K}\sum_{h=1}^{K}\left[\frac{n-m}{mn}\sum_{j\in\boldsymbol{t}_h}\delta_{jh} - \frac{m}{mn}\sum_{j\notin\boldsymbol{t}_h}\delta_{jh}\right] \\ &= \frac{n-m}{n}\left[\frac{1}{K}\sum_{h}\frac{1}{m}\sum_{j\in\boldsymbol{t}_h}\delta_{jh} - \frac{1}{K}\sum_{h}\frac{1}{n-m}\sum_{j\notin\boldsymbol{t}_h}\delta_{jh}\right] = \frac{K-1}{K}\left[\mathrm{err}^{\mathrm{CV}} - \overline{\mathrm{err}}^{(n-m)}\right].\end{aligned} \tag{22}$$

The first term in the square brackets is the mean function's fit to the K test sets, i.e. $\mathrm{err}^{\mathrm{CV}}$, and the second term is the mean function's fit to the K training sets, i.e. the average *apparent error* on the K data sets of size $n-m$.

The Hold-out estimator (HO) is similar to a cross-validation estimator. It is obtained by randomly splitting the sample into a training set to estimate the model and a test set to evaluate the *prediction error*. For a classification problem, Kearns (1997) gives a general bound on the error of the hold-out estimator considering the approximation rate (the accuracy with which the true function can be approximated as a function of the number of model parameters).

The values obtained by HO and, to a smaller extent CV, depend on the initial random partition of the sample. A method to reduce this dependence, thus obtaining a more reliable estimate, consists in repeating the procedure a small number of times with different random splits. The average value obtained is called the Repeated CV (RCV) or the Repeated Corrected CV (RCV*) (Burman, 1989; Molinaro et al., 2005), or the Repeated Hold-Out (RHO), also called the Monte Carlo Cross-Validation (Dudoit and van der Laan, 2005). Note that, if we split the sample into two subsamples of equal size, repeating the HO twice, and calculating the mean, we obtain an RHO estimator that is very similar to a 2-fold cross-validation.

Several authors have applied RCV, but there are no papers showing the real advantage of this approach (except, to some degree Burman, 1989).

5. Estimators based on non-parametric bootstrap

Instead of repeatedly analyzing subsets of the data as in RHO, it is possible to analyze bootstrap samples of data. Given a collection of M samples, $\{\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_M\}$ of size n, drawn with replacement from the training set, the model is estimated on each sample $\boldsymbol{d}_j$, and the corresponding estimated *prediction error*, denoted by err_j^{Bt}, is calculated by considering as a test sample the cases not included in $\boldsymbol{d}_j$.

The *Leave-one-out bootstrap* estimator of Err is defined as:

$$\mathrm{err}^{Bt} = \frac{1}{M}\sum_{j=1}^{M}\mathrm{err}_j^{Bt}. \tag{23}$$

The properties of this estimator, compared with cross-validation and Hold-out estimators, have been analysed in many theoretical and empirical studies (Efron and Tibshirani, 1997). This estimator is biased upwards, but is considered to have lower variability compared to Cross-validation or Hold-out estimators. In a model selection context, Shao (1996) showed that the bootstrap selection procedure is not consistent, but that by modifying the bootstrap sampling, the procedure can be made consistent.

To improve err^{Bt}, Efron (1983) proposed the 0.632 bootstrap estimator:

$$\mathrm{err}^{B632} = 0.632\cdot\mathrm{err}^{Bt} + 0.368\cdot\mathrm{err}, \tag{24}$$

designed to correct the upward bias in err^{Bt} by averaging it with the downwardly biased estimator err. The weights are based on the fact that bootstrap samples include approximately $0.632n$ cases of the training data. However, in all situations of severe overfitting, the estimator err^{B632} is downwardly biased because $\mathrm{err} = 0$. To avoid this problem, Efron and Tibshirani (1997) proposed a new estimator, the 0.632+ bootstrap estimator, err^{B632+}, designed to be a less-biased compromise between err and err^{Bt}. It assigns a greater weight to err^{Bt} when the amount of overfitting is large.

To calculate err^{B632+}, first the *no-information error*, γ, is introduced, which corresponds to the expected *prediction error* of model $\hat{f}(\mathbf{X})$ when Y and $\mathbf{X}$ are independent. Given $\hat{f}(\mathbf{X})$, an estimate of γ is obtained by considering the loss on all n^2 couples $(y_i, \mathbf{x}_j)$:

$$\hat{\gamma} = \frac{1}{n^2}\sum_{i,j}L(y_i, \hat{f}(\mathbf{x}_j)) \quad (i, j = 1, 2, \ldots, n). \tag{25}$$

Thus a relative overfitting rate is defined as:

$$\hat{R} = (\mathrm{err}^{Bt} - \mathrm{err})/\left(\hat{\gamma} - \mathrm{err}\right), \tag{26}$$

with its range forced to be between 0 (no overfitting) and 1 (high overfitting).

The 0.632+ bootstrap estimator is given by:

$$\mathrm{err}^{B632+} = \hat{w}\cdot\mathrm{err}^{Bt} + \left(1 - \hat{w}\right)\cdot\mathrm{err}, \tag{27}$$

where $\hat{w} = 0.632/(1 - 0.368\cdot\hat{R})$. The weight $\hat{w}$ ranges from 0.632 (no-overfitting) to 1 (severe overfitting).

