BOOTSTRAP VARIANCE ESTIMATES FOR NEURAL NETWORKS REGRESSION MODELS

FRANCESCO GIORDANO
MICHELE LA ROCCA
CIRA PERNA[*]
Università degli Studi di Salerno, Italy

SUMMARY. In this paper we investigate the usage of different bootstrap methods to estimate the variance of the fitted values from a neural network regression models with possibly depended errors. We particularly focus on residual bootstrap, moving block bootstrap, sieve bootstrap and post-blackening bootstrap. The performance of the proposed approaches are evaluated by a Monte Carlo experiment.

KEYWORDS: Bootstrap, Artificial Neural Networks, Regression Models; Time Series.

1. Introduction

Let $\{Y_t\}$, $t \in \{0, \pm 1, \pm 2, \mathrm{K}\}$, be a (possibly non stationary) process modelled as :

$$Y_t = f(\mathbf{x}_t) + Z_t , \tag{1}$$

where *f* is a non linear continuous function, $\mathbf{x}_t = (x_{1t}, \mathrm{K}, x_{dt})$ is a vector of *d* non stochastic explanatory variables defined on a compact $\aleph \subset \Re^d$, and $\{Z_t\}$is a stationary noise process with zero mean. The function *f* in the model (1) can be approximated with a *single hidden layer feed-forward neural network*; Hornik *et al*. (1989) have shown that this class of non linear functions can approximate any continuous function uniformly on compact sets, by increasing the size of the hidden layer.

In this context, the use of asymptotic results for estimating the standard errors of fitted values, if possible in principle, become soon very difficult and almost impractical in real problems. This motivates increasing interest in resampling techniques (see Tibshirani, 1995; Refenes and Zapranis, 1999 *inter alia*) as alternative and/or complementary tools to the asymptotic ones.

The aim of the paper is to extend some of the common bootstrap proposals to the context of possibly non stationary time series, specified according to the model (1), to estimate the sampling variability of the neural network estimators. We particularly focus on evaluation of the accuracy of the bootstrap estimates based on four different approaches: the residual bootstrap, the moving block bootstrap, the sieve bootstrap and the post-blackening bootstrap.

[*] *Authors' address* : Dipartimento di Scienze Economiche, Università degli Studi di Salerno, Via Ponte Don Melillo, 84084 Fisciano (SA), Italy. E-mail: perna@unisa.it or larocca@unisa.it .

The paper is organised as follows. In the next section we focus on the use of neural networks in regression context and report some asymptotic results. In section 3 we propose and discuss the use of bootstrap techniques to evaluate the variance of the estimation of the function f by neural network. In the last section, to evaluate the performance of the different proposed bootstrap techniques, we report the results of a Monte Carlo simulation experiment.

2. Neural networks in regression models

Neural Networks have been used in various fields to approximate complex non linear structures. Here we consider the *single hidden layer feedforward network* of the form:

$$g(\mathbf{x}_t;\theta)=\sum_{k=1}^{m} c_k \phi\left(\sum_{j=1}^{d} a_{kj} x_{jt}\right) \qquad (2)$$

where $\theta=(c_1,...,c_m,\mathbf{a}'_1,...,\mathbf{a}'_m)$ with $\mathbf{a}'_k=(a_{k1},\ldots,a_{kd})$; c_k, $k=1,\ldots,m$ is the weight of the link between the *k*-th neuron in the hidden layer and the output; a_{ki} is the weight of the connection between the *j*-th input neuron and the *k*-th neuron in the hidden level.

In the formulation (2) the bias term of the hidden layer and that of the output are both zero. Moreover, we suppose that the activation function of the input is the logistic function $\phi(x)=1/(1+e^{-x})$ and that of the hidden layer is the identity function. Barron (1993) has shown that for sufficiently smooth functions the L_2 approximation with these activation functions is $O(1/m)$.

The vector θ has to be chosen to minimise the least squares criterion:

$$\varphi(Y_t;\theta)=\frac{1}{2T}\sum_{t=1}^{T}(Y_t-g(\mathbf{x}_t;\theta))^2 \qquad (3)$$

that is:

$$\hat{\theta}_T=\arg\min_{\theta}\varphi(Y_t;\theta) \qquad (4)$$

The derivatives of the fit criterion with respect to the weights can be calculated recursively from output to input by using the chain rule, a procedure known as *back-propagation* (see, for example, Haykin, 1994; Lachtermacher and Fuller, 1995). This algorithm can take a large number of iterations to converge and local minima are very common.

After having obtained an estimate of the parameters we have:

$$\hat{f}(Y_t, \mathbf{x}_t) = g\left(\mathbf{x}_t; \hat{\theta}_T\right) = \sum_{k=1}^{m} \hat{c}_k \phi\left(\sum_{j=1}^{d} \hat{a}_{kj} x_{jt} \right) \tag{5}$$

White (1989), using stochastic approximations, derived some asymptotic properties of this recursive procedure. He showed that, under some general hypotheses, the *back-propagation* estimator converges almost surely to the value that minimises the expected mean squared error and it is asymptotically Normal.

In previous papers (Giordano and Perna, 1998; 1999) we proved, using an alternative approach based on the theory of *M*-estimators (Huber, 1981), the consistency of the estimators $\hat{\theta}_T$ and $\hat{f}(Y_t, \mathbf{x}_t)$ and derived the asymptotic distributions both in the case of *iid*. errors that in the case of fourth order stationary and φ-mixing errors.

Let $m = m(T)$, as $T \to \infty$, under the hypotheses that $m(T) \to \infty$ and $[m(T)]^2 / T \to 0$ we proved the following theorems.

Theorem 1. The estimator $\hat{\theta}_T$, defined in (4), converges in distribution to a Normal distribution with zero mean and variance equal to:

$$\frac{1}{T} \frac{\int \psi^2(x;\theta) dF(x)}{\left[\int \frac{\partial}{\partial t} \psi(x;t) \Big|_{t=\theta} dF(x) \right]^2}$$

where

$$\psi(Y_t;\theta) = \frac{\partial}{\partial \theta} \varphi(Y_t;\theta) \tag{6}$$

and $F(x)$ is the uniform distribution.

