Communications for Statistical Applications and Methods
2014, Vol. 21, No. 5, 411–422

DOI: http://dx.doi.org/10.5351/CSAM.2014.21.5.411
Print ISSN 2287-7843 / Online ISSN 2383-4757

Double-Bagging Ensemble Using WAVE

Ahhyoun Kim[a], Minji Kim[a], Hyunjoong Kim[1,a]

[a]Department of Applied Statistics, Yonsei University, Korea

Abstract

A classification ensemble method aggregates different classifiers obtained from training data to classify new data points. Voting algorithms are typical tools to summarize the outputs of each classifier in an ensemble. WAVE, proposed by Kim *et al.* (2011), is a new weight-adjusted voting algorithm for ensembles of classifiers with an optimal weight vector. In this study, when constructing an ensemble, we applied the WAVE algorithm on the double-bagging method (Hothorn and Lausen, 2003) to observe if any significant improvement can be achieved on performance. The results showed that double-bagging using WAVE algorithm performs better than other ensemble methods that employ plurality voting. In addition, double-bagging with WAVE algorithm is comparable with the random forest ensemble method when the ensemble size is large.

Keywords: Ensemble, double-bagging, voting, classification, discriminant analysis, cross-validation.

1. Introduction

Classification is a problem of predicting a categorical target variable. Classification ensemble is a set of trained classifiers whose outcomes are combined to make a final classification (Dietterich, 2000). In classification ensemble, the final predictive result is determined by a combination of individual classifier outputs, so that a new observation can be assigned a class. Many researchers have shown that ensembles have generally higher levels of accuracy than their *base* (the components of the ensemble) models (*e.g.* Ho *et al.*, 1994; Tumer and Oza, 2003). Bagging (Breiman, 1996a), one of the leading ensemble techniques, produces multiple bootstrap sets from the original training data and uses each of them to generate a classifier for inclusion in the ensemble (Oza and Tumer, 2008). By perturbing the training data, bagging gives substantial gains in accuracy (Breiman, 1996a) as well as produces a valuable by-product called the "out of bag" sample (Breiman, 1996b). When bootstrap resamples are generated from the training data, some of the observations may be selected several times while others may not be selected at all. Regarding each bootstrap set, 63% of unique training observations were selected while the size remained unchanged with the original training data. The rest 37%, referred to as the out-of-bag(OOB) sample, can be used to construct accurate estimates of important quantities (Breiman, 1996b).

In many applications, combination method in an ensemble leads to a substantial reduction of misclassification error. Hothorn and Lausen (2003) suggests double-bagging, which is a classification procedure that combines linear discriminant analysis(LDA) and a classification tree as the base model of bagging. They estimated the coefficients of the linear discriminant function from the OOB sample.

This work was supported by the Basic Science Research Fund of the Department of Applied Statistics at Yonsei University.

[1] Corresponding author: Department of Applied Statistics, Yonsei University, Seoul 120-749, South Korea.
E-mail: hkim@yonsei.ac.kr

Published 30 September 2014 / journal homepage: http://csam.or.kr

Then the corresponding discriminant scores computed from the bootstrap sample play a role of additional predictors to be used in classification tree modeling. Hothorn and Lausen (2003) claimed that by considering linear combinations of predictors, variable selection bias has disappeared in double-bagging. Bootstrap aggregation of LDA was already investigated by Skurichina and Duin (1998) and it has been shown that, in a situation when LDA is unstable, bagging can improve the accuracy of LDA. The LDA usually becomes unstable since the sample size of OOB for the LDA is small. As a result, the double-bagging method which generates classification trees with LDA variables incorporated successfully improved the classification performance of the ordinary bagging method.

How to combine the base classifiers is an another issue that can affect the performance of an ensemble method. Bootstrap aggregation with simple majority voting, as in bagging, leads to the reduction of variance compared to a single classifier. Averaging over a variety of classifiers reduces the variance in classification since it allows to detect various aspects of a given data set. Kim *et al.* (2011) proposed a weight-adjusted voting algorithm, called WAVE, for the aggregated classifiers. WAVE can reduce bias compared to bagging, as it improves the simple majority voting scheme by considering more weights on hard-to classify instances. When this new weighted voting algorithm is combined with bootstrap aggregation, the variance would be as low as bagging since the averaging is still involved. Therefore one can achieve the low classification bias while maintaining the low variance. Kim *et al.* (2011) showed that WAVE leads to significantly better performance compared to simple majority voting under a bagging scheme. In addition, they suggested that the WAVE algorithm can be used with other aggregation schemes.

This paper attempts to show that the WAVE algorithm can be used in conjunction with double-bagging as an aggregation scheme. We applied the WAVE algorithm on double-bagging, called DB-WAVE(double-bagging ensemble using WAVE), expecting higher accuracy since double-bagging is a modified version of the bootstrap aggregation. To compare the results between double-bagging with simple majority voting and double-bagging with WAVE, we used one dimensional linear discriminant function when creating an additional variable from an OOB sample. This procedure was implemented based on 25 real datasets.

