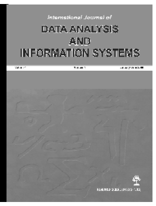
Research Science Press

International Journal of Data Analysis and Information Systems
VOLUME 8 • NUMBER 1 • JUNE 2016

journal homepage : serialsjournals.com

ISSN : 2229-5887



SDP-based Ensemble Pruning Algorithm with Improved Re-sizing Step

Adeseye Agbabiaka[1], **Chiwoo Park**[2]
Department of Industrial and Manufacturing Engineering, Florida State University, Tallahassee FL 32310

Authors emails
aoa11b@my.fsu.edu
cpark5@fsu.edu

ABSTRACT

Ensemble learning is a machine learning technique that takes the fusion or ensemble of many homogeneous or heterogeneous machine learning models to achieve better predictive performance over each individual of the learning models. Considerations of computational cost, memory usage and efficiency discourages an unnecessarily large ensemble. Selecting a subset of the available learning models is called ensemble pruning. In this paper, we present an improved algorithm for the existing semi-definite programming (SDP) pruning. The original SDP pruning first solves a semi-definite programming formulation for the preliminary selection of the promising learning models and then applies the re-sizing step for reducing the preliminary selection to a desired size. The re-sizing step is performed in a greedy fashion toward the increasing direction of the diversity among the selected learning models. The ensemble diversity measure is inversely proportional to the number of the classification errors commonly made by each pair of the selected learning models, which has not been shown to be among the best measures for the overall classification accuracy of the ensemble. Instead of minimizing the total pairwise prediction errors or equivalently maximizing the ensemble diversity, we propose to minimize the overall prediction error of the whole ensemble to improve the accuracy of the SDP ensemble pruning algorithm. Our numerical experiments showed the significant performance improvement of the SDP pruning algorithm with the revision, which makes it outperform the state-of-the-art ensemble pruning algorithms as well as the unpruned full ensemble model for many classification benchmark datasets.

***Keywords**: Ensemble Pruning, Classification, Ensemble Accuracy, Ensemble Diversity*

1. INTRODUCTION

An ensemble is a collection of individually trained learners that are combined together to improve the prediction accuracy. The individual learners known as base learners are usually combined using different methods such as bagging [1] and boosting [2]. The general idea of taking an ensemble of multiple base learners have been shown effective in improving the prediction accuracy in that the ensemble is often more accurate than the best learner in the ensemble [3].

Ensemble methods typically work in two phases; the first phase is to generate individual base learners, and the second phase is to combine all the base learners to make a prediction. The combination of all the base learners is called a full ensemble. When the number of the base learners is huge or when the ensemble should deal with large datasets, the expensive computation cost of evaluating the full ensemble can become an serious issue especially when it is applied for stream mining [4, 5]. In this case, using a subset of the full ensemble is more computationally favorable. In addition, using a subset of the full ensemble has been demonstrated to provide better prediction accuracy over using the full ensemble for some cases [6, 7].

The low computation complexity and the improved prediction accuracy of using a small size ensemble have all necessitated the need for selecting a representative subset of the en- tire collection of base learners. The problem of selecting the representative subset is known as ensemble pruning, ensemble selection, selective ensembling and ensemble thinning. In this paper, we are more concerned about the ensemble pruning for a general classification problem. We use the term base classifier instead of using a base learner.

Several pruning algorithms have been developed over the past few decades. They can be grouped into three main categories based on the similarity in their approaches, ordering based pruning, clustering based pruning, and optimization based pruning. The ordering based methods first rank base classifiers using a certain rank criterion and greedily selects a subset of the better ranked base classifiers. Kappa pruning [8] and orientation pruning [9] fall into this category. The clustering based methods first group all base classifiers into several clusters by their similarity and select a subset of representatives from each cluster to form an ensemble. The hierarchical agglomerative clustering [10] and the k-means clustering, [11] are the popular choices for the clustering task. Last the optimization based pruning formulates the ensemble pruning as an optimization-based subset selection problem which includes the GASEN algorithm [7], the probabilistic pruning techniques [12, 13] and the semi-definite programming (SDP) pruning [14].

The SDP pruning algorithm [14] is based on a global optimization approach unlike most of the other methods which are mostly based on a greedy search and other heuristics that are led to local optimality. Let us brief the original SDP pruning first. The SDP pruning algorithm formulates an ensemble pruning problem as a binary optimization problem of selecting a subset of base classifiers and relaxes the binary optimization problem to a semi-definite programming problem, which can be solved in a polynomial time. Due to the relaxation, the resulting subset does not have a desired size, so an additional re-sizing step should be applied to further reduce the subset to a desired size. The re-sizing step is performed in a greedy fashion that reduces the subset toward the increasing direction of a certain criterion. A part of the criterion used is the degree of the diversity among the selected base classifiers. It has been believed that an ideal ensemble is one having a large diversity of accurate base learners [15].

The diversity criterion used in the re-sizing step of the SDP pruning actually counts the total number of the classification errors made commonly by each pair of the base classifiers in the selected ensemble, which has not been shown to be among the best measures for the overall classification accuracy of the ensemble but is closer to a total pairwise classification error; see our discussion in Section 3. To improve the SDP pruning algorithm, the criterion can be replaced with a better ensemble selection criterion that quantifies the overall classification error of the selected ensemble. In this paper, we propose to use an ensemble accuracy measure as a selection criterion in the resizing step of the SDP pruning algorithm. We considered many different accuracy and diversity measures of an ensemble for the selection criterion. Our numerical study showed that using a weighted voting accuracy gave the better prediction results for many publicly available benchmark datasets; see Section 5.1. Hence, we propose to use a weighted voting accuracy in the re-sizing step. The proposed revision outperforms the state-of-the-art ensemble pruning methods as well as the full ensemble.

The paper is organized as follows. Section 2 briefly introduces the original SDP pruning method, and Section 3 discusses the potential issue in the re-sizing step of the SDP pruning method. Section 4 presents our revised SDP pruning method and Section 5 presents the numerical comparison of our method with the current state-of-the-art. Last, Section 6 concludes the paper.

2. ENSEMBLE PRUNING AS BINARY QUADRATIC PROBLEMS

Given a set of L base classifiers, an ensemble pruning problem can be regarded as a subset selection problem that selects the best subset of size M. It has been believed that an ideal ensemble is one having a large diversity of accurate base learners [15]. Many literatures used different criteria quantifying the overall classification accuracy of an ensemble [8, 16, 17] and the diversity of an ensemble [18, 19, 20].

