Weighted Bagging Predictors

Nino Arsov[1], Martin Pavlovski[1], and Ljupco Kocarev[1,2]

[1] Macedonian Academy of Sciences and Arts
Skopje, Macedonia
[2] Faculty of Computer Science and Engineering,
Ss. Cyril and Methodius University, Skopje, Macedonia
narsov@manu.edu.mk; martin.pavlovski@cs.manu.edu.mk; lkocarev@manu.edu.mk

Abstract. We introduce weighted bagging by combining bootstrap-trained predictors with weighted aggregation. For numerical outputs, we take the weighted average of the predictors, while when the outputs are binary/multi class labels, we apply weighted majority voting. The proposed method is independent of the underlying learning algorithm and is applicable to prediction-ready models of any type. We demonstrate the performance of the method against bagging for regression (12 datasets), binary classification (15 datasets), and multiclass classification (15 datasets). For regression, improvements larger than 5% are observed in 5 out of 12 datasets with average error decrease of 7.5%. For binary classification we found that the generalization error decreases on average 13.18%, with minimum decrease being 5% while maximum decrease being 31.16%. Finally, for multiclass classification error decrease is in the range 1% and 50% with average value of 12.84%; decrease larger than 5% has been observed in 10 out of 15 datasets.

Keywords: bagging, weighted bagging, regression, binary classification, multiclass classification

1 Introduction

For the last two decades, bootstrap aggregating (or bagging) (Breiman, 1996a) and boosting (Schapire and Freund, 2012) have been the two most exploited examples of predictive ensembles. Both approaches manipulate the training data in order to generate different predictors. Bagging produces replicate training sets by sampling with replacement from the training instances. Boosting uses all instances at each repetition, but maintains a weight for each instance in the training set that reflects its importance. Therefore, adjusting the weights causes the learner to focus on different instances and so leads to different predictors. In either case, the multiple predictors are then combined by averaging/voting to form a composite predictor. In bagging, each component predictor has the same weight/vote, while boosting assigns different weighting/voting strengths to component predictors on the basis of their accuracy.

Bagging (Breiman, 1996a) is a simple and effective way to reduce the error rate of many learning algorithms. Whether bagging improves accuracy depends crucially on the stability of the procedure for constructing the predictor: improvement occurs only for unstable procedures (Breiman, 1996a). Bagging generates diverse predictors only if the base leaning algorithm is unstable, that is, if small changes to the training set cause large changes in the learned algorithm. Instability was studied by Breiman (1996c) where it was pointed out that neural nets, classification and regression trees, and subset selection in linear regression are unstable methods. Several explanations why bagging works have been provided for both regression and classification. For regression, Breiman (1996a) explains bagging by considering bias/variance decomposition. This approach has been extended to bias/variance/covariance decomposition for ensembles in (Ueda and Nakano, 1996). Since the correlation between learners can be negative, the covariance term may decrease the expected loss of the ensemble while leaving bias and variance unchanged (Brown, 2004; Brown et al, 2005b; Pisetta, 2012).

In classification, bias/variance/covariance decomposition is hard to obtain and to the best of our knowledge, there are no such decompositions (Pisetta, 2012). An exception may be the work of Zanda (2010) where the bias/variance/covariance decomposition of Ueda and Nakano (1996) is applied to classification by considering probability estimation rather than label estimation. Since the real valued outputs are taken to be the respective probabilities, which are then used in a decision rule to forecast the most likely (categorical) value for the output, the work of Zanda (2010) does not concern strictly classification. Moreover, Friedman (1997) has pointed out that more accurate class probability estimates do not necessarily lead to better classification performance and often can make it worse. Different strategies have been employed to explain why bagging works for classification. Thus, for example, bagging is related to the notion of an order-correct learner (Breiman, 1996a). If a predictor is good in the sense that it is order-correct for most inputs, then aggregation can transform it into a nearly optimal predictor. However, in contrast to regression, for classification poor predictors can be transformed into worse ones. Domingos (1997) shows that bagging works because it effectively shifts the prior to a more appropriate region of model space. Brown et al (2005a) by reviewing diversity in regression and classification provide additional explanation why ensemble approaches to classification and regression outperform single predictors on a wide range of tasks. For recent reviews on ensemble approaches for regression and classification we refer the reader to (Mendes-Moreira et al, 2012) and (Ren et al, 2016).

Several modifications/extensions of bagging have been proposed. Wagging assigns random weights with Gaussian distribution, (Bauer and Kohavi, 1999), and Poisson distribution, (Webb, 2000), to the instances in each training set replicate. The pruning of bagging is proposed in (Hernández-Lobato et al, 2006) in order to reduce the ensemble size without meaningfully reducing the accuracy of the ensemble predictions. Subagging, (Bühlmann, 2012), which obtains each base model using a random subset of the examples, is suggested to reduce the computational cost of bagging. Input smearing, (Frank and Pfahringer, 2006), is yet another technique for improving bagging by increasing the diversity of the ensemble by adding Gaussian noise to the inputs. Breiman (2000) suggests output smearing in which Gaussian noise is added to the target variable of the training set. Breiman (2001) also suggests a technique called iterated bagging in which the training set used to generate the new model in each iteration is obtained by replacing the output targets with the errors of the current ensemble.

In this paper we propose a more effective way of combining bootstrap-trained predictors by weighted aggregation. The proposed approach extends the existing theoretical background established in (Breiman, 1996a) and potentially improves aggregated predictors. It is independent of the underlying learning algorithm and is applicable to prediction-ready models of any type. The basic principle of our approach is to further decrease the mean squared error of the aggregated predictor. When the outputs are numerical, we take the *weighted average* of the predictors. On the other hand, when the outputs are binary class labels, we apply *weighted majority voting.* There are strong arguments to diversify an ensemble by weighting. A calculation in (Efron and Tibshirani, 1994) shows that a bootstrap contains $0.632n$ distinct observations, which is a large portion of n, especially when n is large. It is clear that the bagged predictors are not identical and they have diverse prediction performance. Some generalize better than the others, and therefore, a logical way of re-balancing is to assign a weight to each predictor. On the other hand, bagging is known for influence equalization, discussed in (Grandvalet, 2004), which means that all predictors have the same influence in the joint prediction. With our approach, the generated ensemble is fine-tuned to reduce variance among the predictors further, lower than bagging. However, the approach suffers from the same uncertainties as bagging and its effectiveness depends on how accurately can bootstrap sampling approximate the data distribution. We demonstrate the performance of the method against classical bagging on 42 (12 regression, 15 binary classification, and 15 multiclass classification) datasets. In 30 of these datasets we found that the generalization error decreases in the range from 5% to 50%.

This is the outline of the paper. Section 2 provides brief introduction to bagging and why it works. This part has been replicated here from (Breiman, 1996a) for completeness. In Section 3 we introduce weighted bagging and provide explanation why it works. The weights are calculated first by assigning a sequence of numerical weights corresponding to each learning set and then choosing the weights such that the mean squared prediction error is minimal, by setting the first-order partial derivatives with respect to the weight of each predictor (learning set respectfully) to zero. Section 4 presents our experimental results together with testing strategy and experimental setup. We use a unifying approach for both regression and classification. The problems on which the approach has been tested are described, and then the results are presented for regression, Section 4.1, for binary classification, Section 4.2, and for multiclass classification, Section 4.3. We conclude the paper with Section 5.

2 Bagging and Why It Works

Let $\mathcal{L}$ be a learning set of data $\{(\boldsymbol{x}_i, y_i),\ i = 1, \ldots, n)\}$, where the y's are numerical responses. For brevity and without loss of generalization, we assume that there is no erroneous assignment of numerical responses, i.e. there is no output noise. We have a learning procedure to form a predictor $\varphi(\boldsymbol{x}, \mathcal{L})$ using $\mathcal{L}$, where we use $\varphi(\boldsymbol{x}, \mathcal{L})$ to predict y when the input is $\boldsymbol{x}$.

