Weight-Adjusted Bagging of Classification Algorithms Sensitive to Missing Values

Kuo-Wei Hsu

***Abstract*—Bagging is commonly used to improve the performance of a classification algorithm by first using bootstrap sampling on the given data set to train a number of classifiers and then using the majority voting mechanism to aggregate their outputs. However, the improvement would be limited in the situation where the given data set contains missing values and the algorithm used to train the classifiers is sensitive to missing values. We propose an extension of bagging that considers not only the weights of the classifiers in the voting process but also the incompleteness of the bootstrapped data sets used to train the classifiers. The proposed extension assigns a weight to each of the classifiers according to its classification performance and adjusts the weight of each of the classifiers according to the ratio of missing values in the data set on which it is trained. In experiments, we use two classification algorithms, two measures for weight assignment, and two functions for weight adjustment. The results reveal the potential of the proposed extension of bagging for working with classification algorithms sensitive to missing values to perform classification on data sets having small numbers of instances but containing relatively large numbers of missing values.**

***Index Terms*—Bagging, missing values, multilayer perceptron, sequential minimal optimization.**

I. Introduction

For classification, an ensemble is a group of classifiers and represents an instance of collective intelligence. One of the possible advantages of using an ensemble rather than a single classifier is the enhancement of the classification performance for low-quality data [1], [2].

Bagging (bootstrap aggregating) is an ensemble algorithm and has caused general interests since it was proposed by Breiman in mid-1990's [3]. It has been applied in various applications and also exerts influence on some other ensemble algorithms. Bagging is commonly used to improve the performance of a classification algorithm. Given a data set, it first uses bootstrap sampling to generate data sets that later will be used to train classifiers, and then it uses the majority voting mechanism to aggregate outputs of the classifiers [3], [4].

In many real-world applications, the situation is that the data contain missing values but the classification algorithm considered the best for the data is sensitive to missing values. Two examples are multilayer perceptron (MLP, a type of artificial neural network algorithms) [5] and sequential minimal optimization (SMO, a type of support vector machine algorithms) [6], [7]. In such a situation, the improvement given by bagging would be limited. The goal of this paper is to present an approach to improve the performance of bagging when it is used with the two algorithms in such a situation.

We propose an extension of bagging, weight-adjusted bagging. It assigns a weight to each of the classifiers in an ensemble according to the classification performance achieved by each classifier, and it takes into account the weights of the classifiers in the voting process. Moreover, it takes into account the incompleteness of the data sets generated by bootstrap sampling (sampling with replacement) and used to train the classifiers, and it adjusts the weight of each classifier according to the ratio of missing values in the data set on which each classifier is trained.

For weight assignment, we use accuracy or F1-measure; for weight adjustment, we propose two functions (as introduced later). Neither using weighted voting in an ensemble [8] nor using weight adjustment in an ensemble [9] is a new idea, but the idea presented in this paper is different from others in that it incorporates the information about the characteristics of the data sets used in training into the process in which the individual outputs of classifiers in an ensemble are aggregated to form the final output of the ensemble. That is, it considers the *context* in which the classifiers in an ensemble were trained. From such a point of view, this paper is the foundation for research on context-aware aggregation for ensembles.

The rest of this paper is organized as follows: We present the procedures of the proposed extension of bagging for training and testing in Section II, report the experimental results in Section III, and discuss related papers in Section IV. Finally, we conclude this paper in Section V.

II. Procedures

The procedure to create any classifiers, including ensembles, is called training; the procedure to use a classifier, or an ensemble, in called testing in this paper. These two procedures are the main focus of this section.

Bagging creates or trains an ensemble in 2 steps: First, it uses bootstrap sampling to generate a number of training sets. Second, it applies the pre-specified classification algorithm on the training sets one by one to create a number of classifiers. How bagging creates or trains an ensemble is illustrated in Fig. 1. The proposed extension of bagging also uses the same steps to create or train an ensemble, but it additionally records the information about the characteristics of training sets.

Two major ingredients to train a classifier are the algorithm and the training set. The classification performance

Manuscript received May 10, 2013; revised July 12, 2013. This work was supported by the National Science Council of Taiwan under grant number NSC 101-2221-E-004-011. The support is gratefully acknowledged.

Kuo-Wei Hsu is with the Department of Computer Science, National Chengchi University, Taipei, Taiwan (e-mail: hsu@cs.nccu.edu.tw).

DOI: 10.7763/IJIET.2013.V3.335

of a classifier depends on how good its classification algorithm is and how good its training set is. Since we use only one classification algorithm to train all the classifiers in an ensemble (and there is no classification algorithm that can outperform all others in all applications), we relate the quality of a classifier to the quality of its training set. That is, we consider the situation, or the *context*, in which a classifier is trained by a pre-specified algorithm in a training set.

We further relate the quality of a training set of a classifier to the completeness (or the incompleteness) of the training set. Since in this paper we are interested in the situation where the used classification algorithms, e.g. MLP and SMO, are sensitive to missing values, we relate the quality of a training set of a classifier to the ratio of missing values, or the sparsity, of the training set.

