2012 IEEE 12th International Conference on Data Mining

Ensemble Pruning via Constrained Eigen-Optimization

Linli Xu, Bo Li, Enhong Chen
School of Computer Science and Technology
University of Science and Technology of China
Hefei, Anhui
linlixu@ustc.edu.cn, lib3@mail.ustc.edu.cn, cheneh@ustc.edu.cn

***Abstract*—An ensemble is composed of a set of base learners that make predictions jointly. The generalization performance of an ensemble has been justified both theoretically and in practice. However, existing ensemble learning methods sometimes produce unnecessarily large ensembles, with an expense of extra computational costs and memory consumption. The purpose of ensemble pruning is to select a subset of base learners with comparable or better prediction performance. In this paper, we formulate the ensemble pruning problem into a combinatorial optimization problem with the goal to maximize the accuracy and diversity at the same time. Solving this problem exactly is computationally hard. Fortunately, we can relax and reformulate it as a constrained eigenvector problem, which can be solved with an efficient algorithm that is guaranteed to converge globally. Convincing experimental results demonstrate that this optimization based ensemble pruning algorithm outperforms the state-of-the-art heuristics in the literature.**

***Keywords*-ensemble pruning; optimization**

I. Introduction

Ensemble methods have been a very active research field in the machine learning and data mining communities during the past decade. By definition, an ensemble is composed of a set of base learners that make predictions jointly. The ensemble methods have achieved significant empirical success in many applications, which arises largely from the well-accepted fact that an ensemble usually generalizes better than a single classifier given the same amount of training information. According to Krogh and Vedelsby [1], the performance of an ensemble relies on the accuracy as well as the diversity of its members. That is, the members of an ideal ensemble should produce highly accurate predictions while making different errors as much as possible. On the other hand, an increase in the accuracies of ensemble members implies lower diversity, therefore constructing an effective ensemble involves taking care of the tradeoff between accuracy and diversity.

A number of approaches have been proposed to generate ensembles effectively, such as bagging [2], boosting [3], random forests [4]. One issue with ensemble methods is the tendency to construct ensembles with unnecessarily large sizes. This comes with two costs – the memory required to store all the base learners, and the processing time to get a prediction for an unlabeled test example based on all the base learners. These computational overhead can be critical for large scale problems or online applications [5], [6]. On the other hand, having a large number of models in an ensemble does not guarantee good predictive performance. For example, an ensemble that contains many similar models may have reduced diversity and capability for error correction.

Selecting a fraction of the base learners from the ensemble is typically considered an appropriate recipe to address the issues above. This technique is known as ensemble pruning or ensemble selection, where a subensemble is carefully chosen with a smaller size according to a given criterion. Besides a reduction in the computational overhead, comparable or even better predictive performance is also a potential benefit after pruning.

Ensemble pruning can be thought of as a special case of weighted ensemble learning with 0-1 weights. The goal of the general weighted ensemble learning problem is to optimize over a set of weight values of the base learners with the goal to enhance the generalization performance of the weighted ensemble [1], [7], [8]. These weight-based approaches sometimes can produce sparse solutions and reduce the size of the ensemble. However, the reduced size of the ensemble is not explicitly predefined. This is also a subtlety to be noted in various ensemble pruning and selection techniques.

Assuming the generalization performance of an ensemble can be estimated in terms of a cost function measured on the training set, to select a subensemble with the best performance from m classifiers, intuitively one need to search in the space of $2^m - 1$ non-empty subensembles. This problem is proven to be NP-complete [9], and therefore it is not practical to produce a globally optimal solution.

The majority of the efforts in ensemble pruning are then devoted to solving the problem approximately, where many approaches have been proposed. A straightforward heuristic proposed in [10] starts with training a library of about 2000 classifiers which consist of different models with different parameter settings, and then iteratively adding member classifiers from the library to the ensemble to maximize the performance according to some predefined metric. This method is simple and intuitive, however, it fails to consider the overall information of the original

1550-4786/12 $26.00 © 2012 IEEE
DOI 10.1109/ICDM.2012.97

ensemble. Besides that, there are three rough categories of ensemble pruning techniques: 1) Clustering based: models with similar predictive behaviors are clustered together, each cluster is pruned separately to reduce the overall size of the ensemble [11], [12]; 2) Ordering based: models in the ensemble are ordered based on some predefined evaluation measures, members of the subensemble are then selected according to this order [13]. Examples in this category include Kappa pruning [5], margin distance minimization [14], orientation ordering [15], and individual contribution ordering [16]; 3) Optimization based: ensemble pruning can be viewed as a combinatorial optimization problem with the goal to find a subset of the original ensemble that optimizes a predefined criterion which is an estimate of the generalization performance of the subensemble. Unlike the previous heuristic approaches, using a mathematical formulation, one can model the performance of the subensemble in a more principled way. Unfortunately, as discussed above, solving the optimization problem globally is hard. Therefore, genetic algorithms [17], [7] and semidefinite relaxations [18] have been proposed to solve the problem approximately. However, although the time complexity of these methods is no longer exponential, they still suffer from low scalability.

In this paper, we propose a new optimization based approach to solve the ensemble pruning problem. The goal is to efficiently optimize a criterion that integrates accuracy and diversity at the same time. Unlike the genetic algorithms and semidefinite relaxations discussed above, we formulate the problem as a linearly constrained eigen-optimization problem, which can be solved with an efficient algorithm that is guaranteed to converge globally. More importantly, we can achieve a significant improvement in computational efficiency with comparable or even better predictive performance. This novel optimization based approach is tested on 38 UCI repository data sets and shown to outperform the original ensemble and subensembles produced by the state of the art ensemble pruning methods.

The rest of the paper is organized as follows. We first formulate our problem as a combinatorial optimization problem, and show how to reformulate and relax it into a constrained eigenvector optimization problem which can be solved with an iterative algorithm that is guaranteed to converge to a global solution. After that some related work is discussed, followed by the experimental results to demonstrate the effectiveness of the proposed approach. The paper is then concluded with possible directions for future work.

II. Ensemble pruning with a constrained eigenvector computation

As shown in the literature, a good ensemble should satisfy the condition that its member classifiers be both accurate and diverse [19]. That is, the member classifiers should not only be accurate by themselves, but also make different errors. On the other hand, with the increase of the accuracies of the member classifiers, the diversity of the ensemble will decrease, which implies a trade-off that we should take into account when selecting a subset from the original ensemble.

