Trimmed Bagging

C. Croux* K.U. Leuven
K. Joossens K.U.Leuven
A. Lemmens Erasmus University Rotterdam

Abstract

Bagging has been found to be successful in increasing the predictive performance of unstable classifiers. Bagging draws bootstrap samples from the training sample, applies the classifier to each bootstrap sample, and then averages over all obtained classification rules. The idea of *trimmed bagging* is to exclude the bootstrapped classification rules that yield the highest error rates, as estimated by the out-of-bag error rate, and to aggregate over the remaining ones. In this note we explore the potential benefits of trimmed bagging. On the basis of numerical experiments, we conclude that trimmed bagging performs comparably to standard bagging when applied to unstable classifiers as decision trees, but yields better results when applied to more stable base classifiers, like support vector machines.

*Corresponding author. Faculty of Economics and Management University Center of Statistics, Naamsestraat 69, B-3000 Leuven, Belgium. Email: Christophe.Croux@econ.kuleuven.be

1. Introduction

Originating from the machine learning literature, bagging is based on the principle of classifier aggregation. This idea has been inspired by Breiman (1996) who found gains in accuracy by combining several base classifiers, sequentially estimated from perturbed versions of the training sample. Bagging, the acronym for Bootstrap AGGregatING, consists of sequentially estimating a base classifier from bootstrapped samples of a given training sample. The bootstrapped classifiers form then a committee of component classifiers from which a final classification rule can be derived by simple aggregation. Bagging may substantially reduce the variance of a classifier, without affecting too much its bias. Theoretical results on bagging have been obtained by Bühlmann and Yu (2002) and Buja and Stuetzle (2006), among others. Bagging is conceptually simple and intuitive, and was shown to be successful in several applications (e.g. Lemmens and Croux (2006) for an application in customer retention in marketing).

Although its good performance has been demonstrated on several occasions, some conditions need to be fulfilled to ensure that bagging outperforms the base classifier. The latter needs to be good on average, but unstable with respect to the training set. Unstable classifiers have typically low bias but high variance (Breiman (1998)). Among the different alternatives, decision trees have been proven to be a good choice as a base classifier for bagging. However, as pointed out by several authors, e.g. Dietterich (2000), there is no guarantee that bagging will improve the performance of any base classifier. When using stable base classifiers, like support vector machines, it may even yield a deterioration of the predictive accuracy.

In this paper, we propose a new method to improve the performance of any classifier, called *trimmed bagging*. The idea behind trimmed bagging is simple and intuitive: instead of averaging over all bootstrapped classifiers, we only average over the best performing ones. Hence, we will trim away those bootstrapped classifiers that result in the highest error rates. The remainder of the paper is organized as follows. In the next section, we formally define the trimmed bagging procedure. We also review two other variants of bagging, namely bragging, first introduced by Bühlmann (2003), and the nice bagging procedure of Skurichina and Duin (1998). Section 3 outlines the assessment criterion and Section 4 empirically compares the performance of the various bagging variants. Section 5 contains the conclusions.

2. Methodology

2.1. Bagging and bragging

Suppose that we want to classify an observation into one of two groups, labeled by "$y = 1$" and "$y = 0$", using a multivariate predictor variable x. A classifier is computed on the training set $Z_{\text{tr}} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, for which the values of x_i and y_i are known. The constructed classifier depends on the training data

used, and can be expressed as a mapping $x \to g_{\text{tr}}(x) = g(x; Z_{\text{tr}}) \in [0, 1]$, which assigns a score or probability to each observation. The case that the classifier only takes on the values 0 or 1 is allowed as well.

Bagging has been proposed by Breiman (1996) as a tool to improve the performance of the base classifier $g_{\text{tr}}(\cdot)$. Practically, we construct B bootstrapped versions of the original training sample Z_{tr}, denoted by $Z^{*1}, ..., Z^{*B}$. By construction, some observations will appear several times in a bootstrapped sample, while others will be missing. The latter will form the out-of-bootstrap or out-of-bag samples, denoted as $Z^{-*1}, ..., Z^{-*B}$. Using the B bootstrapped samples, we construct B classifiers $g(\cdot; Z^{*1}), ..., g(\cdot; Z^{*B})$. The bagged classifier is then obtained by averaging over all B bootstrapped classifiers:

$$g_{\text{bag}}(x) = \underset{1 \le b \le B}{\text{average}}\, g(x; Z^{*b}). \tag{1}$$

To determine the optimal number of bootstrapped samples, a strategy is to select B such that the apparent error rates (i.e. error rates on the training data) remain stable for values larger than B. An extensive overview of different supervised classification rules, including bagging, can be found in Hastie, Tibshirani, and Friedman (2001).