The SDP pruning algorithm [14] proposed to use a new subset selection criterion, which are based on a combination of the accuracy of individual base classifiers and the diversity of an ensemble. The accuracy part is measured by the total number of the classification errors made by the individual classifiers in an ensemble, while the diversity part is measured by adding the pairwise statistical correlation in the output vectors of classifiers i and j over all possible pairs of the classifiers in an ensemble. When the result of classifier i on N training data is recorded by a binary vector v_i with $(v_i)_k = 1$ if the classifier is incorrect on the kth training case, the selection criterion is to minimize the individual misclassification error plus the pairwise correlation,

$$\sum_{i=1}^{L} G_{ii}x_i + \sum_{i,j=1;i\neq j}^{L} G_{ij}x_i x_j, \qquad (1)$$

where $x_i = 1$ implies classifier i is chosen, $G_{ii} = 1^T v_i/N$ quantifies the misclassification rate of classifier i, and G_{ij} quantifies the normalized pairwise correlation of v_i and v_j,

$$G_{ij} = \frac{v_i^T v_j}{2}\left(\frac{1}{v_i^T v_i} + \frac{1}{v_j^T v_j}\right).$$

The minimization problem is formulated as a 0-1 binary quadratic problem,

$$\begin{aligned} &\min\ x^T G x \\ &\text{s.t.}\quad 1^T x = M \\ &x \in \{0, 1\}^L, \end{aligned} \tag{3}$$

where x is a column vector of binary variables x_i, and the matrix G is the matrix of G_{ij}'s. The approach was originally applied for classification problems and was later extended for regression problems by Hernandez-Lobato *et al.* [21]. Solving the binary quadratic problem is a NP-hard problem. Zhang et al. [14] solved its semi-definite programming (SDP) relaxation. The solution of the relaxation problem is not binary-valued, so a randomization-based rounding method should be applied to convert the non-binary solution into a binary-valued one [22]. However, by the nature of the rounding method, the size of the finally selected subsets (or equivalently the number of non-zero elements in the solution x) is not equal to a desired size M. Thus, when the desired size was not obtained, a re-sizing step should be applied to reduce the number of the non-zero elements in x to the desired size. The procedure for the SDP ensemble pruning algorithm is shown in Algorithm 1. The re-sizing step is performed in a greedy fashion. However, the greedy selection part is applied after the global optimization (3) is solved for selecting promising base classifiers, so the algorithm generally obtains a less greedy solution than pure greedy-based algorithms do.

Algorithm 1. SDP Pruning Algorithm

Input: matrix G, a desired size of ensemble M

Output: Pruned ensemble

Step 1. SDP: Solve the SDP relaxation of problem (3) to obtain the solution x.

Step 2. Randomized rounding:

Use a randomization approximation scheme [22] to round-off x to a binary-valued vector; If the ith element of the rounded vector is 1, it implies that the ith learner is selected in the ensemble.

Step 3. Re-sizing: The ensemble obtained by Step 2 is re-sized to M through a greedy ensemble selection algorithm that starts with the ensemble obtained by Step 2 and iteratively adds or deletes the base learners one-by-one so that the resulting ensemble reduces the objective value of problem (3) most.

3. DISCUSSION ON THE DIVERSITY MEASURE CRITERION IN THE SDP PRUNING ALGORITHM

We notice that the pairwise correlation term $v_i^T v_j$ in equation (2) is nothing but the number of the common classification errors made by both of classifiers i and j for the training dataset. Please note that is an N-dimensional binary vector with $(v_i)_k = 1$ only if classifier i makes a wrong classification on the kth training case. The term $v_i^T v_j$ counts the number of common wrong classification cases. In the other words, the diversity term G_{ij} of the SDP pruning algorithm essentially quantifies the total pairwise classification error which implies the total number of the classification errors made by each pair of the selected classifiers. Although the pairwise classification error might be related to the overall classification error of the ensemble of the selected classifiers, the correlation is not very strong. As a simple example, we can imagine an ensemble of five binary classifiers combined by a majority voting, and suppose that only the first three classifiers are correct for a training case and the other classifiers are incorrect. In this example, the total pairwise classification error is one since classifiers 4 and 5 make a common error. However, the ensemble of the five classifiers would not make any classification error on the training case because the majority of the individual classifiers are correct, so the overall classification error of the ensemble should be zero.

Solving the SDP pruning with the guidance of the diversity term may result in an ensemble of improved classification accuracy up to certain degree but would not lead to an ensemble of highest classification accuracy, which is the ultimate goal of taking an ensemble. This motivated us the replacement of the subset selection criterion in the SDP pruning algorithm with a different criterion. We had exhaustively search for the selection criteria used for ensemble pruning in literature. Some are based on the classification accuracy of an ensemble [8, 16, 17], and the others are based on diversity measures of an ensemble such as margin distance and complementariness [18], concurrency [19], focused selection diversity [20], the disagreement measure [23], the double fault measure[24] and the weighted error count and correct value [25].

We realized that most of the diversity measures applied in the literature quantify the pairwise classification errors like the diversity term G_{ij} of the SDP pruning algorithm. For example, the disagreement measure [23] quantifies: for every pair of classifier i and j,

$$\textit{Disagreement Measure} = \frac{n^{10} + n^{01}}{N}$$

$$= \frac{N - v_i^T v_j + N - v_i^T v_j}{N}$$

$$= 2 - \frac{2 v_i^T v_j}{N},$$

where n^{10} = the number of instances where classifier i is correct and classifier j is wrong, and n^{01} = the number of instances where classifier i is wrong and classifier j is correct. The double fault measure quantifies

$$\textit{Double Fault} = 1 - \frac{v_i^T v_j}{N}.$$

Both of the measures are determined by the quantity $v_i^T v_j$. Therefore, applying such diûerent diversity measures in the SDP ensemble pruning algorithm will not improve the performance of the algorithm significantly, which was also shown by our numerical study in Section 5. We propose to use a classification accuracy-based criterion instead.