Theorem 2. The random variables $\bar{g}(\mathbf{x}_t) = \sum_{k=1}^{m} \hat{c}_k \phi\left(\sum_{j=1}^{d} a_{kj} x_{jt} \right)$ are asymptotically Normal with

$$E[\bar{g}(\mathbf{x}_t)] = g(\mathbf{x}_t;\theta)$$

$$\operatorname{var}[\bar{g}(\mathbf{x}_t)] = \sum_{k=1}^{m} \phi^2\left(\sum_{j=1}^{d} a_{kj} x_{jt} \right) \operatorname{var}(\hat{c}_k) + \sum_{k \neq h} \phi\left(\sum_{j=1}^{d} a_{kj} x_{jt} \right) \phi\left(\sum_{j=1}^{d} a_{hj} x_{jt} \right) \operatorname{cov}(\hat{c}_k, \hat{c}_h)$$

The latter theorem permits to derive the asymptotic distribution of $\hat{f}(Y_t, \mathbf{x}_t)$ which is distributionally equivalent to $\bar{g}(\mathbf{x}_t)$.

It is evident, from the previous results, the variance of the estimators involved is difficult to evaluate analytically. To overcome the problem, in previous papers

(Giordano and Perna, 1998; 1999), we derived the following upper bounds for the coefficient variance:

$$\operatorname{var}(\hat{c}_k) \le \frac{1}{T} \frac{\sigma^2 + \frac{c_f}{m}}{\left[\int \phi^2\left(\sum_{j=1}^{d} a_{kj} x_j\right) d\mathbf{x}\right]^2};$$

$$\operatorname{cov}(\hat{c}_k, \hat{c}_h) \le \frac{1}{T} \frac{\sigma^2 + \frac{c_f}{m}}{\left[\int \phi^2\left(\sum_{j=1}^{d} a_{kj} x_j\right) d\mathbf{x}\right]\left[\int \phi^2\left(\sum_{j=1}^{d} a_{hj} x_j\right) d\mathbf{x}\right]}$$

in which: $c_f = (2rC)^2$, r is the radius of the compact $\aleph$; $C = \int_{\Re^d} |w| \left|\tilde{f}(w)\right| dw$ and $\tilde{f}(w)$ is the Fourier transform of the function f.

It follows that the variance of $\bar{g}(\mathbf{x}_t)$ can be approximated by:

$$\operatorname{var}[\bar{g}(\mathbf{x}_t)] \le \frac{\sigma^2}{T} \left[\sum_{k=1}^{m} \frac{\phi^2\left(\sum_{j=1}^{d} a_{kj} x_{jt}\right)}{\left(\int \phi^2\left(\sum_{j=1}^{d} a_{kj} x_j\right) d\mathbf{x}\right)^2} + \sum_{k \ne h} \frac{\phi\left(\sum_{j=1}^{d} a_{kj} x_{jt}\right)\phi\left(\sum_{j=1}^{d} a_{hj} x_{jt}\right)}{\left(\int \phi^2\left(\sum_{j=1}^{d} a_{kj} x_j\right) d\mathbf{x}\right)\left(\int \phi^2\left(\sum_{j=1}^{d} a_{hj} x_j\right) d\mathbf{x}\right)} \right]$$

The upper bounds have a quite complex structure not feasible for the applications and for an easy practical usage. This is quite common in the setting of nonparametric estimation where asymptotic techniques, even if available in principle and very useful to study the theoretical properties of the statistics involved, are only rarely used. It is much more common to carry out stochastic simulations such as bootstrapping to provide feasible estimators of the sampling variability. In the context of neural networks the bootstrap technique has been pursued in Tibshirani (1995) and Refenes and Zapranis (1999), *inter alia*. Bootstrap works by creating many pseudo-replicates, bootstrap sample, of the training set and then re-estimating the statistics on each bootstrap sample.

In particular, we compare the residual bootstrap (a typical proposal in neural networks) with different non-parametric bootstrap schemes. They have a wider range of applications and give consistent procedures under some very general and minimal conditions. These are genuine non parametric bootstrap methods which seem the best choice when dealing with non parametric estimates. In our context, no specific and explicit structures for the noise must be assumed. This can be particularly useful in neural networks when the specification of the parameters can heavily affect the structure of the residuals.

3. The bootstrap approach

As first proposed by Efron (1979), bootstrap methods are designed for application to samples of independent data. Under that assumption they implicitly produce an adaptive model for the marginal sampling distribution. Extensions to dependent data are not straightforward and modifications of the original procedures are needed in order to preserve the dependence structure of the original data in the bootstrap samples. In the context of neural networks applied to time series data two alternative groups of techniques are available.

A straightforward approach is model based, where the dependence structure is modelled explicitly and completely by a neural network and the bootstrap sample is drawn from the fitted neural network model. The procedure can be implemented as follows.

Step 1. Compute the neural network estimates $\hat{f}(Y_t, \mathbf{x}_t)$ for $t = 1, \ldots, T$.

Step 2. Compute the residuals $\hat{Z}_t = Y_t - \hat{f}(Y_t, \mathbf{x}_t)$ with $t = 1, \ldots, T$ and the centred residuals $\tilde{Z}_t = \hat{Z}_t - \sum_{t=1}^{T} \hat{Z}_t / T$.

Step 3. Denote by $\hat{F}_{\tilde{Z}}$ the empirical cumulative distribution function of $\tilde{Z}_t$, $t = 1, \ldots, T$. Resample $\{Z_t^*\}$ *iid* from $\hat{F}_{\tilde{Z}}$ with $t = 1, \ldots, T$.

Step 4. Then generate a bootstrap series by $Y_t^* = \hat{f}(Y_t, \mathbf{x}_t) + Z_t^*$ with $t = 1, \ldots, T$.

Such model-based approach is, of course, inconsistent if the model used for resampling is misspecified.

Alternatively, nonparametric, purely model free bootstrap schemes have been proposed. In those procedures blocks of consecutive observations are resampled randomly with replacement, from the original time series and assembled by joining the blocks together in random order in order to obtain a simulated version of the original series (Kunsch, 1989; Politis and Romano, 1992 *inter alia*). These approaches, known as blockwise bootstrap or moving block bootstrap, generally works satisfactory and enjoys the properties of being robust against misspecified models.

The MBB bootstrap procedure can be adapted to possibly non stationary time series, in a neural network context, as follows.

Step 1. Compute the neural network estimates $\hat{f}(Y_t, \mathbf{x}_t)$ for $t = 1, \ldots, T$.

Step 2. Compute the residuals $\hat{Z}_t = Y_t - \hat{f}(Y_t, \mathbf{x}_t)$ with $t = 1, \ldots, T$ and the centred residuals $\tilde{Z}_t = \hat{Z}_t - \sum_{t=1}^{T} \hat{Z}_t / T$.