Bagging is an ensemble-generation method based on predictor aggregation. The aggregated predictors are individually formed using a sequence of learning sets $\{\mathcal{L}_k\}$ from the same underlying distribution of $\mathcal{L}$, called *replicates* of $\mathcal{L}$. The basic idea behind bagging is to construct a better predictor φ_A by aggregating predictors obtained from the learning sets in $\{\mathcal{L}_k\}$. The way in which predictor aggregation is effectuated is dependent on the learning problem type, or more specifically, the type of y. If y is numerical, then a logical way to aggregate predictors is by averaging over k, i.e. taking $\varphi_A(\boldsymbol{x}) = \mathbf{E}_{\mathcal{L}}\varphi(\boldsymbol{x}, \mathcal{L})$ (the subscript A in φ_A denotes aggregation and $\mathbf{E}_{\mathcal{L}}$ is the expectation over $\mathcal{L}$). For multi-class classification problems, where y is categorical, $\varphi(\boldsymbol{x}, \mathcal{L})$ is replaced by majority voting $\varphi_A(\boldsymbol{x}) = \arg\max_j n_j$ to predict a class $j \in \{1, \ldots, J\}$, where $n_j = |\{k; \varphi(\boldsymbol{x}, \mathcal{L}_k) = j\}|$.

In reality, though, we are often stuck with a single learning set without having the luxury of having replicates of $\mathcal{L}$ (Breiman, 1996a). However, we can imitate the process by using a single learning set, from which we repeatedly draw samples to form a sequence of learning sets $\{\mathcal{L}^{(B)}\}$. These sets are uniformly drawn with replacement from $\mathcal{L}$ and each contains n independent observations. Each learning set in $\{\mathcal{L}^{(B)}\}$ is called a *bootstrap*, hence the name *bootstrap aggregation*. In bagging, if y is numerical, we take the aggregated bagging predictor $\varphi_B(\boldsymbol{x})$ to be

$$\varphi_B(\boldsymbol{x}) = av_B\varphi(\boldsymbol{x}, \mathcal{L}^{(B)}),$$

while for classification we let the predictors in $\{\varphi(\boldsymbol{x}, \mathcal{L}^{(B)})\}$ vote to form $\varphi_B(\boldsymbol{x})$. The notations and equations above are adopted from (Breiman, 1996a).

One of the main concepts Breiman used to explain the effectiveness of bagging is that of *model variance*. The variance of a predictor is a component in the Bias-Variance decomposition of the error Err of a predictive model, $Err = Bias^2 + Variance + \eta$, where η is the persisting noise that the error cannot be cleansed of. High-variance predictors are related to overfitting and instability in particular, where small changes in the learning sets result in significant perturbations in hypotheses' predictions. The error from variance is effectively, although not necessarily, reduced by aggregation. The bias and variance are the two most important components of the generalization error of a predictor, and balancing bias against variance provides the best generalization performance (Brown et al, 2005b). In this sense, bagging is particularly effective because it reduces variance, while preserving a relatively unchanged magnitude of bias ((Breiman, 1996a) provides a thorougher insight).

Let each observation $(\boldsymbol{x}, y)$ in $\mathcal{L}$ be independently drawn from a probability distribution D and assume that the y's are numerical. Then, the aggregated predictor over D is given by $\varphi_A(\boldsymbol{x}, D) = \mathbf{E}_{\mathcal{L}}\varphi(\boldsymbol{x}, \mathcal{L})$.

If $\boldsymbol{X}$ and Y are random variables having the distribution D, independent of $\mathcal{L}$, then the mean squared prediction error ε of $\varphi(\boldsymbol{x}, \mathcal{L})$ is

$$\varepsilon = \mathbf{E}_{\mathcal{L}} \mathbf{E}_{Y,\boldsymbol{X}} (Y - \varphi(X, \mathcal{L}))^2.$$

The squared error of the aggregated predictor $\varphi_A(\boldsymbol{X}, D)$ is

$$\varepsilon_A = \mathbf{E}_{Y,\boldsymbol{X}} (Y - \varphi_A(\boldsymbol{X}, D))^2.$$

Using $(\mathbf{E}Z)^2 \leq \mathbf{E}Z^2$, it turns out that

$$\begin{aligned}
\varepsilon &= \mathbf{E}Y^2 - 2\mathbf{E}(Y \mathbf{E}_{\mathcal{L}} \varphi(\boldsymbol{X}, \mathcal{L})) + \mathbf{E}_{Y,\boldsymbol{X}} \mathbf{E}_{\mathcal{L}} \varphi^2(\boldsymbol{X}, \mathcal{L}) \\
&= \mathbf{E}Y^2 - 2\mathbf{E}Y\varphi_A + \mathbf{E}_{Y,\boldsymbol{X}} \mathbf{E}_{\mathcal{L}} \varphi^2(\boldsymbol{X}, \mathcal{L}) \\
&\geq \mathbf{E}Y^2 - 2\mathbf{E}Y\varphi_A + \mathbf{E}_{Y,\boldsymbol{X}} (\mathbf{E}_{\mathcal{L}} \varphi(\boldsymbol{X}, \mathcal{L}))^2 \\
&= \mathbf{E}Y^2 - 2\mathbf{E}Y\varphi_A + \mathbf{E}_{Y,\boldsymbol{X}} \varphi_A^2 = \mathbf{E}_{Y,\boldsymbol{X}} (Y - \varphi_A)^2 = \varepsilon_A
\end{aligned}$$

The intensity by which bagging improves a single predictor depends on how unequal the two sides of $(\mathbf{E}_{\mathcal{L}} \varphi(\boldsymbol{X}, \mathcal{L}))^2 \leq \mathbf{E}_{\mathcal{L}} \varphi(\boldsymbol{X}, \mathcal{L})^2$ are (Breiman, 1996a). Reducing variance by aggregation manifests a clear improvement in cases where instability is an issue leading to poor generalization performance.

3 Weighted Bagging and Why It Works

Bagging uses bootstrap sampling to imitate the process of replicating the learning set $\mathcal{L}$. In this section we propose a more effective way of combining bootstrap-trained predictors by weighted aggregation. Weighting starts with modifying the aggregated predictor φ_A by assigning a sequence of numerical weights $\{\gamma_{\mathcal{L}}\}$ corresponding to each learning set replicate $\mathcal{L}$ used to form a constituent predictor in the bagging ensemble, i.e.

$$\varphi_A(\boldsymbol{x}, D; \{\gamma_{\mathcal{L}}\}) = \mathbf{E}_{\mathcal{L}} \gamma_{\mathcal{L}} \varphi(\boldsymbol{x}, \mathcal{L}).$$

Using random variables, the average squared prediction error of $\varphi_A(\boldsymbol{X}, D)$ is

$$\varepsilon_A = \mathbf{E}_{Y,\boldsymbol{X}} (Y - \mathbf{E}_{\mathcal{L}} \gamma_{\mathcal{L}} \varphi(\boldsymbol{x}, \mathcal{L}))^2.$$

Next, the weights $\{\gamma_{\mathcal{L}}\}$ are chosen such that ε_A is minimal. The method of first partial derivatives w.r.t. each $\gamma_{\mathcal{L}}$ is used to find the extreme points of the mean squared error. Fixing a learning set replicate $\mathcal{L}_k$ yields the following expansion of the first partial derivative of ε_A:

$$\begin{aligned}
&\frac{\partial}{\partial \gamma_{\mathcal{L}_k}} \mathbf{E}_{Y,\boldsymbol{X}} (Y - \mathbf{E}_{\mathcal{L}} \gamma_{\mathcal{L}} \varphi(\boldsymbol{X}, \mathcal{L}))^2 \\
&= \mathbf{E}_{Y,\boldsymbol{X}} \frac{\partial}{\partial \gamma_{\mathcal{L}_k}} (Y - \mathbf{E}_{\mathcal{L}} \gamma_{\mathcal{L}} \varphi(\boldsymbol{X}, \mathcal{L}))^2 \\
&= -2\mathbf{E}_{Y,\boldsymbol{X}} \left[(Y - \mathbf{E}_{\mathcal{L}} \gamma_{\mathcal{L}} \varphi(\boldsymbol{x}, \mathcal{L})) \varphi(\boldsymbol{X}, \mathcal{L}_k) \right] \\
&= -2 \left[\mathbf{E}_{Y,\boldsymbol{X}} Y \varphi(\boldsymbol{X}, \mathcal{L}_k) - \mathbf{E}_{Y,\boldsymbol{X}} (\varphi(\boldsymbol{X}, \mathcal{L}_k) \mathbf{E}_{\mathcal{L}} \gamma_{\mathcal{L}} \varphi(\boldsymbol{x}, \mathcal{L})) \right],
\end{aligned}$$

and the second partial derivative of ε_A is

$$\frac{\partial^2}{\partial \gamma_{\mathcal{L}_k}} \mathbf{E}_{Y,\boldsymbol{X}}(Y - \mathbf{E}_{\mathcal{L}} \gamma_{\mathcal{L}} \varphi(\boldsymbol{X}, \mathcal{L}))^2 = \mathbf{E}_{Y,\boldsymbol{X}} \varphi^2(\boldsymbol{X}, \mathcal{L}_k) \geq 0.$$

