Pareto Ensemble Pruning*

Chao Qian and **Yang Yu** and **Zhi-Hua Zhou**
National Key Laboratory for Novel Software Technology, Nanjing University
Collaborative Innovation Center of Novel Software Technology and Industrialization
Nanjing 210023, China
{qianc,yuy,zhouzh}@lamda.nju.edu.cn

Abstract

Ensemble learning is among the state-of-the-art learning techniques, which trains and combines many base learners. Ensemble pruning removes some of the base learners of an ensemble, and has been shown to be able to further improve the generalization performance. However, the two goals of ensemble pruning, i.e., maximizing the generalization performance and minimizing the number of base learners, can conflict when being pushed to the limit. Most previous ensemble pruning approaches solve objectives that mix the two goals. In this paper, motivated by the recent theoretical advance of evolutionary optimization, we investigate solving the two goals explicitly in a bi-objective formulation and propose the PEP (Pareto Ensemble Pruning) approach. We disclose that PEP does not only achieve significantly better performance than the state-of-the-art approaches, and also gains theoretical support.

Introduction

Ensemble methods (Zhou 2012) are a kind of powerful machine learning approaches, which train and combine multiple base learners for one single learning task. They usually achieve the state-of-the-art prediction performance, and thus have been widely applied. Instead of combining all the trained base learners of an ensemble, ensemble pruning (Tsoumakas, Partalas, and Vlahavas 2009) selects only a subset of base learners to use. Obviously, reducing the compositing base learners can save the storage space and accelerate the prediction speed. Furthermore, it has been shown that the pruned ensemble can have a better generalization performance than the whole ensemble (Zhou, Wu, and Tang 2002; Zhang, Burer, and Street 2006).

Previous ensemble pruning techniques can be categorized into two branches, the ordering-based pruning and the optimization-based pruning. The ordering methods (Martínez-Muñoz, Hernández-Lobato, and Suárez 2009; Partalas, Tsoumakas, and Vlahavas 2012) commonly start from an empty set and then iteratively add a base learner optimizing a certain objective. The sequence of being added into the pruned ensemble gives an order of the base classifiers, the front classifiers of which constitute the final ensemble. Different ordering methods are mainly different on the choice of the objective, which can be minimizing the error (Margineantu and Dietterich 1997), maximizing the diversity (Banfield et al. 2005), or combining the both (Li, Yu, and Zhou 2012). Many studies have shown that they can achieve a good pruned ensemble efficiently (Martínez-Muñoz, Hernández-Lobato, and Suárez 2009; Hernández-Lobato, Martínez-Muñoz, and Suárez 2011). The optimization-based pruning formulates the ensemble pruning as an optimization problem which aims at finding a subset of base learners with the best generalization performance. Different optimization techniques have been employed, e.g., semi-definite programming (Zhang, Burer, and Street 2006), quadratic programming (Li and Zhou 2009) and heuristic optimization such as genetic algorithms (Zhou, Wu, and Tang 2002) and artificial immune algorithms (Castro et al. 2005). The heuristic methods use some trial-and-error style heuristics to directly search in the solution space. They were believed to be powerful, but their performance had no theoretical support. Moreover, empirical results have shown that the size of the pruned ensemble by heuristic methods is often much larger than that by ordering methods (Zhou, Wu, and Tang 2002; Li and Zhou 2009).

Ensemble pruning naturally bears two goals simultaneously, maximizing the generalization performance and minimizing the number of learners. When pushing to the limit, the two goals are conflicting, as overly fewer base learners lead to a poor performance. In order to achieve both a good performance and a small ensemble size, previous ensemble pruning approaches solve some objectives that mix the two goals. But recently, it has been revealed that, when dealing with multiple objectives via evolutionary optimization, explicit consideration of every goal in a multi-objective formulation can be quite helpful (Yu, Yao, and Zhou 2012).

This paper investigates the explicit bi-objective formulation of ensemble pruning, and proposes the PEP (Pareto Ensemble Pruning) approach. PEP solves the bi-objective formulation of ensemble pruning by an evolutionary Pareto optimization method combined with a local search operator. Firstly, we show theoretically that PEP is superior to the ordering methods in both performance and size. Then, the em-

*This research was supported by the NSFC (61333014, 61375061), JiangsuSF (BK2012303), FRF Central Universities (20620140519) and Baidu Fund (181415PO2189).

pirical studies support that PEP is significantly better than the state-of-the-art approaches. Finally, we apply PEP in the application on mobile human activity recognition.

The rest of the paper presents the PEP approach, the theoretical studies, the empirical studies, the application, and the conclusion in subsequent sections.

The PEP Approach

Given a data set $D = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^m$ and a set of n trained base classifiers $H = \{h_i\}_{i=1}^n$, where $h_i : \mathcal{X} \to \mathcal{Y}$ maps the feature space $\mathcal{X}$ to the label space $\mathcal{Y}$, let $H_{\boldsymbol{s}}$ denote a pruned ensemble with the selector vector $\boldsymbol{s} \in \{0,1\}^n$, where $s_i = 1$ means that the base classifier h_i is selected. The ensemble pruning simultaneously optimizes some performance measure f related to the generalization error of $H_{\boldsymbol{s}}$ and minimizes the size of $H_{\boldsymbol{s}}$ which is simply counted as $|\boldsymbol{s}| = \sum_{i=1}^n s_i$.

Instead of optimizing a mixture objective of these two goals, we investigate directly solving the bi-objective ensemble pruning problem, which is formulated as

$$\arg\min_{\boldsymbol{s}\in\{0,1\}^n} (f(H_{\boldsymbol{s}}), |\boldsymbol{s}|).$$

In the bi-objective formulation, the objective function, i.e., $(f(H_{\boldsymbol{s}}), |\boldsymbol{s}|)$, gives to any candidate solution $\boldsymbol{s}$ not a scalar value but a vector. For example, a pruning solution which results in 0.2 value of the employed f function and 10 classifiers will have the objective vector $(0.2, 10)$. Unlike single-objective optimization, the objective vector makes the comparison between two solutions not straightforward, because it is possible that one solution is better on the first dimension while the other is better on the second dimension. Therefore, the domination relationship is usually used for this special situation. Definition 1 introduces the domination relationship in bi-objective (i.e., two objectives) minimization case.

Definition 1 (Domination)**.** *Let $\boldsymbol{g} = (g_1, g_2) : \mathcal{S} \to \mathbb{R}^2$ be the objective vector. For two solutions $\boldsymbol{s}, \boldsymbol{s}' \in \mathcal{S}$:*
(1) $\boldsymbol{s}$ weakly dominates *$\boldsymbol{s}'$ if $g_1(\boldsymbol{s}) \leq g_1(\boldsymbol{s}')$ and $g_2(\boldsymbol{s}) \leq g_2(\boldsymbol{s}')$, denoted as $\succeq_{\boldsymbol{g}}$;*
(2) $\boldsymbol{s}$ dominates *$\boldsymbol{s}'$ if $\boldsymbol{s} \succeq_{\boldsymbol{g}} \boldsymbol{s}'$ and either $g_1(\boldsymbol{s}) < g_1(\boldsymbol{s}')$ or $g_2(\boldsymbol{s}) < g_2(\boldsymbol{s}')$, denoted as $\succ_{\boldsymbol{g}}$.*

Consequently, a bi-objective optimization problem may not have a single optimal solution, but instead have a set of *Pareto optimal* solutions. A solution $\boldsymbol{s}$ is Pareto optimal if there is no other solution in $\mathcal{S}$ that dominates $\boldsymbol{s}$.

