Averaging Regularized Estimators

Michiaki Taniguchi and Volker Tresp*
Siemens AG, Central Research
Otto-Hahn-Ring 6
81730 München, Germany

Abstract

We compare the performance of averaged regularized estimators. We show that the improvement in performance which can be achieved by averaging depends critically on the degree of regularization which is used in training the individual estimators. We compare four different averaging approaches: simple averaging, bagging, variance-based weighting and variance-based bagging. In any of the averaging methods the greatest degree of improvement —if compared to the individual estimators— is achieved if no or only a small degree of regularization is used. Here, variance-based weighting and variance-based bagging are superior to simple averaging or bagging. Our experiments indicate that better performance for both individual estimators and for averaging is achieved in combination with regularization. With increasing degrees of regularization, the two bagging-based approaches (bagging, variance-based bagging) outperform the individual estimators, simple averaging, as well as variance-based weighting. Bagging and variance-based bagging seem to be the overall best combining methods over a wide range of degrees of regularization.

*e-mail: Michiaki.Taniguchi@zfe.siemens.de, Volker.Tresp@zfe.siemens.de

1 Introduction

Several authors have noted the advantages of averaging estimators which were trained either on identical training data (Perrone, 1993) or on bootstrap samples of the training data (a procedure termed "bagging predictors" by Breiman, 1994). Theory and experiments both show that averaging helps most if the errors in the individual estimators are not positively correlated and if the estimators have only small bias. On the other hand, it is well known from theory and experiment that best performance of a single predictor is typically achieved if some form of regularization (weight decay), early stopping or pruning are used. All three methods tend to decrease variance and increase bias of the estimator. Therefore, we expect that the optimal degrees of regularization for a single estimator and for averaging would not necessarily be the same. In this paper we investigate the effect of regularization on averaging. In addition to simple averaging and bagging we also perform experiments using combining principles where the weighting functions are dependent on the input. The weighting functions can be derived by estimating the variance of each estimator for a given input (variance-based weighting, Tresp and Taniguchi, 1995, variance-based bagging, Taniguchi and Tresp, 1995). In the next section we derive some fundamental equations for averaging biased and unbiased estimators. In Section 3 we show how the theory can be applied to regression problems and we introduce the different averaging methods which were used in the experiments described in Section 4. In Section 5 we discuss the results and in Section 6 we present conclusions.

2 A Theory of Combining Biased Estimators

2.1 Optimal Weights for Combining Biased Estimators

We would like to estimate the unknown variable t based on the realizations of a set of random variables $\{f_i\}_{i=1}^M$. The expected squared error between f_i and t

$$\begin{aligned} E(f_i - t)^2 &= E(f_i - m_i + m_i - t)^2 \\ &= E(f_i - m_i)^2 + E(m_i - t)^2 + 2E((f_i - m_i)(m_i - t)) \\ &= var_i + b_i^2 \end{aligned}$$

decomposes into the variance $var_i = E(f_i - m_i)^2$ and the square of the bias $b_i = m_i - t$ with $m_i = E(f_i)$. $E(.)$ stands for the expected value. Note, that $E[(f_i - m_i)(m_i - t)] = (m_i - t)E(f_i - m_i) = 0$.

In the following we are interested in estimating t by forming a linear combination of the f_i

$$\hat{t} = \sum_{i=1}^{M} g_i f_i = g'f$$

where $f = (f_1, \ldots, f_M)'$ and the weighting vector $g = (g_1, \ldots, g_M)'$. The expected

error of the combined system is (Meir, 1995)

$$\begin{aligned} E(\hat{t}-t)^2 &= E(g'f - E(g'f))^2 + E(E(g'f) - t)^2 \\ &= E(g'(f - E(f)))^2 + E(g'm - t)^2 \\ &= g'\Omega g + (g'm - t)^2 \end{aligned} \tag{1}$$

where Ω is an $M \times M$ covariance matrix with

$$\Omega_{ij} = E[(f_i - m_i)(f_j - m_j)]$$

and with $m = (m_1, \ldots, m_M)'$. The expected error of the combined system is minimized for[1]

$$g^* = (mm' + \Omega)^{-1} tm.$$

2.2 Constraints

A commonly used constraint which we also use in our experiments is that

$$\sum_{i=1}^{M} g_i = 1, \qquad g_i \geq 0, \;\; i = 1, \ldots, M.$$

In the following, g can be written as

$$g = (u'h)^{-1} h \tag{2}$$

where $u = (1, \ldots, 1)'$ is an M-dimensional vector of ones, $h = (h_1, \ldots, h_M)'$, and $h_i > 0, \forall i = 1, \ldots, M$.

The constraint can be enforced in minimizing Equation 1 by using the Lagrangian function

$$L = g'\Omega g + (g'm - t)^2 + \mu(g'u - 1)$$

with Lagrange-multiplier μ. The optimum is achieved if we set (Tresp and Taniguchi, 1995)

$$h^* = [\Omega + (m - tu)(m - tu)']^{-1} u.$$

[1] Interestingly, even if the estimators are unbiased i. e. $m_i = t \quad \forall i = 1, \ldots, M$ the minimum error estimator is biased which confirms that a biased estimator can have a smaller expected error than an unbiased estimator. As example, consider the case that $M = 1$ and $m_1 = t$ (no bias). Then

$$g^* = \frac{t^2}{t^2 + var_1}$$

Note that this term is smaller than one if $t \neq 0$ and $var_1 > 0$. Then, $E(\hat{t}) = E(g^* f_1) < m_1$, i. e., the minimum expected error estimator is biased!