Therefore, the $\gamma_{\mathcal{L}}$'s minimize the modified ε_A due to convexity at its extreme points. Setting the first-order partial derivative w.r.t. $\gamma_{\mathcal{L}_k}$ to zero yields

$$\mathbf{E}_{Y,\boldsymbol{X}}(\varphi(\boldsymbol{X}, \mathcal{L}_k) \mathbf{E}_{\mathcal{L}} \gamma_{\mathcal{L}} \varphi(\boldsymbol{x}, \mathcal{L})) = \mathbf{E}_{Y,\boldsymbol{X}} Y \varphi(\boldsymbol{X}, \mathcal{L}_k),$$

which is a linear equation with an unknown variable $\gamma_{\mathcal{L}_k}$. Expanding the remaining partial derivatives and setting them to zero results in a system of $\#\{\gamma_{\mathcal{L}}\}$ linear equations with $\#\{\gamma_{\mathcal{L}}\}$ variables. Finding a solution of the system is the key to weighted bagging.

Since the true underlying distribution D of $\boldsymbol{X}$ and Y is unknown, D is approximated using the available observed data in the given learning set $\mathcal{L}$. The bagged predictor $\varphi_B(\boldsymbol{X}, D_{\mathcal{L}})$ is then, of course, only an estimate of the true aggregated predictor $\varphi_A(\boldsymbol{X}, D)$, where $D_{\mathcal{L}}$ is the bootstrap approximation to D with a mass of $1/N$ at any case $(\boldsymbol{x}, y) \in \mathcal{L}$ (Breiman, 1996a). Given a sequence $\{\mathcal{L}^{(B)}\}$ of n_B bootstrap samples like in classical bagging to approximate D. Then, the bagged estimate $\varphi_B(\boldsymbol{x}, D_{\mathcal{L}}; \{\gamma_{\mathcal{L}^{(B)}}\})$ of $\varphi_A(\boldsymbol{x}, D; \{\gamma_{\mathcal{L}}\})$ is

$$\varphi_B(\boldsymbol{x}, D_{\mathcal{L}}; \{\gamma_{\mathcal{L}^{(B)}}\}) = \mathbf{E}_{\mathcal{L}^{(B)}} \gamma_{\mathcal{L}^{(B)}} \varphi(\boldsymbol{x}, \mathcal{L}^{(B)}),$$

where $\mathbf{E}_{\mathcal{L}^{(B)}} \gamma_{\mathcal{L}^{(B)}} \varphi(\boldsymbol{x}, \mathcal{L}^{(B)})$ is estimated using the first moment around zero of the expectation

$$\begin{aligned}\mathbf{E}_{\mathcal{L}^{(B)}} \gamma_{\mathcal{L}^{(B)}} \varphi(\boldsymbol{x}, \mathcal{L}^{(B)}) &= \sum_{\{\mathcal{L}^{(B)}\}} \gamma_{\mathcal{L}^{(B)}} \varphi(\boldsymbol{x}, \mathcal{L}^{(B)}) p(\mathcal{L}^{(B)}) \\ &= \frac{1}{n_B} \sum_{\{\mathcal{L}^{(B)}\}} \gamma_{\mathcal{L}^{(B)}} \varphi(\boldsymbol{x}, \mathcal{L}^{(B)}),\end{aligned}$$

since $p(\mathcal{L}^{(B)}) = 1/n_B$. The equation above is the bagged estimate of the aggregated predictor φ_A.

Plugging the bagged estimate into the expansion of the first-order partial derivatives of the mean squared errors and fixing a bootstrap sample $\mathcal{L}^{(b)})$ gives a system of n_B linear equations with n_B variables, each being of the form

$$\frac{1}{n_B} \sum_{\{\mathcal{L}^{(B)}\}} \gamma_{\mathcal{L}^{(B)}} \mathbf{E}_{Y,\boldsymbol{X}}(\varphi(\boldsymbol{X}, \mathcal{L}^{(b)}) \varphi(\boldsymbol{X}, \mathcal{L}^{(B)})) = \mathbf{E}_{Y,\boldsymbol{X}} Y \varphi(\boldsymbol{X}, \mathcal{L}^{(b)}),$$

where the $\gamma_{\mathcal{L}^{(B)}}$'s form the linear system's solution. With this approach, the generated ensemble is fine-tuned to reduce variance further, lower than bagging. While error improvement over the learning set $\mathcal{L}$ is ascertained, the approach still suffers from the same uncertainties as bagging and its effectiveness depend on how accurate our estimates of the expectation $\mathbf{E}_{Y,\boldsymbol{X}}$ are - just like in bagging. When equal-influence bagging is optimal for $\mathcal{L}$, the $\gamma_{\mathcal{L}^{(B)}}$'s will all be set to 1.

The coefficient matrix $\mathbf{A}_{[a]}$ of the linear system, of order $n_B \times n_B$ is given by $a_{ij} = \mathbf{E}_{Y,\boldsymbol{X}}(\varphi(\mathcal{L}, \mathcal{L}^{(i)})\varphi(\mathcal{L}, \mathcal{L}^{(j)}))$, $i = 1, \ldots, n_B$, $j = 1, \ldots, n_B$. Therefore, $\mathbf{G}$ is a symmetric square matrix having positive diagonal elements, which is a strongly implying, but not an ascertained precondition for a positive-definite matrix. Symmetric positive-definite matrices are nonsingular and form linear systems with unique solutions.

4 Experimental Results

In this section we demonstrate the proposed weighted bagging method for regression, binary classification, and multiclass classification on different datasets. We use Classification and Regression Trees (CART's), proposed by Breiman et al (1984) as predictors. The trees, in which the depth is not externally controlled nor limited, are applied to regression, binary and multiclass classification problems. Though the application of the proposed method to regression problems is more than clear, we also apply it to both binary and multiclass classification problems where responses are limited to $\{-1, 1\}$ or $\{1, \ldots, J\}$, respectively.

Some multiclass problems can be transformed into binary problems since the underlying nature of the data allows it to (we refer the reader to the work of Allwein et al (2001) that gives an insight). On the other hand, in cases when multiclass cannot be reduced to binary classification in a straightforward manner, statistical learning has unfolded transformation techniques like *one-versus-rest* classification (or *one-versus-one*), where binary models are built separately to distinguish between each class $j = 1, \ldots, J$ and all other $k \neq j$. The usual one-versus-rest approach has also been extended to *pairwise coupling* of classes (Rifkin and Klautau, 2004; Hastie et al, 1998). One-versus-rest introduces a potential class imbalance risk, but requires only J models, while one-versus-one requires substantially more, i.e. $J(J-1)/2$ models, which increases the number of unknown parameters (weights) $n_B(J-1)/2$ times. For all three types of problems (regression, binary and multiclass classification) we use the regression variant of CART to mimic confidence in the decisions. Moreover, a real-valued classification algorithm is a compulsory prerequisite of one-versus-rest classification techniques.