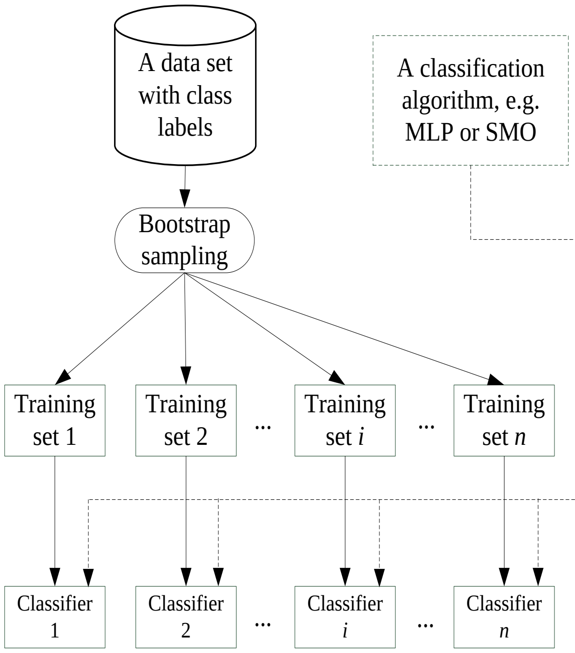

Fig. 1. Procedure for training.

Fig. 2 presents the procedure of weight-adjust bagging for testing. For a new data record, bagging will give it to the classifiers and aggregate their outputs to form the final output. The proposed extension of bagging will do the same but additionally uses the values of the sparsity of the training sets of classifiers when performing aggregation (indicated by the solid lines connecting "sparsity of training set" to "aggregation" in Fig. 2). It performs aggregation in 2 steps, namely weight assignment and weight adjustment. Fig. 2 presents what makes the proposed extension of bagging different from bagging and its other extensions.

For weight assignment, the first option is to use accuracy. The more accurate a classifier is, the better it is, and the more weight it will be given. The second option is to use F1-measure, the harmonic mean of precision and recall. The higher value of F1-measure a classifier achieves, the better it is, and the more weight it will receive. We calculate accuracy and F1-measure on training sets. This makes unnecessary the use of validation sets (while it is not always easy to obtain more or larger data sets in some applications), and this helps us obtain a group of classifiers, each of which is specifically trained on a portion of the given data set (generated by using bootstrap sampling). Assigning weights to classifiers according to their training performance would increase the risk of overfitting. The risk could be decreased by weight adjustment. From the experimental results reported in Section III, we can find data sets on which weighted bagging is no better than bagging but is weight-adjustment bagging.

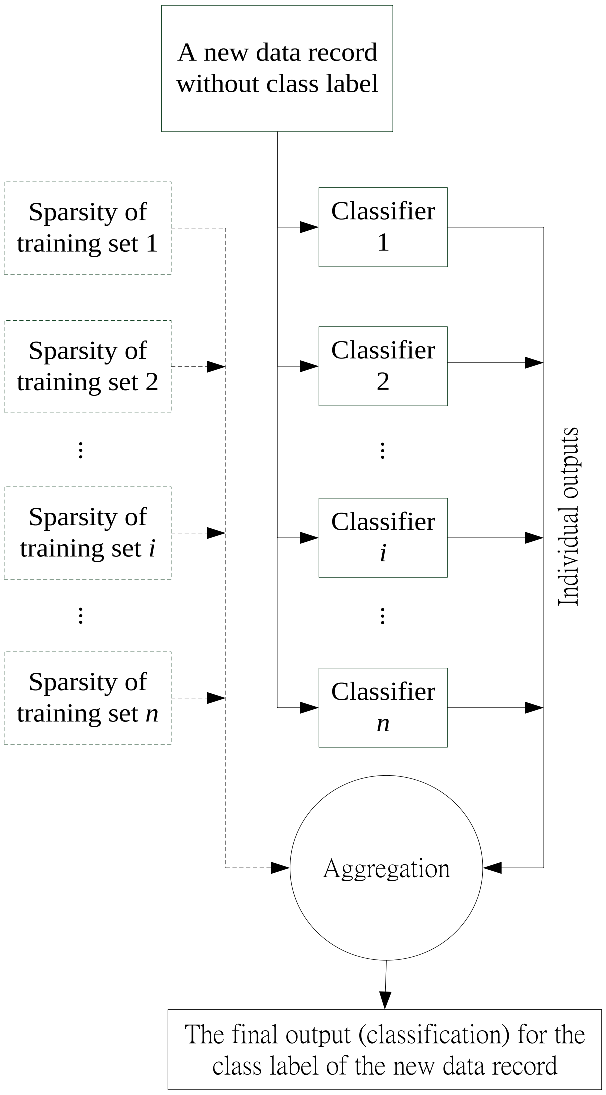
A new data record without class label
Sparsity of training set 1
Sparsity of training set 2
⋮
Sparsity of training set *i*
⋮
Sparsity of training set *n*
Classifier 1
Classifier 2
⋮
Classifier *i*
⋮
Classifier *n*
Individual outputs
Aggregation
The final output (classification) for the class label of the new data record

Fig. 2. Procedure for testing.

We propose to use two functions for weight adjustment. Each adjusts the weight of a classifier in an ensemble created by weighted-adjusted bagging according to the sparsity of its training set. Equation (1) gives the definition of the sparsity of a data set, denoted by s ($0 \leq s \leq 1$) and calculated by dividing the number of missing values by the product of the number of instances and the number of attributes.

$$s = \frac{\#\,missing\ \ values}{\#\,instances\ \ \times\#\,attributes} \tag{1}$$

The core idea of weight adjustment is to help identify effective (or ineffective) classifiers. It can be broken down into 4 parts or points, as given below:

1) If a classifier performs well on a low-quality training set (i.e. a difficult data set), we conclude that it is an effective classifier with a higher degree of confidence, and we would expect it to demonstrate satisfactory classification performance when it is given new and unseen data; therefore, we should increase its weight.