Step 3. Fix $l < n$ and form blocks of length *l* of consecutive observations from the original data, i.e. the bootstrap sample is

$$Z^{*}_{(j-1)l+t} = \tilde{Z}_{S_j+t},\ 1 \le j \le b,\ 1 \le t \le l\,.$$

where $b = [T/l]$ denoting with $[x]$ the smallest integer greater or equal to *x*. Let $S_1, S_2, \ldots, S_b$ are *iid* uniform on $\{0,1,\ldots,T-l\}$. If T is not a multiple of *l,* only $T + l - bl$ observations from the last block are used. Given bootstrap replicate $\{Z_1^*, \ldots, Z_T^*\}$, generate the bootstrap observations by setting. $Y_t^* = \hat{f}\left(Y_t, \mathbf{x}_t\right) + Z_t^*$ with $t = 1, \ldots, T$.

The MBB does not require one to select a model and the only parameter required is the block length. The idea that underlies this block resampling scheme is that if block are long enough the original dependence will be reasonably preserved in the resampled series. Clearly this approximation is better if the dependence is weak and the blocks are as long as possible, thus preserving the dependence more faithfully. On the other hand the distinct values of the statistics must be as numerous as possible to provide a good estimate of the distribution of the statistics and this point towards short blocks. Thus, unless the length of the series is considerable to accommodate longer and more number of blocks the preservation of the dependence structure may be difficult, especially for complex, long range dependence structure. In such cases, the block resampling scheme tend to generate resampled series that are less dependent than the original ones. Moreover, the resampled series often exhibits artifacts which are caused by joining randomly selected blocks. As a consequence, the asymptotic variance-covariance matrices of the estimators based on the original series and those based on the bootstrap series are different and a modification of the original scheme is needed. A possible solution is the matched moving block bootstrap proposed by Carlstein *et al*., (1996). The idea is to align with higher likelihood those blocks which match at their ends. This is achieved by a quite complex procedure which resamples the blocks according to a Markov chain whose transitions depend on the data. A further difficulty, is that the bootstrap sample is not (conditionally) stationary. This can be overcome by taking blocks of random length, as proposed by Politis and Romano (1994), but a tuning parameter, which seems difficult to control, has to be fixed. Anyway, a recent study of Lahiri (1999) shows that this approach is much less efficient than the original one and so no clear choice is possible..

A more effective solution seems to be the sieve bootstrap (see Buhlmann 1998; 1999). It can be implemented in our context as follows.

Step 1. Compute the neural network estimates $\hat{f}\left(Y_t, \mathbf{x}_t\right)$ for $t = 1, \ldots, T$.

Step 2. Compute the residuals $\hat{Z}_t = Y_t - \hat{f}\left(Y_t, \mathbf{x}_t\right)$ with $t = 1, \ldots, T$ and the centred residuals $\tilde{Z}_t = \hat{Z}_t - \sum_{t=1}^{T} \hat{Z}_t / T$.

Step 3. Fit an autoregressive model of order *p* to the residuals $\tilde{Z}_t$ and compute another set of residuals

$$\hat{\varepsilon}_t = \sum_{j=0}^{p} \hat{\phi}_j \, \tilde{Z}_{t-j} \, , \; \hat{\phi}_0 = 1 , \; t = p+1, \ldots ,T \, .$$

A guideline for approximating *p* is given by the Akaike information criterion in the increasing range $\left[0, 10\log_{10}(T)\right]$, the default option of the *S*-plus package.

Compute $\tilde{\varepsilon}_t = \hat{\varepsilon}_t - \sum_{t=p+1}^{T} \hat{\varepsilon}_t /(T-p)$, $t = p+1, \ldots ,T$.

Step 4. Denote by $\hat{F}_{\tilde{\varepsilon}}$ the empirical cumulative distribution function of $\tilde{\varepsilon}_t$, $t = p+1, \ldots ,T$. Resample $\left\{\varepsilon_t^*\right\}$ *iid* from $\hat{F}_{\tilde{\varepsilon}}$ with $t = 1, \ldots ,T$.

Step 5. Generate the bootstrap error series $\left\{Z_t^*\right\}$, $t = 1, \ldots ,T$, defined by

$$\varepsilon_t^* = \sum_{j=0}^{p} \hat{\phi}_j \, Z_{t-j}^* \, , \; \hat{\phi}_0 = 1 , \; t = 1, \ldots ,T \, .$$

Here we start the recursion with some starting value (the initial conditional if available or some resampled innovations) and wait until stationarity is reached.

Step 6. Then generate a bootstrap series by $Y_t^* = \hat{f}\left(Y_t, \mathbf{x}_t\right) + Z_t^*$ with $t = 1, \ldots ,T$.

Observe that even if the sieve bootstrap is based on a parametric model it is basically non parametric in its spirit. The AR(*p*) model here is just used to filter the residuals series.

A different approach can be motivated by observing that if the model used in the sieve bootstrap is not appropriate, the resulting residuals cannot be treated as *iid*. An hybrid approach between the previous two, named post-blackening bootstrap (PBB in the following), was suggested by Davinson and Hinkley (1997) and studied by Srinivas and Srinivasan (2000). The procedure is much similar to the sieve bootstrap but the residuals from the AR(*p*) model are not resampled in an *iid* manner but using the MBB bootstrap. Hence, if some residual dependence structure is still present in the AR residuals this is kept from the blockwise bootstrap. Here, the model, usually a simple linear model, is used to 'pre-withen' the series by fitting a model that is intended to remove much of the dependence present in the observations. A series of innovations is then generated by block resampling of residuals obtained from the fitted model, the innovation series is then 'post-blackened' by applying the estimated model to the resampled innovations.