To handle the accuracy and diversity at the same time in a mathematical formulation, we follow the recipe proposed by [18], which is based on the observation that the performance of an ensemble can be measured by a linear combination of its members' individual accuracy and pairwise diversity. Therefore, one could first record the predictive accuracies of the individual basic learners as well as the comparative performance of the member pairs, and integrate them with a linear combination as an approximation of the ensemble's predictive performance, which will be used as an objective in the optimization problem.

Let $L = \{(\mathbf{x}_i, y_i)|i = 1, ..., t\}$ be our training set, where $\mathbf{x}_i$ is the input feature vector of the i-th example, and $y_i \in \{1, ..., c\}$ is the corresponding class label. We start with an ensemble trained on the training data L with m member classifiers, and the goal is to select k classifiers out of them and form a new subensemble.

To achieve that, firstly a matrix M is used to record the performance of all the member classifiers on the training set:

$$M_{ij} = \begin{cases} 1, & \text{if } j\text{th classifier makes an error on the data } i \\ 0, & \text{otherwise} \end{cases}$$

Thus, $G = M^\top M$ is a matrix with the interesting properties that the diagonal entries G_{ii} represent the misclassification errors made by each classifier i on the training data, while the off-diagonal entries G_{ij} correspond to the number of common errors made by classifier i and j. Normalization is then applied on G to make its elements on the same scale

$$\tilde{G}_{ii} = \frac{G_{ii}}{t}, \quad \tilde{G}_{ij} = \frac{1}{2}\left(\frac{G_{ij}}{G_{ii}} + \frac{G_{ji}}{G_{jj}}\right) \tag{1}$$

where t is the number of training examples. It is obvious that $\tilde{G}_{ii}$ is the misclassification error rate of classifier i, and $\tilde{G}_{ij}$ measures the amount of overlapping errors between classifier i and j. Intuitively, smaller $\tilde{G}_{ii}$ corresponds to a more accurate member classifier, while smaller $\tilde{G}_{ij}$ implies a more different classifier pair. Therefore, it is straightforward that a small value of $\sum_{ij} \tilde{G}_{ij}$ implies a good ensemble; consequentially, to find an ideal subensemble, we would like to select those member classifiers with small values of $\tilde{G}_{ii}$ and $\tilde{G}_{ij}$ as well.

Based on the discussion above, in the optimization model, selecting a subensemble of size k with both accuracy and diversity can now be formulated as

$$\begin{aligned} \min_{\mathbf{w}} \quad & \mathbf{w}^\top \tilde{G} \mathbf{w} \\ \text{s.t.} \quad & \textstyle\sum_i w_i = k \\ & w_i \in \{0, 1\}. \end{aligned} \tag{2}$$

The binary variable w_i serves as a 0-1 weight or an indicator: when $w_i = 1$, the ith classifier will be selected and its

corresponding diagonal and off-diagonal entries in $\tilde{G}_{ij}$ will be counted in the objective. The parameter k controls the size of the pruned subensemble and needs to be specified beforehand.

The problem (2) is a standard 0-1 optimization problem, which is NP-hard in general. However, we will show that after some reformulation and relaxations, it can be solved in an efficient way.

First use $\mathbf{z} = 2\mathbf{w} - 1$ to replace $\mathbf{w}$, we have $\mathbf{z} \in \{-1, +1\}^m$, and the problem (2) can be rewritten as

$$\begin{aligned} \min_{\mathbf{z}} \quad & \frac{(\mathbf{z}+1)^\top \tilde{G}(\mathbf{z}+1)}{4} \\ \text{s.t. } & \mathbf{z}^\top \mathbf{1} = 2k - m \\ & z_i \in \{-1, 1\}. \end{aligned} \tag{3}$$

Now we make a transformation of the variables by letting

$$\hat{\mathbf{z}}_{(m+1)\times 1} = \begin{bmatrix} 1 \\ \mathbf{z} \end{bmatrix}$$

and

$$\tilde{G}'_{(m+1)\times(m+1)} = \begin{bmatrix} \mathbf{1}^\top \tilde{G} \mathbf{1} & \mathbf{1}^\top \tilde{G} \\ \tilde{G}\mathbf{1} & \tilde{G} \end{bmatrix}, \tag{4}$$

where $\mathbf{1}$ is a vector of all 1's. Note here in the new optimization formulation we need to explicitly constrain that the first entry of $\hat{\mathbf{z}}$ equals to 1. Now we can write problem (3) in a more concise form:

$$\begin{aligned} \min_{\hat{\mathbf{z}}} \quad & \hat{\mathbf{z}}^\top \tilde{G}' \hat{\mathbf{z}} \\ \text{s.t} \quad & \hat{\mathbf{z}} \in \{-1, +1\}^{m+1} \\ & \hat{\mathbf{z}}^\top \mathbf{1} = 2k - m + 1 \\ & \hat{z}_1 = 1. \end{aligned}$$

This can be further reformulated as

$$\begin{aligned} \min_{\mathbf{v}} \quad & \mathbf{v}^\top \tilde{G}' \mathbf{v} \\ \text{s.t} \quad & \mathbf{v} \in \{-\tfrac{1}{\sqrt{m+1}}, \tfrac{1}{\sqrt{m+1}}\}^{m+1} \\ & \mathbf{v}^\top \mathbf{1} = \tfrac{2k-m+1}{\sqrt{m+1}} \\ & v_1 = \tfrac{1}{\sqrt{m+1}} \end{aligned} \tag{5}$$

where $\mathbf{v} = \frac{\hat{\mathbf{z}}}{\sqrt{m+1}}$, and positive values in $\mathbf{v}_{2:n}$ indicate the corresponding classifiers being selected in the subensemble.