2) If a classifier performs well on a high-quality training set (i.e. an easy data set), we are less confident that it is an effective classifier because most of its effectiveness may be contributed by the easiness of its training set but not its underlying classification algorithm; we consider increasing its weight, but not as much as we do in the previous one..
3) If a classifier does not performs well on a high-quality training set (i.e. an easy data set), we conclude that it is not an effective classifier with a higher degree of confidence, and we would not expect it to demonstrate satisfactory classification performance when it is given new and unseen data; therefore, we should decrease its weight.
4) If a classifier does not perform well on a low-quality training set (i.e. a difficult data set), we are less confident that it is an ineffective classifier because its poor performance may not be caused by its underlying classification algorithm but the low-quality training set; we consider decreasing its weight, but not as much as we do in the previous one.

Based on the core idea describe above, we propose two functions for weight adjustment. For the *i*-th classifier in an ensemble created by weighted-adjusted bagging, the original and adjusted values of its weight are denoted by $w^{i}_{original}$ and $w^{i}_{adjusted}$, respectively. Equation (2) shows the first function used to adjust weights, where EXP is the exponential function.

$$w^{i}_{adjusted} = w^{i}_{original} \times \mathrm{EXP}(s) \tag{2}$$

Equation (3) shows the second function used to adjust weights, where LOG is the logarithmic function. It is valid only when *s* is lower than 0.5. For a value of *s* closer to 0.5, (3) the adjusted value of the weight of a classifier given by (3) is higher than that given by (2). That is, a classifier whose training set contains missing values is compensated more by (3) than by (2), and the compensation is higher when there are more missing values.

$$w^{i}_{adjusted} = w^{i}_{original} \times \left(1 + \mathrm{LOG}\left(\frac{0.5+s}{0.5-s}\right)\right) \tag{3}$$

There are surely other possible functions suitable for weight adjustment, and an exploration of other possibilities is part of the future work.

III. Experiments

The main aim of this section is to report and discuss experimental results.

A. Data Sets

Data sets used in experiments are downloaded from the Internet [10], [11] and summarized in Table I. Each data set is associated with a binary classification problem. In Table I, the columns from left (the 1st) to right (the 5th) specify the name, the size[1], the dimension (the size of the attribute or feature space), the percentage of the minority class, and the sparsity of a data set, respectively. Data sets are sorted by their values of sparsity (where sparsity is the relative number of empty cells if we view a data set as a data matrix) in an ascending order and their numbers of instances in a descending order.

TABLE I: Summary of Data Sets

name	instances	attributes	minority %	sparsity %
cleveland	303	13	45.9	0.2
breast-w	699	9	34.5	0.3
breast-tumor	286	9	29.7	0.3
credit-a	690	15	44.5	0.6
credit	490	15	44.3	0.6
biomed	209	8	35.9	0.9
audiology-n	226	69	9.7	2
runshoes	60	10	30	2.3
echo-months	130	7	32.3	4.4
vote	435	16	38.6	5.6
hepatitis	155	19	20.6	5.7
schizo	340	13	47.9	18.9
hungarian	294	13	36.1	20.5
colic	368	22	16.2	23.8
labor	57	16	35.1	35.7

Roughly speaking, the levels of quality of the data sets decrease or the levels of difficulty of the data sets increase from top to bottom of Table I. That is, the data sets in the upper part of Table I are associated with the classification problems more difficult than those with which the data sets in the lower part of Table I are associated.

B. Settings

Using WEKA [12], we extend its implementation of bagging, and we use its implementations of MLP (multilayer perceptron) and SMO (sequential minimal optimization) with their default parameters. 5 × 2 CV (5 iterations of 2-fold cross-validation) is commonly used in model comparison. 2-fold cross-validation generates a testing set as large as the corresponding training set, and this helps us examine the generalization capability of the compared models. In what follows, we report and discuss results from 10 × 2 CV, where different seeds of random numbers are used in different iterations.

We compare bagging (B), weighted bagging (WB), and weight-adjusted bagging (WAB) in experiments. The number of the classifiers in an ensemble created by any of the 3 algorithms is set to 10.

C. Results

In this subsection, we report and discuss the experimental results. First of all, we report results in accuracy for MLP in Table II. Then, we report results in F1-measure for MLP, accuracy for SMO, and F1-measure for SMO in Table III, Table IV, and Table V, respectively.

In each of these tables, the 1st (the leftmost) and the 2nd (the second leftmost) columns specify the name of a data set and the algorithm, respectively. The other 4 columns are results corresponding to the following 4 cases:

1) Accuracy is used for weight assignment and (2) is used for weight adjustment (the 3rd column)
2) F1-measure is used for weight assignment and (2) is used for weight adjustment (the 4th column)
3) Accuracy is used for weight assignment and (3) is used for weight adjustment (the 5th column)

[1] We use small data sets because the employed implementations of MLP and SMO take much time when unning on the adopted computing platform. We would resolve the issue in the near future.