6. Estimators based on parametric bootstrap and covariance penalties

As seen in Section 3, a way to evaluate the prediction capability of a model is to estimate its *optimism* and then add to that the *apparent error*. Many methods based on this idea have been proposed. Considering Gaussian linear models and squared error loss, it can be shown (Hastie et al., 2001, p. 203–206) that the well known AIC, BIC or Mallows' C_p estimators could also be interpreted as a weighted sum of apparent error and optimism. In this case, expression (15) can be seen as a generalization of Mallows C_p for non-parametric models.

In the literature, several methods have been proposed to estimate the *optimism*. When considering a nonlinear estimation rule $\hat{\boldsymbol{\mu}} = g(\mathbf{y})$, if $g(.)$ is a smoother function for a homoskedastic Gaussian model, under differentiability conditions, it is possible to use Stein's unbiased risk estimate (Stein, 1981). Other authors have proposed estimating the covariance term by bootstrap.

Efron (2004) introduced a Parametric Bootstrap procedure (PB) consisting of the following steps:

1. Supposing that $y \sim N(\boldsymbol{\mu}, \sigma^2\mathbf{I})$, calculate $\hat{\boldsymbol{\mu}} = g(\mathbf{y})$ and $\hat{\sigma}^2$, by applying a model that is sufficiently complex to have negligible bias.

2. From the bootstrap density $\hat{\mathbf{f}} = N(\hat{\boldsymbol{\mu}}, \hat{\sigma}^2\mathbf{I})$ generate, for each $\mathbf{x}_i$ of the data-set, B new values y_i^{*b} and calculate $\hat{\mu}_i^{*b} = g(y_i^{*b})$.
3. Estimate $\mathrm{cov}_{\mathbf{y}}\left(y_i, \hat{\mu}_i\right)$ by

$$\widehat{\mathrm{cov}}_i = \sum_{b=1}^{B} \hat{\mu}_i^{*b}\left(y_i^{*b} - \bar{y}_i^*\right)/(B-1) \quad \text{with } \bar{y}_i^* = \sum_b \frac{y_i^{*b}}{B}. \tag{28}$$

4. Finally, the estimator is given by:

$$\mathrm{err}^{\mathrm{PB}} = \mathrm{err} + \frac{2}{n}\sum_i \widehat{\mathrm{cov}}_i. \tag{29}$$

To adapt Efron's proposal of Parametric Bootstrap for a complex non-parametric model (e.g., Neural Networks), we need a two-step procedure.

In the first step we need to obtain good estimates of σ^2 and $\boldsymbol{\mu}$. In fact, Efron's proposal to use a "big" model is not applicable when considering models capable of fitting random noise: in this case, it is possible to obtain $\hat{\boldsymbol{\mu}} = \mathbf{y_c}$ and $\hat{\sigma}^2 = 0$.

To overcome this problem, in the first step of our simulation we applied a non-parametric bootstrap procedure in order to obtain preliminary estimates of σ^2 and $\boldsymbol{\mu}$. Once we have obtained these estimates, we can define the bootstrap density $\hat{\mathbf{f}} = N(\hat{\boldsymbol{\mu}}, \hat{\sigma}^2\mathbf{I})$ and then calculate formula (29).

7. The Monte Carlo experiments

We considered two different experiments, illustrated in Sections 8 and 9, with one dependent variable Y and four (simulation A) or five (simulation B) covariates. In both cases, the values of the main parameters were drawn at random, thus obtaining hundreds of different data generating distributions. Finally, from each distribution we drew 30 samples.

To estimate the regression function f we chose three different methods:

- Regression Trees (TREE) as implemented in CART (Breiman et al., 1984). It can fit non-linear data, but is quite unstable and if it has a large number of leaves, it can overfit data. To control the tree size, we did not use pruning but fixed the minimum number of observations for a split (10 for sample size 120, and 20 for sample size 300 or 500).
- Projection Pursuit Regression (PPR), as proposed by Friedman and Stuetzle (1981). We used the SMART algorithm (Friedman, 1985). Given a sufficient number of parameters, PPR can approximate any continuous function, but for a moderate number, it generates quite stable and smooth functions. We fixed the number of terms at 3, as PPR performed adequately for this choice in a pre-simulation analysis.
- Neural Networks (NEU). We considered a simple feedforward Neural Network containing one hidden layer with 6 nodes, which appeared to be adequate for our data.

Our aim is to compare the estimators of Err by carrying out simulations with different sample sizes: 120, 300 and 500. To denote a method, e.g., Regression Tree, applied to the sample size 500, we will use the synthetic notation TREE 500. The loss function considered in these simulations is the usual squared error function (see Section 2).

Given a data generating distribution, we carried out the following steps for each sample:

1. Estimation of the regression function $f(\mathbf{X})$ by TREE, PPR and NEU on all the units of the sample.
2. Using the estimated function, we calculate the predicted values of Y on a very large test-set (50,000 units). Comparing the predicted with the true values of Y, we obtain, with a good approximation, the *extra-sample error* of $\hat{f}_{\mathbf{c}}$.
3. Calculation of all the prediction error estimators described in the previous sections, using only the training data.

