International Journal on Artificial Intelligence Tools
Vol. 26, No. 1 (2017) 1760003 (20 pages)

DOI: 10.1142/S021821301760003X

Induced Subgraph Game for Ensemble Selection

Hadjer Ykhlef
Department of Computer Science, University of Blida, Algeria
ykhlef.hadjer@gmail.com

Djamel Bouchaffra
Division of Design and Implementation of Intelligent Systems
Centre de recherche des technologies avancées, Algeria
djamel.bouchaffra@gmail.com

Received 11 June 2016
Accepted 15 December 2016
Published 23 February 2017

Ensemble methodology has proved to be one of the strongest machine learning techniques. In spite of its huge success, most ensemble methods tend to generate unnecessarily large number of classifiers, which entails an increase in memory storage, computational cost, and even a reduction in the generalization performance of the ensemble. Ensemble selection addresses these shortcomings by searching for a fraction of individual classifiers that performs as good as, or better than the entire ensemble. In this paper, we formulate ensemble selection problem as a coalitional game played on a graph. The proposed game aims at capturing two crucial concepts that affect the performance of an ensemble: *accuracy* and *diversity*. Most importantly, it ranks every classifier based on its contribution in keeping a proper balance between these two notions using Shapley value. To demonstrate the validity and the effectiveness of the proposed approach, we carried out experimental comparisons with some major selection techniques based on 35 UCI benchmark datasets. The results reveal that our approach significantly improves the original ensemble and performs better than the other methods in terms of classification accuracy, pruning ratio, and computational cost.

Keywords: Ensemble of classifiers; ensemble selection; coalitional games; Shapley value; induced subgraph games.

1. Introduction

During the last decade, ensemble learning has attracted a widespread attention from the machine learning and data mining community.[1–3] It refers to creating a collection (also called team, committee, ensemble, and pool) of learning models whose predictions are merged together to produce the final decision. It is widely accepted that a combination of multiple classifiers is a powerful decision making tool, and usually generalizes better than a single classifier.[4] Ensemble methodology has achieved a great success in a broad scope of real-world applications such as: remote sensing,[5] face recognition,[6] and intrusion detection.[7]

Some of the established methods for growing ensembles are boosting,[8] bagging,[4] random subspace,[9] and random forest.[10] It is believed that the key success of these techniques consists of creating an ensemble of accurate and diverse classifiers. However, there is no consensus on what diversity means? It can be perceived as: dependence, orthogonality, or even complementarity.[2] In addition, several studies have demonstrated that the effectiveness of any ensemble method depends on finding the right tradeoff between the individual accuracies and the diversity of the group.[11,12]

Despite their remarkable success, most ensemble methods need to train a large number of classifiers in order to guarantee that the training error rate reaches its minimal value. This necessity might result in overfitting the training set, which in turn causes a reduction in the generalization performance of the ensemble. In addition, an ensemble made of too many members incurs an increase in memory storage and computational cost. Ensemble pruning (also called ensemble shrinking, and ensemble selection) tackles these shortcomings by selecting a subset of classifiers that maintains, or even improves the generalization ability of the entire committee. Given an ensemble made of n classifiers, finding a subset that yields the best predictive performance requires searching the space of $2^n - 2$ non empty subsets, which is unfeasible for large ensembles. To overcome this computational burden, many ensemble pruning approaches have been proposed in the literature. Most of these techniques fall into three primary categories: *clustering-based*, *search-based*, and *ranking-based* approaches. A clustering-based approach invokes a clustering algorithm to extract a number of *representative prototype* component learners that compose the final ensemble.[13] A searched-based technique performs a heuristic search in the space of all possible subsets of classifiers while measuring the importance of a candidate subset; examples of this category include: genetic algorithm[14] and semi definite programming.[15] Nonetheless, their computational costs remain large. A ranking-based approach (aka ordering-based) assigns a rank to every ensemble member according to a certain criterion, and then selects the top classifiers whose ranks are above a predefined threshold. In spite of their simplicity, Martínez-Muñoz *et al.* reported that ranking-based methods are competitive with search-based techniques which are known to be efficient but have high computational costs.[2] *The only challenge a ranking method faces consists of adequately setting the criterion that measures the contribution of every classifier to the ensemble performance.* Other pruning methods are provided in the related work section.

We introduce an induced subgraph game, a class of coalitional games, for ranking classifiers. The proposed game captures both the individual accuracies and the diversity of the group. Most precisely, it measures the utility of every ensemble member in the game using Shapley value. *This latter assigns to each player* (*classifier*) *a payoff* (*rank*), *which corresponds to its marginal contribution in achieving a proper balance between the individual accuracies and the ensemble diversity.* It is worth underscoring that Coalitional Game Theory (CGT) is not new to the machine learning and data mining community. It has been used for modeling problems like: feature selection,[16] clustering,[17] and more recently as a general framework for the ADABOOST technique.[18]

This paper extends our previous work[19] with the following contributions: First, we propose *new formulations* for estimating accuracy and diversity based on probability notions. Moreover, we introduce a *simple test* that automatically determines the appropriate number of classifiers to include in the pruned ensemble. We also demonstrate the effectiveness of the new approach through extensive comparisons with some major state-of-the-art pruning techniques such as semi-definite programming and orientation ordering; then, we support our analysis with statistical tests.

The remainder of this paper is organized as follows: Section 2 reviews some state-of-the-art pruning techniques. Necessary notions from CGT are provided in Section 3. Section 4 introduces the proposed selection game. The experiments are conducted on benchmark datasets, and the results are discussed in Section 5. Finally, conclusions and future work are laid out in Section 6.

2. Related Work

2.1. *Ordering-based methods*

Margineantu and Dietterich were the first to address the ensemble pruning problem.[20] They proposed to rank each classifier based on a diversity measure estimated using *Cohen's Kappa*, and then to select a subset made of the most diverse members of the ensemble.