We propose the PEP (Pareto Ensemble Pruning) approach as presented in Algorithm 1 to solve the bi-objective ensemble pruning problem. PEP is inspired by a multi-objective evolutionary algorithm, which has been shown to be among the best-so-far algorithms for approximating some NP-hard problems (Yu, Yao, and Zhou 2012). It firstly generates a random solution, and puts it into the candidate solution set P; and then follows a cycle to improve the solutions in P iteratively. In each iteration, a solution $\boldsymbol{s}$ is randomly selected from P; and is then perturbed to generate a new solution $\boldsymbol{s}'$; if $\boldsymbol{s}'$ is not dominated by any solution in P, $\boldsymbol{s}'$ is added into P and at the same time solutions in P that are weakly dominated by $\boldsymbol{s}'$ get removed.

As PEP is inspired by evolutionary algorithms, which usually focus on global exploration but may not utilize the local information well, a local search is incorporated into PEP to improve the quality of the new candidate solution in order to improve its efficiency. We employ the VDS (Lin and Kernighan 1973), short for the variable-depth search as described in Algorithm 2. It is known for the TSP problem as the Lin-Kernighan strategy. The VDS performs a sequence of greedy local moves, each of which chooses the best local solution. To prevent loops in the moves, it keeps a set L to record the moved directions. We apply the VDS to the newly included solution $\boldsymbol{s}'$ (i.e., step 8 of Algorithm 1), and the generated solutions are then used to update P.

Algorithm 1 (PEP)**.** *Given a set of trained classifiers $H = \{h_i\}_{i=1}^n$, an objective $f : 2^H \to \mathbb{R}$ and an evaluation criterion $eval$, it contains:*

1. *Let $\boldsymbol{g}(\boldsymbol{s}) = (f(H_{\boldsymbol{s}}), |\boldsymbol{s}|)$ be the bi-objective.*
2. *Let $\boldsymbol{s}$ = randomly selected from $\{0,1\}^n$ and $P = \{\boldsymbol{s}\}$.*
3. ***Repeat***
4. *Select $\boldsymbol{s} \in P$ uniformly at random.*
5. *Generate $\boldsymbol{s}'$ by flipping each bit of $\boldsymbol{s}$ with prob. $\frac{1}{n}$.*
6. ***if** $\nexists \boldsymbol{z} \in P$ such that $\boldsymbol{z} \succ_{\boldsymbol{g}} \boldsymbol{s}'$*
7. *$P = (P - \{\boldsymbol{z} \in P \mid \boldsymbol{s}' \succeq_{\boldsymbol{g}} \boldsymbol{z}\}) \cup \{\boldsymbol{s}'\}$.*
8. *$Q = VDS(f, \boldsymbol{s}')$.*
9. ***for** $\boldsymbol{q} \in Q$*
10. ***if** $\nexists \boldsymbol{z} \in P$ such that $\boldsymbol{z} \succ_{\boldsymbol{g}} \boldsymbol{q}$*
11. *$P = (P - \{\boldsymbol{z} \in P \mid \boldsymbol{q} \succeq_{\boldsymbol{g}} \boldsymbol{z}\}) \cup \{\boldsymbol{q}\}$.*
12. ***Output** $\arg\min_{\boldsymbol{s}\in P} eval(\boldsymbol{s})$.*

Algorithm 2 (VDS Subroutine)**.** *Given a pseudo-Boolean function f and a solution $\boldsymbol{s}$, it contains:*

1. *$Q = \emptyset, L = \emptyset$.*
2. *Let $N(\cdot)$ denote the set of neighbor solutions of a binary vector with Hamming distance 1.*
3. ***While** $V_{\boldsymbol{s}} = \{\boldsymbol{y} \in N(\boldsymbol{s}) \mid (y_i \neq s_i \Rightarrow i \notin L)\} \neq \emptyset$*
4. *Choose $\boldsymbol{y} \in V_{\boldsymbol{s}}$ with the minimal f value.*
5. *$Q = Q \cup \{\boldsymbol{y}\}$.*
6. *$L = L \cup \{i \mid y_i \neq s_i\}$.*
7. *$\boldsymbol{s} = \boldsymbol{y}$.*
8. ***Output** Q.*

It is noticeable that the VDS subroutine only considers the objective f but not the other objective. One could think if this will bias the search toward one objective. Since the selection (i.e., steps 6-7 and 9-11) determines the search direction, the VDS is used only to generate more potential candidate solutions for the selection in PEP, but will not effect the search direction.

After a set of Pareto optimal solutions to the bi-objective formulation of ensemble pruning has been solved, we can select out one final solution according to our preference, which is done through optimizing some evaluation criterion at the last step of Algorithm 1.

On the choice of the performance measure f, since the generalization performance is hard to be measured directly, an alternative way is to use the error directly on the training set or on a validation data set (Margineantu and Dietterich 1997; Zhou, Wu, and Tang 2002; Caruana et al. 2004). Other criteria have also been introduced to guide the pruning, such as the diversity measures. A representative is the κ-statistics

used in the Kappa pruning method (Banfield et al. 2005), which calculates the difference of two classifiers from their disagreements on a data set. Several diversity measures have been proposed (Brown et al. 2005) and have been shown to be connected to the generalization performance (Li, Yu, and Zhou 2012). A measure combining the data error and the diversity measure has been shown to lead to the state-of-the-art performance (Li, Yu, and Zhou 2012).

When selecting the final solution out of the Pareto optimal set, the choice of the $eval$ criterion could be application dependent. When the model size is sensitive, the selection should be among only those with small number of classifiers; otherwise it should lean to the performance measure.

Theoretical Analysis

This section investigates the effectiveness of PEP by comparing it theoretically with the ordering-based ensemble pruning methods, briefly called OEP. Algorithm 3 presents a common structure of OEP, where the objective f is used to guide the search and an evaluation criterion (usually using the validation error) is employed to select the final ensemble.

Algorithm 3 (OEP). *Given trained classifiers $H=\{h_i\}_{i=1}^n$, an objective $f:2^H\to\mathbb{R}$ and a criterion $eval$, it contains:*

1. *Let $H^S=\emptyset$, $H^U=\{h_1,h_2,\ldots,h_n\}$.*
2. ***Repeat until** $H^U=\emptyset$:*
3. $\quad h^*=\arg\min_{h\in H^U} f(H^S\cup\{h\})$.
4. $\quad H^S=H^S\cup\{h^*\}$, $H^U=H^U-\{h^*\}$.
5. *Let $H^S=\{h_1^*,\ldots,h_n^*\}$, where h_i^* is the classifier added in the i-th iteration.*
6. *Let $k=\arg\min_{1\le i\le n} eval(\{h_1^*,\ldots,h_i^*\})$.*
7. ***Output** $\{h_1^*,h_2^*,\ldots,h_k^*\}$.*

In the following subsections, we firstly show that, for any pruning instance, PEP can produce a solution at least as good as that by OEP. We then show that PEP is strictly better than OEP on some cases. Furthermore, we show that traditional heuristic single-objective optimization based pruning methods (briefly called SEP) can be much worse than PEP/OEP.

PEP Can Do All of OEP

We prove in Theorem 1 that for any pruning task, PEP can efficiently produce at least an equally good solution as that by OEP in both the performance and the size. The optimization time is counted as the number of pruned ensemble evaluations, which is often the most time consuming step.

Theorem 1. *For any objective and any size, PEP within $O(n^4\log n)$ expected optimization time can find a solution weakly dominating that generated by OEP at the fixed size.*

The proof idea is that, first PEP can find the special solution $\{0\}^n$ (i.e., none of the classifiers is selected) efficiently; then PEP can apply VDS on $\{0\}^n$ to follow the process of OEP; and thus PEP can produce a solution at least as good as that by OEP. Lemma 1 bounds the time for finding the special solution $\{0\}^n$. Let $P_{\max}$ denote the largest size of P in PEP during the optimization.