The problem above is non-convex and NP-hard, and the NP-hardness is caused by the discrete constraints $\mathbf{v} \in \{-\frac{1}{\sqrt{m+1}}, \frac{1}{\sqrt{m+1}}\}^{m+1}$. This is where we make our only relaxation: we replace the discrete constraints with a norm constraint $\|\mathbf{v}\| = 1$. This gives us

$$\begin{aligned} \min_{\mathbf{v}} \quad & \mathbf{v}^\top \tilde{G}' \mathbf{v} \\ \text{s.t} \quad & \|\mathbf{v}\| = 1 \\ & \mathbf{v}^\top \mathbf{1} = \tfrac{2k-m+1}{\sqrt{m+1}} \\ & v_1 = \tfrac{1}{\sqrt{m+1}}. \end{aligned} \tag{6}$$

The linear constraints above on vector $\mathbf{v}$ can be written in a more compact way as $A\mathbf{v} = \mathbf{b}$, where

$$A = \begin{bmatrix} 1 & 0 & ... & 0 \\ 1 & 1 & ... & 1 \end{bmatrix}$$

and

$$\mathbf{b} = \frac{1}{\sqrt{m+1}} \begin{bmatrix} 1 \\ 2k-m+1 \end{bmatrix}.$$

Problem (6) is still non-convex. Fortunately, we will see that it can be solved exactly in an efficient way.

First we switch the minimization formulation in problem (6) to a maximization problem without affecting the solution:

$$\begin{aligned} \max_{\mathbf{v}} \quad & \mathbf{v}^\top (\alpha I - \tilde{G}') \mathbf{v} \\ \text{s.t} \quad & \|\mathbf{v}\| = 1 \\ & A\mathbf{v} = \mathbf{b} \end{aligned} \tag{7}$$

where α is sufficiently large such that $\alpha I - \tilde{G}'$ is a semidefinite positive matrix. Now the problem is similar to a maximum eigenvalue computation, except that now we have some additional inhomogeneous linear constraints, which complicates the optimization problem. Fortunately, a recent technique called *Projected Power Method* [20] can be applied here to find the exact solution to problem (7) through an iterative procedure with convergence to *global* solution guaranteed. The overall procedure is summarized in Algorithm 1.

Algorithm 1 Ensemble Pruning via Constrained Eigen-Optimization

Input: Matrix M recording the results produced by the ensemble members.

Calculate the matrix $\tilde{G}'$ according to (4)
Solve the optimization problem (7):
$H = \alpha I - \tilde{G}'$
$P = I - A^\top (AA^\top)^{-1} A$, $i = 0$
$\mathbf{n}_0 = A^\top (AA^\top)^{-1} \mathbf{b}$
$\gamma = \sqrt{1 - \|\mathbf{n}_0\|^2}$
$\mathbf{v}_0 = \gamma \frac{PH\mathbf{n}_0}{\|PH\mathbf{n}_0\|} + \mathbf{n}_0$
repeat
  $\mathbf{u}_{i+1} = \gamma \frac{PH\mathbf{v}_i}{\|PH\mathbf{v}_i\|}$
  $\mathbf{v}_{i+1} = \mathbf{u}_{i+1} + \mathbf{n}_0$
  $i = i + 1$
until $\mathbf{v}$ converges
return $\mathbf{v}$

The proposed algorithm is an efficient technique to solve the ensemble pruning problem. More importantly, the iterative approach is guaranteed to converge to a global optimum according to [20]. Once $\mathbf{v}$ is returned, one can apply different rounding schemes to $\mathbf{v}_{2:n}$ and recover the relaxed solution of $\mathbf{z}$ or $\mathbf{w}$, and the ensemble can be pruned according to

that. In this paper, we simply keep the k base learners with higher values in $\mathbf{v}_{2:n}$ to construct the pruned ensemble.

III. Related work

Before presenting the experimental results, we give a brief review of some related work on ensemble pruning. The methods that we are going to discuss here fall into the categories of ordering-based and optimization-based ensemble pruning.

A representative of the ordering based ensemble pruning methods is Orientation Ordering (OO) [15], where a signature vector is first defined for each classifier to indicate whether the classifier is correct on the training examples. A reference vector is then used to measure the direction towards which the signature vector of the ensemble should be modified to achieve a perfect classification performance on the training data. Classifiers are ordered by increasing angles of their signature vectors with the reference vector. Members of the pruned subensemble can then be selected according to this order.

Another ordering based approach is Ensemble Pruning via Individual Contribution (EPIC) [16] where a metric that considers both accuracy and diversity is defined to evaluate each member classifier's contribution, and ordering is based on decreasing individual contribution of each classifier.

In this paper, we are interested in ensemble pruning by solving an optimization problem. Unlike the heuristic approaches discussed above, a mathematical formulation is used to model the expected performance of the ensemble in a more principled way. Representatives include genetic algorithms that evolve voting weights of the member classifiers towards a higher value of fitness that approximates the generalization performance of the ensemble [17], [7]. Another example is a regularized selective ensemble algorithm that solves for the weights of the base classifiers to minimize a regularized risk function, which can be formulated as a quadratic program with an ℓ^1-norm constraint on the weight vector [8].

A more recent and related optimization-based approach reformulates the combinatorial optimization problem (2) in a different way:

$$\begin{aligned} \min_{\hat{\mathbf{z}}} \quad & \hat{\mathbf{z}}^\top \tilde{G}' \hat{\mathbf{z}} \\ \text{s.t.} \quad & \hat{\mathbf{z}} \in \{-1,+1\}^{m+1} \\ & \hat{\mathbf{z}}^\top D \hat{\mathbf{z}} = 4k \\ & \hat{z}_1 = 1 \end{aligned}$$

where $D = \begin{bmatrix} m & \mathbf{1}^\top \\ \mathbf{1} & I \end{bmatrix}$. Note the different quadratic reformulation of the ensemble size constraint: $\hat{\mathbf{z}}^\top D \hat{\mathbf{z}} = 4k$, this facilitates applying semidefinite relaxations and solving the problem approximately with a semidefinite program (SDP) [21]. More specifically, after substituting $Z = \hat{\mathbf{z}}\hat{\mathbf{z}}^\top$ and replacing the constraints with relaxations, the SDP formulation can be written as:

$$\begin{aligned} \min_{Z} \quad & \text{trace}(\tilde{G} Z) \\ \text{s.t.} \quad & \text{trace}(DZ) = 4k \\ & \text{diag}(Z) = \mathbf{1} \\ & Z \succeq 0 \end{aligned} \tag{8}$$

In general, this is a tighter relaxation than the proposed formulation (6). However, although the global solution to an SDP can be found in polynomial time, the complexity is $O(q^2 p^{2.5})$ [22] where p is the size of the matrix variable and q is the number of the constraints. Therefore, the complexity for solving the SDP relaxation of ensemble pruning is $O(m^{4.5})$, which is intensive for large problems. Memory is also a big issue. In practice, solving the problem with an ensemble of size 200 is usually computationally prohibitive for a standard workstation.