4. REVISING THE RE-SIZING STEP OF THE SDP PRUNING WITH A BETTER ENSEMBLE ACCURACY MEASURE

We propose to replace the ensemble selection criterion of the SDP pruning algorithm with an overall ensemble accuracy criterion, a weighted voting accuracy or majority voting accuracy of an ensemble, which is defined as follows. Suppose that we have L base classifiers, where the m th classifier takes an input vector $\boldsymbol{u}$ and returns a binary vector $C_m(\boldsymbol{u})$; when a K-class classification is dealt, the output vector follows an one-of-K representation, i.e. it is a K-dimensional vector in which one of the elements is equal to one and all other elements are 0. Let S denote a subset of $\{1,..., L\}$ that represents an ensemble of the subset of the L base classifiers. Given a training data $D = \{(u_i, y_i); y_i \in \{0, 1\}, i = 1,..., N\}$, the weighted voting accuracy of S is defined by

$$WVA(S;D) = 1 - \frac{1}{N}\sum_{i=1}^{N}\left(y_i \neq \max el\left(\sum_{m\in S} w_m C_m(u_i)\right)\right),$$

and the majority voting accuracy is defined by

$$MVA(S;D) = 1 - \frac{1}{N}\sum_{i=1}^{N} I\left(y_i \neq \max el\left(\sum_{m\in S} C_m(u_i)\right)\right),$$

where maxel(s) is the index of the element in an input vector s that has the maximum value among all the elements in s, and w_m is the weight for the mth classifier that is typically proportional to the classification accuracy of the mth classifier.

We acknowledge that if one replaces the objective function of the SDP pruning algorithm in optimization (3) with either $WVA(S; D)$ or $MVA(S; D)$, the optimization problem becomes a combinatorial problem that cannot be relaxed to a semi-definite programming form because either a weighted voting accuracy or a majority voting accuracy is neither a linear function or a quadratic function of S. Maximizing $WVA(S; D)$ or $MVA(S; D)$ over all possible M-combinations with no relaxation is too computationally expensive. A simple greedy solution or the linear relaxation solution of the problem may not result in a good solution.

Instead of solving the combinatorial problem for maximizing $WVA(S; D)$ or $MVA(S; D)$, we still solve the original SDP problem (3) for selecting promising base classifiers as a preliminary step, and the new selection criterion $WVA(S; D)$ or $MVA(S; D)$ is applied to re-size the preliminary selection to a desired size. Our new re-sizing step is described as follows. Suppose that *M1* classifiers were chosen by Steps 1 and 2 of the original SDP pruning algorithm. As we discussed, *M1* is typically not equal to a desired size *M*. If *M1* is less than a desired size *M*, one should choose and add additional classifiers to the ensemble by the forward selection algorithm in Algorithm 2. If *M1* is larger than *M*, we reduce the initial selection to the ensemble of size *M* by the backward subtraction algorithm in Algorithm 3.

Algorithm 2. Forward selection ensemble pruning algorithm

```
Input:  $H = \{1,...,L\}$, indices for all base
               classifiers
        $D$ = training data
        $w_m$= the weight of base classifier m M = the
               desirable ensemble size
Output: $S$ is a subset of $H$.
Step 1. Initialize $S$ to be a set of the indices for the
        $M1$ classifiers selected by Step 1 and Step
        2 of Algorithm 1.
Step 2. while size($S$) $\neq M$ do
        $m_t = \text{argmax}_{m\in H\setminus S}$ WVA($S \cup \{m\}$; $D$)
        $S = S \cup \{m_t\}$
        end while
```

Algorithm 3. Backward selection ensemble pruning algorithm

```
Input: $H = \{1, \ldots, L\}$, indices for all base classifiers
    $D$ = training data
    $w_m$= the weight of base classifier m M = the desirable ensemble size
Output: $S$ is a subset of $H$.
Step 1. Initialize $S$ to be a set of the indices for the $M1$ classifiers selected by Step 1 and Step 2 of Algorithm 1.
Step 2. while size($S$) $\neq M$ do
    $m_t = \text{argmax}_{m \in S}$ WVA($S \setminus \{m\}$; $D$)
    $S = S \setminus \{m_t\}$
end while
```

5. NUMERICAL STUDY

We evaluate the performance of our revised SDP pruning algorithm against the unpruned ensemble, the best individual classifier, kappa pruning [8], diversity pruning [8], and the original SDP pruning [14] for a number of publicly available bench-mark datasets, where the unpruned ensemble implies the full ensemble of all base classifiers with no pruning, and the best individual implies the classifier in the full ensemble that obtains the smallest training error. We implemented the kappa pruning, the diversity pruning and the SDP pruning for comparison. In implementing the SDP pruning algorithm, we used the CVX toolbox [26] for solving convex optimization problems with the SDPT3 solver [27].

Table 1: Benchmark datasets

Datasets	*# of cases*	*# of attributes*	*# of classes*
haberman	306	4	2
sonar	208	61	2
SpectHeart	267	23	2
tic-tac-toe	958	10	2
Glass	214	10	6
abalone	4177	9	3
car	1728	7	4
cmc	1473	10	3
iris	150	5	3
Dermatology	358	35	6
mfeat-mor	2000	7	10
mfeat-pix	2000	241	10
optdigits	5620	65	10
page	5473	11	5
pendigits	10992	17	10
sat	6435	37	6
segmentation	2310	20	7
semeion	1593	257	10
thyroid	215	6	3
Vehicle	846	19	3
vertebral	310	7	3
vowel	528	11	11
waveforms	5000	41	3
wine	178	14	3
databanknote	1372	5	2
bloodtrans	748	5	2
climate	540	19	2
ILPD	583	11	2
ionosphere	351	35	2
spambase	4601	58	2
wholesalecustomer1	440	8	2
australian	690	15	2
wdbc1	569	32	2
wpbc1	198	33	2
ecoli	336	8	8
zoo	101	17	7
balance	625	5	3
pima	768	9	2

contd. table 1

A total of 38 classification datasets from the UCI repository were used as benchmark datasets in our numerical experiments as shown in Table 1. Each of the datasets is randomly split into training data D_a and test data D_b with the training data containing 70% of whole data. A classification and regression tree with single split was used as a base classifier. An ensemble of L single split classification and regression trees were generated by the standard procedure of the Adaboost. M1 [2]. Next, we applied the kappa pruning [8], diversity pruning [8], the original SDP pruning [14] and our revised SDP pruning on the ensemble to generate an ensemble of size $L \times p$, where p is a fractional number less than 1. For given choices $M \in \{50, 100, 200\}$ and $p \in \{0.1, 0.25, 0.5\}$, each of the methods outputs $S \subset \{1, 2, \ldots, L\}$ that is a set of the indices for the selected base classifiers. The output S was evaluated in terms of the classification accuracy with test data D_b, which is a collection of predictor u_i and the corresponding class label y_i. When base classifier m takes u_i and return output $C_m(u_i)$ in a one-of-K representation, the overall classification accuracy of S is measured by either the weighted voting accuracy,

$$WVA(S;D_b)=1-\frac{1}{|D_b|}\sum_{(u_i,y_i)\in D_b} I\left(y_i \neq \max el\left(\sum_{m\in S} w_m C_m(u_i)\right)\right),$$

and the majority voting accuracy,

$$MVA(S;D_b)=1-\frac{1}{|D_b|}\sum_{(u_i y_i)\in D_b=1} I\left(y_i \neq \max el\left(\sum_{m\in S} C_m(u_i)\right)\right),$$

where $|D_b|$ is the number of the elements in the set D_b, and w_m is the weight for the base classifier m; the weights were obtained during the Adaboost. M1 procedure.