The bootstrap series generated by using one of the previous methods can be used to approximate the sampling distribution, or some particular aspects such as its variability. Given the bootstrap series Y_t^*, $t = 1, \ldots ,T$, compute the bootstrap analogue of the neural network parameters

$$\hat{\theta}_T^* = \arg\min_{\theta} \frac{1}{2T} \sum_{t=1}^{T} \left(Y_t^* - g(\mathbf{x}_t ; \theta)\right)^2$$

and the bootstrap analogue of the neural network estimates

$$\hat{f}^{*}(Y_t,\mathbf{x}_t)=g\left(\mathbf{x}_t;\hat{\theta}_T^{*}\right)=\sum_{k=1}^{m}\hat{c}_k^{*}\phi\left(\sum_{j=1}^{d}\hat{a}_{kj}^{*}x_{jt}\right).$$

Then, estimate the variance $\mathrm{var}\left[\hat{f}(Y_t,\mathbf{x}_t)\right]$ with the bootstrap variance $\mathrm{var}^{*}\left[\hat{f}^{*}(Y_t,\mathbf{x}_t)\right]$, where $\mathrm{var}^{*}\left[\hat{f}^{*}(Y_t,\mathbf{x}_t)\right]$ denotes the variance of $\hat{f}^{*}(Y_t,\mathbf{x}_t)$ conditional on $(Y_t,\mathbf{x}_t)$ $t=1,\ldots,T$, the observed data. As usual the bootstrap variance can be approximated through a Monte Carlo approach by generating B different bootstrap series and estimating the bootstrap variance as

$$\mathrm{var}_B^{*}\left[\hat{f}^{*}(Y_t,\mathbf{x}_t)\right]=\frac{1}{B-1}\sum_{b=1}^{B}\left[\hat{f}_b^{*}(Y_t,\mathbf{x}_t)-\bar{\hat{f}}_{\cdot}^{*}(Y_t,\mathbf{x}_t)\right]^2$$

where

$$\bar{\hat{f}}_{\cdot}^{*}(Y_t,\mathbf{x}_t)=\frac{1}{B}\sum_{b=1}^{B}\hat{f}_b^{*}(Y_t,\mathbf{x}_t).$$

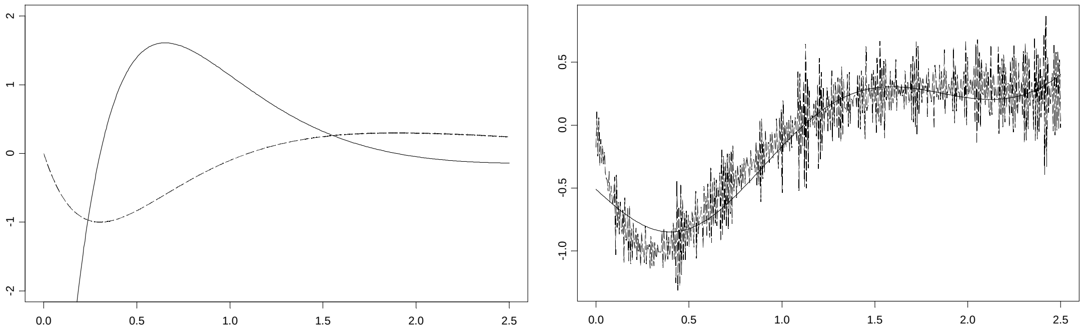
4. Monte Carlo results and some concluding remarks

To study how the proposed bootstrap procedures can be used to produce accurate estimates of sampling variability of the neural network estimates, a Monte Carlo experiment was performed. The simulated data set has been generated as $Y_t = f(x_t)+Z_t$ where the deterministic part is given by the Wahba's function specified as $f(x)=4.26\left(e^{-x}-4e^{-2x}+3e^{-3x}\right)$, with $x\in[0,2.5]$ as in Refenes and Zapranis (1999). Two different specifications for the noise process Z_t have been considered: a white noise and an ARMA(1,1), specified as $Z_t=-0.8Z_{t-1}-0.5\varepsilon_{t-1}+\varepsilon_t$ with the innovations ε_t distributed as a Student-*t* with 6 degrees. All the noise processes have been scaled so that the variability of the of the noise is about 20% of that of the signal. In figure 1 we reported a graph of the function along with its first order derivative and a typical realizations when considering ARMA noise.

The simulations are based on 200 Monte Carlo runs and 50 bootstrap replicates. We fixed $T=\{200,.500\}$. The block length l in the MBB scheme is fixed to $l=T^{1/3}$, a value that seems to work quite well in many cases (Buhlmann and Kunsch, 1999); the number m of neurons in the hidden layer are and $m=T^{1/3}$ (see Perna and Giordano, 1999). As accuracy measure we considered the statistics

$T\left\{\text{var}^*\left[\hat{f}^*(Y_t, x_t)\right] - \text{var}\left[\hat{f}(Y_t, x_t)\right]\right\}$ where the ‘true’ variance, $\text{var}\left[\hat{f}(Y_t, x_t)\right]$, has been computed through 200 Monte Carlo runs.

Figure. 1. *Wahba’s function (dashed line) and its first order derivative on the left panel; a typical realization with an ARMA process with innovations distributed as Student –t on the right panel along with a neural network estimates .*



As stressed by Refenes and Zapranis (1999) the accuracy of the bootstrap estimates of $\text{var}\left[\hat{f}(Y_t, x_t)\right]$ can be affected by computational problems, such as sensitivity of the learning algorithm to initial conditions. In our simulation study, we investigated the impact of four strategies for the choice of the starting values in the learning algorithm, when generating the different bootstrap series. In the first scheme, the local bootstrap, they are fixed to the values that minimise the objective function (3) and equal for all the *B* resampled series (B1 in the following). In the second scheme, the local perturbated bootstrap, the starting values are perturbated by a small random quantity drawn from a a zero mean Gaussian distribution with variance equal to 0.01 (B2 in the following). In the third scheme, the random global bootstrap, they are randomly selected from an uniform interval $[-0.5, 0.5]$ (B3 in the following). Finally, in the last scheme, the fixed global bootstrap, the starting values are randomly selected from $[-0.5, 0.5]$ and remain fixed when generating the resampled series (B4 in the following).

The performance of the proposed procedures have been examined in terms of the distribution of $T\left\{\text{var}^*\left[\hat{f}^*(Y_t, x_t)\right] - \text{var}\left[\hat{f}(Y_t, x_t)\right]\right\}$. In figures 2-6 we reported the median of the Monte Carlo distributions along with the quantities $H_1 = Q_1 - 1.5(Q_3 - Q_1)$ and $H_2 = Q_3 + 1.5(Q_3 - Q_1)$ where Q_1 and $Q_3$are the first and the third quartile. The Monte Carlo distributions were computed on 200 and 500 points, respectively, equally spaced in the interval $[0, 2.5]$.

In all the cases considered it is evident that serious problems arise for critical values of the first order derivative of the Wahba’s function.