In particular, in order to calculate the *in-sample error* (7), for each unit of the training sample we generated 300 new response values y_{is} $(s = 1, 2, \ldots, 300)$ by $N\left(\mu_i, \sigma_\varepsilon^2\mathbf{I}\right)$, with $\mu_i = f(\mathbf{x}_i)$ and σ_ε^2 is the variance of the noise. To estimate the covariance penalties from the sample, first of all we obtained $\hat{\mu}_i$ and $\hat{\sigma}_\varepsilon^2$ by a nonparametric bootstrap, then we drew the new response values from $N\left(\hat{\mu}_i, \hat{\sigma}_\varepsilon^2\mathbf{I}\right)$. Finally we applied expression (28).

For each distribution, we computed the value:

$$v_h = \sqrt{\frac{1}{T}\sum_{t=1}^{T}\left(\widehat{\mathrm{Err}}_{th} - \mathrm{Err}_{th}\right)^2} \quad (h = 1, 2, \ldots, H), \tag{30}$$

where $T = 30$, the number of samples in each distribution, H is the number of distributions generated and $\widehat{\mathrm{Err}}_{th}$ is the generic estimator of Err_{th} in the t-th sample, drawn from the h-th distribution.

We also calculated the mean of *extra-sample error* in a fixed distribution by:

$$\overline{\mathrm{Err}}_h = \frac{1}{T}\sum_{t=1}^{T}\mathrm{Err}_{th}. \tag{31}$$

Thus we have

$$\varphi_h = \text{std}\left(\text{Err}_{th}\right)_h = \sqrt{\frac{1}{T}\sum_{t=1}^{T}\left(\text{Err}_{th} - \overline{\text{Err}_h}\right)^2}. \tag{32}$$

In most cases, we should obtain $\nu_h \geq \varphi_h$.

A relative measure of error for the h-th distribution is

$$\text{rse}_h = \frac{\nu_h}{\varphi_h}. \tag{33}$$

An overall measure of performance on all distributions can be obtained by

$$\overline{\text{rse}} = \frac{1}{H}\sum_{h=1}^{H}\frac{\nu_h}{\varphi_h}. \tag{34}$$

If the estimator is unbiased and has low variability, then rse_h will be small. Its mean $\overline{\text{rse}}$ allows us to evaluate the average performance of an estimator on a large number of different distributions.

As summary measures of the amount of bias of the methods on all the distributions, we computed the following indices:

$$\text{rb}_h = \frac{1}{T}\sum_{t=1}^{T}\frac{\widehat{\text{Err}}_{th} - \text{Err}_{th}}{\widehat{\text{Err}}_{th} + \text{Err}_{th}} \quad \text{(relative bias for the } h\text{-th distribution)}; \tag{35}$$

$$\overline{\text{arb}} = \frac{1}{H}\sum_{h=1}^{H}\left|\text{rb}_h\right| \quad \text{(mean absolute relative bias)}. \tag{36}$$

An unbiased estimator should have $\text{rb}_h \cong 0$ for each h, obtaining a small value of $\overline{\text{arb}}$.

Finally, to be sure of the interpretation of the previous indices and, in particular, of $\overline{\text{rse}}$ as a global measure of performance, we also computed an "estimators' mean rank": for each sample, we ranked the estimators on the strength of their ability to predict Err; the mean of these thousands of ranks is easily interpretable in order to compare the estimators.

The estimators considered in the simulations are:

- err, the resubstitution estimator;
- err^{RCV}, the Repeated 10-fold Cross Validation estimator, with 10 random starts;
- $\text{err}^{\text{RCV}^*}$, the same as err^{RCV} but with Burman's correction;
- err^{RHO}, the Repeated Hold-Out estimator, with 100 random splits;
- err^{LOO}, the Leave-one-out Cross Validation estimator;
- err^{B632}, the Bootstrap 0.632 estimator, with 100 bootstrap samples;
- $\text{err}^{\text{B632+}}$, the Bootstrap 0.632+ estimator, with 100 bootstrap samples;
- err^{PB}, the Parametric Bootstrap estimator, with 100 bootstrap samples.

All the previous estimators require similar computational times, except for err (very quick) and err^{LOO} (very slow for a large sample size). Moreover err^{PB} also required a preliminary estimation of the variance of the noise by a nonparametric bootstrap (see Section 6).

8. Simulation A

Given five variables Y, X_1, X_2, X_3, X_4, we considered 1000 data generating distributions in which $Y = f(X_1, X_2, X_3, X_4) + \varepsilon$ and f is the non-linear function

$$f(\mathbf{X}) = a\left(c_0 + \sum_{j=1}^{4} c_j X_j\right) + \sum_{j=1}^{4}\beta_j\left(X_j - \bar{X}_j\right)^2, \tag{37}$$

where

- X_j $(j = 1, \ldots, 4)$ is the j-th explanatory variable generated from a Beta distribution with parameters (g_j, t_j) drawn from a Uniform distribution in the interval [2, 10];
- a, c_0, c_j and β_j $(j = 1, \ldots, 4)$ are randomly drawn from Uniform distributions in the intervals [0, 4], [−5, 5], [−5, 5] and [−50, 50], respectively;
- ε is generated from a Gaussian distribution $N(0, \sigma_\varepsilon^2)$.