4) F1-measure is used for weight assignment and (3) is used for weight adjustment (the 6th column)

Below is the summary of the results reported in Table II: For all the 4 cases, WB is better than B on 4 data sets, while WAB is better than both WB and B on at least 7 data sets. For Case 3, WAB is better than the others on 8 out of 15 data sets. WAB is better than WB and B on *hepatitis*, *schizo*, *hungarian*, *colic*, and *labor*, which are the data sets of higher values of sparsity.

TABLE II: Results in Accuracy for MLP

data set	algorithm	Acc&(2)	F1&(2)	Acc&(3)	F1&(3)
cleveland	B	0.81	0.81	0.81	0.81
	WB	0.802	0.802	0.802	0.802
	WAB	0.802	0.801	0.802	0.802
breast-w	B	0.954	0.954	0.954	0.954
	WB	0.954	0.954	0.954	0.954
	WAB	0.954	0.954	0.954	0.954
breast-tumor	B	0.555	0.555	0.555	0.555
	WB	0.563	0.563	0.563	0.563
	WAB	0.567	0.566	0.568	0.568
credit-a	B	0.848	0.848	0.848	0.848
	WB	0.847	0.847	0.847	0.847
	WAB	0.848	0.848	0.849	0.849
credit	B	0.856	0.856	0.856	0.856
	WB	0.856	0.856	0.856	0.856
	WAB	0.856	0.856	0.856	0.856
biomed	B	0.867	0.867	0.867	0.867
	WB	0.864	0.864	0.864	0.864
	WAB	0.863	0.864	0.862	0.863
audiology-n	B	0.916	0.916	0.916	0.916
	WB	0.916	0.916	0.916	0.916
	WAB	0.914	0.913	0.914	0.913
runshoes	B	0.708	0.708	0.708	0.708
	WB	0.698	0.698	0.698	0.698
	WAB	0.7	0.695	0.7	0.702
echo-months	B	0.672	0.672	0.672	0.672
	WB	0.668	0.668	0.668	0.668
	WAB	0.673	0.675	0.673	0.672
vote	B	0.95	0.95	0.95	0.95
	WB	0.951	0.951	0.951	0.951
	WAB	0.951	0.951	0.951	0.951
hepatitis	B	0.808	0.808	0.808	0.808
	WB	0.812	0.812	0.812	0.812
	WAB	0.821	0.821	0.823	0.821
schizo	B	0.562	0.562	0.562	0.562
	WB	0.565	0.565	0.565	0.565
	WAB	0.571	0.571	0.569	0.569
hungarian	B	0.811	0.811	0.811	0.811
	WB	0.811	0.811	0.811	0.811
	WAB	0.818	0.818	0.819	0.819
colic	B	0.808	0.808	0.808	0.808
	WB	0.806	0.806	0.806	0.806
	WAB	0.81	0.81	0.81	0.811
labor	B	0.851	0.851	0.851	0.851
	WB	0.851	0.851	0.851	0.851
	WAB	0.865	0.865	0.865	0.865

We report results in F1-measure for MLP in Table III, which is in the same format as Table II.

Below is the summary of the results reported in Table III: For all the 4 cases described earlier, WB outperforms B on 7 data sets, and WAB outperforms both WB and B on 7 data sets. WAB outperforms both WB and B on the data sets *hepatitis*, *schizo*, *hungarian*, *colic*, and *labor*, which are the data sets of higher values of sparsity (i.e. data sets containing larger numbers of missing values).

On the data sets, *breast-tumor*, *runshoes*, and *echo-months*, all 3 algorithms give values of F1-measure lower than 0.5, indicating that the results are not practically meaningful. If we exclude results obtained on the three data sets, we observe that, for all the 4 cases, WB outperforms B on 6 data sets, and WAB outperforms both WB and B on 6 data sets.

TABLE III: Results in F1-Measure for MLP

data set	algorithm	Acc&(2)	F1&(2)	Acc&(3)	F1&(3)
cleveland	B	0.788	0.788	0.788	0.788
	WB	0.784	0.784	0.784	0.784
	WAB	0.784	0.784	0.785	0.784
breast-w	B	0.932	0.932	0.932	0.932
	WB	0.933	0.933	0.933	0.933
	WAB	0.933	0.933	0.933	0.933
breast-tumor	B	0.439	0.439	0.439	0.439
	WB	0.422	0.422	0.422	0.422
	WAB	0.426	0.428	0.429	0.43
credit-a	B	0.831	0.831	0.831	0.831
	WB	0.828	0.828	0.828	0.828
	WAB	0.828	0.828	0.829	0.829
credit	B	0.839	0.839	0.839	0.839
	WB	0.836	0.836	0.836	0.836
	WAB	0.837	0.837	0.836	0.836
biomed	B	0.814	0.814	0.814	0.814
	WB	0.806	0.806	0.806	0.806
	WAB	0.805	0.806	0.803	0.805
audiology-n	B	0.578	0.578	0.578	0.578
	WB	0.586	0.586	0.586	0.586
	WAB	0.589	0.588	0.589	0.588
runshoes	B	0.427	0.427	0.427	0.427
	WB	0.431	0.431	0.431	0.431
	WAB	0.441	0.434	0.444	0.445
echo-months	B	0.445	0.445	0.445	0.445
	WB	0.429	0.429	0.429	0.429
	WAB	0.439	0.442	0.436	0.441
vote	B	0.936	0.936	0.936	0.936
	WB	0.938	0.938	0.938	0.938
	WAB	0.937	0.937	0.937	0.937
hepatitis	B	0.549	0.549	0.549	0.549
	WB	0.548	0.548	0.548	0.548
	WAB	0.566	0.567	0.568	0.565
schizo	B	0.501	0.501	0.501	0.501
	WB	0.529	0.529	0.529	0.529
	WAB	0.534	0.533	0.531	0.531
hungarian	B	0.719	0.719	0.719	0.719
	WB	0.726	0.726	0.726	0.726
	WAB	0.739	0.739	0.741	0.74
colic	B	0.726	0.726	0.726	0.726
	WB	0.729	0.729	0.729	0.729
	WAB	0.736	0.737	0.737	0.737
labor	B	0.794	0.794	0.794	0.794
	WB	0.794	0.794	0.794	0.794
	WAB	0.814	0.814	0.814	0.814