In advance of exploring DB-WAVE procedure, we introduce a double-bagging and WAVE ensemble scheme in Section 2. Section 3 describes the DB-WAVE algorithm is as an extension of the WAVE ensemble method. To evaluate the classification performance in general, we compare DB-WAVE with many other ensemble methods including the double-bagging, bagging-WAVE, bagging, boosting (Freund and Schapire, 1996) and random forest (Breiman, 2001) schemes. The experimental results are discussed in Section 4. Lastly, Section 5 gives the conclusions of this research.

2. Double-Bagging and the WAVE Algorithm

Because DB-WAVE is a combination of double-bagging and the WAVE ensemble method, we need to go over the concepts of both algorithms in detail. We introduce double-bagging first and the WAVE ensemble method is described afterwards.

We assume that we are given a learning sample $\mathfrak{L} = \{(y_i, \mathbf{x}_i),\ i = 1, \ldots, N\}$ composed of N independent observations of p-dimensional vectors of predictor $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip}) \in \mathbb{R}^p$ and class labels $y_i \in \{1, \ldots, J\}$. Let $\mathfrak{L}^* = \{(y_i^*, \mathbf{x}_i^*),\ i = 1, \ldots, N\}$ denote a random bootstrap sample of size N.

2.1. Double-bagging

Double-bagging is based on bagging, considering that double-bagging is an extended version using out-of-bag(OOB) samples. Before investigating double-bagging, we will give brief explanations of

1. Generate B random samples $\mathfrak{L}^{*(1)}, \ldots, \mathfrak{L}^{*(B)}$ with replacement N samples from the original training set $\mathfrak{L}$.
2. Train a base classifier C for each of the B samples.
3. Combine the results of all B trained classifiers by a majority vote and return class that was predicted most often (for classification)

$$C(x) = \arg\max_{y} \sum_{b=1}^{B} I\left(C_b(x) = y\right).$$

Figure 1: *Bagging (Breiman, 1996a) ensemble method*

bagging and OOB samples.

2.1.1. Bagging

Bagging(bootstrap aggregating) predictors are constructed using bootstrap samples from the training data set and then aggregated to form a bagged predictor by taking a majority vote over the class labels estimated by the classifiers (Breiman, 1996a). For classification, when predicting a class for a new instance, the aggregation takes a plurality vote. Bauer and Kohavi (1999) showed that bagging outperforms a single classifier. They also found that the error reduction is due to the decrease in variance but not bias. Figure 1 describes the bagging algorithm.

2.1.2. Out-of-bag sample

In the bagging procedure, while the bootstrap sample has the same size as the original training set, only about 63% of the unique cases are included in a given bootstrap sample. Therefore, each bootstrap sample leaves out roughly 37% of the observations. Breiman (1996b) called these omitted samples collectively the out-of-bag(OOB) sample and suggested that they can be used as a better estimation for the misclassification error of a classifier.

2.1.3. Double-bagging

Hothorn and Lausen (2003) suggested the concept of double-bagging which uses an OOB sample to combine different classifiers. The OOB sample estimates the coefficients of a linear discriminant function. The corresponding linear discriminant scores computed on the bootstrap sample are used as additional predictors for the classification trees. Hothorn and Lausen (2003) showed that double-bagging enhances the classification accuracy (among LDA, CART, and CART with bagging) in a number of artificial examples and applications.

Unlike bagging, double-bagging uses not only bootstrap samples but also OOB samples, which means that all of the information of the training set is employed. Indeed, double-bagging improved upon the results of bagging in several experiments (Hothorn and Lausen, 2003). Figure 2 describes the double-bagging algorithm.

1. Draw B random samples $\mathfrak{L}^{*(1)}, \ldots, \mathfrak{L}^{*(B)}$ with replacement from $\mathfrak{L}$ and let $X^{*(b)}$ denote the matrix of predictors $\mathbf{x}_1^{*(b)}, \ldots, \mathbf{x}_N^{*(b)}$ from $\mathfrak{L}^{*(b)}$.
2. Compute an LDA using the out-of-bag sample $\mathfrak{L} \setminus \mathfrak{L}^*$, that gives a $p \times (J-1)$ matrix $Z^{(b)}$, where the columns are the coefficients of the linear discriminant functions.
3. Construct the classifier C using the original predictor variables as well as the discriminant variables of the bootstrap sample $(\mathfrak{L}^{*(b)}, X^{*(b)}Z^{(b)})$.
4. Iterate steps (2) and (3) for all B bootstrap samples.
5. Classify a new observations $\mathbf{x}$ by majority voting using the predictions of all classifiers, where a classifier $C(\mathbf{x}, \mathfrak{L})$ predicts future y values for a vector of predictors $\mathbf{x}$ based on a learning sample $\mathfrak{L}$

$$C\left(\left(\mathbf{x}, \mathbf{x}Z^{(b)}\right), \left(\mathfrak{L}^{*(b)}, X^{*(b)}Z^{(b)}\right)\right) \quad \text{for } b = 1, \ldots, B.$$

Figure 2: *Double-bagging (Hothorn and Lausen, 2003) ensemble method*

2.2. WAVE algorithm

The WAVE algorithm needs to be explained since WAVE algorithm is closely related to DB-WAVE. Majority voting is a general ensemble combination technique that can be applied to any type of classifier. Two categories of majority voting is available; a simple majority voting and a weighted voting.