Martínez-Muñoz *et al.* introduced the concepts of *signature vector* associated to an individual member and the *ensemble signature vector*.[21,22] Signature vector indicates whether the corresponding classifier is correct or not on each training sample. The ensemble signature vector is defined as the average of all members' signature vectors. An efficient technique that uses these notions is *orientation ordering*. This method ranks the individual members according to the angle between their signature vectors and the reference vector.[22] The reference vector measures the direction that corresponds to perfect classification performance on the training set. The classifiers whose angles are less than $\pi/2$ are chosen to compose the pruned ensemble.

2.2. *Search-based methods*

The investigation carried out by Zhou *et al.* revealed that extracting a subset of learners from an ensemble composed of neural networks could improve the generalization performance.[14] This approach attributes a weight to each component classifier; a low value indicates that the corresponding member should be excluded. These weights are evolved toward an optimal solution following *genetic algorithm*. The evolution is governed by a fitness function which is defined as the classification accuracy on a separate sample set. Only classifiers with weights above a pre-set threshold are selected to compose the final pruned ensemble.

Zhang *et al.* considered ensemble pruning as a quadratic integer programming problem.[15] Their approach selects a sub-ensemble with an optimal accuracy/diversity tradeoff. The estimation of the optimal solution for this problem is computationally difficult; nevertheless, Zhang *et al.* acquired a good approximate solution using *semi-definite programming* (SDP). However, their technique is still computationally expensive. Furthermore, its success depends mainly on setting the appropriate size of the pruned ensemble.

In the same context as SDP, Xu *et al.* formulated ensemble selection as a combinatorial optimization problem with the goal of maximizing both accuracy and diversity.[23] Despite the fact that the original problem is computational expensive, they derived a relaxation of the original problem into *constrained eigen-optimization*, which can be solved efficiently. Although eigen-optimization technique yields better computational costs than SDP, it still requires setting the size of the pruned ensemble, which can considerably affect both the classification accuracy and the running time.

2.3. *Clustering-based methods*

Usually, methods of this category operate in two stages: Initially, they group models with similar performance together (strong correlation); two classifiers from different sets have low agreement (diverse). Subsequently, a set of representative prototypes are selected from each cluster to form the pruned ensemble. For instance, Lazarevic and Obradovic first invoked k-means to discover clusters of base learners.[24] Then, they iteratively pruned the ensemble members starting with the weakest one, and they consider them in descending order of their error rates until the committee overall predictive performance begins to decrease.

2.4. *Other methods*

This section reviews ensemble pruning approaches that do not belong to the aforementioned categories. Martínez-Muñoz *et al.* used ADABOOST to prune an ensemble trained by BAGGING.[25] Boosting-based pruning is a multistage technique: At each iteration, instead of training a base learner, it selects from the pool of classifiers the member with the lowest weighted training error. If no individual learner has a weighted error less than 0.5, this approach restarts the boosting process and resets all instances' weights. Note that the weights associated to the training samples are initialized and updated similarly to the ADABOOST algorithm. This procedure is repeated until the pre-set size of the pruned ensemble is met.

Tsoumakas *et al.* proposed *statistical tests* to prune an ensemble made of heterogeneous members.[26] First, their approach uses statistical procedures like *Turkey* and *Hsu* tests with the goal of identifying pairs of classifiers with significant differences;

then, only the individual learners that achieve significantly better performance constitute the pruned ensemble.

Ykhlef and Bouchaffra considered the problem of ensemble pruning as a *non-monotone simple game*.[27] Their approach extracts sub-ensembles with moderate diversities. First, it estimates the contribution of each classifier to the overall ensemble diversity based on Banzhaf power index. It then maps the pruned ensemble to the notion of the minimal winning coalition. Although the exact estimation of Banzhaf index is intractable for large and moderate ensembles, Ykhlef and Bouchaffra introduced weighted voting games to derive a new formulation that can be computed by a pseudo-polynomial time algorithm.

3. Coalitional Game Theory

Coalitional game theory[28] models situations that involve interactions among decision-makers, called *players*. The key concept behind these games is that players are allowed to bind agreement and form *coalitions*. Furthermore, the focus is on the outcomes achieved by groups rather than by individuals. Formally, a coalitional game with transferable utility (TU-game) $G = (\Omega, v)$ is composed of: (i) a set of n players $\Omega = \{h_1, h_2, \dots, h_n\}$, and (ii) a characteristic function (a.k.a payoff function) $v : 2^\Omega \mapsto \mathbb{R}$, where 2^Ω denotes the set of all possible coalitions. The mapping v assigns to coalition $S \subseteq \Omega$ a real number that expresses the worth or the benefit achieved by its members. The set Ω is called the *grand coalition*, and $\emptyset$ is called the *empty coalition*, such that $v(\emptyset) = 0$.

An outcome of a coalitional game[29] is a pair $(\mathcal{CS}, \tau)$ consisting of: (i) a coalition structure $\mathcal{CS} = \{S^1, S^2, \dots, S^\ell\}$, such that $\bigcup_{j=1}^{\ell} S^j = \Omega$ and $S^i \cap S^j = \emptyset$ for all $i, j \in \{1, 2, \dots, \ell\}, i \neq j$; and (ii) a payoff vector $\tau = (\tau_i)_{h_i \in \Omega}$, where τ_i measures the total utility assigned to player h_i. A solution concept defines for each coalitional game a set of feasible outcomes. It aims at capturing two appealing properties: *fairness* and *stability*. A payoff allocation τ satisfies the fairness criteria if every player receives a value that corresponds to its real contribution in the game, whereas stability guarantees that no subset of players has an incentive to deviate from the current coalition structure and form a coalition on their own. Famous solution concepts of characteristic function games include: Core,[30] Shapley value,[31] Banzhaf value, Nucleolus, and Bargaining set. We focus our study on the core and the Shapley value.