Lemma 1. *The expected iterations of PEP for finding $\{0\}^n$ is $O(P_{\max}n\log n)$.*

Proof. Let $i=\min\{|\boldsymbol{s}|\mid \boldsymbol{s}\in P\}$, i.e., the minimal number of 1 bits for solutions in P. It is easy to see that i cannot increase because a solution with more 1 bits cannot weakly dominate a solution $\boldsymbol{s}$ with $|\boldsymbol{s}|=i$. Once a solution $\boldsymbol{s}'$ with $|\boldsymbol{s}'|<i$ is generated, it will always be accepted, because it is better on the size objective and no other solution in P can dominate it. Thus, i can decrease by 1 in one iteration with probability at least $\frac{1}{P_{\max}}\cdot\frac{i}{n}(1-\frac{1}{n})^{n-1}$, because it is sufficient to select a solution $\boldsymbol{s}$ with $|\boldsymbol{s}|=i$ from P, whose probability is at least $\frac{1}{P_{\max}}$, and then flip just one 1 bit, whose probability is $\frac{i}{n}(1-\frac{1}{n})^{n-1}$. Then, the expected iterations $E[i]$ for decreasing i by 1 is at most $P_{\max}\frac{n}{i}\frac{1}{(1-\frac{1}{n})^{n-1}}\le\frac{eP_{\max}n}{i}$, where the inequality is by $(1-\frac{1}{n})^{n-1}\ge\frac{1}{e}$. By summing up $E[i]$, an upper bound $\sum_{i=1}^n E[i]\le\sum_{i=1}^n eP_{\max}n/i\in O(P_{\max}n\log n)$ on the expected iterations for finding $\{0\}^n$ is derived. □

Proof of Theorem 1. We consider that the solution $\{0\}^n$ is generated in step 5 of Algorithm 1. As $\{0\}^n$ is Pareto optimal, it will go into the VDS process, which actually follows the process of OEP, since OEP starts from the empty set (i.e., $\{0\}^n$), and iteratively adds one classifier (i.e., changes one bit from 0 to 1) minimizing f. Denote $\boldsymbol{s}^*$ as the solution generated by OEP. The set of solutions Q output by VDS thus must contain $\boldsymbol{s}^*$. Then, $\boldsymbol{s}^*$ will be used to update P as in steps 9-11 of Algorithm 1; this will make P always contain a solution weakly dominating $\boldsymbol{s}^*$.

Thus, we only need to analyze the expected iterations until $\{0\}^n$ is generated by step 5. We consider two cases. If the initial solution is $\{0\}^n$ which will never be removed as it is Pareto optimal, $\{0\}^n$ can be regenerated by step 5 with probability at least $\frac{1}{P_{\max}}(1-\frac{1}{n})^n$, since it is sufficient to select $\{0\}^n$ from P, whose probability is at least $\frac{1}{P_{\max}}$, and then flip no bits, whose probability is $(1-\frac{1}{n})^n$. Thus, in this case the expected iterations is at most $P_{\max}/(1-\frac{1}{n})^n\le 2eP_{\max}$. Otherwise, the expected iterations is $O(P_{\max}n\log n)$ by Lemma 1. Since any two solutions in P are incomparable, there exists at most one corresponding solution in P for each possible size. Thus, $P_{\max}\le n+1$ since the size $|\boldsymbol{s}|\in\{0,1,\ldots,n\}$. Then, the expected iterations for generating $\{0\}^n$ is $O(n^2\log n)$.

For each iteration of PEP, the optimization time is at most 1 (i.e., evaluating $\boldsymbol{s}'$) + that of VDS. The VDS takes n local moves since after every local move one index is added to L, and each local move performs at most n evaluations since $|V_{\boldsymbol{s}}|\le n$. Thus, the optimization time of one iteration is $O(n^2)$, which implies that the expected time for finding a solution weakly dominating $\boldsymbol{s}^*$ is $O(n^4\log n)$. □

PEP Can Do Better Than OEP

We consider binary classification (i.e., $\mathcal{Y}=\{-1,1\}$), voting for combining base classifiers, and taking the validation data set error as the performance objective f and also the evaluation criterion. The validation error is calculated as

$$f(H_{\boldsymbol{s}})=\frac{1}{m}\sum_{i=1}^m I(H_{\boldsymbol{s}}(\boldsymbol{x}_i)\ne y_i), \tag{1}$$

where $I(\cdot)$ is the indicator function that is 1 if the inner expression is true and 0 otherwise. Let $f(H_{\boldsymbol{s}=\{0\}^n}) = +\infty$ to ensure that at least one classifier will be selected.

Theorem 2 shows that, for the type of pruning tasks described in Situation 1, PEP can find the optimal pruned ensemble within polynomial time, while OEP only finds a suboptimal one with larger error, or larger size, or both.

In this setting, a pruned ensemble $H_{\boldsymbol{s}}$ is composited as

$$H_{\boldsymbol{s}}(\boldsymbol{x}) = \arg\max_{y\in\mathcal{Y}} \sum_{i=1}^{n} s_i \cdot I(h_i(\boldsymbol{x}) = y).$$

We define the difference of two classifiers as

$$\text{diff}(h_i, h_j) = \frac{1}{m}\sum_{k=1}^{m} \big(1 - h_i(\boldsymbol{x}_k)h_j(\boldsymbol{x}_k)\big)/2,$$

and the error of one classifier as

$$\text{err}(h_i) = \frac{1}{m}\sum_{k=1}^{m} \big(1 - h_i(\boldsymbol{x}_k)y_k\big)/2.$$

Both of them $\in [0, 1]$. If $\text{diff}(h_i, h_j) = 1$ (or 0), h_i and h_j always make the opposite (or same) prediction; if $\text{err}(h_i) = 1$ (or 0), h_i always makes the wrong (or right) prediction.

In situation 1, the optimal pruned ensemble consists of 3 base classifiers (i.e., H'): each makes different mistakes, and the combination leads to zero error. The proof of Theorem 2 is mainly that OEP will first select the base classifier h^* with the smallest error due to the greedy nature and will be misled by it, while PEP can first efficiently find the pruned ensemble $\{h^*\}$, and then applying VDS on it can produce the optimal pruned ensemble H' with a large probability.

Situation 1.

$$\exists H'\subseteq H, |H'|=3 \wedge \forall g,h\in H', \textit{diff}(g,h)=\textit{err}(g)+\textit{err}(h);$$

$$\exists h^*\in H-H', \begin{cases}\textit{err}(h^*) < \min\{\textit{err}(h)|h\in H'\},\\ \forall h\in H', \textit{diff}(h,h^*)<\textit{err}(h)+\textit{err}(h^*);\end{cases}$$

$$\forall g\in H-H'-\{h^*\}, \textit{err}(g)>\max\{\textit{err}(h)|h\in H'\}$$
$$\wedge\, \textit{err}(g)+\textit{err}(h^*)-\textit{diff}(g,h^*) >$$
$$(\min{+}\max)\{\textit{err}(h)+\textit{err}(h^*)-\textit{diff}(h,h^*)|h\in H'\}.$$

Theorem 2. *In Situation 1, OEP using Eq.1 finds a solution with objective vector* $(\geq 0, \geq 3)$ *where the two equalities never hold simultaneously, while PEP finds a solution with objective vector* $(0, 3)$ *in* $O(n^4 \log n)$ *expected time.*

Proof. Without loss of generality, assume $h^* = h_1$, $H' = \{h_2, h_3, h_4\}$ and $\text{err}(h_2) = \min\{\text{err}(h)|h \in H'\}$. Let $d_i = (\text{err}(h_1)+\text{err}(h_i)-\text{diff}(h_1, h_i))/2$, i.e., the ratio of the same mistakes made by h_1 and h_i, and then $d_i > 0$. Assume $d_3 \leq d_4$. Denote $\boldsymbol{B}_j \in \{0,1\}^j$ as a Boolean vector of length j.