An important concern is how the expected values $\mathbf{E}_{Y,\boldsymbol{X}}(\cdot)$ are estimated. To estimate $\mathbf{E}_{Y,\boldsymbol{X}}(Y, \varphi(\boldsymbol{X}, \mathcal{L}^{(b)})), b \in [1, n_B]$ or $\mathbf{E}_{Y,\boldsymbol{X}}(\varphi(\boldsymbol{X}, \mathcal{L}^{(i)})\varphi(\boldsymbol{X}, \mathcal{L}^{(j)}))$, $i \neq j, i, j \in [1, n_B]$, one can only use the available data in the learning set $\mathcal{L}$. The obvious approach is to use the first moment around zero and take the average, but a more critical question is how to approximate $\boldsymbol{X}$ and Y. Although improvement at the learning set $\mathcal{L}$ is guaranteed, we need better strategies to engender improvement at the testing set. Taking the corresponding bootstrap sample $\mathcal{L}^{(b)}, b \in [1, n_B]$ by which a single predictor was constructed would clearly lead the bagging ensemble to overfit the learning set, leading to poor generalization performance. Thus, we use the whole $\mathcal{L}$ to approximate $\boldsymbol{X}$ and Y in all estimations of the expected values above. Taking into account that about a third of the data in $\mathcal{L}$ is not known to a single predictor, the ensemble adjusts the weights $\{\gamma_{\mathcal{L}^{(B)}}\}$ based on unobserved data as well. The reasoning is that it leads to “better” generalization performance. Other strategies involve V-Fold Cross Validation (CV), with V being anywhere between 2 and N, to stabilize $\{\gamma_{\mathcal{L}^{(B)}}\}$ by averaging over the V folds, but are omitted in this paper since they showed to be no better that using $\mathcal{L}$ only once to approximate the expected values.

The datasets are used as provided, that is, no preprocessing techniques like noise/outlier detection, feature normalization or selection have been applied in order to test the robustness and adaptivity of our approach to different kinds of unprocessed data. Without loss of generalization, it is assumed that there is no output noise, i.e. there is no erroneous labeling of the responses provided in each dataset. A number of regression, binary and multiclass classification problems have been selected and elaborated separately. A unified experimenting approach is used for all three, though.

To test our approach, we use 10-Fold CV to evaluate the generalization performance. Each fold is taken to be a testing set $\mathcal{L}_{TS}$ exactly once, while the rest form the learning set $\mathcal{L}$. There are ten different pairs of learning and testing sets. Each fold is randomly drawn and complementary to the remaining nine. For regression, the standard 10-Fold CV is used, where folds are sampled randomly and without replacement. Stratified 10-Fold CV is used for binary and multiclass classification, where stratified sampling without replacement is used to randomly draw the ten folds. Stratified sampling preserves the original class frequencies in all folds the same as in the initial set $\mathcal{L}$ of all available data. In both cases, in each of the ten CV realizations, 10% of the data are used for testing. We used CV for all datasets, except for two cases where additional testing sets were provided by the source. All experiments use $n_B = 30$ predictors (regressors or classifiers) to form the bagging ensemble because it seemed to be a reasonable number, since we want to keep the number of unknown parameters relatively small. There are two additional reasons for this; first, optimizing hundreds of parameters has a potentially adverse effect which eventually leads to overfitting. Second, considering each $\varphi(\boldsymbol{x}, \mathcal{L}^{(B)})$ as an independent observation from a random variable representing the predictor $\varphi(\boldsymbol{x}, \mathcal{L})$, then 30 is a statistically significant sample size for an accurate approximation of the expected value $\mathbf{E}_{\mathcal{L}}\varphi(\boldsymbol{x}, \mathcal{L}) = \varphi_B(\boldsymbol{x}, \mathcal{L}) = 1/n_B \sum_{\{\mathcal{L}^{(B)}\}} \varphi(\boldsymbol{x}, \mathcal{L}^{(B)})$. There has been a substantial debate on how large should this number be, but a decent number of books suggest a value of about 30. For instance, in (Tanis and Hogg, 2001, p. 202) it is stated that, generally, a value greater than 25 or 30 provides a good approximation of the sample's expectation. According to this, 30 predictors would be just enough for us to get the real sense of how bagging performs in order to compare it against our weighted approach in a fair manner, hence $n_B = 30$. There were only a few exceptions where this is not the case, and each one of them is later explained on its own merits.

At this point, qualitative measures of $\varphi_B(\boldsymbol{x}, D_{\mathcal{L}})$ against $\varphi_B(\boldsymbol{x}, D_{\mathcal{L}}; \{\gamma_{\mathcal{L}^{(B)}}\})$ are computed. For brevity, we formulate the $\{\gamma_{\mathcal{L}^{(B)}}\}$-based measures. For regression we consider the mean squared errors of $\varphi_B(\boldsymbol{x}, D_{\mathcal{L}})$ and $\varphi_B(\boldsymbol{x}, D_{\mathcal{L}}; \{\gamma_{\mathcal{L}^{(B)}}\})$, denoted $\varepsilon(\varphi_B(\boldsymbol{x}, D_{\mathcal{L}}),\, \mathcal{L}_{TS}) \equiv \varepsilon(\mathcal{L}_{TS})$ and $\varepsilon(\varphi_B(\boldsymbol{x}, D_{\mathcal{L}}; \{\gamma_{\mathcal{L}^{(B)}}\}),\, \mathcal{L}_{TS}) \equiv \varepsilon_{\{\gamma_{\mathcal{L}^{(B)}}\}}(\mathcal{L}_{TS})$, respectively, using the testing set $\mathcal{L}_{TS}$ in each CV fold. Therefore,

$$\varepsilon_{\{\gamma_{\mathcal{L}^{(B)}}\}}(\mathcal{L}_{TS}) = \frac{1}{N_{TS}} \sum_{(\boldsymbol{x}_{TS}, y_{TS}) \in \mathcal{L}_{TS}} (y_{TS} - \varphi_B(\boldsymbol{x}_{TS}, D_{\mathcal{L}}; \{\gamma_{\mathcal{L}^{(B)}}\}))^2.$$

For binary classification we take sign $[\varphi_B(\boldsymbol{x}, D_{\mathcal{L}})]$ and sign $[\varphi_B(\boldsymbol{x}, D_{\mathcal{L}}; \{\gamma_{\mathcal{L}^{(B)}}\})]$ to predict the class label $y \in \{-1, 1\}$. We then count misclassified cases from $\mathcal{L}_{TS}$ and calculate the misclassification rates $\rho(\varphi_B(\boldsymbol{x}, D_{\mathcal{L}}), \mathcal{L}_{TS}) \equiv \rho(\mathcal{L}_{TS})$ and $\rho(\varphi_B(\boldsymbol{x}, D_{\mathcal{L}}; \{\gamma_{\mathcal{L}^{(B)}}\}), \mathcal{L}_{TS}) \equiv \rho_{\{\gamma_{\mathcal{L}^{(B)}}\}}(\mathcal{L}_{TS})$ given by

$$\rho_{\{\gamma_{\mathcal{L}^{(B)}}\}}(\mathcal{L}_{TS}) = \frac{1}{N_{TS}} \sum_{(\boldsymbol{x}_{TS}, y_{TS}) \in \mathcal{L}_{TS}} I(y_{TS} \neq \text{sign}\,[\varphi_B(\boldsymbol{x}_{TS}, D_{\mathcal{L}}; \{\gamma_{\mathcal{L}^{(B)}}\})]),$$

where $I(\cdot)$ is the indicator function.

For multiclass classification we build $j = 1, \ldots, J$ binary models as part of the one-versus-rest strategy, each having its own parameters $\{\gamma_{\mathcal{L}^{(B)}}\}^{(j)}$. One-versus-rest requires real-valued classification algorithms, and thus we do not take sign $\left[\varphi_B^{(j)}(\boldsymbol{x}, D_{\mathcal{L}})\right]$ and sign $\left[\varphi_B^{(j)}(\boldsymbol{x}, D_{\mathcal{L}}; \{\gamma_{\mathcal{L}^{(B)}}\})\right]$, but rather the real-valued confidence outputs $\varphi_B^{(j)}(\boldsymbol{x}, D_{\mathcal{L}})$ and $\varphi_B^{(j)}(\boldsymbol{x}, D_{\mathcal{L}}; \{\gamma_{\mathcal{L}^{(B)}}\})$. To assign a class label, we take the model with the highest real valued confidence, i.e. the predicted class label $\hat{y}$ to be $\hat{y} = \arg\max_j \varphi_B^{(j)}(\boldsymbol{x}, D_{\mathcal{L}}; \{\gamma_{\mathcal{L}^{(B)}}\})$. The misclassification rate is equivalently computed as formulated above, that is

$$\rho_{\{\gamma_{\mathcal{L}^{(B)}}\}}(\mathcal{L}_{TS}) = \frac{1}{N_{TS}} \sum_{(\boldsymbol{x}_{TS}, y_{TS}) \in \mathcal{L}_{TS}} I(y_{TS} \neq \hat{y}_{TS}).$$