On the other hand, the relaxation proposed in this paper (6) can be solved with an iterative procedure with $O(m^2)$ complexity for each loop [20], and it converges quickly in practice. This implies a significant improvement in scalability.

Regarding the numerical performance, the relaxations made by the SDP formulation and our approach are different, which results in different numerical solutions. In the following section, we will design some experiments to compare the numerical performance between SDP and the proposed approach, including the quality of optimization as well as running time.

IV. Experimental results

In this section we conduct a series of experiments to compare the performance of different ensemble pruning methods discussed above. Data sets are taken from the UCI machine learning repository [23]. In the experiments we use bagging as the original ensemble considering its prediction performance, robustness, and little tuning required in general [24]. More specifically, the base learner we use in the experiments is J48, a Java implementation of C4.5 [25] in Weka [26]. We should note that although here we use bagging as the original ensemble, the proposed method is not limited to bagging, and can be applied to any ensemble methods.

A. Numerical Comparison

To investigate the numerical effects of different relaxations made to the discrete optimization problem (5), we randomly generate a problem of size 25, that is an ensemble of 25 base classifiers, and set the pruning ratio to 30%. We first find the optimal solution to problem (5) by enumerating all possible selections of the base classifiers with size 8, which implies $C_{25}^8 = 1,081,575$ possible subensembles, and choose the one with minimum objective value. We then run

Table I
NUMERICAL COMPARISON OF THE EXACT SOLUTION AND THE SDP/EIGCONS RELAXATIONS.

	Exact solution	SDP relaxation	Eigcons relaxation
Objective	4.28	4.26	4.08
Pruning mistakes	–	0.44 $\pm$0.50	0.96$\pm$0.57

the SDP algorithm and the proposed approach to compare with the optimal solution. SDP is implemented with the semidefinite programming package SeDuMi [27]. The procedure is repeated 50 times. Table I shows the results, where we compare the SDP algorithm and the constrained eigen-optimization approach (EigCons) in terms of the objective values, and the difference of the subensembles found by the two relaxed solutions to the optimal one.

According to optimization theory, the objective value of a relaxed minimization problem should be smaller than or equal to that of the original problem since the solution space is less constrained. This is clearly shown in Table I. Meanwhile, we can see that the objective value of the SDP algorithm is greater than that of the constrained eigen-optimization approach, which is not surprising since the SDP relaxation is tighter. As a consequence, when we compare the subensembles found by the two algorithms to the optimal one, we can see that the proposed constrained eigen-optimization method produces more pruning mistakes than the SDP algorithm, which demonstrates a tradeoff between precision and scalability. However, from the following experiments on real data sets, we will see that the difference in precision here is tolerable especially when we take the issue of computational efficiency into consideration.

B. Comparison with the SDP formulation on UCI data

We then further investigate the different behavior of the two optimization based pruning methods: the SDP formulation and the proposed algorithm (EigCons). Here, due to the high complexity of solving an SDP, we only use one data set for illustration. We take the "autos" data set from the UCI repository, randomly split it into a training set and a testing set, try different ensemble sizes (m values) and set pruning ratio to 30%, then run the two algorithms. This procedure is repeated 5 times and we take the average of the results to compare. Table II and Table III show the classification errors and the running time of the SDP approach and our proposed algorithm respectively.

Although the experiment is not repeatedly run on many different trials due to the computational expenses of SDP optimization, from Table II, we could roughly say that the performance of the constrained eigen-optimization technique is comparable to the SDP formulation given the different relaxations taken by the two approaches.

Table III summarizes the running time of the two algorithms with different ensemble sizes, where one could observe remarkable difference of the two algorithms in terms of computational efficiency. In fact, when the number of classifiers gets over 200, the SDP approach enters a bottleneck of memory space, therefore we are not able to get results with m values bigger than 150. Obviously in practice, the SDP approach is too expensive to be applied to real problems with large ensemble sizes.

Table II
COMPARISON ON THE PERCENTAGES OF CLASSIFICATION ERROR WITH DIFFERENT NUMBER OF CLASSIFIERS.

Number of classifiers	50	100	150
SDP	33.04	27.83	27.54
EigCons	33.33	28.41	24.64

Table III
NUMERICAL COMPARISON ON THE RUNNING TIME (SECONDS) WITH DIFFERENT NUMBER OF CLASSIFIERS.

Number of classifiers	50	100	150
SDP	40.3	2599.6	27819.8
EigCons	0.9	1.6	3.0

C. Comparison with the ordering based pruning methods

Here we mainly focus on comparing the proposed constrained eigen-optimization technique (EigCons) with the two ordering based pruning methods discussed above: orientation ordering (OO) and individual contribution ordering (EPIC). We use 38 UCI data sets summarized in Table IV and follow the experimental setting proposed by the EPIC paper [16]: first randomly divide each data set into three subsets with equal sizes, which include a training set, a pruning set and a testing set respectively. An ensemble is learned based on the training set, and a pruning set is used to evaluate the performance of the member classifiers, then ensemble pruning techniques are applied. After that, the performance of the pruned subensemble is evaluated on the testing set. There are six possible permutations of the three subsets, which implies six sets of sub-experiments. Each set of sub-experiments consists of 50 trials; in every trial, a bagging ensemble of 100 decision trees is trained and pruned, the subensemble is then evaluated. Overall on each data set, there are 300 repeats of experiments. In each repeat we change the random seed to ensure the generated bagging is not the same as previous.

Table IV
A BRIEF DESCRIPTION OF THE DATA SETS USED IN THE EXPERIMENTS.