WAVE(weight-adjusted voting for ensembles of classifiers) proposed by Kim *et al.* (2011) is a new weighted voting algorithm that uses two weight vectors: a weight vector of classifiers and a weight vector of instances. The idea of WAVE is to give more weights to classifiers that can classify hard-to-classify instances correctly.

The WAVE algorithm is different from the weighted voting algorithm of boosting in terms of the formation of weight vectors. Both WAVE and boosting are based on the idea of giving more weights to hard-to-classify instances. However, the meaning of "hard-to-classify" instances in WAVE is not equal to the definition of boosting. While the latter means that observations are misclassified by the previous classifier during the ensemble formation, the former is defined as the most often misclassified observations in an ensemble. The two weight vectors for the instance and the classifier in the weighted voting algorithm are formulated at different time periods. In boosting, the weight vectors are obtained simultaneously as the sizes of the ensembles increase. Meanwhile, the weight vectors in WAVE are determined after the ensemble is completely formed.

Let $\mathbf{X}$ be a performance matrix indicating whether the classification is correct(1) or incorrect(0) (for N instances and B classifiers in an ensemble). Let $\mathbf{J}_{ij}$ be an $i \times j$ matrix consisting of 1's for any dimension i and j. We also define $\mathbf{1}_n$ as $n \times 1$ vectors of 1's and $\mathbf{I}_k$ denote a $k \times k$ identity matrix. Kim *et al.* (2011) investigates the proof of the convergence of the classifier weight $\mathbf{P}^*$ for voting classifiers, which can be obtained directly from $\mathbf{X}'(\mathbf{J}_{NB} - \mathbf{X})(\mathbf{J}_{BB} - \mathbf{I}_B)$. The bagging with WAVE (bagging-WAVE) is performed (Figure 3).

Kim *et al.* (2011) demonstrated that the proposed weighted voting scheme performs significantly better than simple majority voting. Note that the ensemble method adopted in Kim *et al.* (2011) is bootstrap aggregation as in bagging. We apply the WAVE algorithm to double-bagging in the following section.

1. Obtain B bootstrap samples with replacement from a training set made up of N instances.
2. Train B classifiers from each bootstrap sample.
3. Generate an $N \times B$ matrix $\mathbf{X} = [X_1, \ldots, X_B]$ consisting of 0's (incorrect) and 1's (correct) and a $B \times B$ matrix $\mathbf{T} = \mathbf{X}'(\mathbf{J}_{NB} - \mathbf{X})(\mathbf{J}_{BB} - \mathbf{I}_B)$.
4. Calculate the optimal weight vector $\mathbf{P}^*$ from $\mathbf{T}$.

$$\mathbf{P}^* = \frac{(\sum_{i=1}^{r} \mathbf{u}_i \mathbf{u}_i')\mathbf{1}_B}{\mathbf{1}_B'(\sum_{i=1}^{r} \mathbf{u}_i \mathbf{u}_i')\mathbf{1}_B} = [\mathbf{P}_1^*, \ldots, \mathbf{P}_B^*]'$$

   Here $\mathbf{u}_i$ is an eigenvector corresponding to the eigenvalues of $\mathbf{T}$ and r denotes the number of dominating eigenvalues.

5. Combine the outputs of the classifiers using a weight-adjusted voting algorithm

$$C^*(x) = \arg\max_y \sum_{b=1}^{B} \mathbf{P}_b^* \times I\left(C_b(x) = y\right).$$

Figure 3: *Bagging-WAVE (Kim et al., 2011) ensemble method*

3. Double-Bagging Ensemble Using WAVE (DB-WAVE)

In the DB-WAVE algorithm, double-bagging is used to construct an ensemble of classifiers and the WAVE algorithm is then applied to the ensemble to aggregate the outcomes of all classifiers. Let $\mathfrak{L} = \{(y_i, \mathbf{x}_i),\ i = 1, \ldots, N\}$ be a learning sample composed of N independent observations of p-dimensional vectors of predictor $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip}) \in \mathbb{R}^p$ and class labels $y_i \in \{1, \ldots, J\}$. Also, $\mathfrak{L}^* = \{(y_i^*, \mathbf{x}_i^*),\ i = 1, \ldots, N\}$ denotes a random bootstrap sample of size N. The DB-WAVE method is defined in Figure 4.