3.1. *Shapley value*

Shapley value of a characteristic function game $G = (\Omega, v)$ provides a fair manner of distributing the grand coalition's worth among its members. Formally, it is defined by:

$$\phi_i = \sum_{S \subseteq \Omega \setminus \{h_i\}} \frac{|S|!(|\Omega| - |S| - 1)!}{|\Omega|!} (v(S \cup h_i) - v(S)) \,. \tag{1}$$

The payoff allocated to player h_i, denoted ϕ_i, corresponds to its average marginal contribution in the game. Specifically, suppose that a coalition S is formed, starting with an empty set and adding one player at a time. Within any such sequence of additions, we first compute player h_i's marginal contribution $(v(S \cup h_i) - v(S))$; then, we multiply this quantity by $|S|!$ (the number of different ways the coalition S could have been assembled) and by $(|\Omega|-|S|-1)!$ (the number of different ways the remaining players could join S). Finally, we calculate *the average of these marginal contributions* by summing over all possible coalitions and by dividing by $|\Omega|!$ i.e. the number of all possible permutations of n players.

3.2. *Core*

The core is the best-known solution concept for addressing the stability criterion. An outcome is stable if no coalition can obtain a payoff that exceeds the sum of its members' current payoffs. As an illustrative example, let us consider a characteristic function game $G = (\Omega, v)$ and an outcome $(\mathcal{CS}, \tau)$ of this game, where $\mathcal{CS} = (S, \bar{S})$. In addition, suppose that $\sum_{h_i \in S} \tau_i < v(S)$. In this case, the players in S could do better by abandoning the current coalition structure $\mathcal{CS}$ and forming other coalitions of their own. Therefore, the outcome $(\mathcal{CS}, \tau)$ is unstable. The set of payoff allocations in which no group of players can jointly deviate to improve their payoffs, i.e. stable outcomes, forms the core of a coalitional game. Formally, The core of $G = (\Omega, v)$ consists of all outcomes $(\mathcal{CS}, \tau)$ that satisfy $\sum_{h_i \in S} \tau_i \geq v(S), \forall\, S \subseteq \Omega$.

Note that Shapley value assigns to every characteristic function game a unique payoff allocation, whereas the core can be an empty set.[29]

The naïve representation of a coalitional game consists of listing the payoffs associated to all possible coalitions, which is exponential in the number of players, and hence impractical for most problems. To overcome this shortcoming, several representations for coalitional games such as marginal contribution nets,[32] network flow games,[33] induced subgraph games,[34] synergy coalition groups[35] have been proposed in the literature. In the next subsection, we define the induced subgraph game representation.

3.3. *Induced subgraph games*

This representation considers a coalitional game to be played on an undirected weighted graph $\mathcal{G} = (\Omega, E)$, in which every edge $(h_i, h_j) \in E$ is associated with a weight $w_{i,j}$; we write $\mathbf{w} = (w_{i,j})_{(h_i,h_j)\in E}$. In the induced subgraph game $G = (\mathcal{G}, \mathbf{w})$, a node $h_i \in \Omega$ corresponds to a player and the worth of a coalition $S \subseteq \Omega$ is defined as:

$$v(S) = \sum_{\substack{(h_i,h_j)\in E \\ \{h_i,h_j\}\subseteq S}} w_{i,j}\,. \tag{2}$$

This formulation is concise because it suffices to use a $|\Omega| \times |\Omega|$ matrix to represent a coalitional game. Interestingly, induced subgraph games admit an efficient algorithm for computing Shapley value. Formally, given an induced subgraph game $G = (\mathcal{G}, \mathbf{w})$, player h_i's Shapley value is defined as:

$$\phi_i = w_{i,i} + \frac{1}{2} \times \sum_{\substack{(h_i,h_j)\in E \\ h_i \neq h_j}} w_{i,j} \,. \tag{3}$$

The proof of the above formulation can be found in Ref. 29. In addition, when all edge weights are positive $w_{i,j} \geq 0 \,\forall\, (h_i, h_j) \in E$, induced subgraph games are guaranteed to have a non-empty core, and moreover, Shapley value belongs to the core.[34]

4. Induced Subgraph Game for Ensemble Selection

Let $\Omega = \{h_1, h_2, \ldots, h_n\}$ be an ensemble of n classifiers. Each learner is provided with the same training set $\Gamma_{train} = \{(x_i, y_i), i = 1 \ldots m\}$, where $x_i \in \mathcal{X}$ is a feature vector characterizing the ith instance, and $y_i \in \mathcal{Y}$ denotes the true class label. Given a feature vector x, the final decision of the committee Ω combines the predictions of all its members $h_1(x), h_2(x), \ldots, h_n(x)$ following majority vote. We also recorded the oracle outputs of the ensemble members in a Boolean matrix $Z = (z_{ki})_{\substack{1\leq k\leq m, \\ 1\leq i\leq n}}$ with $z_{ki} = 1$ if h_i is correct on the kth sample, and 0 otherwise. The number of correct/incorrect predictions made by two classifiers h_i and h_j on the training set Γ_{train} is defined as:

$$N^{ab}(i,j) = \sum_{k=1}^{m} \mathbb{I}(z_{ki} = a \text{ and } z_{kj} = b)\,, \quad a, b \in \{0, 1\}\,, \tag{4}$$

where $\mathbb{I}(cond)$ designates the indicator function, which equals 1 if the condition $cond$ is satisfied ($cond = true$), and 0 otherwise.