Data set	Classes	Dimensions	Size
Anneal	6	38	898
Arrhythmia	16	279	452
Audiology	24	69	226
Auto-mpg	4	7	398
Autos	7	25	205
Balance Scale	3	4	625
Balloons	2	4	76
Breast-w	2	9	699
Bridges2	6	11	108
Clean1	2	166	476
Cmc	3	9	1473
Credit-g	2	20	1000
Dermatology	6	34	366
Flag	6	27	194
Glass	7	9	214
Hayes-roth	3	4	132
Heart-h	5	13	294
Hypothyroid	4	29	3772
Letter	26	16	20000
Lymph	4	18	148
Machine	8	7	209
Mfeat-factors	10	216	2000
Mfeat-fourier	10	76	2000
Mfeat-karhunen	10	64	2000
Mfeat-morphological	10	6	2000
Mfeat-pixel	10	240	2000
Mfeat-zernike	10	47	2000
Primary-tumor	22	17	339
Promoters	2	57	106
Segment	7	19	2310
Sonar	2	60	208
Soybean	19	35	683
Splice	3	61	3190
Tae	3	5	151
Tic-tac-toe	2	9	958
Vehicle	4	18	846
Vowel	11	13	990
Wine	3	13	178

Table V summarizes the results of different pruning algorithms and the original bagging ensembles. Means and standard deviations of the prediction errors over 300 trials of each experiment on each data set can be found in the table. Here we only report the results of the pruning algorithms with pruning ratio equal to 30% to be clear. Results of bagging ensembles are evaluated with the complete ensembles.

In Table V, the proposed constrained eigen-optimization technique with pruning ratio 30% is compared to the other pruning methods (OO, EPIC) with the same pruning ratio and the full-size bagging ensemble. Statistical significance of the comparison is evaluated. In Table V, the $\oplus$ sign denotes EigCons outperforms the comparing method at significance level of 95%, similarly $\odot$ implies comparable results, and $\ominus$ corresponds to the case that EigCons is outperformed by the competing technique with statistical significance.

From Table V we can observe that in most cases, the ensemble pruning methods outperform the full bagging ensemble, which justifies the motivation of ensemble pruning. Among all the pruning techniques, the proposed constrained eigen-optimization method is superior to the comparing methods most of the times.

Next, we investigate the influence of the pruning ratio on the performance of the pruned subensemble. Here, we use the same experimental setting as above, except that due to the concern of computational expenses, we reduce the number of trials in each sub-experiment to 30. We vary the sizes of subensembles from 1 to the size of the full ensemble, 100, and then plot the error curves of various algorithms with the increase of the number of decision trees included in the subensemble. On each data set we also plot the error curve of the bagging subensemble which is constructed by including decision trees incrementally from the original ensemble. We do not report the standard deviations here to be clear.

Fig. 1-2 summarize the results on 12 representative data sets: "Autos", "Arrhythmia", "Balance Scale", "Cmc", "Flag", "Glass", "Hayes-roth", "Primary-tumor", "Sonar", "Vehicle", "Vowel" and "Wine". From the curves we can first observe that as the number of classifiers increases, the error of a bagging ensemble generally decreases, and the decreasing rate gets smaller and smaller when the number of classifiers gets large. We can also note that in general all the pruning methods outperform the bagging algorithm (except for the comparable performance of OO and bagging on the "Flag" data set), which agrees with what is observed in Table V.

From Fig. 1-2 we can observe that on most of the data sets, the error curves of the constrained eigen technique are below those of the competing pruning methods including EPIC and OO. It can also be noted that on a few data sets, OO can produce small subensembles with relatively lower error rates. Specifically, on the data sets "Arrhythmia", "Autos", "Glass" and "Primary-tumor", the error curves produced by OO drop below the other curves when the number of classifiers is small (roughly less than 10), after that our constrained eigen technique starts to outperform OO as the size of the subensemble grows. On the "Cmc" data set, OO performs quite well with small subensembles, and is comparable to the constrained eigen technique when the number of classifiers increases. When comparing to EPIC, the proposed constrained eigen technique is consistently better on all the data sets.

Overall from the curves one could observe that generally the best performance of ensemble pruning is achieved with pruning ratio between 15% and 30%. Therefore, given an original ensemble, it is safe in general to set the pruning ratio between the values of 15% and 30%.

Table V
PERCENTAGES OF CLASSIFICATION ERROR OF THE FULL-SIZE BAGGING ENSEMBLE AND DIFFERENT ENSEMBLE PRUNING METHODS WITH PRUNING RATIO OF 30%. $\oplus$ DENOTES EIGCONS OUTPERFORMS THE COMPARING METHOD AT SIGNIFICANCE LEVEL OF 95%, $\odot$ IMPLIES COMPARABLE RESULTS, AND $\ominus$ REPRESENTS THAT EIGCONS IS OUTPERFORMED BY THE COMPETING TECHNIQUE WITH STATISTICAL SIGNIFICANCE.