Now the individual biases $(m_i - t)$ appear explicitly in the weights. For the optimal weights

$$E(\hat{t} - t)^2 = \frac{1}{u'(\Omega + (m - tu)(m - tu)')^{-1}u}.$$

Note, that by using the constraint in Equation 2 the combined estimator is unbiased if the individual estimators are unbiased, which is the main reason for employing the constraint. With unbiased estimators we obtain

$$h^* = \Omega^{-1}u$$

and for the optimal weights $E(\hat{t} - t)^2 = (u'\Omega^{-1}u)^{-1}$. If in addition the individual estimators are uncorrelated we obtain

$$h_i^* = \frac{1}{var_i}.$$

3 Averaging Regularized Estimators

3.1 Training

The previous theory can be applied to the problem of function estimation. Let's assume we have a training data set $L = \{(x^k, y^k)\}_{k=1}^K$, $x^k \in \Re^N$, $y^k \in \Re$. Our goal is to estimate the conditional expected value

$$t(x) = E(y|x).$$

In our experiments we used neural networks as estimator.[2] Let $f_i(x)$ denote the response of the i-th neural network at input x. Each neural network was trained on the training set L, consisting of K samples, to minimize the cost function

$$\text{Cost}_i = \sum_{k=1}^{K}(y^k - f_i(x^k))^2 + \lambda \sum_{j=1}^{J} w_{ij}^2, \quad i = 1, \ldots, M \tag{3}$$

where $\{w_{ij}\}_{j=1}^J$ are the weights in the $i-$th neural estimator and J is the number of weights in each network. The first term is the squared error between the prediction of the neural network and the target in the training data L and the second term is a weight-decay penalty weighted by the regularization parameter $\lambda \geq 0$. Weight decay is commonly used to improve network performance by decreasing variance in prediction for the cost of introducing bias.

It is obvious that averaging is only useful if the individual estimators differ in their prediction. Neural networks trained on an identical data set only vary because the

[2] In the experiments we used standard multi-layer perceptrons with one hidden layer. For details, see Section 4.

optimization was initialized with different random initial weights and the optimization procedure terminates in different local minima. To further decorrelate the individual estimators, Breiman suggested to train the estimators using bootstrap replicates $L_1^B, \ldots, L_M^B$, a procedure Breiman calls bagging predictors (Breiman, 1994). The bootstrap replicate L_i^B is generated by randomly sampling K-times from the original training data set L with replacement. For background on bootstrap techniques, see Efron and Tibshirani (1993).

In the experiment we considered combined estimators which can be written as

$$\hat{t}(x) = \sum_{i=1}^{M} g_i(x) f_i(x) = \frac{1}{n(x)} \sum_{i=1}^{M} h_i(x) f_i(x), \tag{4}$$

where $n(x) = \sum_{i=1}^{M} h_i(x)$ is the normalizing factor and $h_i(x) \geq 0, \forall i = 1, \ldots, M$. Note, that we allow for the possibility that the weighting functions $g_i(x)$ depend on the input x. Also, note that it follows that (compare Section 2.2)

$$\sum_{i=1}^{M} g_i(x) = 1, \qquad g_i(x) \geq 0, \;\; i = 1, \ldots, M.$$

In the experiments, we compare the performances of the combined systems for different choices of $g_i(x)$.

Since in Equation 4 averaging is performed for a fixed input x, we can apply the theory of Section 2 to calculate the optimal weighting functions. Unfortunately, it is not straightforward to obtain reliable estimates of the covariance matrices and the biases. We therefore have to rely on experiments to decide if averaging is useful and in which cases which averaging method should be employed. In the experiments, we use four different combining methods which are motivated by the theoretical results in Section 2. The four combining methods are described in the following sections.

3.2 Simple Averaging (AV)

In simple averaging, we set

$$h_i(x) = 1, \quad i = 1, \ldots, M \;\text{ for all } x.$$

It is easy to come up with examples where simple averaging is optimal. For example if all estimators are unbiased and uncorrelated with identical variances or in general, when symmetry indicates that no single estimator should be preferred. Past experiments have shown that simple averaging can improve performance considerably. For experimental results and theoretical background of simple averaging see Perrone (1993), Jacobs (1995), Krogh and Vedelsby (1995) and Wolpert (1992).

3.3 Bagging (BA)

The only difference to AV is that the individual networks are trained on bootstrap replicates. Set again

$$h_i(x) = 1, \quad i = 1, \ldots, M \text{ for all } x.$$

Although the individual estimators are less correlated, we expect that each individual estimator contains more variance (and possibly more bias) because the training set contains fewer distinct data if compared to the case where each estimator is trained on the complete data set.

3.4 Variance-based Weighting (VW)

In variance-based weighting we set

$$h_i(x) = \frac{1}{var(f_i(x))}, \quad i = 1, \ldots, M.$$

The intuitive idea of the variance-based approach is that when an estimator is uncertain about its own prediction for a certain input then this estimator is not competent for this input and obtains a low weight. From our theoretical considerations it is clear that this is optimal if the errors in the individual estimators are uncorrelated and if the estimators are unbiased. The errors of estimators trained on identical data or on bootstrap replicates are very likely correlated and weight decay introduces bias such that — from a theoretical point of view — variance-based weighting is not optimal. Again, experimental results have to show under which conditions variance-based weighting is useful.