Repeating CV, such as 10 × 10-Fold CV has been vigorously advocated in the past. There are, however findings that repeated CV does not provide a much accurate estimate of the error(Vanwinckelen and Blockeel, 2012), thus we choose not to repeat CV. The error measures are averaged over all ten folds to obtain $\bar{\varepsilon}_{\{\gamma_{\mathcal{L}^{(B)}}\}}(\mathcal{L}_{TS})$ and $\bar{\rho}_{\{\gamma_{\mathcal{L}^{(B)}}\}}(\mathcal{L}_{TS})$, and then, since realizations over the folds are mutually independent, we construct a 95% confidence interval for the mean. We have used the t-distribution method of approximating the normal distribution with 9 degrees of freedom. All errors are reported as *mean±havled 95% confidence interval.*

4.1 Regression

We have selected 12 datasets for regression, including linear and nonlinear problems on one hand, and realistic or artificial (simulated) data on the other. Below we list the datasets, including an additional reference for the reader, as well as the type of the data for each dataset. Most of them have been obtained from the University of California, Irvine (UCI) online machine learning repository (Lichman, 2013), while others have been obtained from third-party publicly available repositories including KEEL Dataset Repository (KEEL, 2016) and US government agencies like the National Institute for Science and Technology (NIST). All datasets descriptions have been replicated as provided by the original source.

Abalone is a dataset comprised of physical measurements of abalones (edible sea snails) used to predict their age. The Airfoil dataset comes from NASA, obtained from a series of aerodynamic and acoustic tests of two and three-dimensional airfoil blade sections, while the output represents the sound pressure level in decibels. The Boston Housing data concerns housing values in the suburbs of Boston, where several urban factors are taken as features. Concrete Compressive Strength concerns predicting the compressive strength of concrete, which is a highly nonlinear function of age and ingredients. Forest Fires is a difficult regression task, where the aim is to predict the burned area in forest fires in northeastern Portugal by using meteorological and other data. The Kirby2 data is a nonlinear problem that comes from scanning electron microscope lines, that resulted from a NIST study. The Mammals dataset involves quadruped animals. The Servo dataset comes from a simulation of a servo system, covering an extremely nonlinear phenomenon of predicting the rise time of a servo mechanism. The Bank dataset is generated from a simplistic simulator, which simulates queues in a series of banks, where customers coming from several residential ares choose their preferred bank and tend to change queue if their patience expires. The goal is to predict the rejection rate, i.e. the fraction of customers that are turned away from the bank because all open tellers have full queues. The Friedman datasets are well-known artificial regression problems coming from existing theoretical underlying distributions. All datasets involve univariate regression, i.e. have only one response variable. In addition, we provide short quantitative descriptions of each dataset including the number of all available cases in the initial learning set and the number of features, excluding the output variable. The dataset size varies from less than 100, up to several thousands, while dimensionality lies within the range of tens. The dataset summary is provided in Table 1.Table 2 summarizes the results for the 12 regression problems. For brevity, we present mean errors accompanied by their 95% confidence intervals and measure the decrease. All results are obtained using $n_B = 30$ and 10-Fold CV.

4.2 Binary Classification

Many popular classification algorithms use squared error minimization within, and a typical example is Gentle Boost (Friedman et al, 2000), which combines one-level, single-node regression trees, also known as *regression stumps*. All these algorithms are called real-valued classification algorithms since they output different magnitudes of prediction confidence instead of a single label. Here we use trees of unlimited depth to perform classical binary classification. Another reason to use real-valued outputs for classification is closely related to our approach because it significantly reduces the risk of a singular coefficient matrix of the linear system which leads to approximative solutions.

We have chosen 15 datasets for binary classification, including problems of different extents of predictive difficulty. We use both realistic and simulated

Dataset	**Cases**	**Features**
Abalone (Waugh, 1995) (realistic)	4177	8
Airfoil (Lichman, 2013) (realistic)	1503	6
Boston Housing (Belsley et al, 2005; Quinlan, 1993) (realistic)	506	13
Concrete Compressive Strength (Yeh, 1998) (realistic)	1030	8
Forest Fires (Cortez and Morais, 2007) (realistic)	517	12
Kirby2 (ITL, 2016; Kirby, 1979) (realistic)	151	1
Mammals (Lichman, 2013) (realistic)	97	4
Servo (Quinlan et al, 1992) (realistic)	167	4
Bank (DELVE, 2016) (simulated)	8192	8
Friedman #1 (Friedman, 1991; Breiman, 1996a) (simulated)	1000	10
Friedman #2 (Friedman, 1991; Breiman, 1996a) (simulated)	1000	4
Friedman #3 (Friedman, 1991; Breiman, 1996a) (simulated)	1000	20

Table 1. Summary of the 12 regression datasets.

Dataset	$\boldsymbol{\bar{\varepsilon}(\mathcal{L}_{TS})}$	$\boldsymbol{\bar{\varepsilon}_{\{\gamma_{\mathcal{L}(B)}\}}(\mathcal{L}_{TS})}$	**Decrease**
Abalone	1.61 ± 0.31	1.62 ± 0.30	-0.61%
Airfoil	2.72 ± 0.79	2.69 ± 0.77	1.10%
Boston Housing	3.08 ± 0.95	3.06 ± 0.91	0.65%
Concrete Compressive Strength	5.45 ± 1.63	5.36 ± 1.65	1.65 %
Forest Fires	26.90 ± 7.58	24.13 ± 6.98	10.3%
Kirby2	2.80 ± 0.72	2.74 ± 0.68	2.14%
Mammals	1.17 ± 0.45	0.99 ± 0.38	15.38%
Servo	0.25 ± 0.15	0.16 ± 0.13	36.0%
Bank	0.014 ± 0.01	0.14 ± 0.01	0.00%
Friedman #1	3.48 ± 0.42	3.38 ± 0.41	2.87%
Friedman #2	410.46 ± 94.64	372.25 ± 83.23	9.31%
Friedman #3	3.68 ± 1.40	3.14 ± 1.14	14.67%
Average			**7.5%**

Table 2. Averaged CV mean squared errors for regression datasets, accompanied by the 95% confidence intervals. The last column depicts the decrease in errors.

data. All realistic datasets were obtained from public repositories. The simulated dataset Hastie 10,2 is obtained from a publicly available source, namely the Scikit-Learn machine learning library for Python (Pedregosa et al, 2011).

The Australian Credit Approval data classifies (in)eligible credit card approvals based on customer information. The Breast Cancer dataset contains medical diagnosis data regarding breast cancer. Climate Model Simulation Crashes provides Latin hypercube samples of 18 climate model input parameter values in order to predict climate model simulation crashes and determine the parameter value combinations that cause the failures. The Diabetes (Pima Indians) data contains patient records of female patients at least 21 years old of Pima Indian heritage, a group of Native Americans originating from central and southern Arizona. The goal is to predict the onset of diabetes mellitus. The German Credit data classifies people described by a set of attributes as good or bad credit risks. The Heart dataset is a database in which medical data is stored to classify patients by the presence of a heart disease. The Hepatitis dataset has the same goal, that is, predicting the presence of hepatitis symptoms. The Ionosphere data is used for classification of radar returns from the Earth's ionosphere, collected from a system in Goose Bay, Labrador and the signals are classified as either "good" or "bad" depending on whether they have returned evidence of some type of structure in the ionosphere. The Liver Disorders is a BUPA Medical Research Ltd. database that collects data from blood tests which are thought to be sensitive liver disorders related to excessive alcohol consumption. The Phoneme dataset contains data that is aimed at distinguishing between nasal and oral sounds. The Sonar dataset contains data collected from sonar signals, where the task is to discriminate between ones that were bounced off a metal cylinder and those bounced off a roughly cylindrical rock. The Banana dataset is an artificially generated banana-shaped dataset. More information about the Ringnorm and Twonorm datasets can be found in (Breiman, 1996b). The Ringnorm dataset comes from Normal distribution where the two classes of data form concentric rings, one within the other. The Twonorm dataset contains data coming from two different Normal distributions, where the task is to recognize the correct distribution that a particular sample comes from. The Hastie dataset was proposed by Hastie et al (2009) and contains data coming from a Normal (Gaussian) distribution. Table 3 provides a quantitative summary of each dataset.