Ensemble selection aims at extracting a subset ω that minimizes the error rate on the test set $\Gamma_{test} = \{(x_i, y_i), i = 1 \ldots t\}$. An important criterion for creating accurate ensembles is diversity.[8,4] It is almost always the case that two accurate classifiers have low diversity, while two weak learners (their accuracies are slightly better than random guessing) often disagree with each other. This paradox is known as accuracy/diversity dilemma. Since diversity decreases with the increase in the individual accuracies, the key success of an ensemble method consists of creating a committee which adequately balances diversity and accuracy. We propose a new ranking-based pruning technique that addresses the accuracy/diversity dilemma using CGT principles. We named our approach Induced Subgraph Game Ensemble Pruning (Isgep). We first formulate ensemble selection as an induced subgraph game played among the individual learners. The proposed game is defined based on two notions namely *accuracy* and *diversity. Then, we rank each classifier based on its contribution in keeping a fair balance between accuracy and diversity using Shapley value.*

Definition 4.1. The accuracy of classifier h_i, denoted Acc_i, is given by:

$$Acc_i = \frac{N^{11}(i,i)}{N^{00}(i,i) + N^{11}(i,i)} \,. \tag{5}$$

Definition 4.2. The diversity between two ensemble members h_i and h_j is defined as:

$$Div_{i,j} = \frac{1}{2} \times \left(\frac{N^{10}(i,j)}{N^{00}(i,j) + N^{10}(i,j)} + \frac{N^{01}(i,j)}{N^{00}(i,j) + N^{01}(i,j)} \right) . \tag{6}$$

Recall that $N^{00}(i,j)$, $N^{11}(i,j)$, $N^{01}(i,j)$, and $N^{10}(i,j)$ denote the number of correct/incorrect predictions made by two ensemble members h_i and h_j on the training set. In the first term $N^{10}(i,j)/(N^{00}(i,j) + N^{10}(i,j))$, the nominator corresponds to the number samples on which h_i is correct and its counterpart h_j is incorrect, whereas the denominator measures the total number of errors made by h_j. Therefore, the quotient expresses the *conditional probability* that h_i correctly classifies a sample given that h_j does not. We can derive the same observation regarding the other term. In order to keep the diversity term on the same scale as the accuracy, we take the average of these two probability estimates. This definition elegantly captures the notion of diversity: *pairs of individuals that make uncorrelated errors yield higher diversity.*

Definition 4.3. Let $G = (\mathcal{G}, \mathbf{w})$ be an induced subgraph game, where each node corresponds to an ensemble member h_i and the weights $\mathbf{w} = (w_{i,j})$ are defined as:

$$w_{i,j} = \begin{cases} Acc_i & \text{if } i = j \\ Div_{i,j} & \text{otherwise} \end{cases} . \tag{7}$$

The weight assigned to a self-loop corresponds to the accuracy of h_i, while the weight of an edge linking h_i and h_j expresses the diversity between these two classifiers. In this way, Shapley value measures the rank of each ensemble member by considering its accuracy and diversity. Formally, the rank assigned to a classifier h_i is given by:

$$\phi_i = Acc_i + \frac{1}{2} \times \sum_{h_j \in \Omega \setminus \{h_i\}} Div_{i,j} \,. \tag{8}$$

Since all edge weights are non-negative, the payoff allocation (rank) provided by Shapley value belongs to the core, and hence guarantees stability. Equation (8) consists of two terms: the individual accuracy Acc_i and the diversity contribution $\frac{1}{2} \times \sum_{h_j \in \Omega \setminus \{h_i\}} Div_{i,j}$. From the analysis of this equation, we can derive two important observations: First, when two ensemble members are *similarly accurate*, Shapley value promotes the individual classifier that *induces higher diversity*; second, when two ensemble members have *equal diversity* terms, the one which *performs better* receives higher payoff allocation (rank). Therefore, the focus is on accurate members that contribute considerably to the overall ensemble diversity.

The pruned ensemble is made of the individual classifiers whose Shapley values ϕ_i exceed a pre-set selection threshold σ. Exploratory experiments indicate that a value $\sigma = \sum_{i=1}^{n} \phi_i / n$ is appropriate.

4.1. ISGEP algorithm

The pseudo-code of the proposed approach is depicted by Figure 1. The algorithm takes as an input a training set Γ, an ensemble of classifiers Ω, and a selection threshold σ. It begins with computing the ensemble members' predictions $Preds$ of the training samples [lines 3–7], and uses them to build the adjacency matrix $\mathbf{w}$ [lines 8–13]. Then, it estimates the individual contribution of every classifier using the definition provided by Equation (8) [lines 14–16]. Finally, selection is conducted by amalgamating the ensemble members whose ranks are above the threshold σ [lines 17–21].

1: **Input:** Γ: Training set.
  Ω: Ensemble of classifiers.
  σ: Selection threshold.
2: **Initialize:** $\omega = \emptyset$;
*/*Getting classifiers' predictions*/*
3: **For each** $h_i \in \Omega$
4: **For each** $(x_j, y_j) \in \Gamma$
5: $Preds_j^i = h_i(x_j)$;
6: **End for each** (x_j, y_j)
7: **End for each** h_i
*/*Constructing the adjacency matrix using Preds*/*
8: **For each** $h_i \in \Omega$
9: $w_{i,i} = Acc_i$;
10: **For each** $h_j \in \Omega \setminus \{h_i\}$
11: $w_{i,j} = Div_{i,j}$;
12: **End for each** h_j
13: **End for each** h_i
*/*Computing classifiers' Shapley values*/*
14: **For each** $h_i \in \Omega$
15: $\phi_i = w_{i,i} + \frac{1}{2} \times \sum_{h_j \in \Omega \setminus \{h_i\}} w_{i,j}$;
16: **End for each** h_i
17: **For each** $h_i \in \Omega$
18: **If** $\phi_i \geq \sigma$
19: $\omega = \omega \cup \{h_i\}$;
20: **End if**
21: **End for each** h_i
22: **Output:** ω: Pruned ensemble.

Fig. 1. The ISGEP algorithm.

5. Experiments

5.1. *Setup*

To evaluate the performance of Isgep, we conducted experiments using 35 datasets selected from the UCI benchmark Repository of Machine Learning Databases.[36] The collected datasets cover a wide range of application domains. The dimensions of these databases vary from 4 to 262, and the number of samples ranges from 24 to 20 000. A detailed summary of the datasets properties is shown in Table 1.