We repeated the random data split and the evaluation procedure ten times and took the averages and the sample standard deviations of MVAs or WVAs over the ten runs. The WVA statistics for M = 100 and p = 0.25 are summarized in Table 2. Based on the statistics, we compared the performance of our method with that of the five different ensemble methods, kappa pruning, diversity pruning, the original SDP pruning, the best individual, and the unpruned full ensemble. The comparison was summarized as the win-loss-tie (W/L/T) statistics at the bottom of the table. The absolute W/L/T for each table column counts the number of win, loss and tie cases of our method against the method corresponding to the table column in terms of the average WVA, and the significant W/L/T is the

Table 2: Comparison of weight voting accuracy (WVA) in between our method and one of the kappa pruning, diversity pruning, the original SDP pruning, the best individual, and the original ensemble with no pruning. The number in each cell is the average WVA ± standard deviation of WVA over ten experimental runs. The absolute W/L/T (Win/Loss/Tie) for each table column counts the number of win, loss and tie cases of our method against the method corresponding to the table column in terms of the average WVA, and the significant W/L/T is the same count based on a simple t-test on the significance of the difference in the average WVA.

Datasets	*Our Method*	*Kappa*	*Diversity*	*SDP*	*Best Individual*	*Unpruned*
haberman	76.96 ± 3.11	73.91 ± 6.57	73.70 ± 4.89	75.87 ± 3.47	77.46 ± 2.11	75.44 ± 4.18
sonar	90.79 ± 2.07	93.65 ± 2.51	75.87 ± 4.11	92.06 ± 3.89	77.62 ± 5.34	94.92 ± 2.07
SpectHeart	76.05 ± 2.97	71.36 ± 3.43	73.09 ± 2.68	75.56 ± 2.37	73.62 ± 3.30	72.59 ± 2.68
tic-tac-toe	79.10 ± 3.02	77.36 ± 2.73	76.74 ± 1.68	76.60 ± 2.72	70.59 ± 2.67	79.93 ± 2.28
Glass	67.08 ± 3.37	52.62 ± 2.96	66.46 ± 3.51	65.23 ± 5.72	51.36 ± 2.70	52.62 ± 2.96
abalone	61.31 ± 1.15	60.41 ± 1.30	61.09 ± 0.66	60.77 ± 1.40	57.79 ± 0.58	60.53 ± 1.15
car	85.24 ± 0.78	69.63 ± 0.77	70.06 ± 0.57	84.16 ± 2.23	69.64 ± 0.00	79.69 ± 1.50
cmc	55.48 ± 2.63	48.10 ± 1.61	51.81 ± 1.56	52.22 ± 1.48	46.01 ± 0.55	51.95 ± 1.64
iris	98.22 ± 0.99	98.22 ± 0.99	98.22 ± 0.99	98.22 ± 0.99	66.67 ± 0.00	98.22 ± 0.99
Dermatology	95.56 ± 2.88	84.81 ± 3.25	93.15 ± 3.11	91.30 ± 3.04	51.84 ± 0.36	95.00 ± 2.23
mfeat-mor	39.70 ± 2.19	29.53 ± 1.60	39.53 ± 2.15	39.90 ± 2.10	20.87 ± 0.08	39.53 ± 2.15
mfeat-pix	45.77 ± 2.49	40.27 ± 2.77	44.20 ± 2.27	45.30 ± 2.51	19.98 ± 0.18	40.27 ± 2.77
optdigits	70.88 ± 1.68	43.67 ± 1.07	63.96 ± 0.59	66.71 ± 2.50	20.30 ± 0.05	67.99 ± 0.45
page	96.31 ± 0.40	93.34 ± 0.39	91.05 ± 0.56	94.50 ± 0.90	92.85 ± 0.33	93.79 ± 0.38
pendigits	56.82 ± 0.79	30.07 ± 0.36	50.98 ± 2.31	55.00 ± 1.71	20.56 ± 0.02	52.97 ± 0.79
sat	81.30 ± 1.43	72.07 ± 1.03	74.85 ± 1.14	78.17 ± 1.86	44.16 ± 0.17	79.31 ± 0.94
segmentation	88.31 ± 1.02	80.14 ± 1.45	75.01 ± 5.75	82.86 ± 2.05	29.19 ± 0.00	81.76 ± 0.58
semeion	36.19 ± 1.96	19.04 ± 1.65	32.72 ± 2.36	35.98 ± 2.35	19.88 ± 0.26	32.34 ± 2.42
thyroid	99.69 ± 0.69	99.08 ± 0.84	97.54 ± 0.84	98.15 ± 1.69	83.62 ± 1.62	99.69 ± 0.69
Vehicle	70.31 ± 1.49	68.66 ± 1.64	66.69 ± 2.09	68.90 ± 1.91	57.70 ± 1.46	68.66 ± 1.64
vertebral	83.66 ± 2.89	82.15 ± 2.23	67.53 ± 4.39	82.80 ± 3.04	76.14 ± 0.66	82.37 ± 2.36
vowel	39.37 ± 5.12	25.91 ± 4.16	36.73 ± 4.55	36.98 ± 5.23	19.83 ± 0.74	32.58 ± 5.60
waveforms	81.57 ± 0.99	79.28 ± 0.91	77.92 ± 2.60	80.21 ± 1.32	57.17 ± 0.51	81.76 ± 0.91
wine	99.63 ± 0.83	96.67 ± 2.75	97.41 ± 1.66	98.52 ± 2.03	72.50 ± 1.11	97.78 ± 1.55
databanknote	99.32 ± 0.67	99.17 ± 0.41	67.38 ± 1.35	96.94 ± 0.93	85.03 ± 1.86	100.00 ± 0.00
bloodtrans	81.78 ± 2.81	80.89 ± 3.54	76.44 ± 3.19	81.33 ± 3.30	74.96 ± 0.73	80.89 ± 3.54
climate	96.30 ± 1.51	97.41 ± 0.80	89.75 ± 4.04	96.05 ± 0.34	91.80 ± 0.38	97.90 ± 1.03
ILPD	73.26 ± 1.95	72.46 ± 4.30	69.26 ± 2.47	74.51 ± 2.73	72.43 ± 0.90	76.11 ± 3.29
ionosphere	96.79 ± 2.17	94.91 ± 1.58	48.87 ± 4.97	94.15 ± 1.23	85.13 ± 1.87	97.74 ± 1.43
spambase	92.34 ± 0.73	92.40 ± 0.63	76.84 ± 1.17	91.77 ± 0.80	79.12 ± 0.59	93.38 ± 0.78
wholesalecust1	93.48 ± 1.74	90.91 ± 2.40	79.09 ± 8.58	92.42 ± 2.34	90.68 ± 2.10	92.42 ± 2.78
australian	88.94 ± 1.23	88.36 ± 1.50	49.71 ± 5.47	88.17 ± 1.21	85.65 ± 1.70	87.50 ± 1.23
wdbc1	99.06 ± 0.32	98.48 ± 0.67	80.23 ± 7.80	98.95 ± 0.64	92.72 ± 1.19	99.06 ± 0.32
wpbc1	90.00 ± 5.00	88.33 ± 5.89	86.33 ± 4.31	87.67 ± 5.35	76.88 ± 1.29	95.00 ± 2.36
ecoli	84.16 ± 3.64	63.37 ± 2.89	79.80 ± 3.81	76.44 ± 3.73	64.83 ± 1.06	76.04 ± 3.08
zoo	87.10 ± 9.68	82.58 ± 10.60	82.58 ± 10.60	85.16 ± 12.20	62.15 ± 0.99	82.58 ± 10.60
balance	87.98 ± 1.94	81.49 ± 3.43	82.02 ± 4.31	88.62 ± 2.18	68.11 ± 2.95	86.70 ± 2.58
pima	77.49 ± 1.73	75.41 ± 2.26	76.71 ± 1.87	76.31 ± 1.61	78.53 ± 2.61	77.49 ± 1.73
Absolute W/L/T		33/4/1	37/0/1	33/4/1	37/1/0	25/10/3
Significant W/L/T		22/2/14	28/0/10	14/0/24	34/0/4	16/6/16