The above definitions allow us to generate $f(.)$ with different levels of non-linearity. To produce high generality in the results, the systematic component of the model (the signal) has been scaled so that its standard deviation, σ_s, is equal to a random value drawn from a Uniform distribution in the interval [10, 30]. We then randomly drew the signal-to-noise ratio $r_{s/n}$ (between 0.5 and 4.5). The standard deviation of the noise, σ_ε, was then fixed as $\sigma_\varepsilon = \sigma_s / r_{s/n}$.

Table 1
Extra-sample and *apparent error*. Mean over all generating distributions with respect to method and sample-size.

	Sample size	Mean of *apparent error*	Mean of *extra-sample error*	Ratio
TREE	120	91.37	462.37	0.208
	500	125.97	340.14	0.384
PPR	120	141.33	395.22	0.358
	500	209.78	262.54	0.799
NEU	120	189.36	444.31	0.435
	500	121.53	151.98	0.810

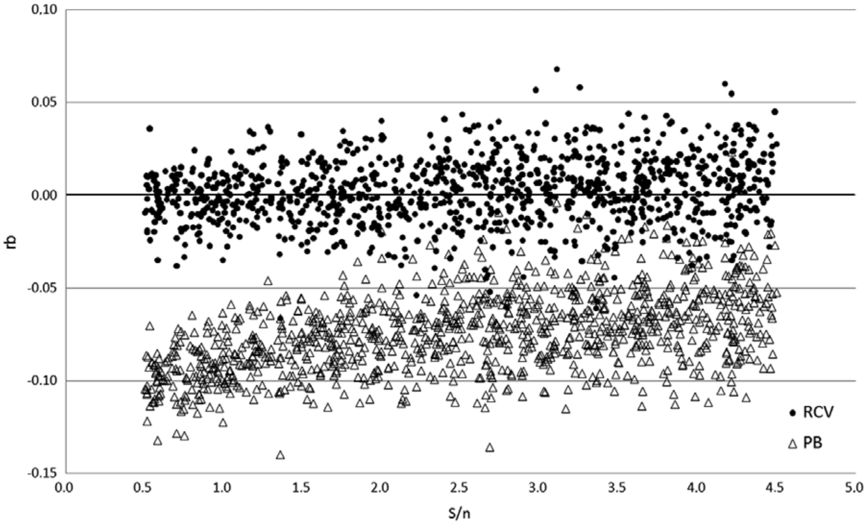

Fig. 1. Plot of rb vs. the S/n ratio. TREE with sample size $n = 120$ (RCV, PB).

Several simulations were carried out as follows. One thousand data generating distributions were obtained, first drawing the values of a, c_0, c_j, β_j, g_j, h_j, σ_s and $r_{s/n}$; then, conditionally on these, creating the large test-set drawing 50,000 values for each X_j ($j = 1, \dots, 4$) and for ε, and finally computing Y.

For each data-generating distribution, in the same way, we extracted 30 independent samples of size 120 or 500.

It is useful to start with a general evaluation of the performance of the methods for the two sample sizes. We can see from Table 1 that, for the smaller sample size, all methods overfit. Indeed, the *apparent error* is much lower than the *extra-sample error*. This is also true for TREE 500.

It should be noted that the presence of overfitting does not necessarily imply that we should use a simpler method; for example, a tree with a smaller number of nodes. If there is a high level of signal compared to error, simplifying the method could deteriorate its predictive power.

Let us now consider the indices defined in the previous section. In the Figs. 1–4 we can see the values of rb_h (35) for RCV, RCV*, PB, B632, B632+, LOO, computed as the mean of 30 samples for each of the 1000 generating distributions. Note the strong dependence of PB, B632, B632+ on the S/n ratio. In the presence of severe overfitting and unstable methods (Fig. 2: TREE 120), we can see the remarkable improvement on B632 obtained by B632+ for small values of the S/n ratio. However, for the other methods, B632+ was worse than B632. Note that, we did not observe any remarkable influence of the level of linearity on the performance of the estimator.

For the more stable method PPR with the larger sample, we obtained the results shown in Figs. 3 and 4. Note the good performance of RCV*, PB and LOO with respect to the biased B632.

In Tables 2 and 3, we show for each estimator the value of $\overline{\text{arb}}$, measuring the average relative bias over all the distributions, and the value of $\overline{\text{rse}}$, which is an overall performance measure. A good estimator should have small values of $\overline{\text{arb}}$ and $\overline{\text{rse}}$. The results are shown for both sample sizes. Note that LOO was not calculated for Neural Networks, due to the high computational resources needed.

These tables clearly reveal the good performance of the repeated cross-validation estimators. In particular, RCV* appears to be the estimator with the smallest bias, except for the overfitting case (TREE). Moreover, note the good behaviour of the Parametric Bootstrap in absence of overfitting. The small bias of PB can also be seen in Fig. 3, while in Fig. 4 we can see the large bias and variability of B632, which therefore obtains a high value of $\overline{\text{rse}}$ (1.773) in Table 3.

In Table 4 we ranked, for each sample, all the estimates with respect to their distance from the true *extra sample error*. Each value in the table represents the mean rank on 30,000 samples. These ranks confirm the results shown in Table 3 and the validity of the indices $\overline{\text{rse}}$ and $\overline{\text{arb}}$.