For SMO, we report results in accuracy and F1-measure in Tables IV and V, respectively. They are also in the same format as Table II.

The results reported in Table IV are summarized as follows: For all the 4 cases described earlier, WB aceieves better performance than B does on 4 data sets, while compared with both WB and B, WAB achieves better performance on at least 9 out of 15 data sets. For Case 2, WAB is better than both WB and B on 11 out of data sets. WAB outperforms both WB and B on the data sets *schizo*, *hungarian*, and *colic*, abd these are data sets of higher values of sparsity, which usually cause problems to SMO (and MLP).

The results reported in Table V are summarized as follows: For all the 4 cases described earlier in this subsection, WB is better than B on 6 data sets, while compared with both WB and B, WAB is better on at least 5 data sets. For Cases 2 and 4, WAB is outperforms both WB and B on 8 out of 15 data sets.

On the data sets *schizo*, *hungarian*, and *colic*, which are those of higher values of sparsity, WAB achieve better performance than do WB and B. Combining the results reported here with those reported in Table III, we conclude that WAB is a better option when MLP or SMO is employed in the situation where the amount of the available data records are not as many as expected and the quality of them are not as high as expected. We often find us in such a situation when dealing with medicine data or survey data.

TABLE IV: RESULTS IN ACCURACY FOR SMO

data set	algorithm	Acc&(2)	F1&(2)	Acc&(3)	F1&(3)
cleveland	B	0.826	0.826	0.826	0.826
	WB	0.825	0.825	0.825	0.825
	WAB	0.825	0.827	0.825	0.827
breast-w	B	0.965	0.965	0.965	0.965
	WB	0.966	0.966	0.966	0.966
	WAB	0.966	0.966	0.966	0.966
breast-tumor	B	0.587	0.587	0.587	0.587
	WB	0.587	0.587	0.587	0.587
	WAB	0.59	0.591	0.589	0.591
credit-a	B	0.851	0.851	0.851	0.851
	WB	0.851	0.851	0.851	0.851
	WAB	0.852	0.853	0.852	0.852
credit	B	0.859	0.859	0.859	0.859
	WB	0.859	0.859	0.859	0.859
	WAB	0.86	0.86	0.859	0.859
biomed	B	0.874	0.874	0.874	0.874
	WB	0.874	0.874	0.874	0.874
	WAB	0.874	0.877	0.875	0.877
audiology-n	B	0.938	0.938	0.938	0.938
	WB	0.937	0.937	0.937	0.937
	WAB	0.937	0.936	0.936	0.936
runshoes	B	0.755	0.755	0.755	0.755
	WB	0.76	0.76	0.76	0.76
	WAB	0.76	0.765	0.75	0.757
echo-months	B	0.678	0.678	0.678	0.678
	WB	0.677	0.677	0.677	0.677
	WAB	0.683	0.69	0.682	0.689
vote	B	0.953	0.953	0.953	0.953
	WB	0.952	0.952	0.952	0.952
	WAB	0.957	0.956	0.957	0.957
hepatitis	B	0.835	0.835	0.835	0.835
	WB	0.835	0.835	0.835	0.835
	WAB	0.838	0.835	0.836	0.835
schizo	B	0.58	0.58	0.58	0.58
	WB	0.581	0.581	0.581	0.581
	WAB	0.584	0.588	0.585	0.586
hungarian	B	0.82	0.82	0.82	0.82
	WB	0.822	0.822	0.822	0.822
	WAB	0.829	0.828	0.829	0.828
colic	B	0.802	0.802	0.802	0.802
	WB	0.801	0.801	0.801	0.801
	WAB	0.808	0.808	0.809	0.808
labor	B	0.863	0.863	0.863	0.863
	WB	0.858	0.858	0.858	0.858
	WAB	0.86	0.861	0.863	0.863

Moreover, on the data sets, *breast-tumor*, *runshoes*, and *echo-months*, all 3 algorithms give values of F1-measure lower than 0.5, indicating that the results are not practically meaningful. If we exclude results obtained on the three data sets, we observe that, for all the 4 cases, WB is better than B on 5 data sets. For Cases 1 and 3, WAB is better than the others on 5 data sets; for Cases 2 and 4, WAB is better than the others on 7 data sets.

Although the experimental results do not indicate the best among the 4 cases, they do give the following suggestions: If we are going to use WAB with MLP, we can use F1-measure for weight assignment and (2) for weight adjustment. If we are going to use WAB with SMO, we can use F1-measure for weight assignment and (2) or (3) for weight adjustment.