We resampled each dataset following Dietterich's 5×2 cross-validation. First, we divided (with stratification) the sample set into two disjoint folds Γ_{train} and Γ_{test}. We then trained the classifiers and computed the adjacency matrix $\mathbf{w}$ using

Table 1. Properties of the datasets used in the experiments.

Datasets	Abbreviations	Samples	Features	Classes
Arrhythmia	*D01*	452	262	16
Australian credit approval	*D02*	690	14	2
Breast cancer wisconsin	*D03*	699	9	3
Car evaluation	*D04*	1728	6	4
Glass identification	*D05*	214	10	6
Hayes-Roth	*D06*	160	5	4
Heart disease cleveland	*D07*	303	13	5
Heart disease hungarian	*D08*	294	3	5
Heart disease switzerland	*D09*	123	13	5
Hepatitis	*D10*	155	19	2
Ionosphere	*D11*	351	34	2
Labor	*D12*	57	16	2
Lenses	*D13*	24	4	3
Letter recognition	*D14*	20 000	16	26
Lymphography	*D15*	148	18	4
Multi-feature fourier	*D16*	2000	76	10
Multi-feature karhunen-love	*D17*	2000	64	10
Multi-feature morphological	*D18*	2000	6	10
Multi-feature profile correlations	*D19*	2000	216	10
Multi-feature zernike	*D20*	2000	47	10
Musk1	*D21*	476	166	2
Musk2	*D22*	6598	166	2
Nursery	*D23*	12 960	8	5
Optical recognition	*D24*	5620	64	10
Pen-based recognition	*D25*	10 992	16	10
Soybean large	*D26*	683	35	19
Soybean small	*D27*	47	35	4
Spambase	*D28*	4601	57	2
Thyroid domain	*D29*	7200	21	3
Waveform (version 1)	*D30*	5000	21	3
Waveform (version 2)	*D31*	5000	40	3
Wine	*D32*	178	13	3
Wisconsin diagnostic breast cancer	*D33*	569	30	2
Wisconsin prognostic breast cancer	*D34*	198	32	2
Yeast	*D35*	1484	8	10

Γ_{train}, whereas the other fold Γ_{test} was employed to measure the classification accuracy, the running time, and the pruning ratio. Next, we swapped the roles of Γ_{train} and Γ_{test} to obtain another estimate of the evaluation measures. This process was repeated five times, and at the end, we obtained ten trained ensembles and ten performance metrics. Note that we only reported the averaged measurements over these iterations.

In order to generate the initial ensemble, we used Bagging with Cart as a base learner to train a set of 100 classifiers. For the implementation, we imported the source code provided by Waikato Environment for Knowledge Analysis "Weka" library version 3.6,[37] and set all the parameters to their default values. We set the selection threshold σ of IsGEP to $\sum_{i=1}^{n} \phi_i/n$, where ϕ_i denotes the rank assigned to h_i, and n is the number of classifiers ($n = 100$). We compared the proposed approach with eight state-of-the-art techniques: Accuracy ordering (BestN), Semi Definite Programming (Sdp),[15] Genetic Algorithm (Ga),[14] Orientation Ordering (OO),[22] Margin Distance Minimization (Mdsq)[21] with a moving reference point p set to $\sqrt{i}$ at the ith iteration, Boosting-Based (BB),[25] Complementarity Measure (CC),[21] and Reduce Error (Re).[20] Accuracy ordering ranks classifiers based on their individual accuracies on the training fold and chooses the first N members. The technique Sdp invokes the Sdpa library version 7.3.6.[38] The parameters of the genetic algorithm search strategy are set to the values that were suggested by Zhou *et al.*[14] Note that BestN, Sdp, Mdsq, BB, CC, and Re methods prune the initial ensemble to a pre-set size L. Therefore, in order to make a fair comparison, we set L to the same size obtained by the proposed approach.

5.2. *Accuracy performance*

Table 2 displays the mean accuracy and standard deviation measurements obtained by the different approaches on each dataset. The last row specifies the averaged rank of each method. Figure 2 displays box plots of various ensemble approaches for six datasets: Lenses, Letter recognition, Multi-feature profile correlations, Pen-based recognition of handwritten digits, Soybean small, and Wisconsin diagnostic breast cancer.

The results given in Table 2 and Figure 2 indicate that IsGEP outperforms the other methods in most cases. In order to confirm the significance of the observed differences, we first carried out a Friedman test to compare these techniques. Under the null hypothesis, we assumed that all techniques are equivalent and the observed differences are due to chance. Friedman test rejects this hypothesis with $F_F = 20.49 > F(9, 306) = 8.06$ for $\alpha = 1 \times 10^{-10}$ (F_F is distributed according to the F distribution with $10-1 = 9$ and $(10-1)\times(35-1) = 306$ degrees of freedom), and confirms the existence of at least one pair of ensemble pruning techniques with significantly different performances. Because we are only interested in comparing our approach with the other alternatives, we therefore proceed with a Bonferroni-Dunn test while considering IsGEP as the control algorithm. Figure 3 shows the

Table 2. Summary of mean accuracy results.