same count based on a simple t-test on the significance of the difference in the average WVA. Our method outperformed all of the compared methods including the unpruned full ensemble.

We also compared our method and the state-of-the-art with the unpruned full ensemble in terms of the significant W/L/T statistics. Table 3 shows the comparison for different experimental settings. Our method outperformed the unpruned ensemble, while the kappa pruning and the diversity pruning underperformed the unpruned ensemble. The original SDP pruning algorithm, while being better than the

Table 3: Significant W/L/T statistics of our method and the state-of-the-art against the unpruned ensemble for different experimental settings. n = the size of original ensemble and k = the size of the selected ensemble.

n,k	*Kappa*	*Diversity*	*SDP*	*Our method*
50,5	1/32/6	3/27/9	8/18/13	12/13/14
50,12	2/31/6	6/21/12	9/14/16	13/10/16
50,25	5/20/14	7/9/13	14/8/17	18/5/16
100,10	1/32/6	5/25/9	5/15/19	11/4/14
100,25	6/20/12	7/24/7	9/9/20	21/3/14
100,50	4/21/14	6/23/10	10/7/22	16/7/16
200,20	0/33/5	4/28/6	5/18/15	17/7/14
200,50	4/21/13	5/27/6	7/9/22	18/3/17
200,100	5/21/12	4/23/11	5/6/27	17/4/17

Table 4 Comparison of majority voting accuracy (MVA) in between our method and one of the kappa pruning, diversity pruning, the original SDP pruning, the best individual, and the original ensemble with no pruning. The number in each cell is the average MVA ± standard deviation of MVA over ten experimental runs.