For OEP, it follows such an optimization path:

$$\{0\}^n \to 1\{0\}^{n-1} \to 11\{0\}^{n-2} \to 111\{0\}^{n-3} \to 1111\{0\}^{n-4} \to \cdots.$$

The corresponding f value changes as: $+\infty \to \text{err}(h_1) \to$

$$(\text{err}(h_1)+\text{err}(h_2))/2 \to d_2+d_3 \to (d_2+d_3+d_4)/2 \to \geq 0.$$

The 1st '$\to$' is because h_1 has the smallest error; the 2nd is because the error of combining two classifiers is their average error and thus h_2 with the smallest error in the remaining classifiers is selected; the 3rd is by $\text{err}(111\{0\}^{n-3}) = d_2+d_3 \leq \text{err}(1101\{0\}^{n-4}) = d_2+d_4$ and $\forall s \in \{1100\boldsymbol{B}_{n-4} \mid |\boldsymbol{B}_{n-4}| = 1\}$, $\text{err}(s) > (\min{+}\max)\{d_2, d_3, d_4\} \geq d_2+d_3$; the 4th is by $\forall s \in \{1110\boldsymbol{B}_{n-4} \mid |\boldsymbol{B}_{n-4}| = 1\}$, $\text{err}(s) > (d_2+d_3+(\min{+}\max)\{d_2, d_3, d_4\})/2 > (d_2+d_3+d_4)/2 = \text{err}(11110^{n-4})$. Since $(\text{err}(h_1) + \text{err}(h_2))/2 > \text{err}(h_1) > d_2 + d_3$, OEP will output a solution with objective vector $(d_2+d_3, 3)$, $((d_2+d_3+d_4)/2, 4)$ or $(\geq 0, \geq 5)$.

For PEP, by Lemma 1, the expected iterations for finding $\{0\}^n$ is $O(P_{\max} n \log n)$. Then, $1\{0\}^{n-1}$ will be generated in one iteration with probability at least $\frac{1}{P_{\max}} \cdot \frac{1}{n}(1-\frac{1}{n})^{n-1} \geq \frac{1}{enP_{\max}}$, since it is sufficient to select $\{0\}^n$ from P and flip just the first 0 bit. Because $1\{0\}^{n-1}$ is Pareto optimal, it will always keep in P. Once it is regenerated by step 5 of Algorithm 1, whose probability is at least $\frac{1}{P_{\max}} \cdot (1-\frac{1}{n})^n$, it will go into the VDS process, and with probability $\Omega(\frac{1}{n})$ the solution path found by VDS on $1\{0\}^{n-1}$ is: $1\{0\}^{n-1} \to 11\{0\}^{n-2} \to 111\{0\}^{n-3} \to 1111\{0\}^{n-4} \to 0111\{0\}^{n-4} \to \cdots$. The corresponding objective vector changes as:

$$(\text{err}(h_1), 1) \to ((\text{err}(h_1)+\text{err}(h_2))/2, 2) \to (d_2 + d_3, 3)$$
$$\to ((d_2 + d_3 + d_4)/2, 4) \to (0, 3) \to (\geq 0, \geq 4).$$

Note that $P_{\max} \leq n + 1$. Thus, the expected time for finding the optimal solution $0111\{0\}^{n-4}$ with objective vector $(0, 3)$ is $O(P_{\max} n \log n + P_{\max} n + P_{\max} n) \cdot O(n^2)$, i.e., $O(n^4 \log n)$. □

PEP/OEP Can Do Better Than SEP

Heuristic optimization methods like evolutionary algorithms (EAs) (Bäck 1996) have been employed for solving the ensemble pruning in a single-objective formulation that mixes the two goals, which are briefly called SEP. GASEN (Zhou, Wu, and Tang 2002) is probably the first such method, and several other methods (e.g., artificial immune algorithms (Castro et al. 2005)) have also been proposed. However, it was unknown theoretically how well these optimization methods can be. Taking an EA presented in Algorithm 4 (He and Yao 2001; Auger and Doerr 2011) as a representative SEP, we prove in Theorem 3 that, for the type of pruning tasks described in Situation 2, OEP (and thus PEP due to Theorem 1) can find the optimal pruned ensemble efficiently but SEP needs at least exponential optimization time.

Algorithm 4 (SEP)**.** *Given a set of trained classifiers* $H = \{h_i\}_{i=1}^n$ *and an objective* $f : 2^H \to \mathbb{R}$*, it contains:*

1. $\boldsymbol{s}$ *= randomly selected from* $\{0, 1\}^n$.
2. ***Repeat until*** *the termination condition is met:*
3. *Generate* $\boldsymbol{s}'$ *by flipping each bit of* $\boldsymbol{s}$ *with prob.* $\frac{1}{n}$.
4. ***if*** $f(H_{\boldsymbol{s}'}) \leq f(H_{\boldsymbol{s}})$ ***then*** $\boldsymbol{s} = \boldsymbol{s}'$.
5. ***Output*** $\boldsymbol{s}$.

In Situation 2, all base classifiers make the same predictions except one that makes fewer mistakes. Without loss of generality, assume $H' = \{h_2, \ldots, h_n\}$. Let $err(h_1) = c_1$ and $err(h \in H') = c_2$, where $c_1 < c_2$. Then, the objective function of Eq.1 is

$$f(H_{\boldsymbol{s}}) = \begin{cases} +\infty, & \text{if } \boldsymbol{s} = \{0\}^n,\\ c_1, & \text{if } \boldsymbol{s} = 1\{0\}^{n-1},\\ (c_1 + c_2)/2, & \text{if } |\boldsymbol{s}| = 2 \wedge s_1 = 1,\\ c_2, & \text{otherwise.}\end{cases}$$

The main proof idea of Theorem 3 is that OEP can easily find the optimal solution $1\{0\}^{n-1}$ by the first greedy step, while SEP almost performs a random walk on a plateau and thus is inefficient. The proof needs Lemma 2, which is the *drift analysis* tool for deriving the optimization time of Markov chains, and note that the analyzed SEP (i.e., EA) is commonly modeled as a Markov chain (He and Yao 2001; Auger and Doerr 2011).

Situation 2.

$$\exists H' \subseteq H,\ |H'| = n-1 \ \wedge\ \forall g,h \in H', \mathit{diff}(g,h) = 0;$$
$$\mathit{err}(H - H') < \mathit{err}(h \in H').$$

Lemma 2. *(Oliveto and Witt 2011) Let X_t $(t \geq 0)$ be the random variable describing a Markov process over a finite state space $S \subseteq \mathbb{R}_0^+$ and denote $\Delta_t(i) = (X_{t+1} - X_t \mid X_t = i)$ for $i \in S$ and $t \geq 0$. Suppose there exist an interval $[a,b]$, two constants $\delta, \epsilon > 0$ and, possibly depending on $l := b - a$, a function $r(l)$ satisfying $1 \leq r(l) = o(l/\log l)$ such that for all $t \geq 0$ the following two conditions hold:*

1. $E(\Delta_t(i)) \geq \epsilon$ *for* $a < i < b$;
2. $P(|\Delta_t(i)| \geq j) \leq r(l)/(1+\delta)^j$ *for* $i > a$ *and* $j \in \mathbb{N}_0$.

Then there is a constant $c^* > 0$ such that for $T^* = \min\{t \geq 0 : X_t \leq a \mid X_0 \geq b\}$ it holds

$$P(T^* \leq 2^{\frac{c^* l}{r(l)}}) = 2^{-\Omega(\frac{l}{r(l)})}.$$

Theorem 3. *In Situation 2, OEP using Eq.1 finds the optimal solution in $O(n^2)$ optimization time, while the time of SEP is at least $2^{\Omega(n)}$ with probability $1 - 2^{-\Omega(n)}$.*

Proof. For OEP, according to the objective f, it will follow such an optimization path:

$$\{0\}^n \to 1\{0\}^{n-1} \to 1\boldsymbol{B}_{n-1}, |\boldsymbol{B}_{n-1}| = 1 \to \cdots.$$

The corresponding objective value changes as

$$+\infty \to c_1 \to (c_1 + c_2)/2 \to c_2.$$

Thus, it outputs the optimal solution $1\{0\}^{n-1}$. Its optimization time is fixed. In the $i\,(1 \leq i \leq n)$-th iteration of Algorithm 3, it needs to evaluate and compare $n-i+1$ pruned ensembles, which are generated by combing the current pruned ensemble with any of the $n - i + 1$ unselected classifiers. Thus, the total optimization time is $\sum_{i=1}^{n}(n - i + 1) = n(n+1)/2$, i.e., $O(n^2)$.