Also, for calculation of the variance it is important to be clear about which random process we average. If the expected value is taken over all random initial weights the variances of all estimators are identical and we would obtain simple averaging. On the other hand if we consider that each estimator has found a different local minimum we can consider each estimator as a distinct model. Now, we only average over the noise on the targets. We consider that the targets were generated by the random process

$$y^k = t(x^k) + \gamma$$

where $t(x^k) = E(y^k|x^k)$ and γ is independent zero-mean noise with variance σ^2.

The variance of an individual predictor for input x can be estimated by a number of different methods (Tibshirani, 1994). We use

$$var(f_i(x)) \approx \sigma^2 \;\; \theta_i(x)^T H_i^{-1} \theta_i(x)$$

where $\theta_i(x) = \frac{\partial f_i(x)}{\partial w_i}$ is the output sensitivity of the neural estimator $f_i(x)$ w.r.t the weights w_i at the input x and where $w_i = (w_{i1}, \ldots, w_{iJ})'$ are the weights in the i-th

estimator. H_i is the Hessian, which can be approximated (for the unregularized estimator) as

$$H_i \approx \sum_{k=1}^{K} \frac{\partial f_i(x^k)}{\partial w_i} \frac{\partial f_i(x^k)}{\partial w_i}^T . \tag{5}$$

x_k is the $k-$th sample in the training data set L.

3.5 Variance-based Bagging (VB)

The only difference to VW is that the neural networks are trained on the bootstrap replicates $\{L_i^B\}_{i=1}^M$ of the original training set L. The Hessian matrix in Equation 5 is now calculated using the bootstrap samples.

4 Experiments

In this section, we present experimental results using two real-world data sets: the Breast Cancer data and the DAX data. The first data set can be obtained from the UCI repository (ftp:ics.uci.edu/pub/machine-learning-databases). The second data set can be obtained by contacting the authors. We compare the four different combining methods described in the previous sections using these two databases with varying degrees of regularization.

In the experiments $M = 25$ neural networks were combined. All neural networks had the same fixed architecture, i.e. a multilayer perceptrons with a single hidden layer of 10 hidden units. The initial weights of the estimators were chosen randomly out of a uniform distribution between -0.2 and +0.2.

We divided the data bases randomly in two independent sets: the training set L and the test set T. For AV and VW each neural estimator was trained on the whole training set. For the bagging-based approaches BA and VB each estimator was trained on the bootstrap replicates L_i^B of the training set L. For training, a quasi-Newton-method with a fixed number of iterations (400 for the Breast Cancer data and 300 for the DAX data) was used. To obtain statistically significant results we repeated each experiment 10 times ($R = 10$ runs) for both data sets. For the Breast Cancer data with relatively few data, we chose a different division into test data and training data for each run.

4.1 Performance Criteria

The averaged summed squared error $ASSE^C(\lambda)$ is the squared test set error of the individual estimators trained on complete data and with weight decay parameter λ averaged over all M estimators and averaged over all R runs. $ASSE^B(\lambda)$ is the equivalent measure for networks trained on bootstrap samples.

$ASSE^{comb}(\lambda)$ with $comb \in \{AV, BA, VW, VB\}$ is the squared test set error of the combining methods with weight decay parameter λ averaged over all R runs.

Furthermore, we define $ASTD^C(\lambda)$ as the averaged standard deviation of the prediction of the neural networks trained with complete data (averaged over all estimators and all runs). $ASTD^B(\lambda)$ is the equivalent measure for networks trained on bootstrap samples.

The mathematical formulas describing the different measures can be found in Appendix A.

4.2 Breast Cancer data

The data base has been recorded at the University of Wisconsin Hospitals, Madison. The data set contains 699 samples with 9 input variables consisting of cellular characteristics and one binary output with 458 benign and 241 malignant cases. All input variables were normalized to zero mean and a standard deviation of one. For every run we divided the data base randomly in $K = 599$ training samples and $P = 100$ test samples.

The Figure 1 (top) shows the averaged performance of the individual estimators as a function of the regularization parameter λ. The large values of both $ASSE^C(\lambda)$ and $ASSE^B(\lambda)$ for $\lambda = 0$ indicate that neural networks without regularization extremely overfit the data. For $\lambda \approx 1.25$ the networks trained on complete data obtain best performance. $ASSE^B(\lambda)$ reaches a minimum at $\lambda = 2$. As expected, $ASSE^B(\lambda)$ is always larger than $ASSE^C(\lambda)$ since the networks trained on bootstrap replicates have seen fewer distinct data. The difference between $ASSE^B(\lambda)$ and $ASSE^C(\lambda)$ decreases with increasing λ.

The Figure 1 (bottom) shows $ASTD^C(\lambda)$ and $ASTD^B(\lambda)$. Both $ASTD^C(\lambda)$ and $ASTD^B(\lambda)$ decrease monotonously with growing λ and $ASTD^B(\lambda)$ is consistently larger. Even for large λ, $ASTD^B(\lambda)$ is still greater than 0.02 whereas $ASTD^C(\lambda)$ is close to zero. The figure clearly demonstrates that bagging increases variance in the prediction.