Some extensions of the original bagging algorithm have already been proposed. Instead of computing an average over the outcomes of the B bootstrapped classifiers, as in (1), one could compute a robust location estimator instead. As such, Bühlmann (2003) proposed to take the median

$$g_{\text{brag}}(x) = \underset{1 \le b \le B}{\text{median}}\, g(x; Z^{*b}), \tag{2}$$

and called the resulting procedure *bragging*. Instead of the median, one could also take a trimmed average in (2). We stress, however, that this does not correspond to the trimmed bagging procedure proposed in this paper. Our proposal is to sort the different classifiers with respect to their corresponding error rate, and not with respect to the numerical values of the outcomes.

2.2. Trimmed Bagging

We start by ordering the different bootstrapped classifiers by increasing error rate (ER):

$$\text{ER}(g(\cdot, Z^{*(1)})) \le \text{ER}(g(\cdot, Z^{*(2)})) \le ... \le \text{ER}(g(\cdot, Z^{*(B)})).$$

Hence, $Z^{*(b)}$ denotes the bootstrapped training set resulting in a classification rule with the bth smallest error rate. The idea of trimmed bagging is to trim off the portion α of "worst" classifiers, in the sense of having the largest error rates, and to average only over the most accurate classifiers. This results in

$$g_{\text{trimbag}}(x) = \underset{1 \le b \le \lfloor (1-\alpha)B \rfloor}{\text{average}}\, g(x, Z^{*(b)}). \tag{3}$$

(Here $\lfloor z \rfloor$ indicates the largest integer value smaller or equal to z.) Note that the trimming in (3) is one-sided, since we want to keep bootstrapped classifiers with low error rate into the final aggregation scheme, and only trim the bad classification rules. Instead of a trimmed average one could also consider other kinds of weighted averages, with the weights inversely proportional to the error rate. We did experiments with several of these weighting schemes, but it turned out that the simple trimmed average, as in (3), behaves as good as more complicated weighted averages.

To compute the error rates $\mathrm{ER}(g(\cdot, Z^{*b}))$, for $b = 1, \ldots, B$, we used the out-of-bag error rate, being the proportion of misclassified observations in the out-of-bootstrap sample, which we denote by Z^{*-b}. The advantage of the *out-of-bag error rate* is that it results in an unbiased estimate of the true error rate, in contrast to the apparent error rate. The latter is the proportion of misclassified observation of Z_{tr} when using $g(\cdot, Z^{*b})$. One also needs to choose the trimming portion α. Since bagging is a variance reduction technique, it is of importance to take α small enough, such that we still aggregate over a substantially large number of component classifiers. On the other hand, taking α too small implies that the risk of including "bad" classifiers in the trimmed sum increases. As a compromise, we selected $\alpha = 0.25$ throughout this paper.

Another possibility would be to take a data driven choice of α. One could only average over these base classifiers $g(\cdot, Z^{*(b)})$ for which

$$\mathrm{ER}(g(\cdot, Z^{*(b)})) < \mathrm{ER}(g(\cdot, Z_{\mathrm{tr}})). \tag{4}$$

Hence we only incorporate bootstrapped classifiers performing better than the initial base classifier $g(\cdot, Z_{\mathrm{tr}})$ in the trimmed average. The resulting aggregated classifier corresponds then to the *nice bagging* procedure, already proposed by Skurichina and Duin (1998). Formally, let B' be the largest b for which (4) holds. Then

$$g_{\mathrm{nicebag}}(x) = \underset{1 \le b \le B'}{\mathrm{average}}\, g(x, Z^{*(b)}).$$

In the remainder of the paper we will compare the classification performance of bagging, bragging, trimmed bagging and nice bagging on a number of real data sets. Note that the computation of all the aggregated classifiers is straightforward, and requires B times the computation of the base classifier. Calculation of the error rates $\mathrm{ER}(g(\cdot, Z^{*(b)}))$, for $b = 1, \ldots, B$ is immediate.