We perform a canonical LDA under the one-dimensional constraint of a canonical linear discriminant function using the OOB samples. The corresponding linear discriminant variable can be computed by multiplying canonical coefficients and the original predictor variables. Then, we add the new discriminant variable to the bootstrap sample. Let this bootstrap sample including the new variable be called double-bagging training set. At each double-bagging training set, we construct the classifier C and apply the WAVE algorithm to combine the outputs of all classifiers. Figure 5 describes this process of the DB-WAVE ensemble formation. Similar to Hothorn and Lausen (2003), we adopted classification trees as the classifiers.

4. Experimental Results

DB-WAVE is compared to double-bagging and other popular ensemble methods in an empirical study. The experiment is conducted based on 25 actual datasets, most of which come from UCI Data Repository (Asuncion and Newman, 2007). Table 1 gives the descriptions of the datasets. All the methods are run on the same 10-folded cross-validation data in order to compare the performances fairly. We also repeat the cross-validation 100 times for eliminating the effect of random seeds. An average of 100 cross-validation accuracy assessments for each ensemble method is calculated in Table 2 (we only

1. Draw B random samples $\mathfrak{L}^{*(1)}, \ldots, \mathfrak{L}^{*(B)}$ with replacement from $\mathfrak{L}$ and let $X^{*(b)}$ denote the matrix of predictors $\mathbf{x}_1^{*(b)}, \ldots, \mathbf{x}_N^{*(b)}$ from $\mathfrak{L}^{*(b)}$.

2. Perform a one-dimensional canonical LDA using the OOB(out-of-bag) sample $\mathfrak{L} \setminus \mathfrak{L}^*$ that gives a $p \times 1$ column vector $Z^{(b)}$ (restricted to one column for all examples), where the columns are the coefficients of the linear discriminant functions.

3. Build the classifier C using the original variables as well as the discriminant variable of the bootstrap sample $(\mathfrak{L}^{*(b)}, X^{*(b)}Z^{(b)})$.

4. Repeat steps (2) and (3) for all B bootstrap samples.

5. Generate an $N \times B$ matrix $\mathbf{X} = [X_1, \ldots, X_B]$ consisting of 0's (incorrect) and 1's (correct) and a $B \times B$ matrix $\mathbf{T} = \mathbf{X}'(\mathbf{J}_{NB} - \mathbf{X})(\mathbf{J}_{BB} - \mathbf{I}_B)$.

6. Calculate the optimal weight vector $\mathbf{P}^*$ from $\mathbf{T}$.

$$\mathbf{P}^* = \frac{(\mathbf{u}_1\mathbf{u}_1')\mathbf{1}_B}{\mathbf{1}_B'(\mathbf{u}_1\mathbf{u}_1')\mathbf{1}_B} = [\mathbf{P}_1^*, \ldots, \mathbf{P}_B^*]'$$

   Here $\mathbf{u}_1$ is the eigenvector corresponding to the largest eigenvalue of $\mathbf{T}$.

7. Combine the outputs of the classifiers using the weighted voting algorithm of WAVE

$$C^*(x) = \arg\max_y \sum_{b=1}^{B} \mathbf{P}_b^* \times I\left(C_b(x) = y\right).$$

Figure 4: *DB-WAVE (Double-Bagging Ensemble using WAVE) ensemble method*

present the most representative result in which 64 base classifiers are combined). Table 3 summarizes the results of accuracy tables for other ensemble sizes. Throughout the experiment, CART (Breiman *et al.*, 1984), one of the famous decision tree algorithms, is employed as a base classifier, except that LDA is used in conjunction with CART in double-bagging and DB-WAVE methods. RPART (Therneau and Atkinson, 1997), an implementation of CART, was run under the R environment. For boosting, we use SAMME (Zhu *et al.*, 2009), which is a modified version of AdaBoost for multiclass problems. We used random forest (Liaw and Wiener, 2002) function under the R environment to implement the random forest (Breiman, 2001). The options needed for RPART, boosting and random forest are equal to those in Kim *et al.* (2011).

4.1. Accuracy comparison

The accuracies in Table 2 represent the average of 100 accuracies obtained when the ensemble size is 64. The first and second best accuracies of each dataset are highlighted in boldface. It was found that DB-WAVE has the most highlights. Random forest has the second most highlights. Bagging-WAVE, a bagging with WAVE algorithm, has more highlights than the bagging with simple majority voting. This finding is in good agreement with that of Kim *et al.* (2011), who showed that bagging-WAVE performs better than bagging quite consistently. Hothorn and Lausen (2003) found that double-bagging improves upon the results of bagging in many experiments. Their finding matches our results.