For SEP where a new solution is generated based on the current solution, we can model it by a Markov chain and use Lemma 2 to prove. Let $a = 2$, $b = n/3$, and let $X_t = |\boldsymbol{s}|$ be the number of 1 bits of the current solution. Let $mut(i \to j)$ be the probability of generating $|\boldsymbol{s}'| = j$ from $|\boldsymbol{s}| = i$ by step 3 of Algorithm 4. For any $a < i < b$, we have

$$E(\Delta_t(i)) = \sum_{j=1}^{n} mut(i \to j) \cdot j + mut(i \to 0) \cdot i - i$$
$$\geq \sum_{j=0}^{n} mut(i \to j) \cdot j - i = (n-i)\frac{1}{n} + i(1 - \frac{1}{n}) - i \geq \frac{1}{3},$$

where the 1st equality is because any solution except $\{0\}^n$ has at most the same objective value as a solution $\boldsymbol{s}$ with $|\boldsymbol{s}| = i > 2$, and then is always accepted; the 2nd equality is by applying the linearity of expectation on $\sum_{j=0}^{n} mut(i \to j) \cdot j = E(\sum_{i=1}^{n} s'_i)$, which is actually the expected number of 1 bits of the new solution generated by step 3 of Algorithm 4.

To make $|\Delta_t(i)| \geq j$, it is necessary to flip at least j bits. Thus, we have

$$P(|\Delta_t(i)| \geq j) \leq \binom{n}{j}\frac{1}{n^j} \leq \frac{1}{j!} \leq \frac{2}{2^j}.$$

Thus, the conditions of Lemma 2 hold with $\epsilon = \frac{1}{3}$, $\delta = 1$ and $r(l) = 2$. We can then get $P(T^* \geq 2^{\Omega(n)}) = 1 - 2^{-\Omega(n)}$, where $T^* = \min\{t \geq 0 : X_t \leq 2 \mid X_0 \geq \frac{n}{3}\}$. By Chernoff bounds, $P(X_0 \geq \frac{n}{3}) = 1 - 2^{-\Omega(n)}$ due to the uniform distribution of initial solution. Thus, with probability $1 - 2^{-\Omega(n)}$, the optimization time for finding a solution $\boldsymbol{s}$ with $|\boldsymbol{s}| \leq 2$ is $2^{\Omega(n)}$, which is obviously an optimization time lower bound for finding the optimal solution $1\{0\}^{n-1}$. □

Empirical Study

We conducted experiments on 20 binary and 10 multiclass data sets (Blake, Keogh, and Merz 1998), pruning the base classifiers trained by Bagging (Breiman 1996). To assess each method on each data set, we repeat the following process 30 times. The data set is randomly and evenly split into three parts, each as the training set, the validation set and the test set. A Bagging of 100 C4.5 decision trees (Quinlan 1993) is trained on the training set, then pruned by a pruning method using the validation set, and finally tested on the test set.

For PEP, the first goal is to minimize the validation error, which is also used as the evaluation criterion for the final ensemble selection. Two baselines are the full Bagging, which uses all the base classifiers, and the Best Individual (**BI**), which selects the best classifier according to the validation error. Five state-of-the-art ordering methods are compared, including Reduce-Error (**RE**) (Caruana et al. 2004), **Kappa** (Banfield et al. 2005), Complementarity (**CP**) (Martínez-Muñoz, Hernández-Lobato, and Suárez 2009), Margin Distance (**MD**) (Martínez-Muñoz, Hernández-Lobato, and Suárez 2009), and **DREP** (Li, Yu, and Zhou 2012) methods. They mainly differ in their considered objectives relating to the generalization performance, and they all use the validation error as the evaluation criterion for selecting the final ensemble. As a representative heuristic single-objective optimization method, an **EA** (Bäck 1996) is compared, which is similar to Algorithm 4 except that it generates and maintains n solutions in each iteration, minimizing the validation error. The parameter p for MD is set to be 0.075 (Martínez-Muñoz, Hernández-Lobato, and Suárez 2009), and the parameter ρ of DREP is selected from $\{0.2, 0.25, \ldots, 0.5\}$ (Li, Yu, and Zhou 2012). The number of iterations for PEP is set to be $\lceil n^2 \log n \rceil$ (the total number of evaluations $O(n^4 \log n)$ divided by the number of evaluations in each iteration $O(n^2)$, as suggested by Theorems 1&2). For the fairness of comparison, the number of iterations for EA is set to be $\lceil n^3 \log n \rceil$ so that it costs the same number of evaluations as PEP.

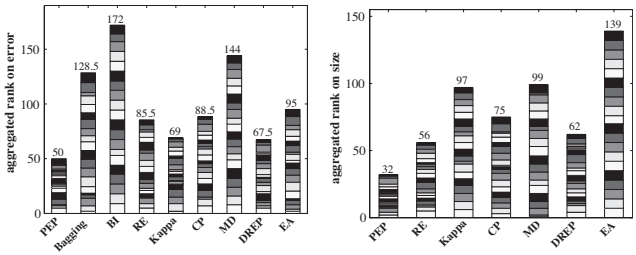
Table 1: The test errors and the sizes (mean+std.) of the compared methods on 20 binary data sets. In each data set, the smallest values are bolded, and '•/○' denote respectively that PEP is significantly better/worse than the corresponding method by the *t-test* with confidence level 0.05. In the rows of the count of the best, the largest values are bolded. The count of direct win denotes the number of data sets on which PEP has a smaller test error/size than the corresponding method (1 tie is counted as 0.5 win), where significant cells by the *sign-test* (Demšar 2006) with confidence level 0.05 are bolded.