3. Assessment Criterion

In line with the classification literature, we assess the performance of the different methods on a validation sample that has not been used during the training step. To do so, we randomly split the data sets in a training sample (80%) and a validation sample (20%). To prevent the results to rely on this splitting decision, we repeat the splitting 10 times, and average the results over the 10 samples. The percentage of misclassified observations in the validation sample is called the validated error rate.

Table 1: Sample size n and number of predictor variables p for the various data sets

Data set	n	p	Data set	n	p
Austral	690	14	Spambase	4601	57
Balloon	156	5	Tictacto	958	9
Breast	699	9	Wdbc	569	30
Cmc	1473	9	Wpbc	198	31
Crx	653	15	Spect	267	23
Iono	351	33	Spectf	369	45

In the following section, we want to test whether bagging (or any bagging variant) improves or degrades the performance of the base classifier $g(\cdot; Z_{\text{tr}})$. To do so, we compute the relative improvement in predictive accuracy, by measuring the decrease in the validated error rate when using bagging (or any variant),

$$\text{Relative Improvement} = \frac{\text{ER}_{\text{base classifier}} - \text{ER}_{\text{bagging}}}{\text{ER}_{\text{base classifier}}},$$

where $\text{ER}_{\text{base classifier}}$ and $\text{ER}_{\text{bagging}}$ are, respectively, the validated error rates for the base classifier and for the bagging variant. The average value of the Relative Improvement over the 10 repetitions of the cross-validation will be reported. We will also indicate whether this average value is significantly different from zero, in the positive or negative sense. Significance testing is doen with a standard two-sided t-test for the nullity of an average (based on a sample of size 10 here).

4. Results

To evaluate the performance of trimmed bagging, we apply the aforementioned bagging variants to 12 of the well-known UCI (University of California Irvine) Machine Learning Repository data sets.[1] Characteristics of the data sets can be found in Table 1.

To demonstrate the ability of trimmed bagging in improving the predictive performance of any base classifier, stable or unstable, we consider the following base classifiers: (a) decision trees (Table 2), as an example of an unstable classifier (b) support vector machines (SVM) (Table 3), linear discriminant analysis (Table 4), and logistic regression (Table 5) as examples of stable classifiers. All these base classifiers are well-known and routineously used. The decision tree was taken to be the default "tree" from the MASS library in the R software package, and the support vector machine was the default "svm" of the e1071 package of R, with the RBF kernel. We compare trimmed bagging with other

[1] See http://www.ics.uci.edu/~mlearn/MLRepository.html

Table 2: Mean relative improvements in error rate of Bagging, Bragging, Nice and Trimmed Bagging with respect to a standard classification tree (significant improvements at the 5% level are indicated by *). The last two rows give the number of significant improvements (deteriorations) for the 12 considered data sets.

Data set	Bagging	Bragging	Nice	Trimmed
Austral	0.13^*	0.14^*	0.04	0.10
Balloon	-0.14	-0.10	-0.22	-0.63
Breast	0.27^*	0.26^*	0.18^*	0.23^*
Cmc	0.04^*	0.04^*	0.02	0.02
Crx	0.17^*	0.18^*	0.10^*	0.15^*
Iono	0.27^*	0.29^*	0.15^*	0.28^*
Spambase	0.11^*	0.12^*	0.12^*	0.12^*
Tictacto	0.41^*	0.36^*	0.45^*	0.45^*
Wdbc	0.15	0.18	-0.42	0.10
Wpbc	0.10	0.12	-0.07	0.11
Spect	0.07	0.06	0.02	0.04
Spectf	0.25^*	0.27^*	-0.05	0.21^*
positive	8	8	5	6
negative	0	0	0	0

bagging variants: bagging, bragging, and nice bagging, taking $B = 250$ replicates throughout. The mean relative improvements over the base classification rule are reported in Tables 2, 3, 4, and 5. The last two rows in each table give the number of significant improvements (deteriorations) in relative improvement for the 12 considered data sets, allowing us to quickly assess the global performance of each bagging variant.