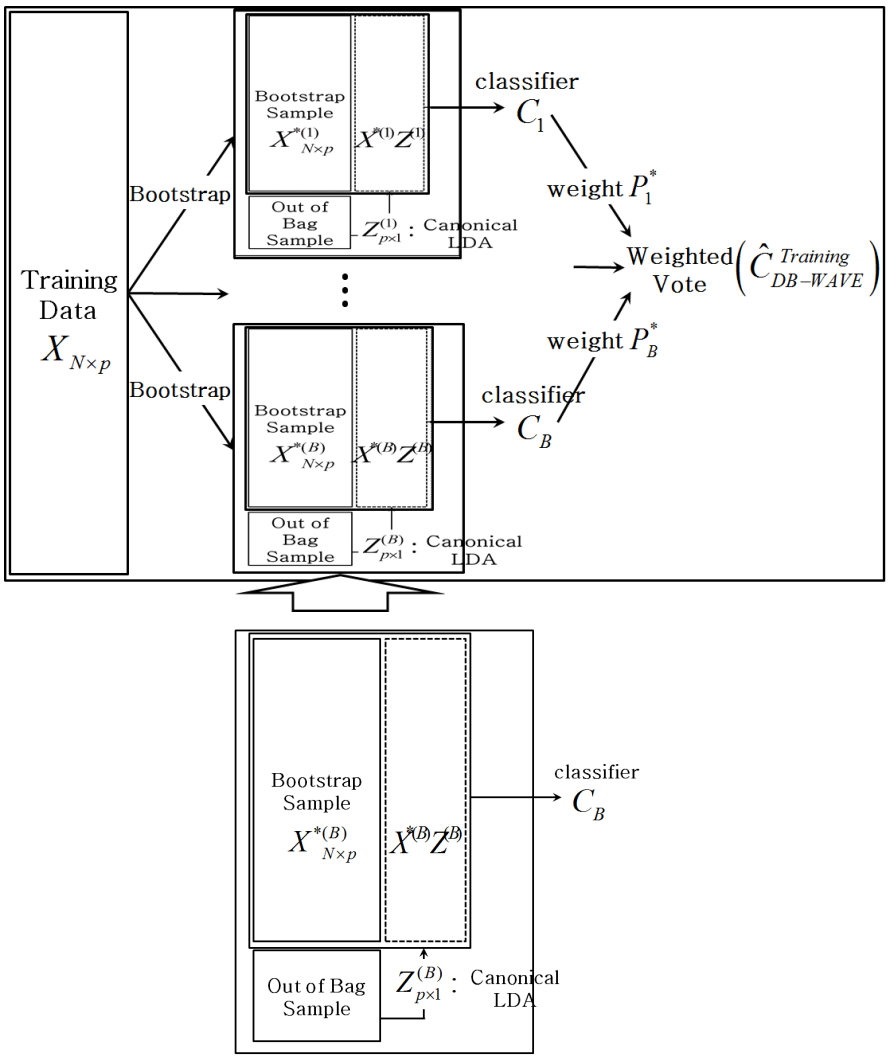

Figure 5: *The DB-WAVE algorithm*

The performance of boosting is dependent on the characteristic of dataset as Opitz and Maclin (1999) has demonstrated. The DB-WAVE ensemble method has a potential to be a superior one since both the WAVE algorithm and double-bagging ensemble method can boost the performance of bagging. It appears that DB-WAVE can improve both bagging-WAVE and double bagging because it has more highlights than others. The statistical significance of the comparison will be discussed in section 4.2.

Table 3 shows that the most frequent method among the first and second best ensemble methods is DB-WAVE when we change the ensemble size. DB-WAVE has the most highlights in all ensemble sizes except 2 in which DB-WAVE has the second most highlights. A single tree, which is used as a base learner, has the best performance when 2 classifiers are used in an ensemble. The unique instances are only about 63% in the bootstrap training data. Therefore, with only two ensembles, the

Table 1: Description of the dataset.

Dataset	Description	# instance	# variables	# classes	Source
Bcw	Breast cancer Wisconsin	699	10	2	UCI
Bld	Liver disorders	345	6	2	UCI
Bod	Body dimension	507	24	2	Heinz *et al.* (2003)
Bos	Boston housing	506	14	3	UCI
Cmc	Contraceptive method choice	1473	9	3	UCI
Col	Horse Colic	368	27	3	UCI
Cre	Credit approval	690	15	2	UCI
Cyl	Cylinder bands	540	35	2	UCI
Dia	Diabetes	532	7	2	Loh (2009)
Fis	Fish	159	7	7	Kim and Loh (2003)
Ger	German credit	1000	20	2	UCI
Gla	Glass	214	9	6	UCI
Hea	Statlog (Heart)	270	13	2	UCI
Int	Chessboard	1000	10	2	Kim *et al.* (2011)
Ion	Ionosphere	351	34	2	UCI
Iri	Iris	150	4	3	UCI
Led	LED display domain	6000	7	10	UCI
Pid	Pima Indians diabetes	768	8	2	UCI
Pov	Poverty	97	6	6	Kim and Loh (2001)
Sea	Vocalizations of harp seals	3000	7	3	Terhune (1994)
Spe	SPECTF heart	267	44	2	UCI
Usn	Usnews	1302	27	3	Statlib (2010)
Veh	Statlog (Vehicle Silhouettes)	946	18	4	UCI
Vol	Volcano	1521	6	6	Loh (2009)
Vow	Vowel recognition	990	10	11	UCI

bootstrap training set has far fewer unique observations than the original training set. As a result, the two-classifier ensemble does not perform better than a single tree.