Test Error									
Data set	PEP	Bagging	BI	RE	Kappa	CP	MD	DREP	EA
australian	.144±.020	**.143±.017**	.152±.023•	.144±.020	**.143±.021**	.145±.022	.148±.022	.144±.019	**.143±.020**
breast-cancer	**.275±.041**	.279±.037	.298±.044•	.277±.031	.287±.037	.282±.043	.295±.044•	**.275±.036**	**.275±.032**
disorders	**.304±.039**	.327±.047•	.365±.047•	.320±.044•	.326±.042•	.306±.039	.337±.035•	.316±.045	.317±.046•
heart-statlog	.197±.037	.195±.038	.235±.049•	**.187±.044**	.201±.038	.199±.044	.226±.048•	.194±.044	.196±.032
house-votes	.045±.019	**.041±.013**	.047±.016	.043±.018	.044±.017	.045±.017	.048±.018•	.045±.017	**.041±.012**
ionosphere	.088±.021	.092±.025	.117±.022•	.086±.021	**.084±.020**	.089±.021	.100±.026•	.085±.021	.093±.026
kr-vs-kp	**.010±.003**	.015±.007•	.011±.004	**.010±.004**	**.010±.003**	.011±.003	.011±.005	.011±.003	.012±.004
letter-ah	.013±.005	.021±.006•	.023±.008•	.015±.006•	**.012±.006**	.015±.006	.017±.007•	.014±.005	.017±.006•
letter-br	**.046±.008**	.059±.013•	.078±.012•	.048±.012	.048±.014	.048±.012	.057±.014•	.048±.009	.053±.011•
letter-oq	.043±.009	.049±.012•	.078±.017•	.046±.011	.042±.011	.042±.010	.046±.011	**.041±.010**	.044±.011
optdigits	**.035±.006**	.038±.007•	.095±.008•	.036±.006	**.035±.005**	.036±.005	.037±.006•	**.035±.006**	**.035±.006**
satimage-12v57	**.028±.004**	.029±.004	.052±.006•	.029±.004	**.028±.004**	.029±.004	.029±.004	.029±.004	.029±.004
satimage-2v5	**.021±.007**	.023±.009	.033±.010•	.023±.007	.022±.007	**.021±.008**	.026±.010•	.022±.008	**.021±.008**
sick	**.015±.003**	.018±.004•	.018±.004•	.016±.003	.017±.003•	.016±.003•	.017±.003•	.016±.003	.017±.004•
sonar	**.248±.056**	.266±.052	.310±.051•	.267±.053•	.249±.059	.250±.048	.268±.055•	.257±.056	.251±.041
spambase	**.065±.006**	.068±.007•	.093±.008•	.066±.006	.066±.006	.066±.006	.068±.007•	**.065±.006**	.066±.006
tic-tac-toe	.131±.027	.164±.028•	.212±.028•	.135±.026	.132±.023	.132±.026	.145±.022•	**.129±.026**	.138±.020
vehicle-bo-vs	**.224±.023**	.228±.026	.257±.025•	.226±.022	.233±.024•	.234±.024•	.244±.024•	.234±.026•	.230±.024
vehicle-b-v	**.018±.011**	.027±.014•	.024±.013•	.020±.011	.019±.012	.020±.011	.021±.011•	.019±.013	.026±.013•
vote	.044±.018	.047±.018	.046±.016	.044±.017	**.041±.016**	.043±.016	.045±.014	.043±.019	.045±.015
count of the best	**12**	2	0	2	7	1	0	5	5
PEP: count of direct win		**17**	**20**	**15.5**	12.5	**17**	**20**	12.5	**15.5**
Ensemble Size									
australian	10.6±4.2	–	–	12.5±6.0	14.7±12.6	11.0±9.7	**8.5±14.8**	11.7±4.7	41.9±6.7•
breast-cancer	8.4±3.5	–	–	8.7±3.6	26.1±21.7•	8.8±12.3	**7.8±15.2**	9.2±3.7	44.6±6.6•
disorders	14.7±4.2	–	–	**13.9±4.2**	24.7±16.3•	15.3±10.6	17.7±20.0	**13.9±5.9**	42.0±6.2•
heart-statlog	**9.3±2.3**	–	–	11.4±5.0•	17.9±11.1•	13.2±8.2•	13.6±21.1	11.3±2.7•	44.2±5.1•
house-votes	**2.9±1.7**	–	–	3.9±4.0	5.5±3.3•	4.7±4.4•	5.9±14.1	4.1±2.7•	46.5±6.1•
ionosphere	**5.2±2.2**	–	–	7.9±5.7•	10.5±6.9•	8.5±6.3•	10.7±14.6•	8.4±4.3•	48.8±5.1•
kr-vs-kp	**4.2±1.8**	–	–	5.8±4.5	10.6±9.1•	9.6±8.6•	7.2±15.2	7.1±3.9•	45.9±5.8•
letter-ah	**5.0±1.9**	–	–	7.3±4.4•	7.1±3.8•	8.7±4.7•	11.0±10.9•	7.8±3.6•	42.5±6.5•
letter-br	**10.9±2.6**	–	–	15.1±7.3•	13.8±6.7•	12.9±6.8	23.2±17.6•	11.3±3.5	38.3±7.8•
letter-oq	**12.0±3.7**	–	–	13.6±5.8	13.9±6.0	12.3±4.9	23.0±15.6•	13.7±4.9	39.3±8.2•
optdigits	22.7±3.1	–	–	25.0±9.3	25.2±8.1	**21.4±7.5**	46.8±23.9•	25.0±8.0	41.4±7.6•
satimage-12v57	**17.1±5.0**	–	–	20.8±9.2•	22.1±10.3•	21.2±10.0•	37.6±24.3•	18.1±4.9	42.7±5.2•
satimage-2v5	**5.7±1.7**	–	–	6.8±3.2	7.6±4.2•	10.9±7.0•	26.2±28.1•	7.7±3.5•	44.1±4.8•
sick	**6.9±2.8**	–	–	7.5±3.9	10.9±6.0•	11.5±10.0•	8.3±13.6	11.6±6.7•	44.7±8.2•
sonar	11.4±4.2	–	–	**11.0±4.1**	20.6±9.3•	13.9±7.1	20.6±20.7•	14.4±5.9•	43.1±6.4•
spambase	17.5±4.5	–	–	18.5±5.0	20.0±8.1	19.0±9.9	28.8±17.0•	**16.7±4.6**	39.7±6.4•
tic-tac-toe	14.5±3.8	–	–	16.1±5.4	17.4±6.5	15.4±6.3	28.0±22.6•	**13.6±3.4**	39.8±8.2•
vehicle-bo-vs	16.5±4.5	–	–	15.7±5.7	16.5±8.2	**11.2±5.7**○	21.6±20.4	13.2±5.0○	41.9±5.6•
vehicle-b-v	**2.8±1.1**	–	–	3.4±2.1	4.5±1.6•	5.3±7.4	**2.8±3.8**	4.0±3.9	48.0±5.6•
vote	**2.7±1.1**	–	–	3.2±2.7	5.1±2.6•	5.4±5.2•	6.0±9.8	3.9±2.5•	47.8±6.1•
count of the best	**12**	–	–	2	0	2	3	3	0
PEP: count of direct win		–	–	**17**	**19.5**	**18**	**17.5**	**16**	**20**

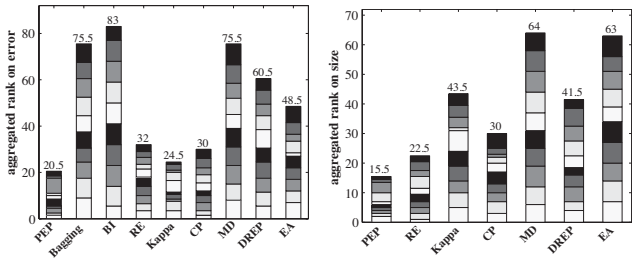
On Binary Classification

Some of the binary data sets are generated from the original multiclass data sets: from *letter*, *letter-ah* classifies 'a' against 'h', and alike *letter-br* and *letter-oq*; *optdigits* classifies '0∼4' against '5∼9'; from *satimage*, *satimage-12v57* classifies labels '1' and '2' against '5' and '7', and alike *satimage-2v5*; from *vehicle*, *vehicle-bo-vs* classifies 'bus' and 'opel' against 'van' and 'saab', and alike *vehicle-b-v*.

Table 1 lists the detailed results. Since it is improper to have a single summarization criterion over multiple data sets and methods, we employ the *number of best*, *number of direct win* that is a pairwise comparison followed by the *sign-test* (Demšar 2006), the *t-test* for pairwise comparison on each data set, and the *rank* (Demšar 2006). PEP achieves the smallest test error (or size) on 60% (12/20) of the data sets, while the other methods are less than 35% (7/20). By the *sign-test* with confidence level 0.05, PEP is significantly better than all the compared methods on size and all the methods except Kappa and DREP on test error, indicated by the rows "PEP: count of direct win". Though the *sign-test* shows that Kappa and DREP are comparable, PEP is still better on more than 60% (12.5/20) data sets. From the *t-test* with significance level 0.05, of which significant better and worse are indicated by '•' and '○', respectively, PEP is never significantly worse than the compared methods on test error, and has only two losses on size (on *vehicle-bo-vs* to CP and DREP). We also compute the rank of each method on each data set as in (Demšar 2006), which are stacked in



(a) on 20 binary data sets



(b) on 10 multiclass data sets

Figure 1: The aggregated rank on the test error and on the size for each method (the smaller the better).