From Table 2 we see that bagging, as expected, indeed significantly improves the error rate if a decision tree is used as a base classifier (in 8 out of 12 cases). The results for bragging are very close to those for bagging, and this for all data sets. Moreover, as we will see from the subsequent Tables, this in not only the case for decision trees, bus also for the other base classifiers we consider. As a first conclusion we already state that bragging and bagging have very similar performance in this study.

Another conclusion we can draw from Table 2 is that nice and trimmed bagging also yield significant improvements with respect to the base classifier (in 6 out of 12 cases), and that in none of the examples a significant deterioration in error rate is observed. Trimmed bagging behaves slightly worse than standard bagging, but the difference is rather marginal.

The better performance of bagging becomes questionable when using stable classifiers. Results in Table 3 indeed confirm that bagging does not work with a support vector machine. In only one case bagging provides a significant increase in the predictive performance of SVMs. Even worse, bagging leads to significant

Table 3: As in Table 2, but now for support vector machines.

Data set	Bagging	Bragging	Nice	Trimmed
Austral	-4.88^*	-4.85^*	-0.02	0.06
Balloon	-2.59	-2.53	0.01	0.71^*
Breast	0.00	0.00	0.00	0.96^*
Cmc	-0.01	0.07	-0.01	-0.02
Crx	-4.15	-4.09	0.01	0.17
Iono	-17.11^*	-17.11^*	0.10	0.19
Spambase	-3.39^*	-3.38^*	-0.01	0.66^*
Tictacto	0.22	0.25	0.21	0.25^*
Wdbc	0.98^*	0.98^*	0.98^*	0.98^*
Wpbc	-0.40	-0.40	0.00	0.64^*
Spect	-0.05	-0.03	-0.07^*	-0.03
Spectf	0.03	0.01	0.05	0.05^*
positive	1	1	1	7
negative	3	3	1	0

Table 4: As in Table 2, but now for linear discriminant analysis.

Data set	Bagging	Bragging	Nice	Trimmed
Austral	0.01	0.01	0.00	0.00
Balloon	0.02	0.02	0.54^*	0.44^*
Breast	-0.02	-0.02	0.03	0.03
Cmc	0.01^*	0.01^*	0.01	0.01^*
Crx	0.00^*	0.00^*	0.00^*	-0.01
Iono	-0.02	-0.02	0.03	0.03
Spambase	0.01	0.01	0.05^*	0.04^*
Tictacto	0.00	0.00	0.02	0.02^*
Wdbc	-0.08^*	-0.08^*	-0.03	-0.01
Wpbc	0.00	0.00	-0.05	-0.05
Spect	0.04^*	0.04^*	0.10	0.05^*
Spectf	0.06	0.06	0.19	0.07
positive	3	3	3	5
negative	1	1	0	0

Table 5: As in Table 2, but now for logistic regression.

Data set	Bagging	Bragging	Nice	Trimmed
Austral	0.00	0.00	-0.01	-0.01
Balloon	0.07	0.07	0.43	0.41
Breast	0.07	0.07	0.07	0.04
Cmc	0.00	0.00	0.01	0.00
Crx	-0.01	-0.01	-0.02	-0.02
Iono	0.05	0.05	-0.11	0.06
Spambase	0.04	0.04	0.12*	0.05
Tictacto	0.01	0.01	0.01	0.01
Wdbc	0.41*	0.41*	0.27	0.31*
Wpbc	0.01	0.01	-0.14	0.00
Spect	0.15	0.15	0.13	0.09
Spectf	0.43*	0.43*	-0.05	0.37*
positive	2	2	1	2
negative	0	0	0	0

loss in accuracy for one quarter of the data sets here. While bragging does not bend these results, nice bagging alleviates somehow the accuracy losses, but trimmed bagging performs best. We observe a significant relative improvement in 5 out of 12 cases. Also important, trimmed bagging never causes a significant decrease in the accuracy of SVMs.

Tables 4 and 5 present results for the linear base classifiers, i.e. linear discriminant analysis and logistic regression. We see from Table 4 that there is one case where bagging yields a significant increase of the error rate. Trimmed bagging, on the other hand, is behaving the best among the bagging variants. Although the improvements when using trimmed bagging for the linear classifiers are modest, we observe that if there is a loss in error rate when applying trimmed bagging, it is very small and not significant.