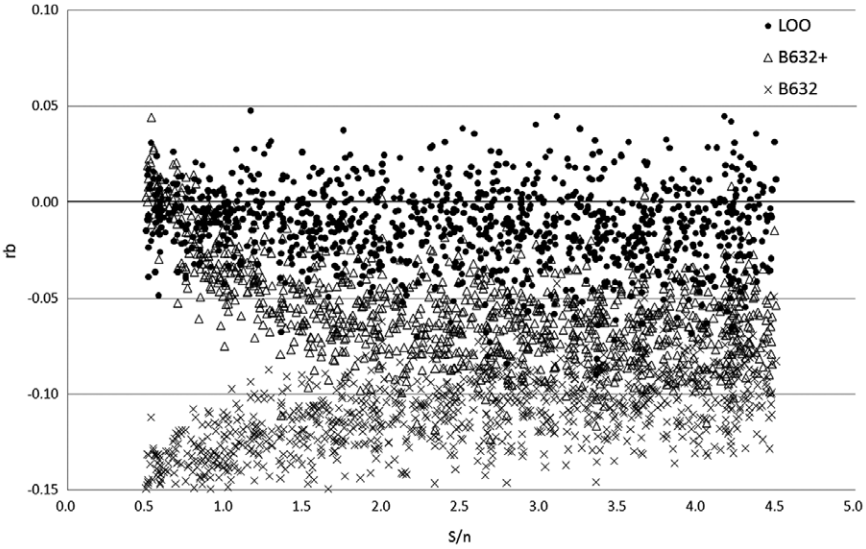

Fig. 2. Plot of rb vs. the S/n ratio. TREE with sample size $n = 120$ (B632, B632+, LOO).

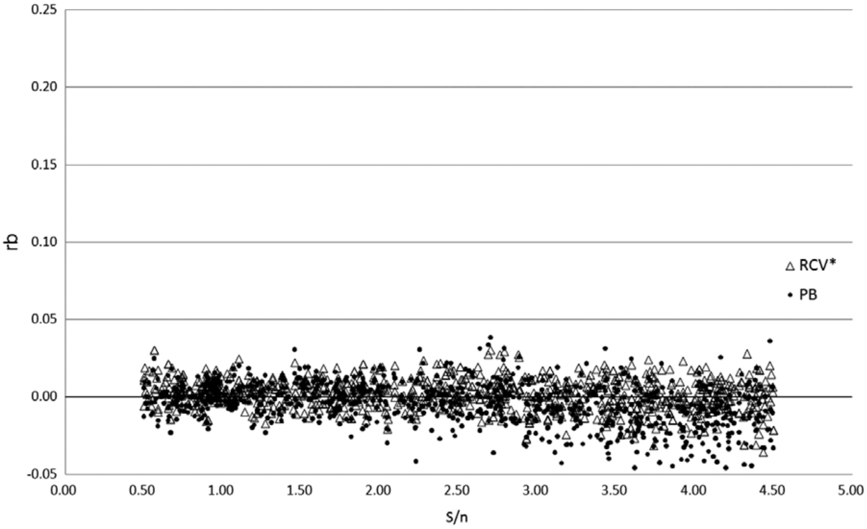

Fig. 3. Plot of rb vs. the S/n ratio. PPR with sample size $n = 500$ (RCV*, PB).

Table 2
Mean absolute relative bias $\overline{\text{arb}}$. Mean over all generating distributions with respect to estimator, method and sample-size.

$\overline{\text{arb}}$	TREE		PPR		NEU	
	Sample size		Sample size		Sample size	
	120	500	120	500	120	500
RCV	**0.014**	**0.010**	0.030	0.019	0.079	0.023
RCV*	0.043	0.030	**0.022**	**0.008**	**0.036**	**0.010**
PB	0.074	0.045	0.062	0.012	0.057	**0.010**
B632	0.109	0.080	0.124	0.074	0.057	0.021
B632+	0.056	0.055	0.182	0.082	0.080	0.022
LOO	**0.018**	**0.008**	**0.023**	0.013	–	–
RHO	0.037	0.034	0.122	0.045	0.210	0.071

Note that B632+ is always worse than B632, but in the case of strong overfitting (TREE), then we should be cautious about using the latter.

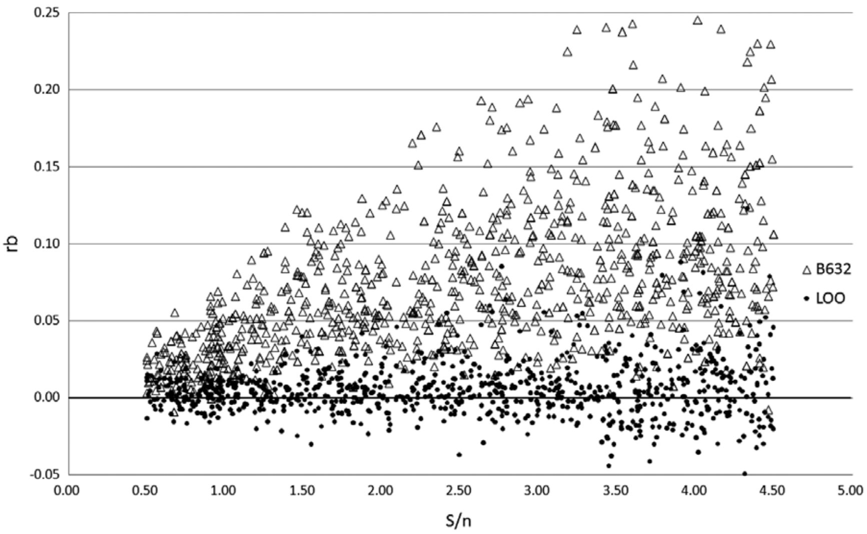

Fig. 4. Plot of rb vs. the S/n ratio. PPR with sample size $n = 500$ (B632, LOO).