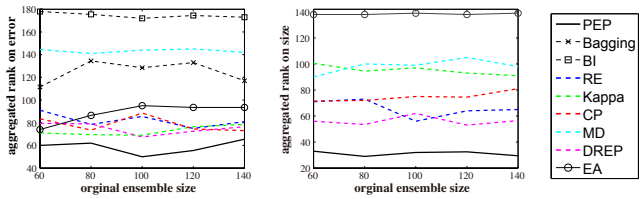
Figure 1(a).

All the criteria agree that BI is the worst on test error, which coincides with the fact that ensemble is usually better than a single classifier. Compared with RE, which greedily minimizes the validation error, PEP minimizes the validation error and the size simultaneously. PEP achieves significant improvement on the test error as well as the ensemble size. This observation supports the theoretical analysis that PEP is more powerful than OEP. As a type of SEP, EA produces ensembles with large sizes, which has been observed in previous studies (Zhou, Wu, and Tang 2002; Li and Zhou 2009). This also confirms our theoretical result that PEP/OEP can be better than SEP. Kappa, CP and MD are also OEP methods but optimizing diversity-like objectives. These methods leave the validation error alone. But since we find that the base classifiers have similar performance as the average coefficient of variation (i.e., the ratio of the standard deviation to the mean) for the validation errors of 100 base classifiers is 0.203, optimizing the diversity may work alone. DREP is an OEP method optimizing a combined error and diversity objective, which is shown to be better than the OEP methods optimizing only the diversity-like objectives, in both test error and ensemble size from Figure 1(a). PEP is better than DREP and the diversity-optimization methods, which may be because PEP achieves smaller sizes that prevent the overfitting problem.

Figure 2 investigates the effect of the original Bagging size n. We can observe that PEP always has the smallest error and size; and the ranking order of the methods is consistent with Figure 1(a).

On Multiclass Classification

We then compare these methods on 10 multiclass UCI data sets. Note that Kappa, CP, MD and DREP are originally designed for binary classification, we extend them for multiclass classification by generalizing their "equal" and "unequal" tests on multiple classes.



(a) test error (b) size

Figure 2: The aggregated rank of each method pruning the Bagging of $\{60, \ldots, 140\}$ base classifiers on 20 binary data sets (the smaller the better).

The detailed results are shown in Table 2. The overall performance is shown in Figure 1(b). We can observe that the compared methods have the similar performance rank as in binary classification except DREP. DREP performs much worse in multiclass classification than in binary classification, which may be because its performance is proved only in the binary classification scenario (Li, Yu, and Zhou 2012).

Application to Mobile Human Activity Recognition

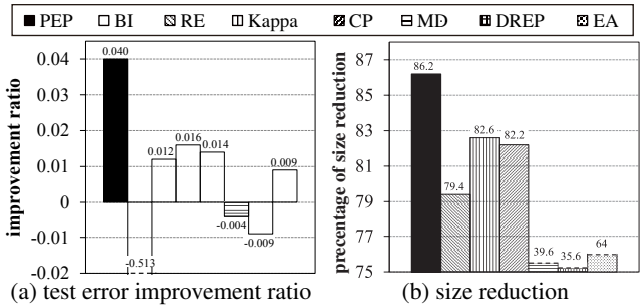
We then apply PEP to the task of Mobile Human Activity Recognition (MHAR) using smartphones. As smartphones have become more and more popular and essential in everyday life, human body signals can be easily retrieved from embedded inertial sensors. Learning from these information can help us better monitor user health and understand user behaviors. Specifically, MHAR using smartphones is to identify the actions carried out by a person according to the context information gathered by smartphones. Besides the accuracy of the classifier, it is also important to consider that smartphones only have limited storage and computation resources for doing predictions. Therefore, ensemble pruning is particularly appealing in the MHAR task.

We employ a lately available MHAR data set, published in (Anguita et al. 2012). The data set is collected from 30 volunteers wearing the smartphone on the waist who performed 6 activities (walking, upstairs, downstairs, standing, sitting, laying). The embedded 3D-accelerometer and 3D-gyroscope of a Samsung Galaxy S2 smartphone were used to collect data at a constant rate of 50 Hz. Then the records build a multiclass classification data set with 10299 instances and 561 attributes. The data set was further randomly partitioned into two parts: 70% as the training set and 30% as the test set. For evaluating the performance of one pruning method on MHAR, we repeat 30 independent runs. In each run, we fix the test set and randomly split the training set into two parts: 75% as the training and 25% as the validation. Bagging of 100 C4.5 decision trees are firstly trained on the training set, then pruned by the pruning methods using the validation set, and finally tested on the test set.

Figure 3(a) depicts the improvement ratio of the pruning methods to the test error of the full Bagging, and Figure 3(b) shows the reduction percentage of the number of classifiers from the full Bagging. It is clear that PEP achieves the best accuracy, about 3 times more than the

Table 2: The test errors and the sizes (mean+std.) of the compared methods on 10 multiclass data sets. In each data set, the smallest values are bolded, and '•/○' denote respectively that PEP is significantly better/worse than the corresponding method by the *t-test* with confidence level 0.05. In the rows of the count of the best, the largest values are bolded. The count of direct win denotes the number of data sets on which PEP has a smaller test error/size than the corresponding method (1 tie is counted as 0.5 win), where significant cells by the *sign-test* (Demšar 2006) with confidence level 0.05 are bolded.

Test Error									
Data set	PEP	Bagging	BI	RE	Kappa	CP	MD	DREP	EA
anneal	**.017±.006**	.032±.013•	.020±.008•	.018±.006	.018±.006	**.017±.007**	.027±.010•	.020±.008•	.025±.010•
audiology	**.360±.036**	.403±.044•	.403±.043•	.365±.040	.370±.035•	.364±.036	.401±.045•	.385±.037•	.383±.036•
balance-scale	.162±.018	.170±.020•	.240±.027•	.165±.026	**.160±.018**	.165±.021	.174±.023•	.167±.020	.166±.023
glass	**.307±.049**	.322±.051	.377±.054•	.310±.053	.308±.046	.309±.051	.334±.056•	.331±.048•	.312±.041
lymph	.231±.044	.254±.052•	.264±.035•	.235±.045	**.221±.040**	.227±.039	.255±.052•	.252±.037•	.251±.050•
primary-tumor	**.604±.031**	.618±.039•	.655±.036•	.610±.032	.612±.038	.610±.030	.615±.038•	.622±.035•	**.604±.039**
soybean	**.096±.019**	.127±.022•	.150±.021•	.100±.019•	.101±.015•	.101±.020	.125±.023•	.120±.025•	.106±.019•
vehicle	.280±.021	.281±.024	.340±.031•	.277±.027	**.275±.022**	.277±.023	.280± .025	.279±.021	.277±.022
vowel	**.200±.030**	.222±.030•	.396±.028•	.203±.028	.203±.028	.205±.027	.224±.030•	.214±.027•	.206±.028•
zoo	.129±.047	.175±.045•	.150±.038•	.132±.042	**.125±.052**	.135±.048	.177±.052•	.144±.038•	.160±.042•
count of the best	**6**	0	0	0	4	1	0	0	1
PEP: count of direct win		**10**	**10**	**9**	6	7.5	**9.5**	**9**	8.5
Ensemble Size									
anneal	3.3±1.8	–	–	**3.0±2.5**	7.0±6.0•	4.8±3.8•	12.0±12.2•	5.1±7.3	45.5±7.0•
audiology	**7.8±3.0**	–	–	11.2±7.5•	14.9±11.9•	13.0±10.9•	25.5±27.1•	12.0±14.5	45.3±4.5•
balance-scale	**16.3±3.0**	–	–	18.3±6.1	26.2±7.7•	19.9±10.5•	48.6±28.0•	30.2±17.1•	44.7±6.7•
glass	**11.9±3.5**	–	–	13.1±6.1	22.2±15.6•	13.6±7.4	27.4±20.6•	19.4±17.3•	42.0±5.2•
lymph	**6.9±1.9**	–	–	7.7±3.2	9.9±4.6•	9.2±5.3•	18.3±24.5•	7.7±10.0	46.0±4.7•
primary-tumor	**17.8±4.8**	–	–	18.4±8.2	48.5±21.0•	19.6±13.2	44.4±24.8•	25.1±24.4	41.5±6.2•
soybean	14.4±3.4	–	–	14.8±6.0	**12.9±4.9**	13.6±6.5	48.5±30.6•	24.8±17.1•	42.3±7.4•
vehicle	20.6±4.7	–	–	19.5±5.7	20.6±9.5	**17.8±10.1**	49.1±29.9•	38.3±26.6•	41.4±6.4•
vowel	**24.9±4.0**	–	–	26.8±6.5	34.8±10.5•	25.4±9.8	61.4±22.8•	44.6±21.6•	39.6±5.1•
zoo	**3.5±1.7**	–	–	3.8±3.8	7.0±5.6•	7.5±7.0•	19.6±22.6•	6.9±8.2•	47.1±5.4•
count of the best	**7**	–	–	1	1	1	0	0	0
PEP: count of direct win		–	–	8	8.5	8	**10**	**10**	**10**