As expected, for normal *iid* innovations (Fig. 2 and Fig 4) , the RB outperforms all the other methods. It is interesting to observe that the MBB yield reasonable overall performances while the SB and the PBB exhibit much more variability for the estimates in correspondence of the critical points of the regression function. In any case the performance of all methods become similar for increasing sample sizes.

This ranking is completely different when considering noise with a much more complex structure, namely an ARMA with student-*t* innovations (Fig. 3 and Fig. 4). In this case, the MBB definitely seems the best choice. The variability of the MBB bootstrap estimates are much better than those obtained by the RB. It is quite surprising that the SB and the PBB behave poorly not only with respect to the MBB but also to the RB, which does not consider any kind of dependence in the residuals of the fitted model. A possible explanation can be given considering that the neural network estimates catch part of the dependence structure of the noise and so the residuals of the fitted model do not allow an accurate estimate of the AR models on which the SB and the PBB are based.

In our simulations it seems to be confirmed that a local bootstrap approach (namely schemes B1 and B2) should be preferred to the global ones (schemes B3 and B4). In these cases all the methods fails (see Fig. 6). Results, not reported here, are even worse when considering a noise with an ARMA structure with Student-t innovations.

Several different aspects should be further explored to get a better insight of the joint usage of neural networks and bootstrap methods. An interesting point arise when considering the relationships between the block length of the MBB and the hidden layer size. In any case, these first results, and the others reported in the literature, are quite encouraging. Of course, the resulting combined procedure is really computer intensive, but this does not seem to be a serious limit due the increasing power computing available even on PC desktops.

Acknowledgements
The paper is supported by MURST98 “Modelli statistici per l’analisi delle serie temporali”.

References

Barron, A.R. (1993) Universal Approximation Bounds for Superpositions of a Sigmoidal Function, *IEEE Transactions on Information Theory*, 39, 930-945.

Buhlmann, P. (1998) Sieve bootstrap for smoothing in nonstationary time series, *The Annals of Statistics*, 26, 48-83.

Bühlmann P. (1999) Bootstrap for Time Series, *Research report n. 87*, ETH, Zürich.

Buhlmann, P.; Kunsch, H. R. (1999) Block length selection in the bootstrap for time series, Computational Statistics and Data Analysis, 31, 295-310

Efron, B. (1979) Bootstrap methods: another look at the jackknife, *The Annals of Statistics*, 7, 1-26.

Giordano F.; Perna C. (1998) Proprietà asintotiche degli stimatori neurali nella regressione non parametrica, *Atti della XXXIX Riunione Scientifica SIS*, 2, 235-242

Giordano F.; Perna C. (1999) Large Sample Properties of Neural Estimators in a Regression Model with φ-mixing errors, to appear

Haykin, S (1994) *Neural Networks: a comprehensive foundation*, Macmillan, New-York.

Hornik, K.; Stinchcombe, M.; White, H. (1989) Multy-Layer Feedforward Networks Are Universal Approximators, *Neural Networks*, 2, 359-366.

Huber P. (1981) *Robust Statistics*, J.Wiley & Sons, New-York

Lachtermacher, G.; Fuller, J.D. (1995) Backpropagation in Time-series Forecasting, *Journal of Forecasting*, 14, 881-393.

Kunsch, H.R. (1989) The jackknife and the bootstrap for general stationary observations, *The Annals of Statistics*, 17, 1217-1241.

Lahiri, S. N. (1999): Theoretical comparisons of block bootstrap methods, *The Annals of Statistics*, 27, 386-404

Perna C., Giordano, F. (1999) The hidden layer size in feed-forward neural networks: a statistical point of view, *Atti del Convegno SCO99, "Modelli complessi e metodi computazionali intensive per la stima e la previsione*, 95-100

Politis, D. N. and Romano, J. P. (1992) A circular block-resampling procedure for stationary data, in *Exploring the limits of the bootstrap* (eds. C. Page and R. LePage), Springer-Verlag, NY.

Politis, D. N. and Romano, J. P. (1994) The stationary bootstrap, *JASA*, 1303-1313.

Refenes, A.P.N.; Zapranis, A.D. (1999) Neural model identification, variable selection and model adequacy, *Journal of Forecasting*, 18, 299-332

Srinivas, V.V.; Srinivasan, K. (2000) Post-blackening approach for modelling dependent annual streamflows, *Journal of Hydrology*, 230, 86-126

Tibshirani, R. (1985) *A comparison of some error estimates for neural network models*, Research Report, Department of Preventive and Biostatistics, University of Toronto

Figure 2. *Median (dashed line), H_1 and H_2 (solid line) of the accuracy measure* $T\left\{\mathrm{var}^*\left[\hat{f}^*(Y_t,x_t)\right]-\mathrm{var}\left[\hat{f}(Y_t,x_t)\right]\right\}$*; Bootstrap scheme B1; normal iid innovations; T=200 and T=500.*