Datasets	Bagging	Ga	BB	OO	BestN	CC	Mdsq	Re	Sdp	Isgep
D01	71.99±2.56	71.59±2.43	71.90±3.01	71.90±2.94	71.64±2.37	71.95±2.60	71.68±2.41	**72.17±2.67**	**72.17±2.78**	**72.17±2.86**
D02	85.88±1.51	85.74±1.51	85.88±1.69	85.83±1.29	85.65±1.24	85.80±1.17	85.77±1.48	85.97±1.09	**86.03±1.43**	86.00±1.33
D03	96.11±0.65	95.99±0.52	96.11±0.69	96.02±0.77	95.88±0.88	96.02±0.75	95.82±0.99	**96.28±0.69**	96.25±0.44	**96.28±0.42**
D04	95.71±1.04	95.82±0.65	96.08±0.74	96.20±0.69	95.87±0.70	95.98±0.92	95.88±0.71	96.04±0.84	95.79±0.94	**96.23±0.84**
D05	71.59±4.52	70.65±3.77	72.34±4.43	72.90±4.34	72.90±4.01	72.06±4.37	72.71±3.46	72.90±4.72	69.63±4.67	**73.36±3.92**
D06	80.25±6.34	80.88±4.00	82.00±2.78	**82.50±4.08**	81.13±3.65	82.00±4.09	81.13±3.65	80.88±4.64	81.63±4.60	**82.50±3.91**
D07	80.72±3.08	80.86±2.85	81.85±1.84	81.85±2.47	80.99±2.94	81.25±2.68	80.99±3.06	81.52±2.82	81.38±2.73	**82.44±2.55**
D08	79.12±2.87	78.78±2.86	79.25±2.67	78.91±2.90	78.50±3.50	78.64±3.16	78.64±3.18	79.25±3.47	79.12±2.87	**79.46±2.77**
D09	**56.31±2.89**	52.26±3.85	53.71±5.14	54.03±5.62	54.69±4.01	52.40±5.35	54.37±4.47	54.19±4.37	55.01±5.37	55.97±4.87
D10	**81.16±4.35**	79.73±5.04	80.64±4.25	79.99±5.51	79.86±6.32	80.39±4.84	79.99±6.23	80.77±4.24	80.38±5.23	80.77±5.54
D11	91.68±1.62	91.62±1.48	**92.65±2.43**	92.42±2.07	91.96±2.53	91.56±1.77	92.08±2.57	92.31±1.89	92.37±2.33	**92.65±1.91**
D12	85.90±5.93	84.53±3.92	85.59±5.46	85.58±5.75	84.52±3.96	82.81±3.89	83.50±4.72	85.90±6.56	84.56±5.81	**86.63±7.10**
D13	65.83±8.29	72.50±13.64	72.50±10.43	75.00±13.61	69.17±12.45	72.50±7.91	74.17±12.08	73.33±12.30	71.67±8.96	**80.00±8.05**
D14	91.56±0.62	91.44±0.50	91.63±0.47	**91.82±0.47**	91.27±0.65	91.66±0.55	91.29±0.66	91.65±0.47	89.25±4.95	91.75±0.40
D15	78.78±4.37	77.97±3.55	**79.73±4.08**	79.46±4.67	78.24±4.20	78.24±5.11	78.51±4.39	78.92±4.28	78.51±4.43	79.59±4.83
D16	79.40±1.22	79.01±1.09	79.28±1.03	79.48±1.25	79.31±1.26	79.23±0.97	79.24±1.32	79.27±1.03	79.14±1.14	**79.50±1.13**
D17	89.36±1.48	89.26±1.39	89.87±1.37	90.14±1.26	89.36±1.23	88.88±1.03	89.35±1.15	89.59±1.11	88.89±1.28	**90.33±1.01**
D18	**71.40±1.06**	70.94±1.24	71.35±1.26	71.22±1.14	71.36±1.26	71.30±1.21	71.23±1.22	71.04±1.33	71.26±1.29	71.35±1.17
D19	94.16±0.87	93.98±0.97	94.21±0.96	94.43±1.05	94.03±0.95	93.25±1.08	94.04±0.98	94.11±0.99	94.48±0.92	**94.52±1.00**
D20	75.69±1.60	75.63±1.53	75.71±1.66	76.27±1.47	75.78±1.83	75.42±1.62	75.85±1.72	75.81±1.70	74.64±4.06	**76.29±1.41**
D21	82.65±3.26	81.97±2.16	83.53±2.05	83.28±2.92	83.28±3.66	81.26±2.97	83.32±3.54	82.98±3.08	**84.08±2.56**	83.78±2.55
D22	96.33±0.31	96.44±0.34	96.44±0.32	96.48±0.36	96.46±0.41	96.49±0.33	96.46±0.37	96.49±0.34	96.47±0.42	**96.52±0.33**
D23	99.08±0.16	99.13±0.19	99.14±0.13	99.15±0.14	99.07±0.13	99.13±0.16	99.07±0.13	99.16±0.13	99.09±0.15	**99.18±0.16**
D24	95.54±0.49	95.55±0.43	95.79±0.39	96.02±0.48	95.41±0.50	95.65±0.44	95.43±0.50	95.60±0.48	95.53±1.05	**96.03±0.55**
D25	97.64±0.24	97.59±0.19	97.75±0.19	97.84±0.19	97.60±0.21	97.69±0.17	97.60±0.21	97.74±0.20	97.67±0.37	**97.90±0.19**
D26	91.54±1.57	92.04±1.39	92.01±1.71	91.89±1.60	91.33±1.44	91.80±1.39	91.36±1.41	91.74±1.60	91.68±1.73	**92.09±1.52**
D27	**98.71±2.91**	98.70±2.93	**98.71±2.91**	**98.71±2.91**	98.28±3.02	96.54±5.72	98.28±3.02	**98.71±2.91**	96.54±5.72	**98.71±2.91**
D28	92.96±0.69	93.11±0.51	93.23±0.67	93.18±0.63	92.96±0.66	**93.25±0.68**	92.99±0.68	93.11±0.63	93.01±0.74	**93.25±0.64**
D29	99.54±0.09	99.58±0.09	99.59±0.09	99.58±0.11	99.57±0.10	99.59±0.11	99.57±0.10	99.59±0.11	**99.61±0.10**	99.59±0.12
D30	**83.65±0.68**	83.38±0.63	83.62±0.65	83.54±0.67	83.32±0.63	83.60±0.60	83.31±0.69	83.54±0.76	82.30±1.74	83.57±0.63
D31	82.86±0.64	82.78±0.61	82.56±0.69	82.79±0.80	82.53±0.73	82.89±0.56	82.59±0.71	**82.90±0.58**	82.51±0.77	82.83±0.73
D32	94.49±3.69	94.38±3.39	95.62±2.51	95.73±3.03	93.48±3.78	92.70±4.28	93.48±3.78	93.82±3.87	94.61±4.39	**95.84±3.05**
D33	93.71±1.83	93.85±1.87	94.13±1.64	94.13±1.69	93.88±1.83	93.92±1.86	93.88±1.93	93.74±1.79	93.71±2.02	**94.73±1.36**
D34	**77.17±1.97**	75.45±2.13	75.56±3.26	75.96±3.04	76.46±2.13	75.66±2.45	76.46±2.82	76.06±1.58	76.06±2.52	76.26±3.20
D35	**59.92±1.20**	59.57±1.47	59.54±1.48	59.35±1.79	59.29±1.45	59.69±1.39	59.33±1.51	59.51±1.55	59.39±1.41	59.80±1.67
Average ranks	**5.76**	**7.76**	**4.21**	**4.10**	**7.43**	**6.24**	**7.07**	**4.49**	**6.19**	**1.76**