The test set performances of the different combination methods is plotted in Figure 2 (top). With no regularization, all averaging methods show dramatically better performance if compared to the individual estimators. The variance-based approaches VW and VB are both better than the other averaging methods if λ is close to zero. This result seems to indicate that the variance-based methods successfully recognize the large variance in networks due to local overtraining. With increasing λ the individual networks as well as all averaging methods improve performance. With increasing λ, the performances of AV and VW — the approaches in which the networks were trained on complete data — converge to $ASSE^C(\lambda)$.

In particular for $0.3 < \lambda < 1$ the performance of bagging is impressive and is superior to AV and VW. Most strikingly however, VB shows very good performance for any λ and seems to combine the advantages of both VW and BA.

Note that the optimum for the individual estimators is at a larger degree of regularization than the optimum of the bagging approaches BA and VB. For AV and VW,

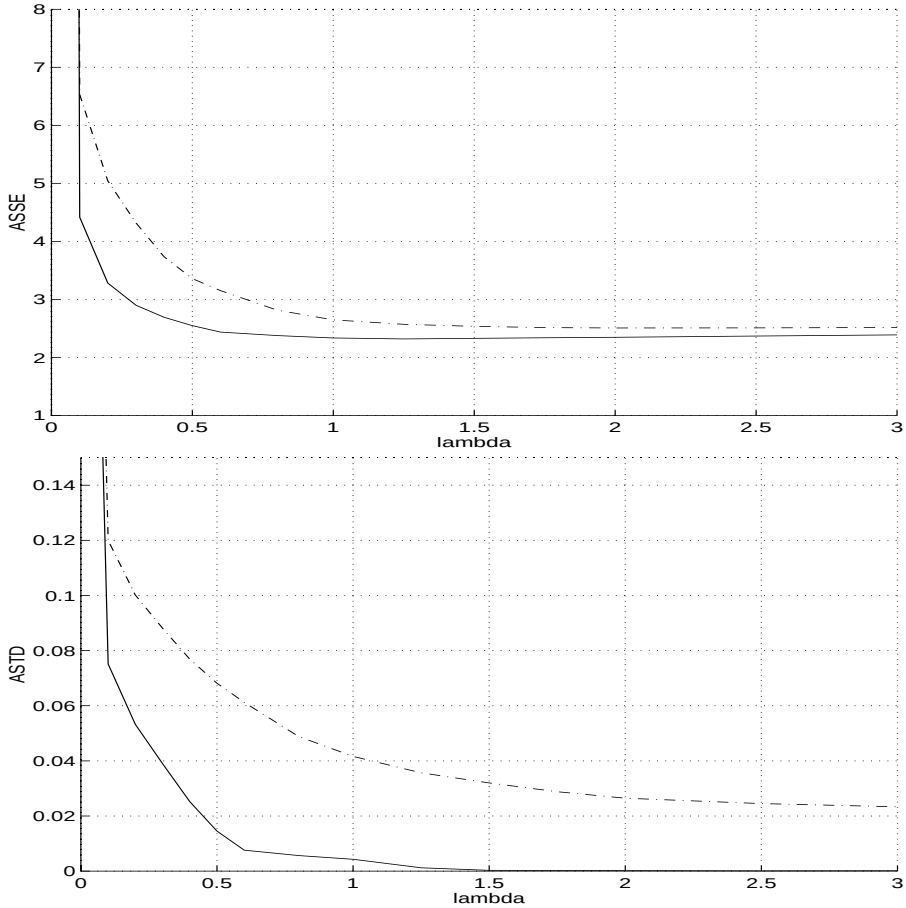

Figure 1: Top: $ASSE^C$ (continuous) and $ASSE^B$ (dash-dotted) as a function of λ for the Breast Cancer data. For $\lambda = 0$, $ASSE^C = 746$ and $ASSE^B = 133$ are out of scale. Bottom: $ASTD^C$ (continuous) and $ASTD^B$ (dash-dotted) as a function of λ for the Breast Cancer data. For $\lambda = 0$, $ASTD^C = 0.49$ and $ASTD^B = 0.55$ are out of scale.