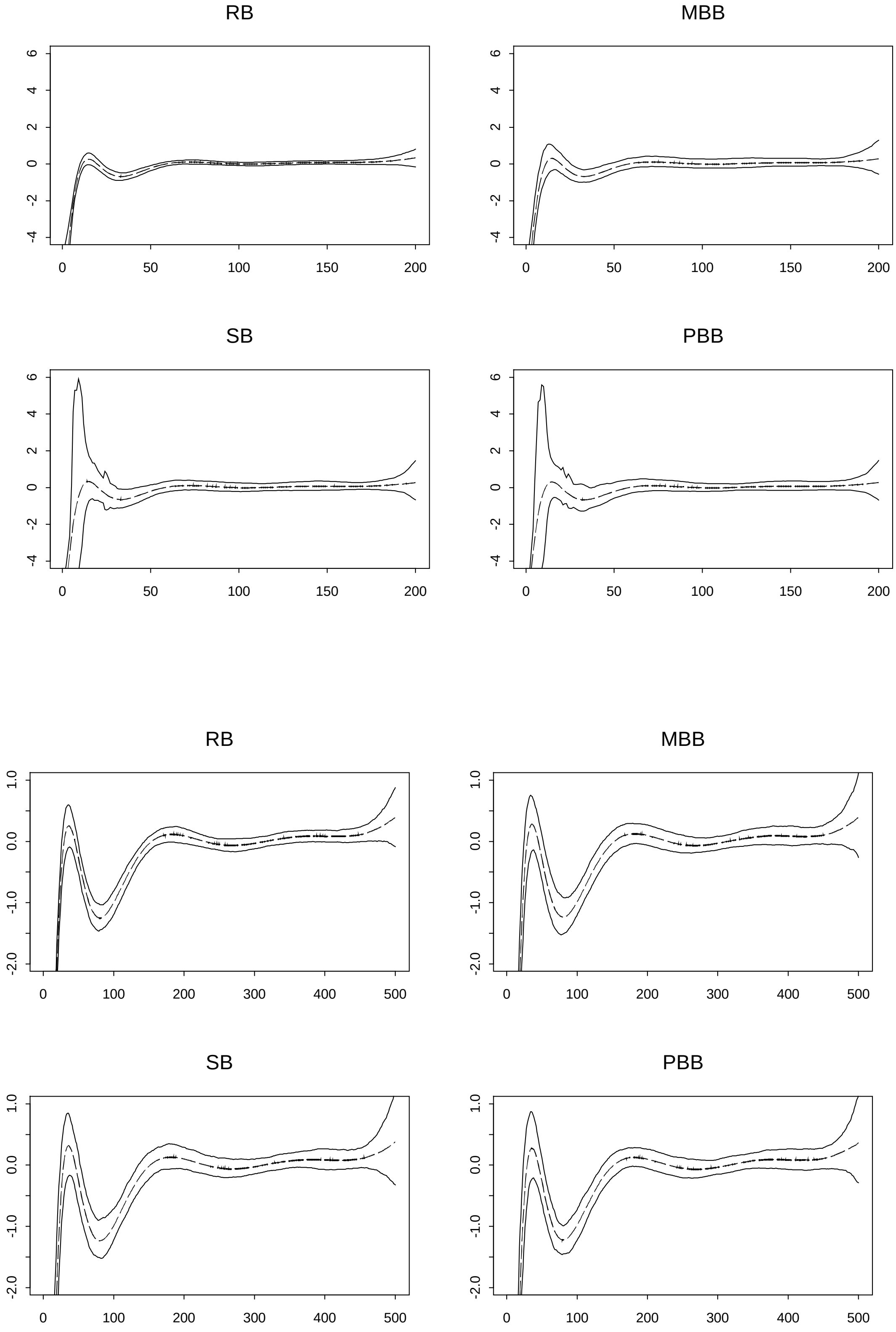


Figure 3. *Median (dashed line), H_1 and H_2 (solid line) of the accuracy measure* $T\left\{\mathrm{var}^*\left[\hat{f}^*(Y_t,x_t)\right]-\mathrm{var}\left[\hat{f}(Y_t,x_t)\right]\right\}$*; Bootstrap scheme B1; ARMA with Student-t innovations; T=200 and T=500.*

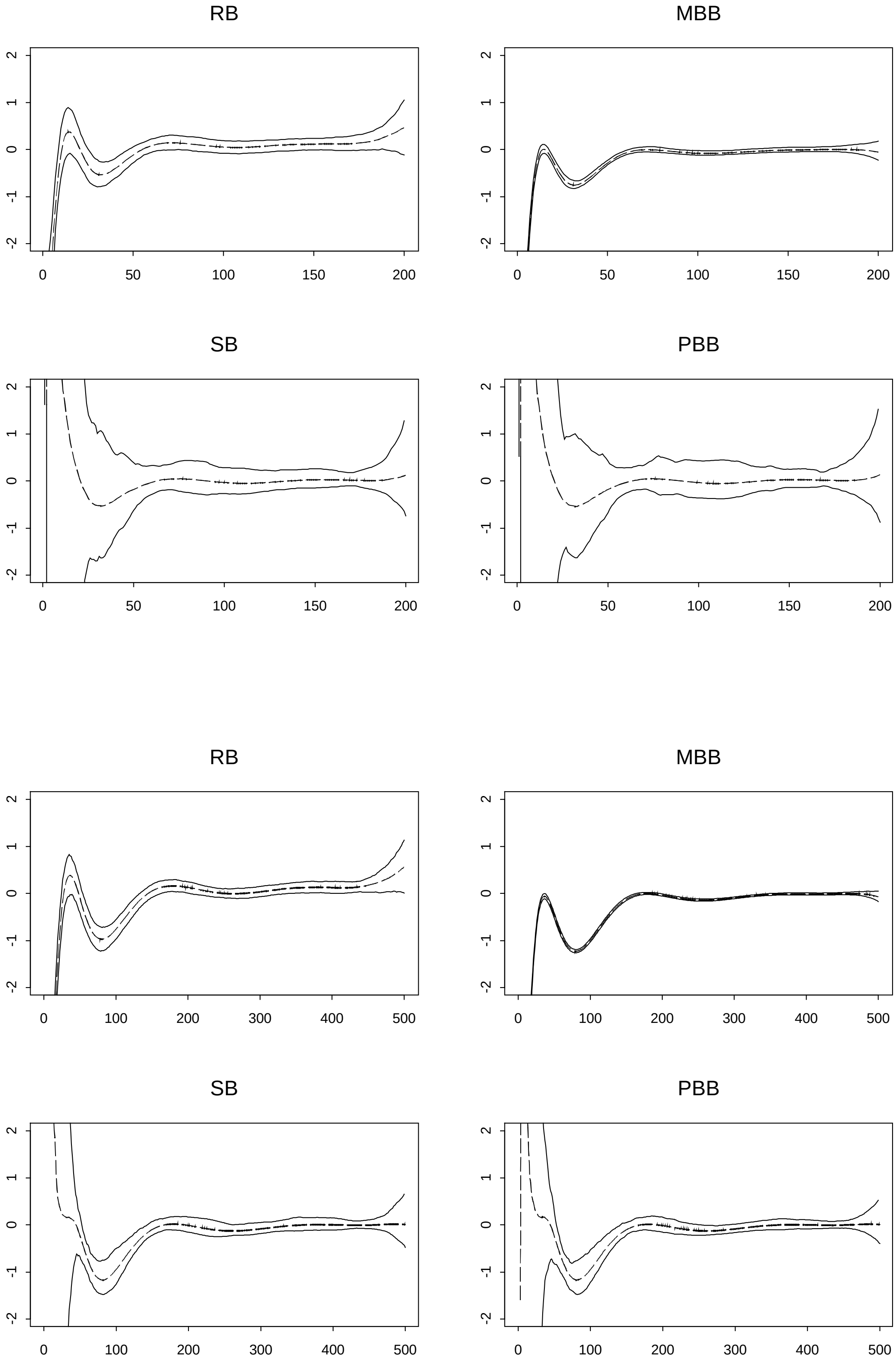


Figure 4. *Median (dashed line), H_1 and H_2 (solid line) of the accuracy measure* $T\left\{\mathrm{var}^*\left[\hat{f}^*(Y_t,x_t)\right]-\mathrm{var}\left[\hat{f}(Y_t,x_t)\right]\right\}$*; Bootstrap scheme B2; normal iid innovations; T=200 and T=500.*