Datasets	*Our Method*	*Kappa*	*Diversity*	*SDP*	*Best Individual*	*Unpruned*
haberman	75.65 ± 3.89	70.44 ± 2.48	53.48 ± 14.89	75.65 ± 2.62	75.28 ± 1.49	70.22 ± 4.71
sonar	86.35 ± 4.29	80.64 ± 4.40	65.08 ± 3.72	83.81 ± 4.68	76.02 ± 2.80	81.91 ± 4.71
SpectHeart	70.62 ± 1.61	63.21 ± 3.20	54.32 ± 3.60	69.63 ± 2.84	73.64 ± 2.62	57.04 ± 3.43
tic-tac-toe	78.82 ± 2.39	71.94 ± 5.71	58.26 ± 3.60	76.81 ± 3.36	69.28 ± 2.06	78.47 ± 3.63
Glass	54.77 ± 8.33	44.00 ± 10.80	48.31 ± 10.41	51.38 ± 8.26	48.48 ± 3.53	46.15 ± 4.74
abalone	60.57 ± 1.41	57.03 ± 1.16	57.83 ± 1.34	59.30 ± 0.77	57.89 ± 0.89	55.25 ± 1.00
car	82.27 ± 3.98	56.53 ± 1.97	61.31 ± 11.76	83.28 ± 3.49	70.36 ± 0.61	42.27 ± 1.56
cmc	55.16 ± 1.56	42.31 ± 2.68	50.36 ± 2.26	54.03 ± 1.49	46.37 ± 0.97	51.27 ± 2.70
iris	76.00 ± 14.09	66.22 ± 3.98	47.11 ± 12.31	80.44 ± 13.91	68.19 ± 1.44	66.22 ± 3.98
Dermatology	62.41 ± 14.85	13.89 ± 3.53	33.89 ± 13.71	62.04 ± 13.16	49.19 ± 1.06	16.30 ± 2.82
mfeat-mor	17.07 ± 1.55	17.03 ± 1.51	17.03 ± 1.55	17.20 ± 1.64	21.09 ± 0.60	17.03 ± 1.51
mfeat-pix	23.80 ± 3.97	17.67 ± 0.59	19.63 ± 6.24	21.73 ± 4.09	20.37 ± 0.47	17.67 ± 0.59
optdigits	44.55 ± 6.36	23.13 ± 5.75	38.33 ± 10.55	40.51 ± 5.77	20.40 ± 0.43	26.26 ± 4.40
page	95.20 ± 0.67	89.57 ± 1.13	62.28 ± 17.39	94.64 ± 0.36	93.47 ± 0.32	74.18 ± 20.02
pendigits	32.93 ± 5.91	19.85 ± 0.23	22.92 ± 9.69	27.60 ± 6.60	20.76 ± 0.27	20.81 ± 7.47
sat	70.84 ± 2.68	23.86 ± 1.07	60.80 ± 3.53	61.14 ± 7.67	44.09 ± 0.42	39.67 ± 9.03
segmentation	41.62 ± 10.42	21.36 ± 8.12	27.39 ± 13.61	35.04 ± 12.18	29.33 ± 0.46	22.60 ± 6.49
semeion	23.26 ± 2.92	19.16 ± 1.05	19.71 ± 3.28	22.80 ± 4.87	19.90 ± 0.51	19.16 ± 1.05
thyroid	84.31 ± 2.53	79.08 ± 8.74	82.15 ± 6.40	84.61 ± 3.61	83.31 ± 1.83	75.69 ± 10.63
Vehicle	62.76 ± 4.87	51.50 ± 0.85	57.80 ± 5.69	60.95 ± 3.45	57.38 ± 1.91	51.50 ± 0.85
vertebral	78.92 ± 3.37	76.56 ± 2.07	58.49 ± 7.92	77.63 ± 3.98	79.28 ± 1.70	66.24 ± 3.19
vowel	23.27 ± 3.24	14.34 ± 2.79	18.36 ± 6.45	20.50 ± 7.21	18.59 ± 0.48	13.46 ± 2.38
waveforms	80.20 ± 0.49	69.11 ± 4.13	79.80 ± 0.78	79.64 ± 0.91	57.72 ± 0.43	71.81 ± 1.72
wine	97.04 ± 1.66	90.37 ± 7.79	93.70 ± 2.81	96.30 ± 3.93	69.23 ± 1.14	74.07 ± 18.84
databanknote	99.17 ± 0.37	97.04 ± 1.18	75.34 ± 5.24	97.48 ± 0.74	86.11 ± 1.17	99.56 ± 0.36
bloodtrans	80.89 ± 1.04	74.49 ± 5.03	57.60 ± 11.92	79.73 ± 2.56	76.00 ± 0.88	73.24 ± 6.04
climate	94.20 ± 1.42	90.49 ± 2.33	57.16 ± 8.08	94.07 ± 1.42	91.59 ± 0.78	92.47 ± 2.02
ILPD	74.40 ± 2.75	67.09 ± 5.81	52.91 ± 7.54	70.97 ± 2.90	71.15 ± 1.51	60.11 ± 5.32
ionosphere	93.21 ± 2.15	93.77 ± 2.55	77.55 ± 7.17	92.83 ± 2.17	85.43 ± 1.55	93.96 ± 1.96
spambase	92.89 ± 0.84	91.14 ± 1.11	61.55 ± 11.89	92.57 ± 1.29	78.53 ± 0.76	84.40 ± 5.19
wholesale customer1	90.00 ± 1.46	84.09 ± 6.38	62.27 ± 15.13	90.00 ± 1.12	89.87 ± 1.10	81.67 ± 6.32
australian	85.77 ± 1.94	82.60 ± 1.94	61.15 ± 7.03	85.67 ± 2.66	86.13 ± 2.06	78.65 ± 3.20
wdbc1	97.78 ± 0.76	96.37 ± 1.05	66.90 ± 4.31	97.78 ± 0.49	92.57 ± 1.09	96.96 ± 1.12
wpbc1	81.00 ± 2.24	71.33 ± 7.58	60.33 ± 9.08	80.00 ± 2.04	77.89 ± 2.63	71.67 ± 5.89
zoo	76.77 ± 10.05	70.32 ± 10.05	67.74 ± 23.04	72.90 ± 8.72	64.32 ± 6.22	12.90 ± 2.28
balance	87.02 ± 1.22	81.92 ± 2.71	85.00 ± 2.99	87.34 ± 1.57	65.72 ± 0.79	85.75 ± 1.74
pima	76.88 ± 2.09	72.64 ± 3.60	49.35 ± 4.46	76.71 ± 1.69	73.06 ± 1.52	7.27 ± 9.49
Absolute W/L/T		36/2/0	38/0/0	30/5/3	34/4/0	35/3/0
Significant W/L/T		33/0/5	31/0/7	6/0/32	29/2/7	33/1/4

Table 5: Weight voting accuracy performance of the SDP pruning with the revised re-sizing step with different selection criteria. The W/L/T statistics are the comparison of the original SDP against the methods corresponding to the table columns