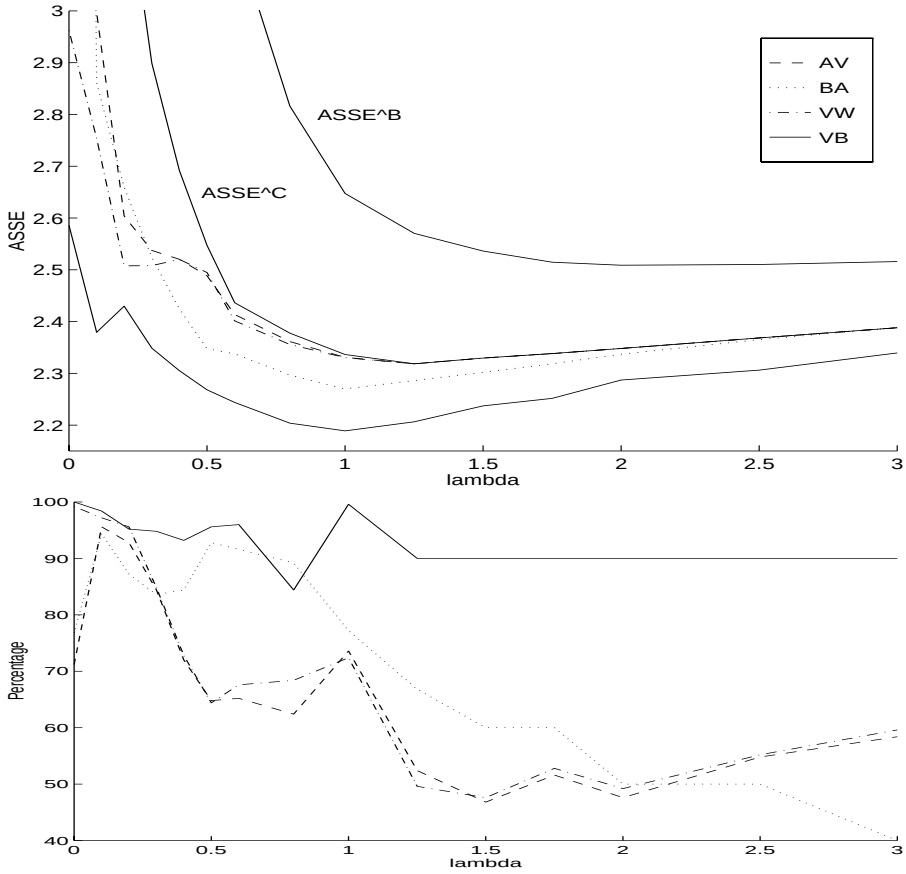

Figure 2: Top: $ASSE^{comb}(\lambda)$ of the different averaging approaches for the Breast Cancer data. Displayed is AV (dashed), BA (dotted), VW (dash-dotted), and VB (lower continuous line). The highest continuous line shows the average performance of the individual bagging networks and the second highest line shows the average performance of the networks trained on the complete data. Bottom: The figure shows the number of single estimators trained on complete data which are worse than the averaging methods for the Breast Cancer data (in percent).

best performance coincides with the best performance of the averaged individual networks trained on complete data.

Figure 2 (bottom) shows the number of single estimators which were worse than the averaging methods in percent. The impressive performance of VB is also apparent: in the large majority of settings for λ, VB is better than all individual networks! Bagging shows excellent performance up to $\lambda \approx 1$ and AV and VW are best for small values of $\lambda <\approx 0.5$.

4.3 DAX data

The DAX data consist of 2564 samples of the Deutschen Aktien Index(DAX)[3] from March 5, 1984 to December 30, 1993. The goal is to predict the value of the DAX of the following day based on past measurements of the DAX. As input variables 12 indicators were calculated (see Appendix B). All inputs and the output were normalized with zero mean and variance of one. The training data set consisted of $K = 2150$ samples (from March 5, 1984 to May 29, 1992) and the test set consisted of $P = 414$ test samples (from June 1, 1992 to December 30, 1993).

The Figure 3 (top) shows $ASSE^C(\lambda)$ and $ASSE^B(\lambda)$ as a function of the regularization parameter λ. The graph shows qualitatively the same characteristics as in the previous experiment. Again, for small λ we see overfitting. $ASSE^C(\lambda)$ is optimal at $\lambda \approx 30$ and $ASSE^B(\lambda)$ is optimal at $\lambda \approx 50$. Again, $ASSE^B(\lambda)$ is always larger than $ASSE^C(\lambda)$ since the networks trained on bootstrap replicates have seen fewer distinct data.

Figure 3 (bottom) shows the average standard deviation of the prediction ($ASTD^C(\lambda)$, $ASTD^B(\lambda)$). Again, for increasing λ, $ASTD^C(\lambda)$ quickly approaches zero whereas $ASTD^B(\lambda)$ still assumes relative large values.

The test set performances of the different combination methods is plotted in Figure 4 (top and center). With no regularization all averaging methods show much better performance if compared to the individual estimators. The variance-based approaches VW and VB are slightly better to the other averaging methods at at $\lambda = 0$. Interestingly, with increasing λ the performances of the averaging methods first decrease but have a global optimum for intermediate values of λ. This can be explained by the drastic improvement of the individual estimators with optimal λ. All averaging methods show excellent performance for $30 < \lambda < 40$. Best performance is achieved for VB at $\lambda = 40$. Note that for $\lambda > 50$, AV and VW are no better than the average of the individual networks whereas BA and VB are still considerably better than the mean of the networks trained on bootstrap samples.

Figure 4 (bottom) shows the number of single estimators which were worse than the averaging methods in percent. Apparent is the impressive performance of all averaging methods if no regularization is used. For $\lambda < 50$ all averaging methods

[3]The DAX represents the average value of the stock of a set of representative companies, analogously to the Dow Jones index in the US.