For the data set *labor*, having a small number of instances but containing a relatively large number of missing values, the best performance measured by either accuracy or F1-measure is given by using WAB with MLP. This reveals the potential of weighted-adjusted bagging for dealing with small data sets of large values of sparsity.

TABLE V: RESULTS IN F1-MEASURE FOR SMO

data set	algorithm	Acc&(2)	F1&(2)	Acc&(3)	F1&(3)
cleveland	B	0.804	0.804	0.804	0.804
	WB	0.805	0.805	0.805	0.805
	WAB	0.805	0.807	0.805	0.807
breast-w	B	0.95	0.95	0.95	0.95
	WB	0.95	0.95	0.95	0.95
	WAB	0.951	0.95	0.951	0.95
breast-tumor	B	0.407	0.407	0.407	0.407
	WB	0.383	0.383	0.383	0.383
	WAB	0.391	0.42	0.389	0.42
credit-a	B	0.844	0.844	0.844	0.844
	WB	0.842	0.842	0.842	0.842
	WAB	0.843	0.844	0.843	0.843
credit	B	0.849	0.849	0.849	0.849
	WB	0.846	0.846	0.846	0.846
	WAB	0.849	0.848	0.848	0.847
biomed	B	0.813	0.813	0.813	0.813
	WB	0.81	0.81	0.81	0.81
	WAB	0.81	0.813	0.811	0.813
audiology-n	B	0.654	0.654	0.654	0.654
	WB	0.664	0.664	0.664	0.664
	WAB	0.662	0.659	0.658	0.659
runshoes	B	0.454	0.454	0.454	0.454
	WB	0.467	0.467	0.467	0.467
	WAB	0.467	0.488	0.467	0.501
echo-months	B	0.327	0.327	0.327	0.327
	WB	0.317	0.317	0.317	0.317
	WAB	0.314	0.392	0.314	0.39
vote	B	0.939	0.939	0.939	0.939
	WB	0.939	0.939	0.939	0.939
	WAB	0.944	0.944	0.945	0.945
hepatitis	B	0.572	0.572	0.572	0.572
	WB	0.556	0.556	0.556	0.556
	WAB	0.569	0.566	0.562	0.565
schizo	B	0.535	0.535	0.535	0.535
	WB	0.547	0.547	0.547	0.547
	WAB	0.55	0.559	0.552	0.556
Hungarian	B	0.729	0.729	0.729	0.729
	WB	0.734	0.734	0.734	0.734
	WAB	0.745	0.745	0.745	0.744
Colic	B	0.722	0.722	0.722	0.722
	WB	0.727	0.727	0.727	0.727
	WAB	0.738	0.737	0.738	0.737
Labor	B	0.801	0.801	0.801	0.801
	WB	0.783	0.783	0.783	0.783
	WAB	0.79	0.79	0.793	0.792

IV. DISCUSSIONS

In this section, we discuss some other papers that are related to or can get benefit from this paper.

Despite its simplicity, the majority voting mechanism adopted by bagging has been found effective in practice. One can assign more weights to classifiers showing better classification performance and use weighted voting, and one can find situations where weighted voting is better than the simple majority voting [13]. The weight of a classifier depends on its classification performance [14]. However, most people overlook the fact that the classification performance of a classifier depends on its underlying classification algorithm and the data set used in its training. When a classifier shows good classification performance, we need to identify the source of its effectiveness; when a classifier shows poor classification performance, we need to identify the source of its ineffectiveness – is it from the underlying algorithm or the training set? Both the idea of weight adjustment proposed by Kim et al. [9] and the one presented in this paper are proposed to reduce (as much as possible) the effects from the differences in the training *contexts* of classifiers. This paper is unique in that it explicitly takes the sparsity of the training set into account.

Ensembles of neural networks have been studied by many researchers. For example, Optiz and Shavlik studied how to generate an ensemble composed of more accurate neural networks [15]; Optiz and Maclin empirically evaluated algorithms (including bagging) to create ensembles of neural networks [16]; Zhou, Wu, and Tang studied the impact of the size of an ensemble of neural networks [17]; Chen and Yu proposed to use particle swarm optimization to determine optimal weights for the classifiers in an ensemble of neural networks created by bagging and then use weighted averaging [18].

Ensembles of support vector machines (SVMs) have also been studied by many researchers. Below are examples: Kim et al. studied bagging with SVM and concluded that it is better than single SVM in terms of classification accuracy [19], [20]; Wang and Lin proposed an extension of bagging that trains SVM classifiers specifically for classes [20]; Wang et al. empirically studied the performance of ensembles of SVMs and concluded that "*although SVM ensembles are not always better than a single SVM, the SVM bagged ensemble performs as well or better than other methods with a relatively higher generality* " [22].

In addition, ensembles of neural networks have been applied in applications as diverse as, for example, ozone concentration prediction [23], image classification [24], [25], cancer cell identification [26], financial decision making [27], credit risk analysis [28] and credit scoring [29], computer virus detection [30], precision fertilization modeling [31], and pulmonary nodule classification [32]. Ensembles of SVMs have also been applied in various applications, such as face image classification [33], human splice site identification [34], fault diagnosis [35], [36], malware detection [37], document summarization [38], protein structural class prediction [39], and visual object recognition [40]. These papers indicate the potential applications in which this paper can be applied.

V. Conclusions and Future Work

One of the possible advantages provided by ensembles is the improvement of the classification performance for low-quality data. Bagging is an algorithm for ensemble creation and has been applied in various applications. In many real-world applications, the obtained data sets contain missing values while the classification algorithm considered the best for the applications is sensitive to missing values.