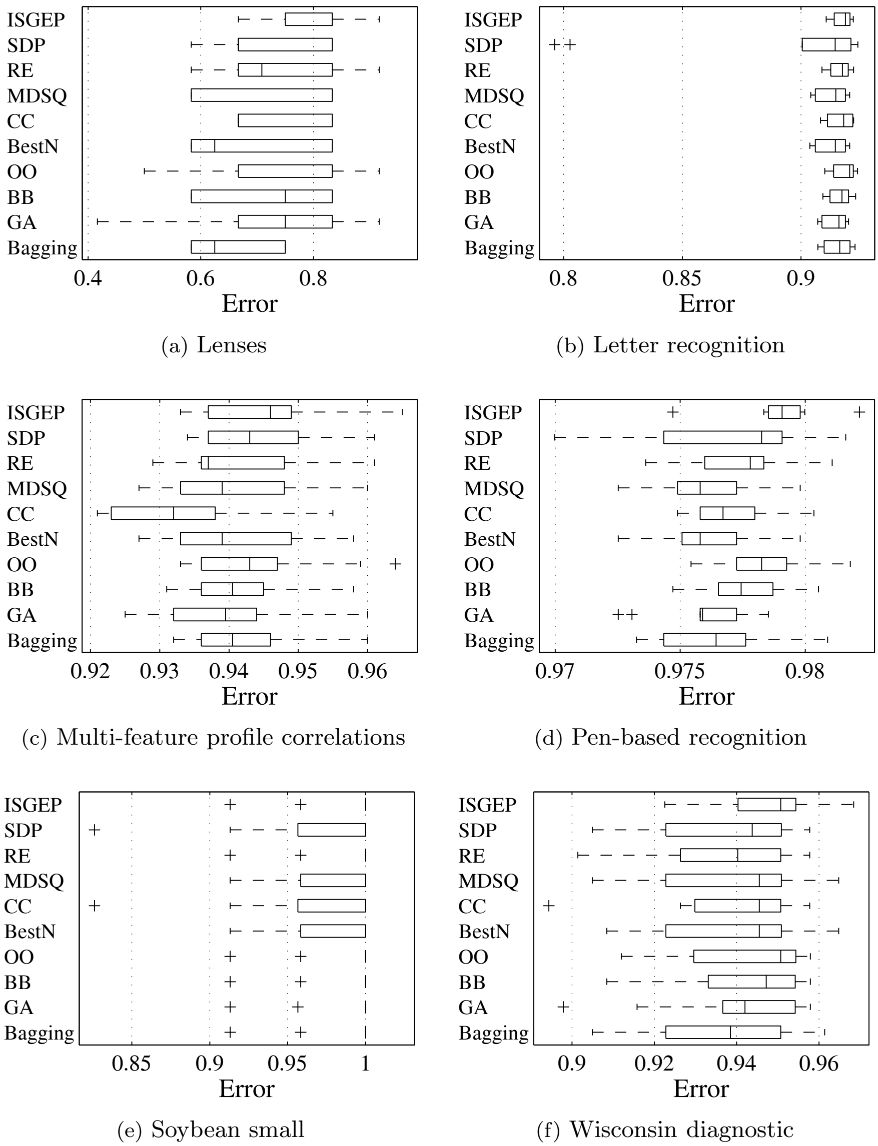


Fig. 2. Box plots of various ensemble approaches. The box edges specify the 1st/3rd quartiles, the horizontal line within each box indicates the median value, and the dotted crossbars correspond to the maximum/minimum values.

results of a Bonferroni-Dunn test at a 5% significance level with the critical value $q_{0.05} = 2.77$ and the critical difference $CD = 2.01$. On the horizontal axis, we represent the average rank of each selection technique given in the last row of Table 2, and mark using a thick line an interval of $2 \times CD$, one on the right and the other to the left of the ISGEP's average rank.

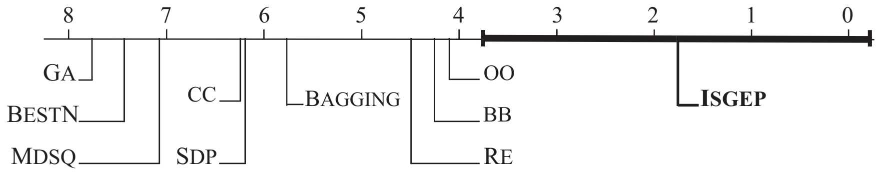


Fig. 3. Comparison of ISGEP with the other pruning approaches using Bonferroni-Dunn test.