Datasets	*SDP*	*WVA*	*Double Fault*	*Weighted Error Count*	*Disagreement*
haberman	75.87 ± 3.47	76.96 ± 3.11	75.65 ± 2.38	77.17 ± 2.77	76.30 ± 3.11
sonar	92.06 ± 3.89	90.79 ± 2.07	86.98 ± 3.05	92.06 ± 4.49	91.11 ± 2.41
SpectHeart	75.56 ± 2.37	76.05 ± 2.97	72.59 ± 5.19	72.35 ± 6.09	72.59 ± 4.57
tic-tac-toe	76.60 ± 2.72	79.10 ± 3.02	74.93 ± 1.02	77.15 ± 2.42	76.94 ± 1.19
Glass	65.23 ± 5.72	67.08 ± 3.37	51.69 ± 4.82	51.69 ± 4.82	49.85 ± 6.67
abalone	60.77 ± 1.40	61.31 ± 1.15	59.98 ± 1.06	59.74 ± 0.96	59.62 ± 1.00
car	84.16 ± 2.23	85.24 ± 0.78	78.57 ± 1.20	80.46 ± 0.87	80.46 ± 0.87
cmc	52.22 ± 1.48	55.48 ± 2.63	49.68 ± 2.80	52.81 ± 2.68	54.30 ± 1.75
iris	98.22 ± 0.99	98.22 ± 0.99	98.67 ± 1.22	98.22 ± 2.90	98.67 ± 1.22
Dermatology	91.30 ± 3.04	95.56 ± 2.88	85.93 ± 4.41	92.78 ± 4.87	93.15 ± 5.14
mfeat-mor	39.90 ± 2.10	39.70 ± 2.19	20.83 ± 0.57	20.70 ± 0.72	20.83 ± 0.57
mfeat-pix	45.30 ± 2.51	45.77 ± 2.49	47.10 ± 2.07	47.20 ± 2.09	47.27 ± 2.11
optdigits	66.71 ± 2.50	70.88 ± 1.68	69.11 ± 2.17	69.13 ± 0.88	69.30 ± 1.50
page	94.50 ± 0.90	96.31 ± 0.40	93.63 ± 0.49	94.30 ± 0.60	93.58 ± 0.44
pendigits	55.00 ± 1.71	56.82 ± 0.79	53.01 ± 1.29	54.45 ± 0.91	54.03 ± 1.96
sat	78.17 ± 1.86	81.30 ± 1.43	78.27 ± 1.01	79.05 ± 1.63	78.55 ± 1.99
segmentation	82.86 ± 2.05	88.31 ± 1.02	82.37 ± 0.60	83.41 ± 1.36	82.83 ± 0.64
semeion	35.98 ± 2.35	36.19 ± 1.96	33.93 ± 2.88	36.86 ± 3.11	37.16 ± 2.46
thyroid	98.15 ± 1.69	99.69 ± 0.69	96.31 ± 2.33	97.85 ± 0.84	96.31 ± 0.84
Vehicle	68.90 ± 1.91	70.31 ± 1.49	65.35 ± 1.55	65.12 ± 1.84	65.59 ± 1.38
vertebral	82.80 ± 3.04	83.66 ± 2.89	79.36 ± 2.33	85.16 ± 3.17	81.72 ± 4.37
vowel	36.98 ± 5.23	39.37 ± 5.12	31.20 ± 3.46	31.57 ± 3.00	30.44 ± 1.30
waveforms	80.21 ± 1.32	81.57 ± 0.99	78.43 ± 1.53	80.45 ± 0.87	80.28 ± 1.53
wine	98.52 ± 2.03	99.63 ± 0.83	98.52 ± 1.55	99.63 ± 0.83	99.63 ± 0.83
databanknote	96.94 ± 0.93	99.32 ± 0.67	97.09 ± 1.03	96.80 ± 1.10	96.46 ± 1.43
bloodtrans	81.33 ± 3.30	81.78 ± 2.81	76.44 ± 3.13	79.11 ± 1.37	77.96 ± 1.93
climate	96.05 ± 0.34	96.30 ± 1.51	96.17 ± 1.77	97.28 ± 1.20	95.93 ± 1.42
ILPD	74.51 ± 2.73	73.26 ± 1.95	73.83 ± 2.54	74.97 ± 3.09	76.00 ± 3.31
ionosphere	94.15 ± 1.23	96.79 ± 2.17	90.19 ± 2.07	92.45 ± 3.59	91.70 ± 3.68
spambase	91.77 ± 0.80	92.34 ± 0.73	92.30 ± 0.52	92.01 ± 0.49	91.90 ± 0.50
wholesalecust1	92.42 ± 2.34	93.48 ± 1.74	92.73 ± 1.74	93.79 ± 1.73	93.18 ± 1.86
australian	88.17 ± 1.21	88.94 ± 1.23	87.21 ± 1.30	87.79 ± 1.11	87.79 ± 0.73
wdbc1	98.95 ± 0.64	99.06 ± 0.32	97.89 ± 0.78	97.89 ± 0.67	98.25 ± 0.58
wpbc1	87.67 ± 5.35	90.00 ± 5.00	88.00 ± 2.98	88.00 ± 5.06	87.67 ± 3.03
ecoli	76.44 ± 3.73	84.16 ± 3.64	80.00 ± 3.60	81.58 ± 3.87	77.23 ± 4.43
zoo	85.16 ± 12.20	87.10 ± 9.68	91.61 ± 5.86	91.61 ± 1.77	91.61 ± 1.77
balance	88.62 ± 2.18	87.98 ± 1.94	82.02 ± 1.38	85.64 ± 2.52	85.53 ± 2.53
pima	76.71 ± 1.87	77.49 ± 1.73	77.49 ± 1.98	76.80 ± 2.02	77.58 ± 1.69
Absolute W/L/T		4/33/1	25/12/1	16/20/2	20/17/1
Significant W/L/T		0/14/24	17/2/19	9/4/25	12/3/23

other two benchmark ensemble pruning method, does not perform better than the unpruned ensemble. For many of the test datasets, the original SDP pruning algorithm actually ties with the unpruned ensemble; this was also discussed in Zhang *et al.* [14].

Table 4 shows the MVA statistics for the compared methods for M = 100 and p = 0.25. For this comparison, the majority voting accuracy was used as a criterion of selecting base classifiers in our method. We could observe a few findings that are different from what was observed in Table 2. First, for many datasets, the unpruned ensemble is no better than the best individual classifier. This implies that combining weak classifiers by a majority voting scheme may not give much performance gain over using a good individual classifier. Second, although our method is still better than the original SDP pruning algorithm in terms of the absolute W/L/T statistics, they are comparable in terms of the significant W/L/T, which means that the performance gap between the two methods is not significant for many datasets.

5.1 Effect of different selection criteria in the greedy re-sizing step of SDP pruning algorithm

In this section, we numerically study the eûect of using different selection criteria in the greedy re-sizing step of the SDP pruning algorithm. For the study, we considered four different selection criteria, the disagreement measure [23], the double fault measure [24], the weighted error count and correct value [25], and weighted voting accuracy measure (our method). The first three measures and the measure used in the original SDP pruning are ensemble diversity measures, while a weighted voting accuracy is an ensemble accuracy measure.

We computed the WVA performance of the SDP pruning algorithm when each of the four diûerent selection criteria is used. Table 5 summarizes the results. Applying either the disagreement measure [23], the double fault measure [24] or the weighted error count and correct value [25] does not appear to improve the original SDP pruning algorithm, while applying a weighted voting accuracy improved the SDP pruning algorithm significantly in terms of the WVA accuracy. This supports our argument that using an ensemble accuracy measure instead of using an ensemble diversity measure in the re-sizing step improves the performance of the SDP pruning algorithm more significantly.

6. CONCLUSION

In this paper we proposed to revise the resizing step of the SDP pruning algorithm for classification problems. Given a set of many base classifiers, the current SDP pruning algorithm first solves a semi-definite problem of taking a subset of some promising base classifiers as a preliminary selection and reduce the size of the subset to a desired size through a greedy algorithm. The greedy step discards some of the base classifiers in the preliminary selection in a greed fashion towards the increase of a chosen criterion. We showed the criterion does not really reflect the overall classification accuracy of the selected ensemble so the greedy step can be guided for better accuracy if a better criterion is chosen for the step. We proposed to use a weight voting accuracy measure as a selection criterion.