This leads to the conclusion that DB-WAVE performs better or as good as other ensemble methods across the ensemble size. Random forest (Breiman, 2001), a powerful ensemble method with high prediction accuracy, is comparable to DB-WAVE only when the ensemble size is 64.

4.2. Significance

Since the ensemble methods were run on the same 10-fold cross-validated data for 100 times, 100 accuracies were available for a statistical comparison in each dataset. We carried out the paired t-tests to examine the pairwise differences of ensemble methods. The paired t-tests are completed separately for each of 25 dataset. Figure 6 presents the test-statistics of the paired t-tests. In the boxplot, a larger t-statistic indicates better accuracy for one method over the other. Various sizes of ensemble (2, 4, 8, 16, 32, and 64) was used to determine if a relationship between the t-statistics and the ensemble sizes exist. These numbers are selected because bootstrap replications between 20 and 100 are said to be reasonable depending on the classifier and the data (Skurichina and Duin (1998); Efron and Tibshirani (1986)).

The comparison of double-bagging and DB-WAVE equates to the differences between simple majority voting and weight-adjusted voting under the same double-bagging scheme. The t-statistics consistently have large positive values across the ensemble size. Thus, the performance of double-bagging improved when the WAVE algorithm is employed.

DB-WAVE is compared to bagging-WAVE, a weighted-adjusted voting on bagging. It turned out that DB-WAVE and bagging-WAVE are quite comparable with slight margin for DB-WAVE when ensemble sizes are 32 and 64. In the comparison of DB-WAVE and boosting, it was noted that DB-

Table 2: Comparison of prediction accuracy over 25 datasets (ensemble size = 64).

Data	Bagging-WAVE	Bagging	DB-WAVE	Double-bagging	Boosting	Random Forest	Rpart
Bcw	0.962	0.962	**0.972**	**0.972**	0.962	0.971	0.945
Bld	0.723	0.723	**0.742**	**0.741**	0.723	0.725	0.681
Bod	0.936	0.934	**0.987**	**0.987**	0.978	0.942	0.914
Bos	0.776	0.776	0.782	0.782	**0.782**	**0.784**	0.741
Cmc	**0.561**	**0.561**	0.558	0.557	0.547	0.551	0.550
Col	0.709	0.708	**0.710**	0.709	0.683	**0.714**	0.661
Cre	**0.863**	0.863	0.858	0.858	0.851	**0.865**	0.852
Cyl	**0.749**	0.747	0.740	0.738	**0.751**	0.740	0.646
Dia	**0.763**	**0.763**	0.761	0.761	0.756	0.762	0.748
Fis	0.823	0.822	**0.830**	**0.832**	0.825	0.812	0.803
Ger	0.755	0.755	**0.759**	**0.758**	0.751	0.746	0.725
Gla	0.758	0.757	0.747	0.746	**0.779**	**0.767**	0.702
Hea	0.827	0.826	**0.833**	0.832	0.788	**0.833**	0.776
Int	**0.824**	0.798	**0.814**	0.789	0.631	0.532	0.800
Ion	0.911	0.909	0.912	0.910	**0.924**	**0.926**	0.873
Iri	0.956	0.954	**0.959**	**0.958**	0.951	0.957	0.954
Led	**0.717**	0.714	0.715	0.713	0.706	**0.726**	0.682
Pid	**0.781**	**0.781**	0.774	0.774	0.766	0.780	0.749
Pov	0.671	0.672	**0.673**	**0.673**	0.610	0.632	0.655
Sea	0.593	0.592	0.595	0.591	**0.646**	**0.607**	0.581
Spe	0.807	**0.807**	0.806	0.806	0.797	**0.809**	0.738
Usn	**0.751**	**0.751**	0.750	0.750	0.717	0.737	0.732
Veh	0.713	0.713	**0.739**	0.739	**0.747**	0.719	0.682
Vol	0.884	0.881	**0.884**	0.882	**0.887**	0.874	0.872
Vow	0.706	0.704	0.712	0.710	**0.942**	**0.726**	0.601
# highlights	8	5	12	7	8	10	0
# best	3	2	6	1	6	7	0
# worst	0	0	0	0	5	1	19

The first and second best results are highlighted in boldface.
The numbers of best and worst results are shown at the bottom. Duplication count is allowed.

Table 3: The two best ensemble methods.