(a) test error improvement ratio (b) size reduction

Figure 3: Average performance improvement and size reduction from the Bagging of 100 base classifiers for the MHAR task (the larger the better).

runner-up on the improvement ratio. Meanwhile, PEP has the best ensemble size reduction, which saves more than 20% (i.e., $(17.4-13.8)/17.4$) storage space than the runner-up. Furthermore, compared with the previous reported accuracy 89.3% achieved by the multiclass SVM (Anguita et al. 2012), PEP achieves a better one 90.4%.

Conclusion

Ensemble pruning can further improve the generalization performance of an ensemble, while reducing the cost for storage and running the ensemble model. There are naturally two goals in ensemble pruning, minimizing the error and minimizing the size. Most previous ensemble pruning approaches solve objectives that mix the two goals, which are conflicting when being pushed to the limit. In this work, we study solving the explicit bi-objective formulation, and propose a Pareto optimization approach, PEP.

Firstly, we derive theoretical results revealing the advantage of PEP over the ordering-based pruning methods as well as the single-objective heuristic optimization methods. We then conduct experiments, which disclose the superiority of PEP over the compared state-of-the-art pruning methods. Finally, we apply PEP in the application of mobile human activity recognition, where the prediction accuracy gets improved while the cost of storage gets saved.

There could be several directions to explore in the future. On the Pareto optimization algorithm, more effective operators can be employed, such as crossover that has been proved useful (Qian, Yu, and Zhou 2013). On the ensemble pruning, as diversity has been recognized as an important factor, it is necessary to investigate combining diversity into PEP. Moreover, since many machine learning tasks naturally involve multiple objectives, it is interesting to investigate new learning approaches by solving the multi-objective formulation explicitly using evolutionary algorithms, with theoretical and empirical advantages expected.

References

Anguita, D.; Ghio, A.; Oneto, L.; Parra, X.; and Reyes-Ortiz, J. L. 2012. Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine. In *Proceedings of the 4th International Workshop on Ambient Assisted Living and Home Care*, 216–223.

Auger, A., and Doerr, B. 2011. *Theory of Randomized Search Heuristics: Foundations and Recent Developments*. Singapore: World Scientific.

Bäck, T. 1996. *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. Oxford, UK: Oxford University Press.

Banfield, R. E.; Hall, L. O.; Bowyer, K. W.; and Kegelmeyer, W. P. 2005. Ensemble diversity measures and their application to thinning. *Information Fusion* 6(1):49–62.

Blake, C. L.; Keogh, E.; and Merz, C. J. 1998. UCI Repository of machine learning databases. [http://www.ics.uci.edu/~mlearn/MLRepository.html].

Breiman, L. 1996. Bagging predictors. *Machine Learning* 24(2):123–140.

Brown, G.; Wyatt, J.; Harris, R.; and Yao, X. 2005. Diversity creation methods: A survey and categorisation. *Information Fusion* 6(1):5–20.

Caruana, R.; Niculescu-Mizil, A.; Crew, G.; and Ksikes, A. 2004. Ensemble selection from libraries of models. In *Proceedings of the 21st International Conference on Machine Learning*, 18–25.

Castro, P. D.; Coelho, G. P.; Caetano, M. F.; and Von Zuben, F. J. 2005. Designing ensembles of fuzzy classification systems: An immune-inspired approach. In *Proceedings of the 4th International Conference on Artificial Immune Systems*, 469–482.

Demšar, J. 2006. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research* 7:1–30.

He, J., and Yao, X. 2001. Drift analysis and average time complexity of evolutionary algorithms. *Artificial Intelligence* 127(1):57–85.

Hernández-Lobato, D.; Martínez-Muñoz, G.; and Suárez, A. 2011. Empirical analysis and evaluation of approximate techniques for pruning regression bagging ensembles. *Neurocomputing* 74(12):2250–2264.

Li, N., and Zhou, Z.-H. 2009. Selective ensemble under regularization framework. In *Proceedings of the 8th International Workshop on Multiple Classifier Systems*, 293–303.

Li, N.; Yu, Y.; and Zhou, Z.-H. 2012. Diversity regularized ensemble pruning. In *Proceedings of the 23rd European Conference on Machine Learning*, 330–345.

Lin, S., and Kernighan, B. W. 1973. An effective heuristic algorithm for the traveling-salesman problem. *Operations Research* 21(2):498–516.

Margineantu, D. D., and Dietterich, T. G. 1997. Pruning adaptive boosting. In *Proceedings of the 14th International Conference on Machine Learning*, 211–218.

Martínez-Muñoz, G.; Hernández-Lobato, D.; and Suárez, A. 2009. An analysis of ensemble pruning techniques based on ordered aggregation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31(2):245–259.

Oliveto, P. S., and Witt, C. 2011. Simplified drift analysis for proving lower bounds in evolutionary computation. *Algorithmica* 59(3):369–386.

Partalas, I.; Tsoumakas, G.; and Vlahavas, I. 2012. A study on greedy algorithms for ensemble pruning. Technical report, Aristotle University of Thessaloniki, Greece.

Qian, C.; Yu, Y.; and Zhou, Z.-H. 2013. An analysis on recombination in multi-objective evolutionary optimization. *Artificial Intelligence* 204:99–119.

Quinlan, J. R. 1993. *C4.5: Programs for Machine Learning*. San Francisco, CA: Morgan kaufmann.

Tsoumakas, G.; Partalas, I.; and Vlahavas, I. 2009. An ensemble pruning primer. In *Applications of Supervised and Unsupervised Ensemble Methods*, volume 245 of *Studies in Computational Intelligence*. Berlin, Germany: Springer. 1–13.

Yu, Y.; Yao, X.; and Zhou, Z.-H. 2012. On the approximation ability of evolutionary optimization with application to minimum set cover. *Artificial Intelligence* 180-181:20–33.

Zhang, Y.; Burer, S.; and Street, W. N. 2006. Ensemble pruning via semi-definite programming. *Journal of Machine Learning Research* 7:1315–1338.

Zhou, Z.-H.; Wu, J.; and Tang, W. 2002. Ensembling neural networks: Many could be better than all. *Artificial Intelligence* 137(1):239–263.

Zhou, Z.-H. 2012. *Ensemble Methods: Foundations and Algorithms*. Boca Raton, FL: Chapman & Hall/CRC.