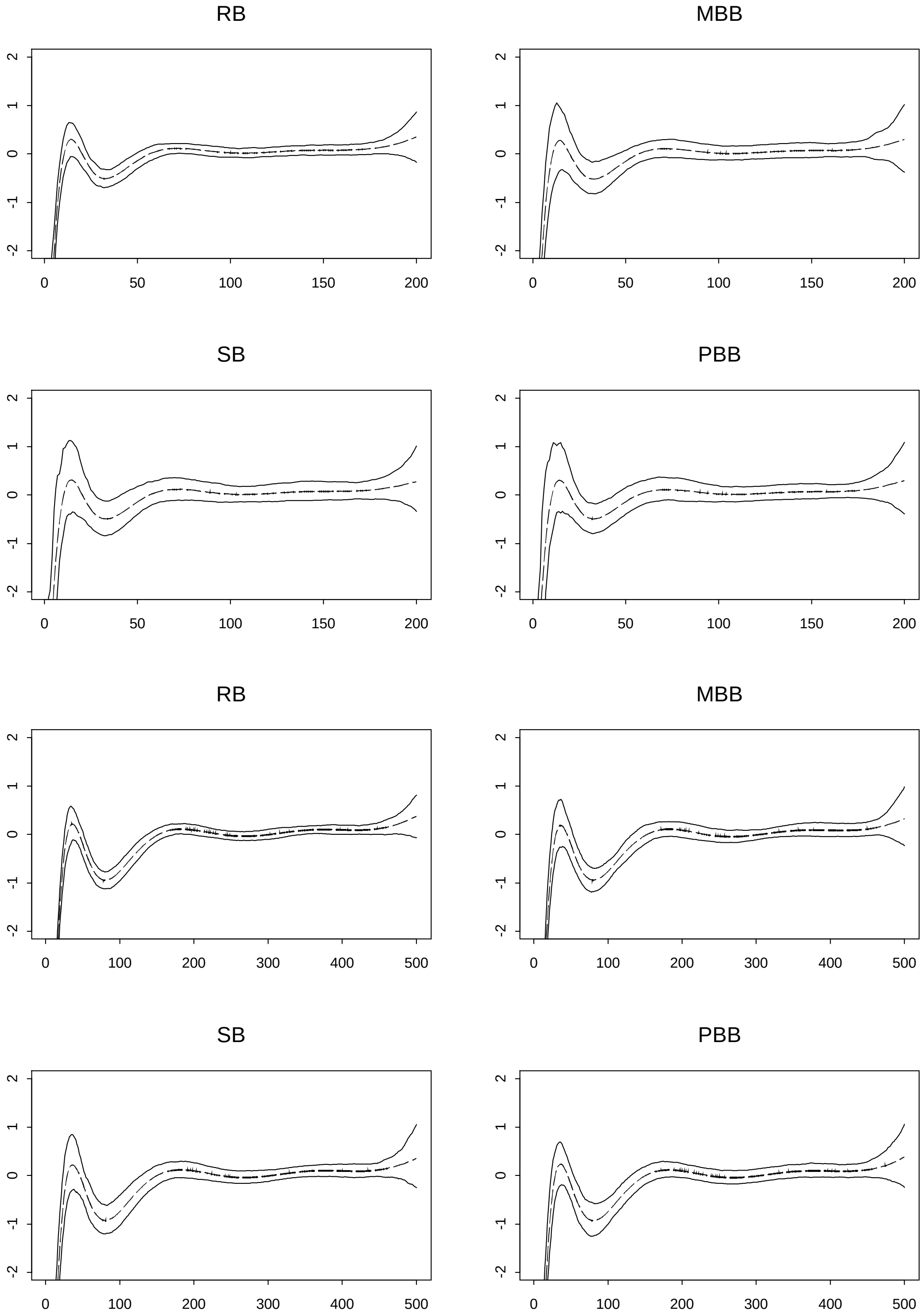


Figure 5. *Median (dashed line), H_1 and H_2 (solid line) of the accuracy measure* $T\left\{\mathrm{var}^*\left[\hat{f}^*(Y_t,x_t)\right]-\mathrm{var}\left[\hat{f}(Y_t,x_t)\right]\right\}$*; Bootstrap scheme B2; ARMA with Student-t innovations; T=200 and T=500.*