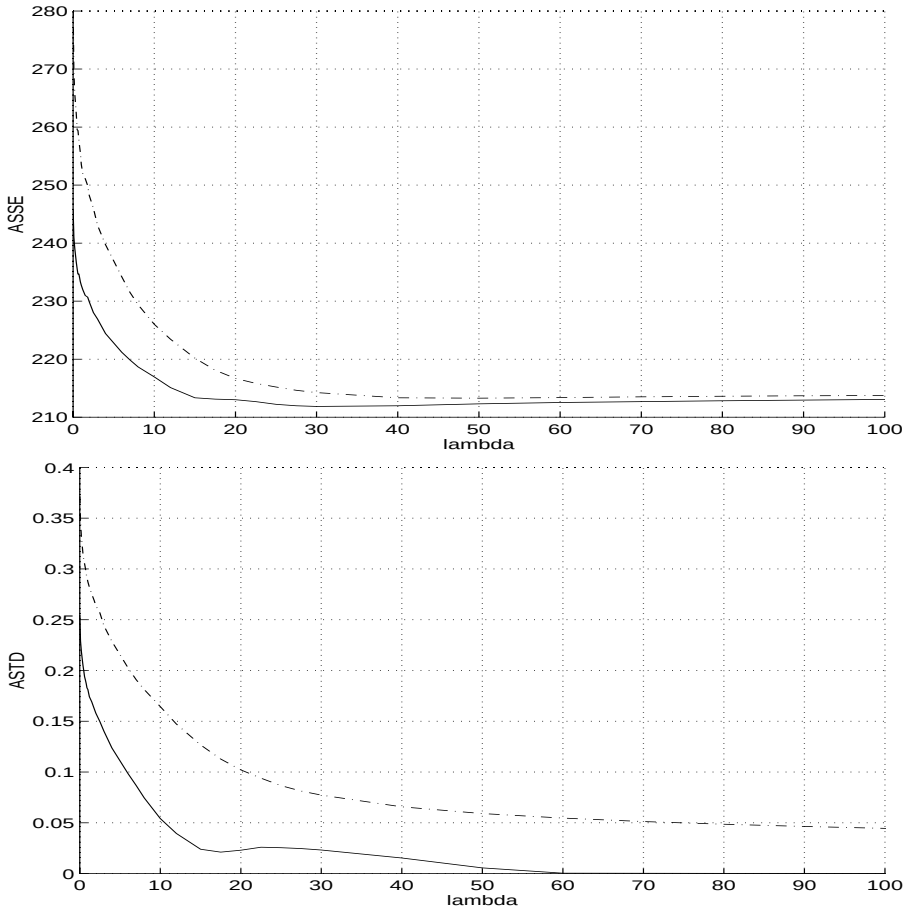

Figure 3: Top: $ASSE^C$ (continuous) and $ASSE^B$ (dash-dotted) as a function of λ for the DAX data. Bottom: $ASTD^C$ (continuous) and $ASTD^B$ (dash-dotted) as a function of λ for the DAX data.