Table 3
Mean relative root squared error $\overline{\text{rse}}$. Mean over all generating distributions with respect to estimator, method and sample-size.

$\overline{\text{rse}}$	TREE		PPR		NEU	
	Sample size		Sample size		Sample size	
	120	500	120	500	120	500
RCV	**1.658**	**1.828**	1.181	1.315	1.271	1.271
RCV*	1.733	2.149	**1.091**	**0.962**	**0.937**	0.829
PB	1.956	2.485	1.183	**0.945**	**0.948**	**0.780**
B632	2.286	3.486	1.563	1.773	1.051	1.054
B632+	1.732	2.669	2.185	1.961	1.206	1.079
LOO	1.856	2.038	1.260	1.323	–	–
RHO	1.807	2.341	1.674	1.562	2.182	1.947

Table 4
The estimators' comparative performance: estimator mean ranks. Lower values indicate better estimators.

	TREE		PPR		NEU	
	Sample size		Sample size		Sample size	
	120	500	120	500	120	500
RCV	**2.87**	**2.63**	3.21	3.54	3.73	3.93
RCV*	3.16	**2.87**	**2.89**	**2.74**	**2.63**	2.64
PB	3.67	3.40	3.30	**2.75**	**2.71**	**2.45**
B632	4.88	5.42	3.99	4.52	3.09	3.16
B632+	3.13	3.86	5.14	5.39	3.68	3.60
LOO	3.30	**2.82**	3.35	3.50	–	–
AE	6.99	7.00	6.11	5.57	5.17	5.22

9. Simulation B

In Section 8 we used a quadratic function, fixing randomly the coefficients to obtain several levels of non-linearity. Now consider the following case (inspired by Friedman, 1991):

$$Y = 10\sin(\pi X_1 X_2) + 20\left(X_3 - \frac{1}{2}\right)^2 + 0.2\mathrm{e}^{4X_4} + 5X_5 + \varepsilon, \tag{38}$$

where the five covariates were randomly generated from a multivariate uniform distribution. To make the simulation more realistic we considered dependent covariates and three levels of signal to noise ratio. In particular, $S/n = 1$, i.e. the variability of Y is due to signal and noise in equal proportion, $S/n = 2.5$, i.e. 86.2% of the variability of Y is due to the signal, $S/n = 5$, i.e. 96.2% of the variability of Y is due to the signal. We analysed 200 generating distributions for each S/n ratio and for each method, where each distribution has a different random covariance matrix. These multiple distributions were generated by using copulas (Nelsen, 2006). As in the simulation A, from each distribution we drew 30 samples (size = 300) and a large test set (size = 50,000), while ε is generated from a Gaussian distribution $N(0, \sigma_\varepsilon^2)$.

Table 5
Extra-sample and apparent error. Mean over all generating distributions with respect to method and S/n ratio.

	S/n	Mean of *apparent error*	Mean of *extra-sample error*	Ratio
	1	13.88	38.14	0.36
TREE	2.5	4.43	11.40	0.39
	5	3.08	7.84	0.39
	1	21.43	33.05	0.65
PPR	2.5	5.89	8.87	0.66
	5	3.39	5.08	0.67
	1	24.38	29.98	0.81
NEU	2.5	5.08	6.72	0.76
	5	2.18	2.92	0.75

Table 6
Mean absolute relative bias $\overline{\text{arb}}$. Mean over all generating distributions with respect to estimator, method and S/n ratio.

$\overline{\text{arb}}$	TREE			PPR			NEU		
	S/n ratio			S/n ratio			S/nratio		
	1	2.5	5	1	2.5	5	1	2.5	5
RCV	**0.008**	**0.013**	0.019	0.014	0.028	0.045	0.016	0.031	0.047
RCV*	0.030	0.025	0.020	**0.009**	**0.013**	**0.021**	**0.012**	**0.012**	**0.017**
PB	0.044	0.039	0.039	0.015	0.024	0.053	0.016	0.018	**0.017**
B632	0.082	0.071	0.068	0.014	0.084	0.148	0.015	0.025	0.046
B632+	0.023	0.049	0.051	0.047	0.099	0.160	0.022	0.029	0.049
LOO	**0.008**	**0.008**	**0.008**	**0.009**	0.020	0.037	–	–	–
RHO	0.013	0.043	0.062	0.048	0.081	0.115	0.043	0.080	0.089

Table 7
Mean relative root squared error $\overline{\text{rse}}$. Mean over all generating distributions with respect to estimator, method and S/n ratio.