We presented an approach to improve the performance of bagging when it is used with the multilayer perceptron and sequential minimal optimization algorithms and the given data set for training contains missing values. The presented approach is named weighted-adjusted bagging. For each of the classifiers in a created ensemble, weight-adjusted bagging assigns a weight to it according to its classification performance and adjusts its weight according to the sparsity of its training set. For weight assignment, we use accuracy or F1-measure; for weight adjustment, we use two functions each of which assigns more weights to classifiers whose training sets contain more missing values. The experimental results reveal that weight-adjusted bagging can outperform both bagging and weighted bagging, especially on data sets that have small numbers of instances but contain relatively large numbers of missing values.

The extensions of bagging that use weighted voting or weight adjustment are not new, but what is presented in this paper is new in the sense that it incorporates the information about the characteristics of the training sets into the voting process. In other words, it considers the *context* in which the classifiers in an ensemble were trained, and therefore it serves as the foundation for research on context-aware aggregation for ensembles.

The future work of this paper includes 1) using measures other than accuracy and F1-measure in weight assignment, 2) using factors other than sparsity of the training set in weight adjustment, 3) using functions different from those proposed in this paper in weight adjustment, and 4) using weight adjustment in ensemble algorithms other than bagging.

Acknowledgment

The author would like to thank the National Science Council of Taiwan for the support under grant number NSC 101-2221-E-004-011. The author would also like to thank anonymous reviewers for their precious time.

References

[1] P. Melville, N. Shah, L. Mihalkova, and R. J. Mooney. "xperiments on Ensembles with Missing and Noisy Data," in *Proc. Multiple Classifier Systems*, 2004, pp. 293-302.

[2] L. Rokach, "Ensemble-based classifiers," *Artificial Intelligence Review*, vol. 33, no. 1-2, pp. 1-39, 2010.

[3] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123-140, 1996.

[4] P. Buhlmann and B. Yu, "Analyzing Bagging," *The Annals of Statistics*, vol. 30, no. 4, pp. 927-961, 2002.

[5] I. H. Witten and E. Frank, *Data Mining – Practical Machine Learning Tools and Techniques*, 2nd ed., Elsevier, 2005, sec. 6.3.

[6] J. Platt, "Fast Training of Support Vector Machines using Sequential Minimal Optimization," in *Advances in Kernel Methods - Support Vector Learning*, B. Schoelkopf, C. Burges, and A. Smola, Ed., 1998.

[7] S. S. Keerthi, S. K. Shevade, C. Bhattacharyya, and K. R. K. Murthy, "Improvements to Platt's SMO Algorithm for SVM Classifier Design," *Neural Computation*, vol. 13, no. 3, pp. 637-649, 2001.

[8] L. I. Kuncheva and J. J. Rodriguez," A weighted voting framework for classifiers ensembles," *Knowledge and Information Systems*, pp. 1-17, 2012.

[9] H. Kim, H. Kim, H. Moon, and H. Ahn," A weight-adjusted voting algorithm for ensembles of classifiers," *Journal of the Korean Statistical Society*, vol. 40, pp. 437-449, 2011.

[10] K. Bache and M. Lichman. (2013). UCI Machine Learning Repository. [Online]. Available: http://archive.ics.uci.edu/ml.

[11] P. Vlachos. (2005). StatLib datasets archive. [Online]. Available: http://lib.stat.cmu.edu/datasets.

[12] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA Data Mining Software: An Update," *ACM SIGKDD Explorations*, vol. 11, no. 1, pp. 10-18, 2009.

[13] N. C. Oza and K. Tumer, "Classifier ensembles: Select real-world applications," *Information Fusion*, vol. 9, no. 1, pp. 4-20, 2008.

[14] G. Tsoumakas, I. Partalas, and I. Vlahavas,. "A taxonomy and short review of ensemble selection," *Workshop on Supervised and Unsupervised Ensemble Methods and Their Applications*, 2008.

[15] D. W. Opitz, and J. W. Shavlik, "Generating accurate and diverse members of a neural-network ensemble," *Advances in neural information processing systems*, pp. 535-541, 1996.

[16] D. W. Opitz and R. F. Maclin, "An empirical evaluation of bagging and boosting for artificial neural networks," *International Conference on Neural Networks*, 1997.

[17] Z. H. Zhou, J. Wu, and W. Tang, "Ensembling neural networks: many could be better than all," *Artificial intelligence*, vol. 137, no. 1, pp. 239-263, 2002.

[18] R. Chen and J. Yu, "An improved bagging neural network ensemble algorithm and its application," *International Conference on Natural Computation*, 2007.

[19] H.-C. Kim, S. Pang, H.-M. Je, D. Kim, and S.-Y. Bang. "Support vector machine ensemble with bagging," *Pattern recognition with support vector machines*, Berlin, Heidelberg: Springer, pp. 397-408, 2002.

[20] H. C. Kim, S. Pang, H.-M. Je, D. Kim, and S.-Y. Bang, "Constructing support vector machine ensemble," *Pattern Recognition*, vol. 36, no. 12, pp. 2757-2767, 2003.

[21] Y. Wang and C. D. Lin, "Learning by Bagging and Adaboost based on Support Vector Machine," *International Conference on Industrial Informatics*, 2007.