Our numerical study showed the new re-sizing step improved the overall ensemble accuracy of the SDP pruning algorithm. We also tested the performance of the SDP pruning algorithm with different selection criteria used in the re-sizing step. The test showed that maximizing a weighted voting accuracy measure in the re-sizing step produced better results than using the other selection criteria.

Acknowledgement

This work was supported in part by the Air Force Office of Scientific Research FA9550-13-1-0075 and the National Science Foundation NSF-1334012.

Biographical Notes

Adeseye Agbabiaka is PhD student of Industrial and Manufacturing Engineering at Florida State University. Agbabiaka's current research interests are ensemble learning.

Chiwoo Park received B.S. in industrial engineering at Seoul National University and Ph.D. degree in industrial engineering at Texas A&M University in 2011. He is currently Assistant Professor in the Department of Industrial and Manufacturing Engineering at Florida State University and a principal investigator at High Performance Material Institute. His research is data analytics for science and engineering problems, especially nanoscience and manufacturing engineering. His work is being supported by the National Science Foundation and the Air Force Office of Scientific Research. He received the best student paper award at IEEE Conferences on Automation Science and Engineering in 2008, the Ralph E. Powe Junior Faculty Award from the Oak Ridge Associated Universities in 2013, and the IIE best application paper in 2014. He is a active member of IIE, IEEE and INFORMS.

References

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

[2] Y. Freund, R. E. Schapire, et al., "Experiments with a new boosting algorithm," *in Proceedings of the Thirteenth International Conference on Machine Learning*, vol. 96, pp. 148–156, 1996.

[3] L. K. Hansen and P. Salamon, "Neural network ensembles," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 10, pp. 993–1001, 1990.

[4] W. N. Street and Y. Kim, "A streaming ensemble algorithm (sea) for large-scale classification," *in Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 377– 382, ACM, 2001.

[5] H. Wang, W. Fan, P. S. Yu, and J. Han, "Mining concept-drifting data streams using ensemble classifiers," *in Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 226–235, ACM, 2003.

[6] Z.-H. Zhou, W. Tang, Z.-H. Zhou, and W. Tang, "Selective ensemble of decision trees," *in Lecture Notes in Artificial Intelligence*, pp. 476–483, Springer, 2003.

[7] 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.

[8] D. D. Margineantu and T. G. Dietterich, "Pruning adaptive boosting," *in Proceedings of the Fourteenth*

International Conference on Machine Learning, vol. 97, pp. 211–218, 1997.

[9] G. Mart´ýnez-Mun˜oz and A. Sua´rez, "Pruning in ordered bagging ensembles," *in Proceedings of the 23rd International Conference on Machine Learning*, pp. 609–616, ACM, 2006.

[10] G. Giacinto, F. Roli, and G. Fumera, "Design of effective multiple classifier systems by clustering of classifiers," *in Proceedings of the Fifteen International Conference on Pattern Recognition*, vol. 2, pp. 160–163, IEEE, 2000.

[11] A. Lazarevic and Z. Obradovic, "Effective pruning of neural network classifier ensembles," *in Proceedings of the International Joint Conference on Neural Networks*, vol. 2, pp. 796–801, IEEE, 2001.

[12] H. Chen, P. Tino, and X. Yao, "A probabilistic ensemble pruning algorithm," *in Proceedings of the Sixth IEEE International Conference on Data Mining Workshops*, pp. 878–882, IEEE, 2006.

[13] H. Chen, P. Tiho, and X. Yao, "Predictive ensemble pruning by expectation propagation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 7, pp. 999–1013, 2009.

[14] Y. Zhang, S. Burer, and W. N. Street, "Ensemble pruning via semi-definite programming," *Journal of Machine Learning Research*, vol. 7, pp. 1315– 1338, 2006.

[15] A. Chandra, H. Chen, and X. Yao, "Trade-off between diversity and accuracy in ensemble generation," *in Multi-objective machine learning*, pp. 429–464, Springer, 2006.

[16] W. Fan, F. Chu, H. Wang, and P. S. Yu, "Pruning and dynamic scheduling of cost-sensitive ensembles," *in Proceedings of the Eighteenth National Conference on Artificial Intelligence*, pp. 146–151, American Association for Artificial Intelligence, 2002.

[17] R. Caruana, A. Niculescu-Mizil, G. Crew, and A. Ksikes, "Ensemble selection from libraries of models," *in Proceedings of the 21st International Conference on Machine learning*, p. 18, ACM, 2004.

[18] G. Martýnez-Munoz and A. Sua´rez, "Aggregation ordering in bagging," *in Proceedings of the IASTED International Conference on Artificial Intelligence and Applications*, pp. 258–263, 2004.

[19] R. E. Banfield, L. O. Hall, K. W. Bowyer, and W. P. Kegelmeyer, "Ensemble diversity measures and their application to thinning," *Information Fusion*, vol. 6, no. 1, pp. 49–62, 2005.

[20] I. Partalas, G. Tsoumakas, and I. P. Vlahavas, "Focused ensemble selection: A diversity-based method for greedy ensemble selection," *in Proceedings of the 18th European Conference on Artificial Intelligence*, pp. 117–121, 2008.

[21] D. Herna´ndez-Lobato, G. Mart´ýnez-Mun˜oz, and A. Sua´rez, "Empirical analysis and evaluation of approximate techniques for pruning regression bagging ensembles," *Neurocomputing*, vol. 74, no. 12, pp. 2250–2264, 2011.

[22] Q. Han, Y. Ye, and J. Zhang, "An improved rounding method and semidefinite programming relaxation for graph partition," *Mathematical Programming*, vol. 92, no. 3, pp. 509–535, 2002.

[23] D. B. Skalak, "The sources of increased accuracy for two proposed boosting algorithms," *in Proceedings of the American Association for Artificial Intelligence, Integrating Multiple Learned Models Workshop*, pp. 120– 125, 1996.

[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. Aksela and J. Laaksonen, "Using diversity of errors for selecting members of a committee classifier," *Pattern Recognition*, vol. 39, no. 4, pp. 608–623, 2006.

[26] M. Grant and S. Boyd, "Cvx: Matlab software for disciplined convex programming, version 2.1.," 2014.

[27] K.-C. Toh, M. J. Todd, and R. H. Tu¨tu¨ncu¨, "Sdpt3a matlab software package for semidefinite programming, version 1.3," *Optimization Methods and Software*, vol. 11, no. 1-4, pp. 545–581, 1999.