	Bagging	Eigcons+30%	OO+30%	EPIC+30%
Anneal	2.78±0.97⊕	2.51±0.96	2.62±1.05⊙	2.7±1.03⊕
Arrhythmia	28.84±3.62⊕	27.58±3.43	27.77±3.84⊙	28.69±3.82⊕
Audiology	32.16±4.5⊕	30.85±5.2	32.34±6.07⊕	32.36±4.5⊕
Auto-mpg	29.2±4.32⊕	26.84±5.12	27.66±4.63⊕	28.39±4.33⊕
Autos	36.36±5.48⊕	31.92±4.21	34.02±5.23⊕	35.15±4.84⊕
Balance Scale	20.17±1.14⊕	18.20±1.41	18.45±1.39⊕	19.71±1.51⊕
Balloons	38.4±11.83⊕	29.87±8.20	31.48±9.69⊕	34.10±11.05⊕
Breast-w	4.3±1.18⊕	4.07±1.08	4.32±1.16⊕	4.38±1.02⊕
Bridges2	45.13±6.04⊕	40.69±6.29	42.24±6.27⊕	42.19±6.54⊕
Clean1	19.51±2.78⊕	16.64±2.49	16.9±2.58⊙	18.91±2.94⊕
Cmc	49.36±0.99⊕	48.97±1.30	48.78±1.19⊙	49.3±1.27⊕
Credit-g	26.42±1.83⊕	25.95±1.73	25.8±1.89⊙	26.47±1.74⊕
Dermatology	6.91±2.11⊕	3.83±1.18	4.87±1.34⊕	5.28±2.11⊕
Flag	43.80±4.67⊕	41.58±3.88	44.3±4.23⊕	44.17±4.6⊕
Glass	38.12±6.32⊕	36.0±5.73	37.22±7.41⊕	37.64±6.54⊕
Hayes-roth	30.08±8.81⊕	25.90±5.15	27.15±6.27⊕	27.06±5.87⊕
Heart-h	20.27±2.33⊕	19.0±2.34	19.59±2.75⊕	19.46±2.43⊕
Hypothyroid	0.54±0.16⊕	0.48±0.15	0.46±0.12⊙	0.5±0.14⊕
Letter	10.21±0.41⊕	9.84±0.31	9.97±0.31⊕	10.81±0.43⊕
Lymph	21.10±3.83⊙	20.71±4.05	22.61±3.47⊕	22.32±3.52⊕
Machine	18.25±4.76⊕	15.50±4.11	15.82±3.82⊙	16.55±4.24⊕
Mfeat-factors	7.89±1.8⊕	7.06±1.6	7.33±1.64⊕	8.22±1.76⊕
Mfeat-fourier	21.87±2.05⊕	21.41±2.09	21.62±2.0⊙	22.2±2.03⊕
Mfeat-karhunen	12.46±1.49⊕	11.4±1.44	11.81±1.43⊕	13.12±1.51⊕
Mfeat-morphological	28.25±0.93⊕	28.04±0.92	27.9±0.95⊙	28.33±1.05⊕
Mfeat-pixel	21.41±4.29⊕	19.04±4.19	20.1±4.27⊕	21.78±3.91⊕
Mfeat-zernike	25.35±1.41⊙	25.14±1.43	25.46±1.47⊕	25.82±1.41⊕
Primary-tumor	64.23±3.18⊕	62.71±2.87	63.0±3.28⊙	64.03±3.03⊕
Promoters	26.50±5.88⊕	17.21±6.61	21.32±7.03⊕	23.82±6.36⊕
Segment	4.60±0.58⊕	4.06±0.6	4.05±0.55⊙	4.5±0.63⊕
Sonar	25.51±2.95⊕	24.11±3.58	24.44±4.29⊙	25.03±3.89⊕
Soybean	13.34±2.61⊕	11.32±2.62	11.63±2.59⊙	12.53±2.35⊕
Splice	7.7±1.1⊕	7.3±0.88	7.25±0.87⊙	7.65±1.01⊕
Tae	55.02±6.02⊕	53.02±6.4	55.41±7.62⊕	54.70±6.6⊕
Tic-tac-toe	18.95±2.09⊕	15.71±2.42	16.6±2.22⊕	18.31±2.29⊕
Vehicle	28.58±3.08⊕	27.91±2.88	28.13±2.86⊙	28.83±3.14⊕
Vowel	22.69±3.16⊕	21.16±2.92	22.41±2.95⊕	23.77±3.18⊕
Wine	13.44±4.03⊕	9.15±4.07	11.4±3.34⊕	12.21±3.46⊕
win/tie/loss	36/2/0		23/15/0	38/0/0

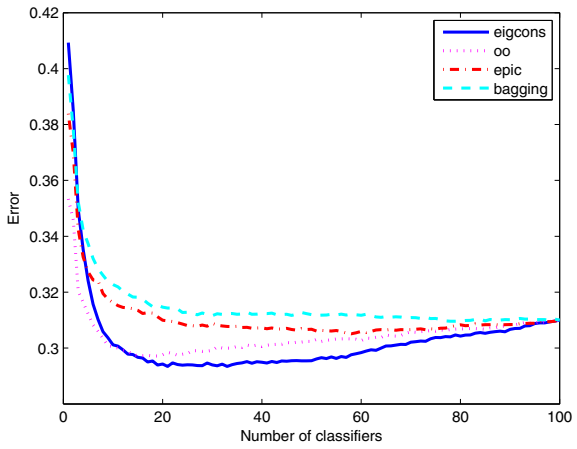
V. CONCLUSION

This paper presents a novel and efficient algorithm for pruning an ensemble to achieve comparable or even better prediction performance with a smaller number of base classifiers. We formulate the task as an optimization problem, and the pruning criterion for our technique is to maximize the individual accuracy and the pairwise diversity of the subensemble members, which are two important factors that influence the performance of an ensemble. We derive a novel relaxation of the original integer programming into a constrained eigen-optimization problem, which can be solved efficiently with an iterative algorithm with global convergence guarantee. This approach can be applied to general ensemble techniques. Experimental results demonstrate the effectiveness of the proposed method.

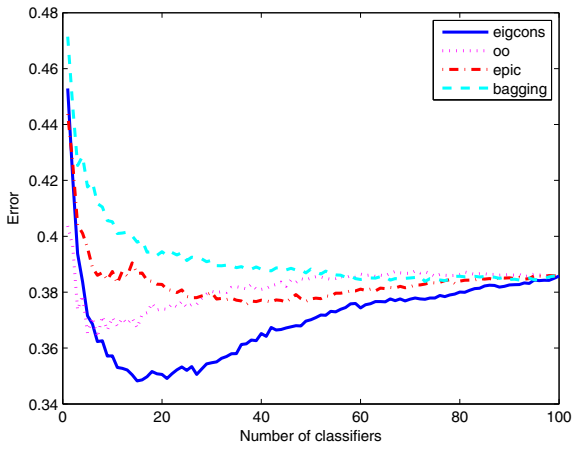
This paper mainly focuses on supervised learning on labeled data. A possible direction for further investigation is to augment the proposed approach by exploiting unlabeled data to help selecting the classifiers. Automatic selection of the pruning ratio is also an interesting research topic. Moreover, applying the pruning technique on data with more complex structure is another direction to pursue.