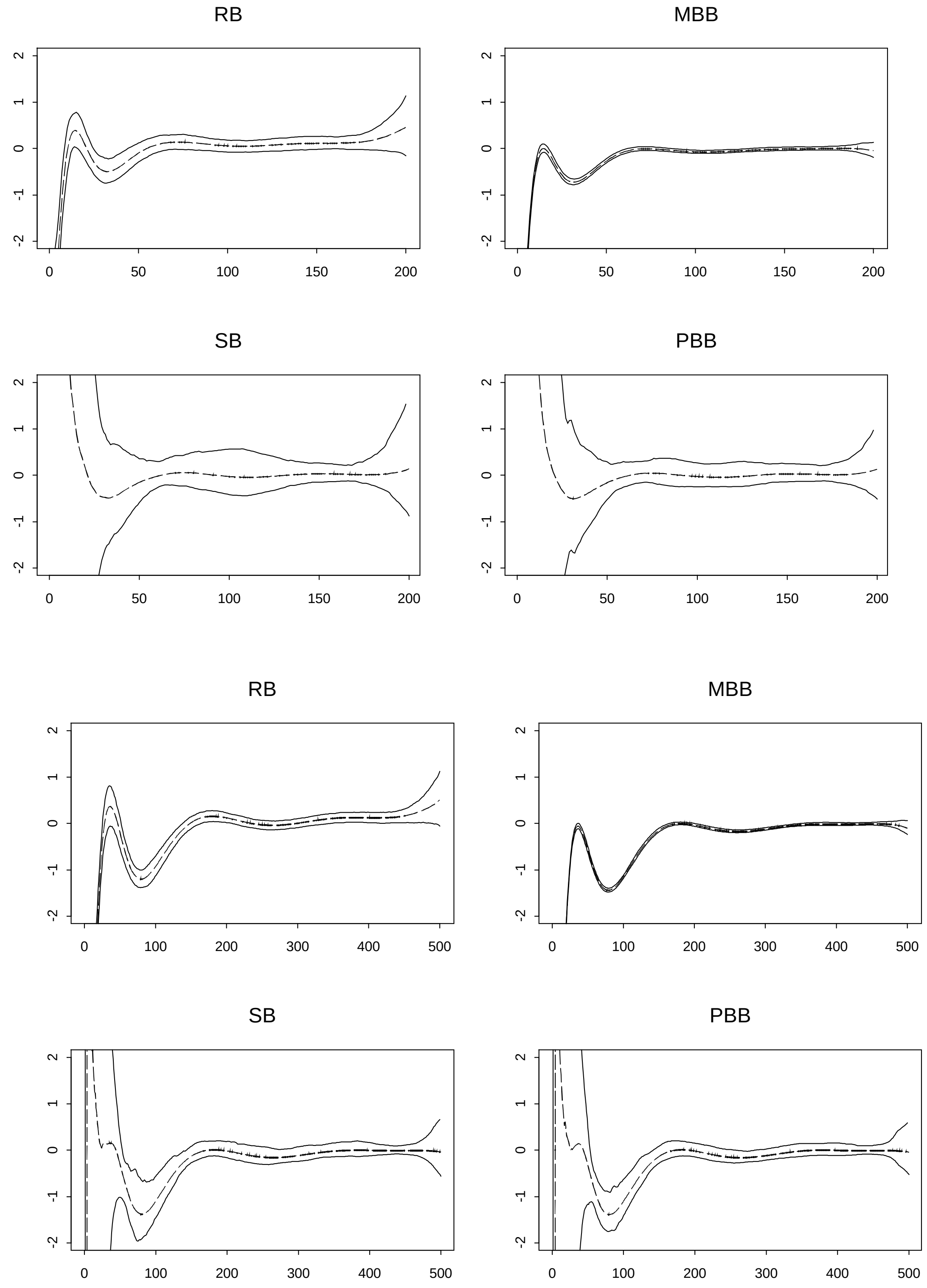
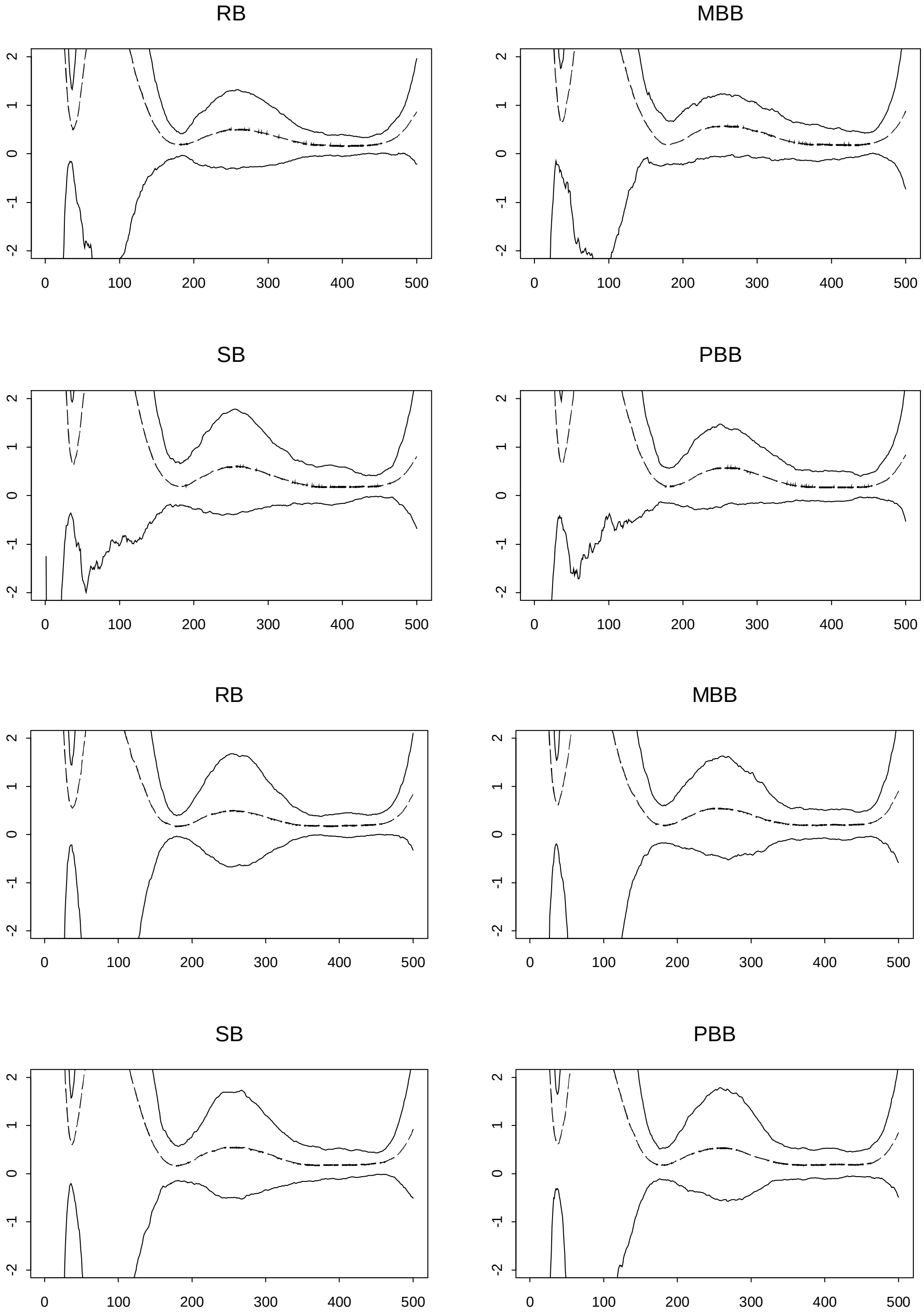


Figure 6. *Median (dashed line), H_1 and H_2 (solid line) of the accuracy measure* $T\left\{\mathrm{var}^*\left[\hat{f}^*(Y_t,x_t)\right]-\mathrm{var}\left[\hat{f}(Y_t,x_t)\right]\right\}$*; Bootstrap scheme B3 and B4; normal iid innovations; T=500.*


RB
MBB
SB
PBB
RB
MBB
SB
PBB