[22] S. J. Wang, A. Mathew, Y. Chen, L. F. Xi, L. Ma, and J. Lee, "Empirical analysis of support vector machine ensemble classifiers," *Expert Systems with Applications*, vol. 36, no. 3, pp. 6466-6476, 2009.

[23] A. J. Cannon and E. R. Lord, "Forecasting summertime surface-level ozone concentrations in the Lower Fraser Valley of British Columbia: An ensemble neural network approach," *Journal of the Air & Waste Management Association*, vol. 50, no. 3, pp. 322-339, 2000.

[24] G. Giacinto and F. Roli, "Design of effective neural network ensembles for image classification purposes," *Image and Vision Computing*, vol. 19, no. 9, pp. 699-707, 2001.

[25] M. Han, X. Zhu, and W. Yao, "Remote sensing image classification based on neural network ensemble algorithm," *Neurocomputing*, vol. 78, no. 1, pp. 133-138, 2012.

[26] Z. H. Zhou, Y. Jiang, Y. B. Yang, and S. F. Chen, "Lung cancer cell identification based on artificial neural network ensembles," *Artificial Intelligence in Medicine*, vol. 24, no. 1, pp. 25-36, 2002.

[27] D. West, S. Dellana, and J. Qian, "Neural network ensemble strategies for financial decision applications," *Computers & Operations Research*, vol. 32, no. 10, pp. 2543-2559, 2005.

[28] K. K. Lai, L. Yu, S. Wang, and L. Zhou, "Credit risk analysis using a reliability-based neural network ensemble model," *Artificial Neural Networks–ICANN*, Berlin, Heidelberg: Springer, pp. 682-690, 2006.

[29] C.-F. Tsai and J.-W. Wu, "Using neural network ensembles for bankruptcy prediction and credit scoring," *Expert Systems with Application*s, vol. 34, no. 4, pp. 2639-2649, 2008.

[30] G. Liu, F. Hu, and W. Chen, "A neural network ensemble based method for detecting computer virus," *International Conference on Computer, Mechatronics, Control and Electronic Engineering*, pp. 391-393, 2010.

[31] H. Yu, D. Liu, G. Chen, B. Wan, S. Wang, and B. Yang, "A neural network ensemble method for precision fertilization modeling," *Mathematical and Computer Modelling*, vol. 51, no. 11, pp. 1375-1382, 2010.

[32] H. Chen, W. Wu, H. Xia, J. Du, M. Yang, and B. Ma, "Classification of pulmonary nodules using neural network ensemble," *Advances in Neural Networks–ISNN*, Berlin, Heidelberg: Springer, pp. 460-466, 2011.

[33] S. Pang, D. Kim, and S. Y. Bang, "Membership authentication in the dynamic group by face classification using SVM ensemble," *Pattern Recognition Letters*, vol. 24, no. 1, pp. 215-225, 2003.

[34] A. C. Lorena and A. C. de Carvalho, "Human splice site identification with multiclass support vector machines and bagging," *Artificial Neural Networks and Neural Information Processing–ICANN/ICONIP*, Berlin, Heidelberg: Springer, pp. 234-241, 2003.

[35] Y. Li, Y. Z. Cal, R. P. Yin, and X. M. Xu, "Fault diagnosis based on support vector machine ensemble," in *Proc. International Conference on Machine Learning and Cybernetics*, 2005, pp. 3309-3314.

[36] T. Jingyuan, S. Yibing, and Z. Wei, "Analog circuit fault diagnosis based on support vector machine ensemble," *Chinese Journal of Scientific Instrument*, vol. 29, no. 6, pp. 1216-1220, 2008.

[37] Y. Ye, L. Chen, D. Wang, T. Li, Q. Jiang, and M. Zhao, "SBMDS: an interpretable string based malware detection system using SVM ensemble with bagging," *Journal in Computer Virology*, vol. 5, no. 4, pp. 283-293, 2009.

[38] Y. Chali, S. A. Hasan, and S. R. Joty, "A SVM-Based Ensemble Approach to Multi-Document Summarization," *Advances in Artificial Intelligence*, Berlin, Heidelberg: Springer, pp. 199-202, 2009.

[39] J. Wu, M. L. Li, L. Z. Yu, and C. Wang, "An ensemble classifier of support vector machines used to predict protein structural classes by fusing auto covariance and pseudo-amino acid composition," *The Protein Journal*, vol. 29, no. 1, pp. 62-67, 2010.

[40] Z. Xie, Y. Xu, and Q. Hu, "Visual Object Recognition with Bagging of One Class Support Vector Machines, "*International Conference on Innovations in Bio-inspired Computing and Applications*, pp. 99-102, 2011.

Kuo-Wei Hsu earned his Ph.D. from the Department of Computer Science and Engineering, the University of Minnesota, Minneapolis, Minnesota, USA, 2011; he obtained his M.S. degree in Computer Science and Information Engineering from the National Taiwan University, Taipei, Taiwan, 2001; he obtained his B.S. degree in Electrical Engineering from the National Chung Hsing University, Taichung, Taiwan, 1999.

He is currently an assistant professor in the Department of Computer Science at the National Chengchi University, Taipei, Taiwan. His current research interests include data mining, database systems, and software engineering. Before he entered the Ph.D. program, he worked as an Information Engineer in the National Taiwan University Hospital, Taipei, Taiwan. Dr. Hsu is currently a member of ACM and IEEE.