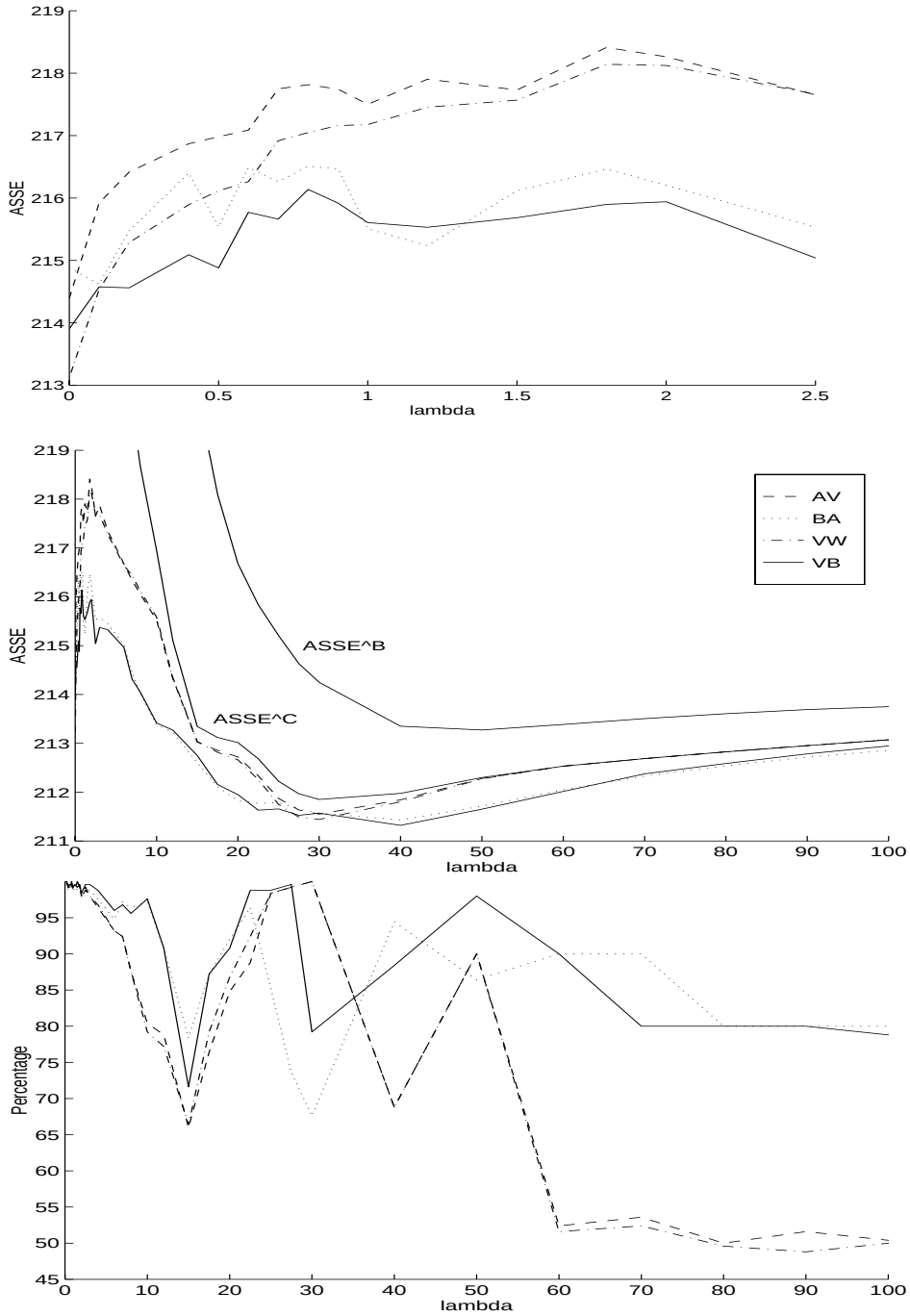

Figure 4: Top and center: $ASSE$ of the different averaging approaches for the DAX data for different scales of λ. Displayed is AV (dashed), BA (dotted), VW (dash-dotted), and VB (lower continuous line). The highest continuous line shows the average performance of the individual bagging networks and the second highest line shows the average performance of the networks trained on the complete data. Bottom: The figure shows the number of single estimators trained on complete data which are worse than the averaging methods for the DAX data (in percent).

are better than 65% of all individual estimators. For $\lambda > 50$ BA and VB are considerably better that the individual estimators whereas the improvement achieved with AV and VW decreases quickly.

5 Discussion

The experiments confirmed the theory but also showed some unexpected results. As predicted, the variance in the networks decreases rapidly with increasing weight-decay parameter λ if networks are trained on complete data as shown in Figures 1 and 3. On the other hand, if networks are trained on bootstrap replicates, we obtain large variance in the networks even at relatively large values of λ. The performance of the individual networks is worse for networks trained on bootstrap replicates since each estimator has seen a smaller number of distinct data. The relative improvement in performance by averaging increases with increasing variance in the estimates and bias hurts. Therefore for all averaging methods the relative improvement is maximum if no weight decay is used. On the other hand the performances of the networks improve with weight decay. So —as confirmed by experiment— regularization also improves the averaged systems. Simple averaging (AV) shows good performance at small values of λ. Even better at small λ is variance-based weighting (VW). The reason is that local overtraining is reflected in large variance. The corresponding estimator obtains consequently a small weight. With increasing λ, the performance of both AV and VW become comparable and both approach the performance of an average individual estimator trained on complete data for large λ. This confirms that all estimators are highly correlated for large λ as already noted in Figures 1 and 3. Bagging (BA) displays better performance than AV and VW up to intermediate values of λ except when λ is extremely small or zero. This can be explained by the fact that training on bootstrap samples results in considerable variance in the networks even for large λ. Variance-based bagging (VB) seems to combine the advantages of both variance-based weighting and bagging. If networks are overtrained they locally have large variance and obtain a small weight locally. Training on bootstrap replicates introduces additional variance in the networks which is particularly useful for large λ. In our first experiment (Breast Cancer data), variance-based bagging was the overall best combining method over a wide range of degrees of regularization. In the second experiment (DAX data) BA and VB show similar performance for intermediate and large values of λ but VB shows superior performance for small λ.

6 Conclusions

Based on our experiments we can conclude that — in comparison with the individual estimators — averaging improves performance at all levels of regularization. In particular we also obtain improvements with respect to optimally regularized estimators, although the degree of improvement is application specific. Averaging is less sensitive with respect to the regularization parameter λ if compared to the

individual estimators. Especially if the individual estimators overfit, averaging still gives excellent performance. Overall, bagging and variance-based bagging which both use networks trained with bootstrap replicates work well for a wide range of values of λ. At extremely small values of λ, variance-based weighting and variance-based bagging are clearly superior to the other averaging approaches.

Acknowledgements

This research was partially supported by the Bundesministerium für Bildung, Wissenschaft, Forschung und Technologie, grant number 01 IN 505 A.

Appendix

A: Performance Criteria

Let $f_{i,j}^{C,\lambda}$ be the i-th estimator on the test set in the j-th run trained on complete training data with regularization parameter λ. The summed squared error for $f_{i,j}^{C,\lambda}$ is defined as

$$SSE_{i,j}^{C}(\lambda) = \sum_{p=1}^{P} \left(y_j^p - f_{i,j}^{C,\lambda}(x_j^p)\right)^2$$

where $T_j = \{(x_j^p, y_j^p)\}_{p=1}^{P}$ are the test data in the j-th run. As performance criterion for the individual estimators we use the averaged summed squared error where we average over all M estimators and all R runs

$$ASSE^{C}(\lambda) = \frac{1}{R}\sum_{j=1}^{R}\frac{1}{M}\sum_{i=1}^{M} SSE_{i,j}^{C}(\lambda).$$

The performance measures for networks trained on bootstrap replicates $SSE_{i,j}^{B}(\lambda)$ and $ASSE^{B}(\lambda)$ are defined analogously.

To measure the performance of combining methods we define similarly

$$SSE_{j}^{comb}(\lambda) = \sum_{p=1}^{P} \left(y_j^p - \hat{t}_j^{\lambda}(x_j^p)\right)^2$$

where $\hat{t}_j^{\lambda}$ is the response of the combined system from the Equation 4 in the j-th run for regularization parameter λ. The averaged SSE for combining systems is defined as

$$ASSE^{comb}(\lambda) = \frac{1}{R}\sum_{j=1}^{R} SSE_{j}^{comb}(\lambda)$$

where $comb \in \{AV, BA, VW, VB\}$.