$\overline{\text{rse}}$	TREE			PPR			NEU		
	S/n ratio			S/n ratio			S/nratio		
	1	2.5	5	1	2.5	5	1	2.5	5
RCV	**1.64**	**1.72**	**1.72**	**1.62**	1.12	1.03	1.74	1.20	1.10
RCV*	1.96	1.84	**1.66**	1.73	**0.79**	**0.57**	1.68	**0.75**	**0.53**
PB	2.20	1.99	1.82	1.76	**0.83**	**0.67**	**1.54**	**0.72**	**0.53**
B632	3.07	2.65	2.36	**1.65**	1.43	1.44	1.76	0.99	0.87
B632+	1.74	2.11	2.00	2.34	1.63	1.57	2.18	1.04	0.89
LOO	1.96	1.98	1.91	1.71	1.13	1.04	–	–	–
RHO	**1.60**	2.21	2.60	2.31	1.52	1.34	2.41	1.59	1.25

Table 8
The estimators' comparative performance: estimator mean ranks. Lower values indicate better estimators.

Mean ranks	TREE			PPR			NEU		
	S/n ratio			S/n ratio			S/n ratio		
	1	2.5	5	1	2.5	5	1	2.5	5
RCV	**3.25**	**3.19**	**3.28**	**3.41**	3.84	4.06	3.69	4.37	4.78
RCV*	3.83	3.38	**3.20**	3.79	**2.84**	**2.49**	3.53	**2.85**	**2.56**
PB	4.31	3.68	3.54	3.85	3.04	**2.97**	**3.49**	**2.84**	**2.56**
B632	6.09	5.71	5.43	3.52	4.76	5.37	**3.39**	3.40	3.63
B632+	3.38	4.08	4.03	5.01	5.84	6.31	3.84	3.70	3.97
LOO	3.81	3.63	3.61	3.72	3.87	4.05	–	–	–
RHO	3.32	4.32	4.92	5.06	5.21	5.14	4.63	5.20	5.29
AE	8.00	8.00	8.00	7.65	6.59	5.61	5.43	5.65	5.21

Table 5 shows the relevant overfitting of TREE, without pruning, as we observed also in Table 1. Tables 6–8 correspond to the Tables 2–4 of the previous simulation, but, in this case, we have highlighted the effect of the S/n ratio. By analysing these tables, we can substantially confirm all the previous considerations, with some differences only for a very low signal ($S/n = 1$). In particular, B632 appears interesting only with a weak signal and absence of overfitting.

10. Conclusions

We have presented, in a unified way, several approaches to estimate the *prediction error* of a non-linear model. Then, using extensive simulations, we have analyzed the performance of several *prediction error* estimators. In particular we have highlighted the effect of the S/n ratio and the model overfitting level, which has not been considered in the literature before. The most remarkable result is that the quite unknown repeated-corrected 10-fold cross-validation RCV* proposed by Burman (1989), often outperformed, on our data, the other estimators. The only estimator which appeared to be competitive is the Parametric Bootstrap (Efron, 2004), especially with large samples and stable methods. In fact, PB is an estimator of *in-sample error*, which is very close to *extra-sample error* for large sample size.

In our simulations, a poor performance appears evident for the estimators based on non-parametric bootstrap, B632 and B632+, or on a repeated Hold-Out RHO, except in the case of a small S/n ratio. Unsatisfactory results were also obtained by the leave-one-out CV. Despite the substantial computational resources required, LOO appears to be quite unbiased but it is never the most convenient estimator to use due to its high variability. It could be argued that B632 and B632+ would have a better performance with a larger number of bootstrap sub-samples, but this would also be true for Parametric Bootstrap. Moreover, by increasing the number of repetitions, also the other estimators RCV, RCV*, RHO should have a better performance. In any case, B632+ outperformed B632 only in the case of severe overfitting.

The positive performance of RCV on Regression Trees is due, paradoxically, to a known defect of K-fold cross-validation which tends to overestimate the *extra-sample error*. Since Regression Trees highly overfit our data and all estimators, in this case, tend to underestimate Err, this behavior makes RCV the least biased estimator.

The S/n ratio has an evident effect on the performance of some estimators, in particular for a low signal level. In this situation, RCV* and PB are less competitive.

Finally it should be noted that when analyzing non-Normal heteroskedastic data, that is relaxing the assumption $\varepsilon \sim N(0, \sigma_\varepsilon^2)$ made in our simulations, the performance of Parametric Bootstrap downgrades more than for the other estimators (Borra and Di Ciaccio, 2008).

Acknowledgements

The authors wish to thank the anonymous referees for their valuable comments and suggestions.

References

Akaike, H., 1973. Information theory and an extension of the maximum likelihood principle, second intern. Symposium on Information Theory 267–281.

Bengio, Y., Grandvalet, Y., 2003. No unbiased estimator of the variance of K-fold cross-validation. Journal of Machine Learning Research 5, 1089–1105.

Borra, S., Di Ciaccio, A., 2008. Estimators of extra-sample error for non-parametric methods. A comparison based on extensive simulations. Tech. Rep. 2008/19. Dept. of Statistics, Prob. and Appl. Statistics, Univ. of Roma La Sapienza.

Breiman, L., 1992. The little bootstrap and other methods for dimensionality selection in regression: x-fixed prediction error. Journal of American Statistical Association 87 (419), 738–754.