Table 4 summarizes the results for the 15 Binary Classification problems. For brevity, we present mean errors accompanied by their 95% confidence intervals and measure the decrease. All results are obtained using $n_B = 30$ and 10-Fold Stratified CV. The misclassification rates are given in percent in Table 4.

4.3 Multiclass Classification

We have adopted the one-versus-rest methodology to construct a multiclass classifier using our approach since it is compatible with binary classification and real-valued outputs. Given a J-class problem, we construct J binary classification models on the basis of Section 4.2. As in Sections 4.1 and 4.2, we use $n_B = 30$ predictors for each ensemble.

We have used 15 datasets to test the approach. They come from two public sources – the UCI repository (Lichman, 2013) and the KEEL dataset repository (KEEL, 2016). The Satimage and Shuttle datasets provide testing sets, and therefore we present one realization in which we have used $n_B = 20$ for Satimage

Dataset	**Cases**	**Features**
Australian (Lichman, 2013)	690	14
Breast Cancer (Prognostic) (Mangasarian et al, 1995, 1990)	699	9
Climate Model Simulation Crashes (Lucas et al, 2013)	540	18
Diabetes (Pima Indians) (Smith et al, 1988)	768	8
German Credit (Lichman, 2013)	1000	19
Heart (Statlog) (Lichman, 2013)	270	12
Hepatitis (Lichman, 2013)	155	18
Ionosphere (Lichman, 2013)	351	33
Liver Disorders (Lichman, 2013)	345	6
Phoneme (Phoneme, 2002)	5404	5
Sonar (binarized) (LIBSVM, 2016; Lichman, 2013)	208	60
Banana (KEEL, 2016)	5300	2
Ringnorm (Breiman, 1996b)	7400	20
Twonorm (Breiman, 1996b)	7400	20
Hastie 10,2 (Hastie et al, 2009; Pedregosa et al, 2011)	1200	10

Table 3. Summary of 15 binary classification datasets. The last four datasets (Banana, Ringnorm, Twonorm, and Hastie 10,2) are simulated, while the rest are realistic.

and $n_B = 30$ for Shuttle. Balance Scale was generated to model psychological experimental results. Each example is classified as having the balance scale tip to the right, tip to the left, or be balanced. The Car data is used for evaluating cars based on standard characteristics of a car. Flare is a modified version of Solar Flare presented at the UCI repository. The target is to distinguish between certain types of solar flares occurred in a 24-hour period. Glass contains data for six types of glass, defined in terms of their oxide content. Hayes-Roth is an artificial database from a study of human subjects. Image Segmentation contains image data described by high-level numeric-valued attributes; the instances were drawn randomly from a database of seven outdoor images and the goal is to classify each pixel. The Iris dataset is a well-known problem of distinguishing between three types of the iris plant. The Letter dataset consists of character image features, where the goal is to try to identify the given capital alphabetic letter. Opt Digits concerns optical recognition of handwritten digits. The original data was collected by Ethem Alpaydin. Satimage contains data from the Landsat Satellite. Given multi-spectral values of pixels in 3×3 neighborhoods, the goal is to predict the classification associated with the central pixel in each neighborhood. Seeds contains measurements of geometrical properties of kernels belonging to three different varieties of wheat, obtained by a soft X-ray technique. Shuttle is a large and highly imbalanced database (80% of the data are in class 1). The data originally comes from NASA and is therefore scarcely documented.

Dataset	$\bar{\rho}(\boldsymbol{\mathcal{L}_{TS}})$	$\bar{\rho}_{\{\gamma_{\mathcal{L}(B)}\}}(\boldsymbol{\mathcal{L}_{TS}})$	**Decrease**
Australian	15.38 ± 4.60	13.05 ± 3.65	15.15%
Breast Cancer (Prognostic)	4.14 ± 2.52	3.57 ± 2.22	13.77%
Climate Model	7.22 ± 1.81	5.92 ± 1.85	17.92%
Diabetes (Pima Indians)	25.52 ± 3.53	23.57 ± 4.06	7.64%
German Credit	25.50 ± 2.00	23.30 ± 2.05	8.63%
Heart (Statlog)	19.63 ± 6.37	17.77 ± 7.68	9.47%
Hepatitis	20.54 ± 4.51	16.66 ± 7.27	18.89%
Ionosphere	9.05 ± 3.49	6.23 ± 2.95	31.16%
Liver Disorders	34.55 ± 6.46	31.08 ± 6.79	10.04%
Phoneme	9.73 ± 1.08	8.88 ± 0.95	8.73%
Sonar	31.87 ± 13.02	27.10 ± 11.92	14.97%
Banana	11.40 ± 1.08	10.83 ± 1.13	5.00%
Ringnorm	5.35 ± 0.82	4.81 ± 0.83	10.09%
Twonorm	4.70 ± 0.73	4.12 ± 0.69	12.34%
Hastie 10,2	20.9 ± 3.83	17.99 ± 4.08	13.92%
Average			**13.18%**

Table 4. Averaged CV misclassification rates (in percent) for binary classification datasets, accompanied by the 95% confidence intervals. The last column depicts the decrease in errors.

Tae consists of evaluations of teaching performance over three regular semesters and two summer semesters of 151 Teaching Assistant (TA) assignments at the Statistics Department of the University of Wisconsin-Madison. Performance of a TA is evaluated as low, medium or high. Zoo contains mostly boolean-valued artificially generated attributes of animals, where the goal is to classify animals in seven predefined classes. Table 5 provides a quantitative summary of each dataset.

Table 6 summarizes the results for the 15 multiclass classification problems. We obtained the mean misclassification rates in percent, accompanied by their 95% confidence intervals from 10-Fold Stratified CV. The only exceptions were Satimage and Shuttle, since the original source had provided predefined testing sets. For that reason, we did not perform averaging. We used $n_B = 20$ for Satimage and $n_B = 30$ for Shuttle. All other datasets were evaluated using $n_B = 30$ predictors in the bagging ensemble.

5 Discussions

The idea of bagging is to inject model diversity at the data level, by training the ensemble predictors on bootstrap replicates (sampling with replacement) of the

Dataset	**Cases**	**Features**	**Classes**
Balance Scale (Lichman, 2013)	625	4	5
Car (KEEL, 2016)	1728	6	4
Flare (KEEL, 2016)	1066	11	6
Glass (Lichman, 2013)	214	10	6
Image Segmentation (KEEL, 2016)	210	19	7
Iris (Lichman, 2013)	150	4	3
Letter (Lichman, 2013)	20,000	16	26
Opt Digits (Lichman, 2013)	3823	64	10
Satimage (Lichman, 2013)	4435 (2000)	36	6
Seeds (Lichman, 2013)	210	7	3
Shuttle (Lichman, 2013)	43,500 (14,500)	9	7
Tae (KEEL, 2016)	151	5	3
Vehicle Silhouette (Lichman, 2013)	846	18	4
Hayes-Roth (KEEL, 2016)	160	4	3
Zoo (KEEL, 2016)	101	16	7

Table 5. Summary of the 15 multiclass classification datasets. Hayes-Roth and Zoo are simulated, while the rest are realistic.

training data set. Then, the outputs of the predictors/models are combined using majority vote (in the case of classification), or averaging (in the case of regression). However, in general, combining the output of predictors/classifiers/models can be performed using several fusion functions, such as majority vote, weighted majority vote, Bayesian combination, average rule, max rule, min rule, median rule or even a more advanced approach such as stacked generalization. This paper introduces weighted bagging predictors by choosing the weights such that the mean squared prediction error is minimal. With weighted bagging, the generated ensemble is fine-tuned to reduce variance among the predictors further, lower than bagging. However, the approach suffers from the same uncertainties as bagging and its effectiveness depends on how accurate our estimates of the expectation are. We have demonstrated the performance of the weighted bagging against bagging for various datasets (12 for regression, 15 for binary classification, and 15 for multiclass classification). For regression, improvements larger than 5% are observed in 5 out of 12 datasets with an average error decrease of 7.5%. For binary classification we found that the generalization error decreases on average 13%, with minimum decrease being 5% while maximum decrease is slightly above 31%. Finally, in the case of multiclass classification, the error decrease ranges from 1% to 50% with the average one being somewhat below 13%; error decreases larger than 5% have been observed in 10 out of 15 datasets.