Ensemble size	The two best methods
2	CART, DB-WAVE
4	DB-WAVE, bagging-WAVE
8	DB-WAVE, bagging-WAVE
16	DB-WAVE, bagging-WAVE
32	DB-WAVE, double-bagging
64	DB-WAVE, Random Forest

WAVE is more effective than boosting algorithm. It was found that DB-WAVE and random forest are quite comparable in large ensemble sizes, while DB-WAVE is better for small ensemble sizes. Figure 6 shows that DB-WAVE is consistently superior to a single decision tree except when the ensemble size is 2. Very large t-statistic values observed in Figure 6 suggest DB-WAVE is dominant in those dataset: 'bod', 'int', and 'bcw'.

5. Concluding Remarks

This study assessed a significant improvement in performance with a combination of double-bagging and the WAVE algorithm. Our results have something in common with the earlier study by Kim *et al.* (2011) that used WAVE on bagging.

Experiments comparing DB-WAVE with other methods(including double-bagging) were run to

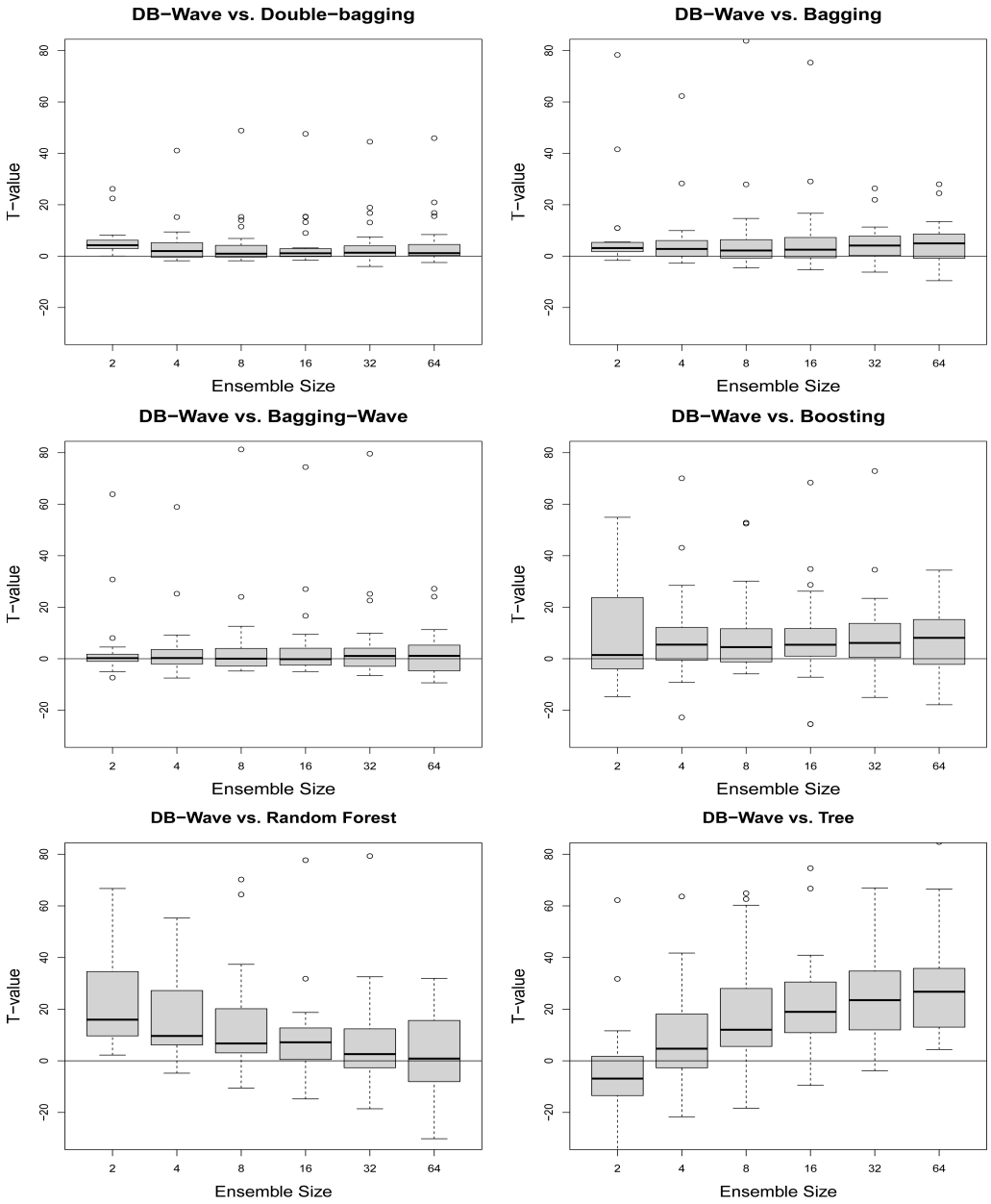

Figure 6: *Paired t-statistics for comparing DB-WAVE and the other methods.*

determine their accuracy levels using 25 actual datasets. In conclusion, the performance of double-bagging improves when employed with the WAVE algorithm. DB-WAVE outperforms most ensemble methods and appears comparable with the random forest method when the ensemble size is large.