The averaged standard deviation of the prediction of the neural networks trained with complete data is defined as

$$ASTD^C(\lambda) = \frac{1}{R}\sum_{j=1}^{R}\frac{1}{P}\sum_{p=1}^{P}\sqrt{\frac{1}{M}\sum_{i=1}^{M}(f_{i,j}^{C,\lambda}(x_j^p) - \bar{f}_j^{C,\lambda}(x_j^p))^2}$$

with

$$\bar{f}_j^{C,\lambda}(x_j^p) = \frac{1}{M}\sum_{i=1}^{M} f_{i,j}^{C,\lambda}(x_j^p).$$

$ASTD$ is a measure of the degree of variance between the individual networks. The corresponding measure for networks trained on bootstrap replicates $ASTD^B(\lambda)$ is defined analogously.

B: The DAX inputs

Table 1: Input variables for predicting the DAX at day $t+1$.

Input Variable	**Description**
1	$y(t)/y(t-1)$
2	$\log(y(t)) - 1/5\sum_{i=0}^{4}\log(y(t-i))$
3	$\log(y(t)) - 1/10\sum_{i=0}^{9}\log(y(t-i))$
4	$(\log(y(t)) - \log(y(t-5))) - (\log(y(t)) - \log(y(t-6)))$
5	$(\log(y(t)) - \log(y(t-10))) - (\log(y(t)) - \log(y(t-11)))$
6	$\mathrm{rsi}(\log(y(t)), 5)$
7	$\mathrm{rsi}(\log(y(t)), 10)$
8	$\frac{\log(y(t)) - \min_{i=1,\ldots,4}(\ln(y(t-i)))}{\max_{i=1,\ldots,4}(\log(y(t-i))) - \min_{i=1,\ldots,4}(\ln(y(t-i)))}$
9	$\frac{\log(y(t)) - \min_{i=1,\ldots,9}(\ln(y(t-i)))}{\max_{i=1,\ldots,9}(\log(y(t-i))) - \min_{i=1,\ldots,9}(\ln(y(t-i)))}$
10	$1/3\sum_{i=0}^{2}\frac{\log(y(t-i)) - \min_{i=1,\ldots,4}(\ln(y(t-i)))}{\max_{i=1,\ldots,4}(\log(y(t-i))) - \min_{i=1,\ldots,4}(\ln(y(t-i)))}$
11	$1/3\sum_{i=0}^{2}\frac{\log(y(t-i)) - \min_{i=1,\ldots,9}(\ln(y(t-i)))}{\max_{i=1,\ldots,9}(\log(y(t-i))) - \min_{i=1,\ldots,9}(\ln(y(t-i)))}$
12	$\mathrm{vol}(t)/\mathrm{vol}(t-1)$

Table 1 describes the input variables used in the DAX data. $y(t)$ is the DAX at day t. $\mathrm{vol}(t)$ is the total volume of the transactions at the German stock market at day t. Furthermore,

$$\mathrm{rsi}(y(t), n) = \frac{\sum_{i=0}^{n-1} b(y(t-i) - y(t-i-1))}{\sum_{i=0}^{n-1} |y(t-i) - y(t-i-1)|} \quad \text{with} \quad b(y(t)) = \begin{cases} 0 & \text{if } y(t) \le 0 \\ y(t) & \text{if } y(t) > 0 \end{cases}$$

For a motivation of the preprocessing and further details see Dichtl (1995).

References

Breiman, L. (1994). Bagging Predictors. TR No. 421, Department of Statistics, University of California.

Dichtl, H. (1995). Zur Prognose des Deutschen Aktienindex DAX mit Hilfe von Neuro-Fuzzy-Systemen. *Beiträge zur Theorie der Finanzmärkte,* 12, Institut für Kapitalmarktforschung, J. W. Goethe-Universität, Frankfurt.

Efron, B. and Tibshirani, R. (1993). *An Introduction to the Bootstrap.* Chapman and Hall.

Jacobs, R. A. (1995). Methods for Combining Experts' Probability Assessment. *Neural Computation,* 7, pp. 867-888.

Krogh, A. and Vedelsby, J. (1995). Neural Network Ensembles, Cross Validation, and Active Learning. *Advances in Neural Information Processing Systems 7.* Cambridge MA: MIT Press.

Meir, R. (1995). Bias, Variance and the Combination of Least Squares Estimators. *Advances in Neural Information Processing Systems 7.* Cambridge MA: MIT Press.

Perrone, M. P. (1993). *Improving Regression Estimates: Averaging Methods for Variance Reduction with Extensions to General Convex Measure Optimization.* PhD thesis. Brown University.

Taniguchi, M. and Tresp, V. (1995). Variance-based Combination of Estimators trained by Bootstrap Replicates. *Proc. Inter. Symposium on Artificial Neural Networks*, Hsinchu, Taiwan.

Tibshirani, R. (1994). A Comparison of Some Error Estimates for Neural Network Models. TR Department of Statistics, University of Toronto.

Tresp, V. and Taniguchi, M. (1995). Combining Estimators Using Non-Constant Weighting Functions. *Advances in Neural Information Processing Systems 7.* Cambridge MA: MIT Press.

Wolpert, D. H. (1992). Stacked Generalization. *Neural Networks*, Vol. 5, pp. 241-159.