Dataset	$\bar{\rho}(\mathcal{L}_{TS})$	$\bar{\rho}_{\{\gamma_{\mathcal{L}(B)}\}}(\mathcal{L}_{TS})$	**Decrease**
Balance Scale	68.17 ± 2.84	67.20 ± 2.82	1.42%
Car	11.99 ± 6.39	9.12 ± 5.05	23.93%
Flare	28.50 ± 2.59	27.47 ± 2.39	3.61%
Glass	8.99 ± 11.99	8.03 ± 11.36	10.68%
Image Segmentation	11.91 ± 2.89	10.48 ± 5.28	11.00%
Iris	4.0 ± 3.33	2.67 ± 2.46	33.25%
Letter	5.40 ± 0.55	5.30 ± 0.52	1.86%
Opt Digits	4.03 ± 1.02	3.99 ± 0.95	1.00%
Satimage	10.40 ± 0.00	9.95 ± 0.00	4.33%
Seeds	8.57 ± 6.18	8.09 ± 6.43	5.61%
Shuttle	0.14 ± 0.00	0.07 ± 0.00	50.00%
Tae	32.48 ± 21.77	29.71 ± 19.39	8.53%
Vehicle Silhouette	24.81 ± 1.94	23.16 ± 1.94	6.65%
Hayes-Roth	19.15 ± 9.73	17.89 ± 7.98	6.58%
Zoo	3.76 ± 4.66	2.85 ± 3.32	24.20%
Average			**12.84%**

Table 6. Averaged CV misclassification rates (in percent) for multiclass classification datasets, accompanied by the 95% confidence intervals. The last column depicts the decrease in errors.

The results show that the model has best performance on binary classification problems. Regression problems are typically more sensitive, compared to classification, especially the binary case, where only the sign of the prediction is of interest. Multiclass classification problems are solved considerably well, but no better than binary ones. Our sense is that this is caused by class imbalance, introduced by the one-versus-rest approach. For J classes, each model $j = 1, \dots, J$ is trained on data highly biased towards the remaining classes $k \neq j$, since generally, these cases largely outnumber those of class j. However, there is a trade-off between having numerous models, retaining the original pairwise class frequency ratios (one-versus-one) and keeping the number of models reasonable, while impelling class imbalance.

The performance in binary classification was examined further. The term *margin* of a classifier refers to the distance from the decision boundary optimized by the classifier, which is often a hyperplane, to any specific case in the learning set $\mathcal{L}$. The "distance" itself is portrayed by the real-valued output of the classification algorithm. The margin is taken to be $y\varphi(\boldsymbol{x})$, where $(\boldsymbol{x}, y) \in \mathcal{L}$. A decisive condition is that $y \in \{-1, 1\}$ in order to consider *positive* against *negative* margins. The former represent correctly classified cases, which is algebraically clear, while the latter represent a misclassification, while the magnitude of the margin represents the decision confidence. The concept of margins is the basic principle of Support Vector Machines (we refer the reader to (Cortes and Vapnik, 1995)), while the general reasoning in the machine learning community is that increased margins of cases in $\mathcal{L}$ lead to lower generalization error rates. For these reasons, we performed tests on binary classification problems and observed that weighted bagging increases the margins both quantitatively and qualitatively, that is, while increasing the fraction of positive margins, it also increases their magnitudes. We define the margin distribution at $\mu \in (-1, 1)$ to be the fraction of margins being at most μ.

To see this, we first focus on the fraction of positive margins. The tests showed similar results on all datasets from Table 3, and Liver Disorders is presented here for brevity. Figure 1 shows that weighted bagging has a continuously larger fraction of positive margins, reaching 1.0. Another phenomenon that came from all tests was the implication that using 30 predictors in the bagged ensemble is more than enough to stabilize the aggregated predictor, where the curves saturate to 1.0 after adding approximately 20 predictors in the ensemble. Weighted bagging reaches that point earlier. Interestingly, it does not lead to overfitting as figures above show. Moreover, an elevated margin distribution has been found to be responsible for a changing generalization error rate after the training error rate drops down to 0. The most significant examples involve voting methods, especially AdaBoost, while the phenomenon and its effects were explained in the work of Schapire et al (1998). The margin distribution of Liver Disorders is

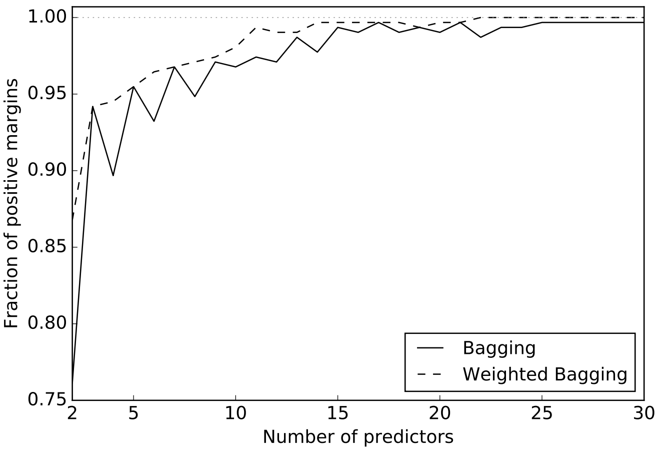

Fig. 1. Fraction of positive margins in bagging and weighted bagging on the Liver Disorders data. The ensemble consisted of two and then subsequently increased to at most 30 predictors .

shown in Figure 2. The phenomenon of larger margins comes from minimizing the squared error of weighted bagging, which can be algebraically rewritten as

$$\begin{aligned}\varepsilon_A &= \mathbf{E}_{Y,\boldsymbol{X}}(Y - \mathbf{E}_{\mathcal{L}}\gamma_{\mathcal{L}}\varphi(\boldsymbol{x},\mathcal{L}))^2 \\ &= \mathbf{E}_{Y,\boldsymbol{X}}(1 - \mathbf{E}_{\mathcal{L}}Y\gamma_{\mathcal{L}}\varphi(\boldsymbol{x},\mathcal{L}))^2 \\ &\leq \mathbf{E}_{Y,\boldsymbol{X}}(1 - \mathbf{E}_{\mathcal{L}}Y\varphi(\boldsymbol{x},\mathcal{L}))^2\end{aligned}$$

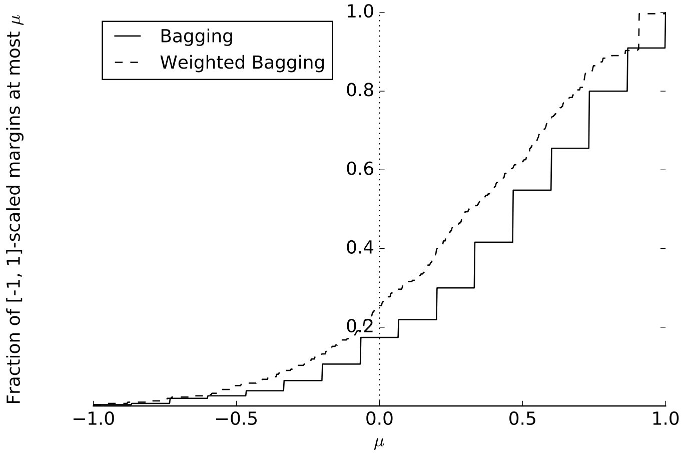

Fig. 2. Margin distribution at $\mathcal{L}$ of Liver Disorders using 30 predictors. The generalization (testing) error is reduced by 15.4%, while the training error on $\mathcal{L}$ decreases from 0.3% to 0%.

Experiments have also shown that the execution time overhead imposed by weighted bagging has an important characteristic; it does not grow linearly with the size of the bagging ensemble. We noticed that the overhead saturates at some point and is depicted by a curve resembling the logarithm function. We attested this using 1000 predictors. Figure 3 shows weighted-bagging-imposed overhead in as percentage of the total execution time. Here we present results for Liver Disorders, while other datasets manifested very similar results.