Currently, only one-dimensional canonical LDA was considered to produce an additional variable

in double-bagging setting. However, although not included in this article, it is also possible to apply higher dimensional canonical LDA to get additional predictors. In addition, we also note that other classification method can be utilized with the OOB sample as a variant of double-bagging.

The ensemble size was limited to 64. This is enough size to show an improvement acquired by WAVE algorithm on double-bagging ensemble method. However, a caution is necessary when comparing with other ensemble methods such as random forest because it generally requires more ensemble sizes than 64. In terms of computation time, we note that WAVE algorithm would make the ensemble procedures slow due to its weight calculation process; however, the difference is not noticeable.

As a future study, we plan to apply the WAVE algorithm to other ensemble methods that use simple majority voting. For example, random forest with WAVE algorithm is under investigation. It is not clear yet how the WAVE algorithm will work on the boosting method which is not simple majority voting.

References

Asuncion, A. and Newman, D. J. (2007). UCI machine learning repository, University of California, Irvine, School of Information and Computer Sciences, http://archive.ics.uci.edu/ml/.

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

Breiman, L. (1996a). Bagging predictors, *Machine Learning*, **24**, 123–140.

Breiman, L. (1996b). Out-of-bag estimation, Technical Report, Statistics Department, University of California Berkeley, Berkeley, California 94708, http://www.stat.berkeley.edu/ breiman/ OOBestimation.pdf.

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

Breiman, L., Friedman, J. H., Olshen, R. A. and Stone, C. J. (1984). *Classification and Regression Trees*, Chapman and Hall, New York.

Dietterich, T. (2000). *Ensemble Methods in Machine Learning*, Springer, Berlin.

Efron, B. and Tibshirani, R. (1986). Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy, *Statistical Science*, **1**, 54–75.

Freund, Y. and Schapire, R. (1996). Experiments with a new boosting algorithm, In *Proceedings of the Thirteenth International Conference on Machine Learning*, **96**, 148–156.

Heinz, G., Peterson, L. J., Johnson, R. W. and Kerk, C. J. (2003). Exploring relationships in body dimensions, *Journal of Statistics Education*, **11**, http://www.amstat.org/publications/jse/v11n2/datasets.heinz.html.

Ho, T. K., Hull, J. J. and Srihari, S. N. (1994). Decision combination in multiple classifier systems, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **20**, 832–844.

Hothorn, T. and Lausen, B. (2003). Double-bagging: Combining classifiers by bootstrap aggregation, *Pattern Recognition*, **36**, 1303–1309.

Kim, H. and Loh, W. Y. (2001). Classification trees with unbiased multiway splits, *Journal of American Statistical Association*, **96**, 589–604.

Kim, H. and Loh, W. Y. (2003). Classification trees with bivariate linear discriminant node models, *Journal of Computational and Graphical Statistics*, **12**, 512–530.

Kim, H., Kim, H., Moon, H. and Ahn, H. (2011). A weight-adjusted voting algorithm for ensembles of classifiers, *Journal of the Korean Statistical Society*, **40**, 437–449.

Liew, A. and Wiener, M. (2002). Classification and regression by random forest, *R News*, **2**, 18–22.

Loh, W. Y. (2009). Improving the precision of classification trees, *The Annals of Applied Statistics*, **3**, 1710–1737.

Opitz, D. and Maclin, R. (1999). Popular ensemble methods: An empirical study, *Journal of Artificial Intelligence Research*, **11**, 169–198.

Oza, N. C. and Tumer, K. (2008). Classifier ensembles: Select real-world applications, *Information Fusion*, **9**, 4–20.

Skurichina, M. and Duin, R. P. (1998). Bagging for linear classifiers, *Pattern Recognition*, **31**, 909–930.

Statlib (2010). Datasets archive, Carnegie Mellon University, Department of Statistics, http://lib.stat.cmu.edu.

Terhune, J. M. (1994). Geographical variation of harp seal underwater vocalisations, *Canadian Journal of Zoology*, **72**, 892–897.

Therneau, T. and Atkinson, E. (1997). An introduction to recursive partitioning using the RPART routines, Mayo Foundation, Rochester, New York. http://eric.univlyon2.fr/ricco/cours/didacticiels/r/long_ doc_rpart.pdf.

Tumer, K. and Oza, N. C. (2003). Input decimated ensembles, *Pattern Analysis and Applications*, **6**, 65–77.

Zhu, J., Zou, H., Rosset, S. and Hastie, T. (2009). Multi-class AdaBoost, *Statistics and Its Interface*, **2**, 349–360.

Received June 8, 2014; Revised July 28, 2014; Accepted July 29, 2014