5. Conclusions

In this paper we propose a modification of the bagging algorithm, called trimmed bagging. Although bagging has been found to be successful in increasing the predictive performance of unstable classifiers (like decision trees), it may lead to serious losses in accuracy when used with stable classifiers (like support vector machines). Trimmed bagging works well for both decision trees and support vector machines. Also for linear discriminant analysis and logistic regression it never leads to a significant decrease in the predictive accuracy of the base classifier in the examples we considered. The trimmed bagging method seems to be applicable, without much caution, to any stable or unstable base classifier. Trimmed bagging is based on the idea to restrict the sample of bootstrapped classifiers to the best ones. In doing so, we hope that the aggregate classifier

will outperform, or at least perform comparably to, the base classifier.

Trimmed bagging is related to nice bagging, but outperforms it in the examples we considered. One of the reasons for this is that we implemented nice bagging exactly as in Skurichina and Duin (1998), using the apparent error rate to rank the different component classifiers. When using the out-of-bag error rate instead, one is less subject to overfitting, and the nice bagging procedure performs almost as good as trimmed bagging.

We also stress the simplicity and general applicability of the trimmed bagging procedure. For instance, Valentini and Dietterich (2003), introduced *Lobag* (Low bias bagging), for aggregating Support Vector Machines. They identify low-biased classifiers, and use these as component classifiers for bagging. This approach combines the low-bias properties of SVMs with the low-variance properties of bagging. This method, however, is restricted to support vector machines only.

This paper focuses on trimmed bagging in the context of binary classification. However, the bagging concept can be applied in many different settings. For example, recent applications of bagging in functional data analysis are given in Holländer and Schumacher (2006) and Nerini and Ghattas (2007). The idea of trimmed bagging could easily be adapted to other situations. In a regression context, for example, one could use the out-of-bag mean squared prediction error (instead of the error rate) to rank the different bootstrapped regression functions. Another field for future research is the application of the trimming idea to more sophisticated classifier aggregation schemes, like those corresponding to Boosting (e.g. Hastie, Tibshirani, and Friedman (2001)) or Random Forests (Breiman (2001)). Finally, let us emphasize that no statistical theory for the trimmed bagging concept has been developed in this paper, and that we only have numerical evidence for its good performance. We do believe, however, that the trimming bagging idea is simple, easy to put in practice, and may have major benefits with respect to standard bagging.

References

Breiman, L., 1996. Bagging predictors. Machine Learning 24, 123–140.

Breiman, L., 1998. Arcing classifiers. The Annals of Statistics 26, 201-849.

Breiman, L., 2001. Random forests. Machine Learning 45, 5–32.

Bühlmann, P., Yu, B., 2002. Analyzing Bagging. The Annals of Statistics 30, 927–61.

Bühlmann, P., 2003. Bagging, subbagging and bragging for improving some prediction algorithms. In: Akritas, M.G. and Politis, D.N. (Eds.), Recent Advances and Trends in Nonparametric Statistics. Elsevier, pp. 9–34.

Buja, A., Stuetzle, W., 2006. Observations on bagging. Statistica Sinica 16,323–351.

Dietterich, T.G., 2000. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. Machine Learning 40, 139-158.

Hastie, T., Tibshirani, R., Friedman, J., 2001. The elements of statistical learning: data mining, inference and prediction. Springer-Verlag, New York.

Holländer, N., Schumacher, M., 2006. Estimating the functional form of a continuous covariate's effect on survival time. Computational Statistics & Data Analysis 50, 1131–1151

Lemmens, A., Croux, C., 2006. Bagging and Boosting Classification Trees to Predict Churn. Journal of Marketing Research 43, 276–286.

Nerini, D., Ghattas, B., 2007. Classifying densities using functional regression trees: Applications in oceanology. Computational Statistics & Data Analysis 51, 4984–4993.

Skurichina, M., Duin, B., 1998. Bagging for linear classifiers. Pattern Recognition 31, 909-930.

Valentini, G., Dietterich, T.G., 2003. Low bias bagged support vector machines. In: Fawcett, T., Mishra, N. (Eds), 20th International Conference on Machine Learning (ICML 2003), Washington DC, USA.