References

Allwein EL, Schapire RE, Singer Y (2001) Reducing multiclass to binary: A unifying approach for margin classifiers. *J Mach Learn Res* 1:113–141, DOI 10.1162/15324430152733133, URL http://dx.doi.org/10.1162/15324430152733133

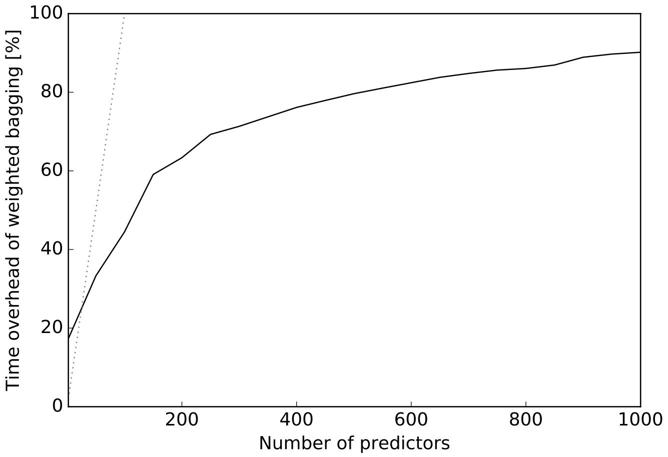

Fig. 3. Sub-linear time overhead of weighted bagging of Liver Disorders as ensemble size increases to 1000 predictors. The referent linear overhead is represented by a gray dotted line.

Bauer E, Kohavi R (1999) An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine learning* 36(1-2):105–139

Belsley DA, Kuh E, Welsch RE (2005) *Regression diagnostics: Identifying influential data and sources of collinearity*, vol 571. John Wiley & Sons

Breiman L (1996a) Bagging predictors. *Machine learning* 24(2):123–140

Breiman L (1996b) *Bias, variance, and arcing classifiers*. Tech. Rep. 460, Statistics Department, University of California, Berkeley, CA, USA

Breiman L (1996c) Heuristics of instability and stabilization in model selection. *The annals of statistics* 24(6):2350–2383

Breiman L (2000) Randomizing outputs to increase prediction accuracy. *Machine Learning* 40(3):229–242

Breiman L (2001) Using iterated bagging to debias regressions. *Machine Learning* 45(3):261–277

Breiman L, Friedman J, Stone CJ, Olshen RA (1984) *Classification and regression trees*. CRC press

Brown G (2004) *Diversity in neural network ensembles*. Citeseer

Brown G, Wyatt J, Harris R, Yao X (2005a) Diversity creation methods: a survey and categorisation. *Information Fusion* 6(1):5–20

Brown G, Wyatt JL, Tiňo P (2005b) Managing diversity in regression ensembles. *Journal of Machine Learning Research* 6(Sep):1621–1650

Bühlmann P (2012) Bagging, boosting and ensemble methods. In: Handbook of Computational Statistics, Springer, pp 985–1022

Cortes C, Vapnik V (1995) Support-vector networks. *Machine learning* 20(3):273–297

Cortez P, Morais AdJR (2007) A data mining approach to predict forest fires using meteorological data. In: 13th Portuguese Conference on Artificial Intelligence (EPIA 2007), New Trends in Artificial Intelligence, Associação Portuguesa para a Inteligência, pp 512–523

DELVE (2016) Data for Evaluating Learning in Valid Experiments (DELVE). URL `http://www.cs.utoronto.ca/~delve/`, (visited on 2016-07-18)

Domingos P (1997) Why does bagging work? a bayesian account and its implications. In: In Proceedings of the Third International Conference on Knowledge Discovery and Data Mining, AAAI Press, pp 155–158

Efron B, Tibshirani RJ (1994) *An introduction to the bootstrap*. CRC press

Frank E, Pfahringer B (2006) Improving on bagging with input smearing. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, pp 97–106

Friedman J, Hastie T, Tibshirani R, et al (2000) Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The annals of statistics* 28(2):337–407

Friedman JH (1991) Multivariate adaptive regression splines. *The annals of statistics* pp 1–67

Friedman JH (1997) On bias, variance, 0/1-loss, and the curse-of-dimensionality. *Data Mining and Knowledge Discovery* 1:55–77

Grandvalet Y (2004) Bagging equalizes influence. *Machine Learning* 55(3):251–270

Hastie T, Tibshirani R, et al (1998) Classification by pairwise coupling. *The annals of statistics* 26(2):451–471

Hastie T, Tibshirani R, Friedman J (2009) *The Elements of Statistical Learning*. Springer

Hernández-Lobato D, Martínez-Muñoz G, Suárez A (2006) Pruning in ordered regression bagging ensembles. In: The 2006 IEEE International Joint Conference on Neural Network Proceedings, IEEE, pp 1266–1273

ITL N (2016) Information Technology Laboratory. URL `http://www.itl.nist.gov/div898/strd/nls/data/kirby2.shtml`, (visited on 2016-07-18)

KEEL (2016) KEEL dataset repository. URL `http://sci2s.ugr.es/keel/datasets.php`, (visited on 2016-07-24)

Kirby R (1979) Scanning electron microscope line width standards. *NIST, unpublished*

LIBSVM (2016) LIBSVM Data: Classification (binary class). `https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html`, accessed: 2016-07-10

Lichman M (2013) UCI machine learning repository. `http://archive.ics.uci.edu/ml`, accessed 18 July 2016

Lucas D, Klein R, Tannahill J, Ivanova D, Brandon S, Domyancic D, Zhang Y (2013) Failure analysis of parameter-induced simulation crashes in climate models. *Geoscientific Model Development* 6(4):1157–1171

Mangasarian OL, Setiono R, Wolberg W (1990) Pattern recognition via linear programming: Theory and application to medical diagnosis. *Large-scale numerical optimization* pp 22–31

Mangasarian OL, Street WN, Wolberg WH (1995) Breast cancer diagnosis and prognosis via linear programming. *Operations Research* 43(4):570–577

Mendes-Moreira J, Soares C, Jorge AM, Sousa JFD (2012) Ensemble approaches for regression: A survey. *ACM Computing Surveys (CSUR)* 45(1):10

Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12:2825–2830

Phoneme (2002) Phoneme data. URL `https://www.elen.ucl.ac.be/neural-nets/Research/Projects/ELENA/databases/REAL/phoneme/`, (visited on 2016-07-18)

Pisetta V (2012) New insights into decision trees ensembles. PhD thesis, University of Lyon

Quinlan JR (1993) Combining instance-based and model-based learning. In: Proceedings of the Tenth International Conference on Machine Learning, pp 236–243

Quinlan JR, et al (1992) Learning with continuous classes. In: 5th Australian joint conference on artificial intelligence, Singapore, vol 92, pp 343–348

Ren Y, Zhang L, Suganthan P (2016) Ensemble classification and regression-recent developments, applications and future directions [review article]. *IEEE Computational Intelligence Magazine* 11(1):41–53

Rifkin R, Klautau A (2004) In defense of one-vs-all classification. *Journal of machine learning research* 5(Jan):101–141

Schapire RE, Freund Y (2012) *Boosting: Foundations and algorithms*. MIT press

Schapire RE, Freund Y, Bartlett P, Lee WS (1998) Boosting the margin: A new explanation for the effectiveness of voting methods. *Annals of statistics* pp 1651–1686

Smith JW, Dickson WC, Everhart JE, Knowler WC, Johannes RS (1988) Using the adap learning algorithm to forecast the onset of diabetes mellitus. *Johns Hopkins APL Technical Digest* 10:262–266

Tanis EA, Hogg RV (2001) *Probability and Statistical Inference*. Prentice Hall Upper Saddle River, NJ

Ueda N, Nakano R (1996) Generalization error of ensemble estimators. In: Neural Networks, 1996., IEEE International Conference on, IEEE, vol 1, pp 90–95

Vanwinckelen G, Blockeel H (2012) On estimating model accuracy with repeated cross-validation. In: BeneLearn 2012: Proceedings of the 21st Belgian-Dutch Conference on Machine Learning, pp 39–44

Waugh S (1995) Extending and benchmarking cascade-correlation. *Dept of Computer Science, University of Tasmania, Ph D Dissertation*

Webb GI (2000) Multiboosting: A technique for combining boosting and wagging. *Machine learning* 40(2):159–196

Yeh IC (1998) Modeling of strength of high-performance concrete using artificial neural networks. *Cement and Concrete research* 28(12):1797–1808

Zanda M (2010) A probabilistic perspective on ensemble diversity. PhD thesis, University of Manchester