Breiman, L., 1996. Heuristic of instability and stabilization in model selection. The Annals of Statistics 24 (6), 2350–2383.

Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J., 1984. Classification and Regression Trees. Wadsworth and Brooks-Cole, Monterey, New York.

Breiman, L., Spector, P., 1992. Submodel selection and evaluation in regression: the X-random case. International Statistical Review 60, 291–319.

Burman, P., 1989. A comparative study of ordinary cross-validation, v-fold cross-validation and repeated learning-testing methods. Biometrika 76 (3), 503–514.

Daudin, J.J., Mary-Huard, T., 2008. Estimation of the conditional risk in classification: the swapping method. Computational Statistics & Data Analysis 52, 3220–3232.

Davison, A.C., Hinkley, D.V., 1997. Bootstrap Methods and their Applications. Cambridge University Press, Cambridge.

Dietterich, T.G., 1998. Approximate statistical tests for comparing supervised classification learning algorithms. Neural Computation 10 (7), 1895–1924.

Dudoit, S., van der Laan, M., 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: improvement on cross-validation. Journal of American Statistical Association 78, 316–331.

Efron, B., 1986. How biased is the apparent error rate of a prediction rule? Journal of American Statistical Association 81, 461–470.

Efron, B., 2004. The estimation of prediction error: covariance penalties and cross-validation. Journal of American Statistical Association 99 (467), 619–632.

Efron, B., Tibshirani, R., 1997. Improvements on cross-validation: the 0.632+ bootstrap method. Journal of American Statistical Association 92 (438).

Elisseeff, A., Pontil, M., 2003. Leave-one-out error and stability of learning algorithms with applications. In: Suykens, , et al. (Eds.), Advanced in Learning Theory: Methods, Models and Applications. In: NATO Science Series III: Computer and Systems Sciences, vol. 190. IOS Press.

Friedman, J.H., 1985. SMART user's guide. Dept. of Statistics. Technical Report LCS 1. Stanford University.

Friedman, J.H., Stuetzle, W., 1981. Projection pursuit regression. Journal of American Statistical Association 76, 817–823.

Friedman, J.H, 1991. Multivariate adaptive regression splines. Annals of Statistics 19, 1–67.

Hastie, T., Tibshirani, R., Friedman, J.H., 2001. The Elements of Statistical Learning: Data Mining, Inference and Prediction, 1st ed. Springer-Verlag, New York.

Jiang, W., Simon, R., 2007. A comparison of bootstrap methods and an adjusted bootstrap approach for estimating the prediction error in microarray classification. Statistics in Medicine 26, 5320–5334.

Kearns, M., 1997. A bound on the error of cross validation using the approximation and estimation rates, with consequences for the training-test split. Neural Computation 9 (5), 1143–1161.

Kearns, M., Ron, D., 1999. Algorithmic stability and sanity-check bounds for leave one-out cross-validation. Neural Computation 11 (6), 1427–1453.

Kim, J., 2009. Estimating classification error rate: repeated cross-validation, repeated hold-out and bootstrap. Computational Statistics & Data Analysis 53 (11), 3735–3745.

Kohavi, R., 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. International Joint Conference on Artificial Intelligence 2 (12), 1137–1143.

Mallows, C., 1973. Some comments on *Cp*. Technometrics 15, 661–675.

Molinaro, A.M., Simon, R., Pfeiffer, R.M., 2005. Prediction error estimation: a comparison of resampling methods. Bioinformatics 21 (15), 3301–3307.

Nelsen, R.B., 2006. An Introduction to Copulas, 2nd ed. In: Springer Series in Statistics, Springer, New York.

Schwarz, G., 1978. Estimating the dimension of a model. The Annals of Statistics 6, 461–464.

Shao, J., 1993. Linear model selection by cross validation. Journal of American Statistical Association 88 (422), 486–494.

Shao, J., 1996. Bootstrap model selection. Journal of American Statistical Association 91, 655–665.

Shen, X., Ye, J., 2002. Adaptive model selection. Journal of American Statistical Association 97, 210–221.

Stein, C., 1981. Estimation of the mean of a multivariate normal distribution. The Annals of Statistics 9, 1135–1151.
Stone, M., 1977. Asymptotics for and against cross-validation. Biometrika 64 (1), 29–35.
Tibshirani, R., Knight, K., 1999. Model search and inference by bootstrap bumping. Journal of Computational and Graphical Statistics 8, 671–686.
Wehberg, S., Schumacher, M., 2004. A comparison of nonparametric error rate estimation methods in classification problems. Biometrical Journal 46 (1), 35–47.
Wisnowski, J.W., Simpson, J.R., Montgomery, D.C., Runger, G.C., 2003. Resampling methods for variable selection in robust regression. Computational Statistics & Data Analysis 43, 341–355.
Ye, J., 1998. On measuring and correcting the effects of data mining and model selection. Journal of American Statistical Association 93, 120–131.
Zhang, P., 1993. Model selection via multifold cross validation. The Annals of Statistics 21 (1), 299–313.
Zhang, C., 2008. Prediction error estimation under Bregman divergence for non-parametric regression and classification. Scandinavian Journal of Statistics 35 (3), 496–523.