ACKNOWLEDGMENT

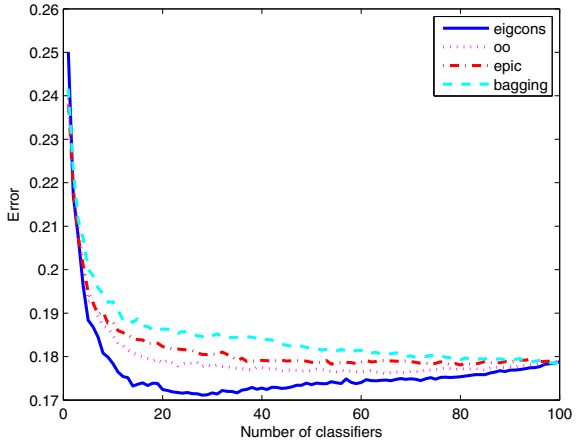
Research supported by the National Natural Science Foundation of China (No. 61003135, 60775037) and NSFC Major Program (No. 71090401/71090400)



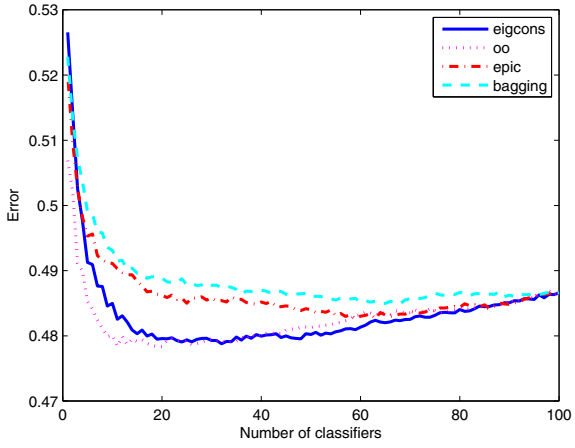
(a) Arrhythmia



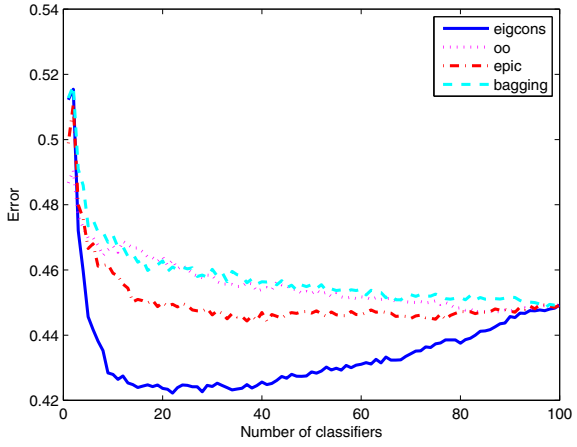
(b) Autos



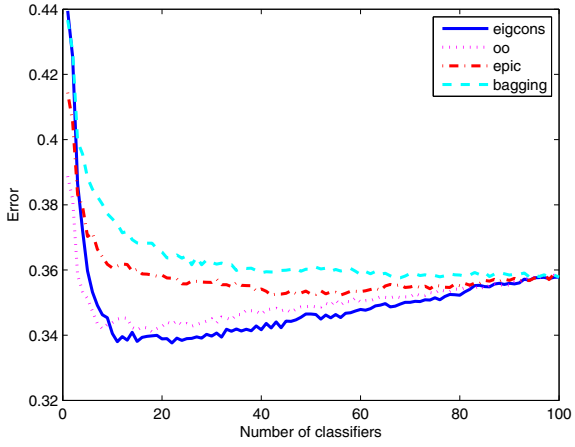
(c) Balance Scale



(d) Cmc

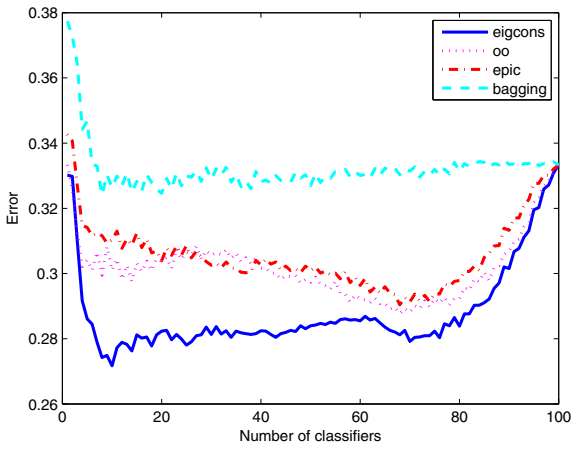


(e) Flag

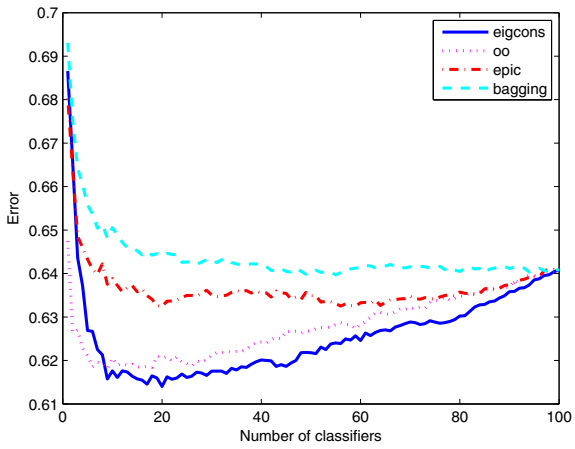


(f) Glass

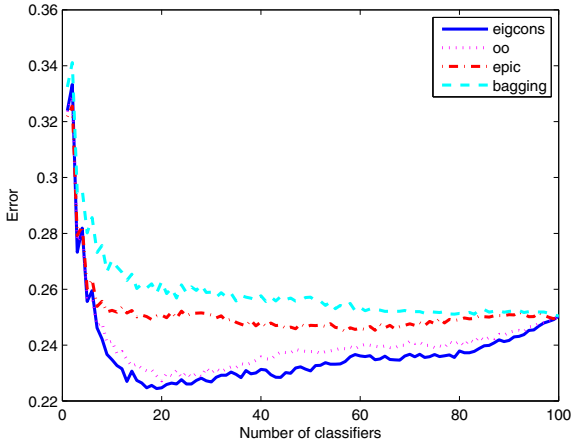
Figure 1. Comparison of prediction errors on data sets “Arrhythmia”, “Autos”, “Balance Scale”, “Cmc”, “Flag” and “Glass”.



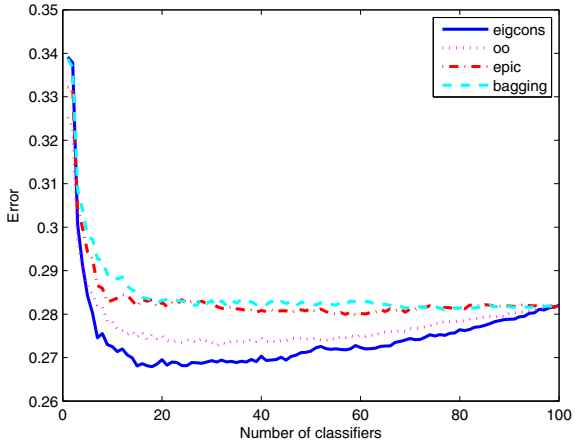
(a) Hayes-roth



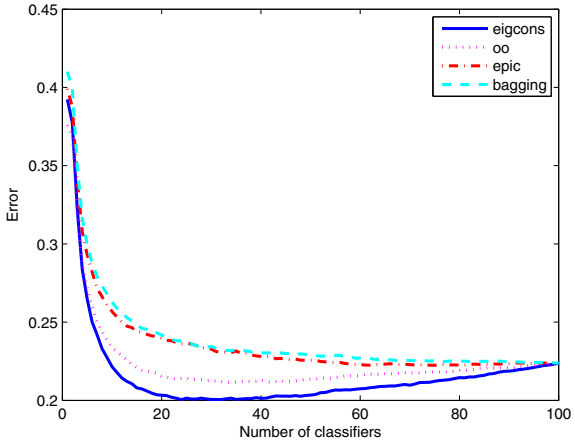
(b) Primary-tumor



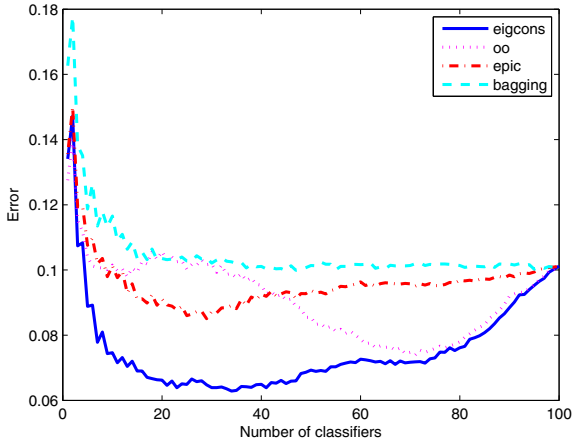
(c) Sonar



(d) Vehicle



(e) Vowel



(f) Wine

Figure 2. Comparison of prediction errors on data sets “Hayes-roth”, “Primary-tumor”, “Sonar”, “Vehicle”, “Vowel” and “Wine”.

REFERENCES

[1] A. Krogh and J. Vedelsby, "Neural network ensembles, cross validation, and active learning," in *Advances in Neural Information Processing Systems*, 1995.

[2] L. Breiman, "Bagging predictors," *Machine Learning*, 1996.

[3] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, 1997.

[4] L. Breiman, "Random forests," *Machine Learning*, 2001.

[5] D. D. Margineantu and T. G. Dietterich, "Pruning adaptive boosting," in *Proceedings of the 14th International Conference on Machine Learning*, 1997.

[6] A. Prodromidis and S. Stolfo, "Cost complexity-based pruning of ensemble classifiers," *Knowledge and Information Systems*, vol. 3, 2001.

[7] Z.-H. Zhou and W. Tang, "Selective ensemble of decision trees," in *Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing*, ser. Lecture Notes in Computer Science, 2003, vol. 2639.

[8] N. Li and Z.-H. Zhou, "Selective ensemble under regularization framework," in *Proceedings of the 8th International Workshop on Multiple Classifier Systems*, ser. Lecture Notes in Computer Science, 2009.

[9] C. Tamon and J. Xiang, "On the boosting pruning problem," in *Proceedings of the 11th European Conference on Machine Learning*, 2000.

[10] 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*, 2004.

[11] B. Bakker and T. Heskes, "Clustering ensembles of neural network models," *Neural Networks*, vol. 16, 2003.

[12] G. Giacinto, F. Roli, and G. Fumera, "Design of effective multiple classifier systems by clustering of classifiers," in *Proceedings of the 15th International Conference on pattern recognition*, 2000.

[13] G. Martínez-Muñoz, D. Hernández-Lobato, and A. Suárez, "An analysis of ensemble pruning techniques based on ordered aggregation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, 2009.

[14] G. Martínez-Muñoz and A. Suárez, "Aggregation ordering in bagging," in *Proceedings of the IASTED International Conference on Artificial Intelligence and Applications*, 2004.

[15] G. Martínez-Muñoz and A. Suárez, "Pruning in ordered bagging ensembles," in *Proceedings of the 23rd International Conference on Machine Learning*, 2006.

[16] Z. Lu, X. Wu, X. Zhu, and J. Bongard, "Ensemble pruning via individual contribution ordering," in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2010.

[17] Z.-H. Zhou, J. Wu, and W. Tang, "Ensembling neural networks: Many could be better than all," *Artificial Intelligence*, vol. 137, 2002.

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

[19] T. G. Dietterich, "Ensemble methods in machine learning," in *Proceedings of the 1st International Workshop on Multiple Classifier Systems*, 2000.

[20] L. Xu, W. Li, and D. Schuurmans, "Fast normalized cut with linear constraints," in *Proceedings of the 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2009.

[21] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge U. Press, 2004.

[22] Y. Nesterov and A. Nemirovskii, "Interior-point polynomial algorithms in convex programming," *SIAM*, 1994.

[23] A. Frank and A. Asuncion, "UCI machine learning repository," 2010. [Online]. Available: http://archive.ics.uci.edu/ml

[24] R. Caruana and A. Niculescu-Mizil, "An empirical comparison of supervised learning algorithms," in *Proceedings of the 23rd international conference on Machine learning*, 2006.

[25] J. R. Quinlan, *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., 1993.

[26] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *SIGKDD Explorations Newsletter*, vol. 11, no. 1, 2009.

[27] J. Sturm, "Using SeDuMi 1.02, a Matlab toolbox for optimization over symmetric cones." *Optimization Methods and Software*, vol. 11-12, 1999.