The analysis of Bonferroni-Dunn test results illustrated by Figure 3 can be summarized by two primary observations:

(1) We notice that ISGEP has the lowest rank and all the other pruning techniques fall outside the marked interval. Therefore, we can conclude that the proposed approach significantly improves the original ensemble and outperforms the other alternatives, which is consistent with our initial observations.
(2) The technique GA exhibits very poor performance, which is not expected since search-based approaches are slow but generally very effective and more accurate than ranking-based methods. A possible cause of this behavior might be related to the size of the ensemble found by GA that will be investigated in Section 5.3.

5.3. *Pruning ratio*

Table 3 reports the pruning ratios obtained by GA, OO and ISGEP on each dataset. We specify in the last row the average pruning ratios over all datasets. We excluded BB, BESTN, CC, MDSQ, RE, and SDP from this comparison because these approaches yield the same pruning ratio as ISGEP.

Table 3 indicates that GA achieves the best pruning ratio followed by ISGEP and OO. In addition, the reported results support our previous claim with regard to the behavior of GA (refer to Section 5.2): the analysis of both Tables 2 and 3 reveals that GA fails to extract the appropriate number of classifiers, which causes a drop in its performance.

5.4. *Pruning time*

Table 4 presents the average pruning time (in milliseconds) required by each ensemble technique over all datasets. We conducted the experiments on a desktop with 3.6 GHZ Intel Core $i7 - 4790$ processor with 8 GB of system memory.

The ensemble techniques BESTN and OO yield the lowest pruning times, whereas the second-best result is attributed to ISGEP, BB, and MDSQ. Although the proposed approach does not achieve the best running time, it succeeds in extracting very accurate sub-ensembles, requiring relatively low computational costs. Furthermore, as one should expect, search-based schemes GA and SDP deliver the worst pruning times.

Table 3. Pruning ratios.

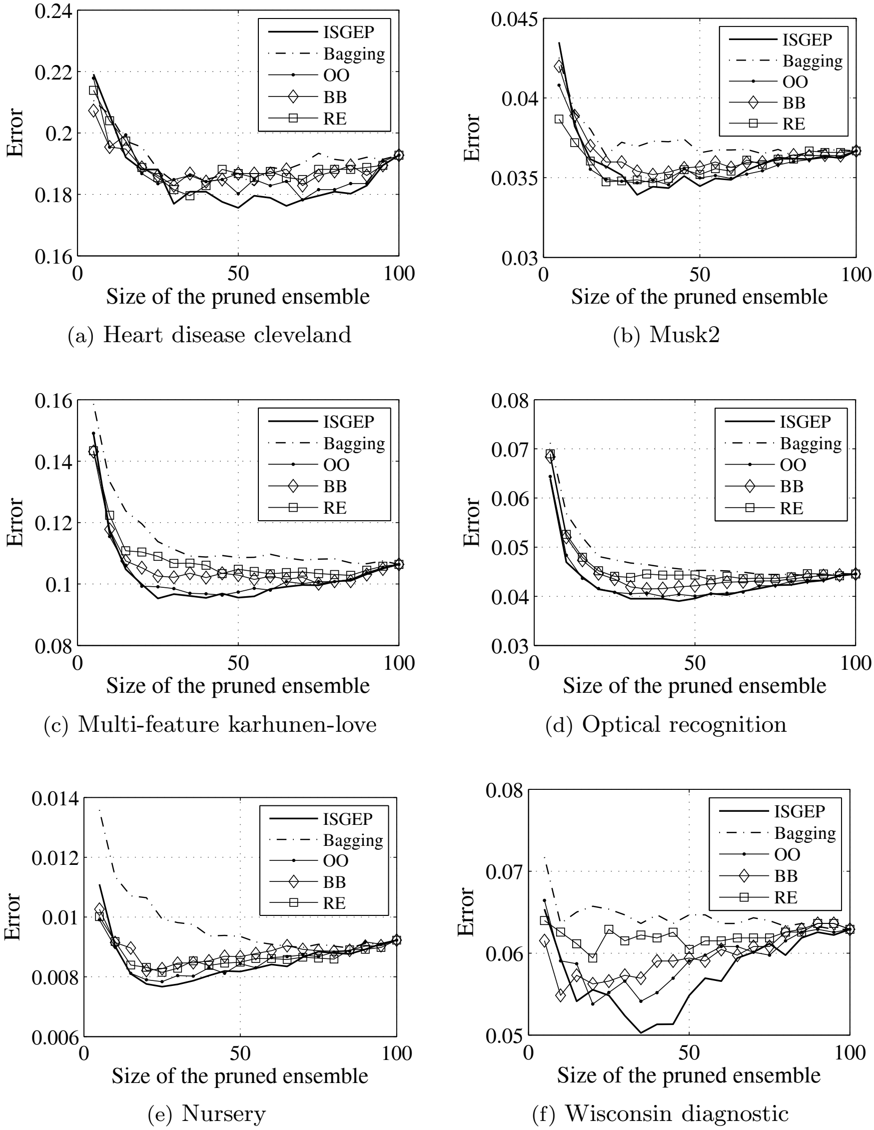
Datasets	GA	OO	ISGEP
D01	69.60±10.08	44.90±3.93	47.40±2.63
D02	76.90±14.26	46.70±4.16	50.50±3.92
D03	80.50±11.40	49.70±4.40	55.50±4.06
D04	69.00±10.68	47.10±2.77	48.20±2.15
D05	78.40±10.73	47.80±2.74	50.50±2.07
D06	64.00±15.64	48.90±2.69	54.50±3.47
D07	64.80±15.82	48.30±3.80	52.00±3.13
D08	80.20±14.09	47.00±2.67	50.60±4.38
D09	60.70±10.34	46.00±2.26	48.10±3.73
D10	79.70±10.56	48.20±3.52	54.50±6.62
D11	83.30±8.33	52.10±3.31	54.70±3.06
D12	75.10±10.66	48.30±8.19	53.50±6.33
D13	77.30±12.43	58.30±11.83	64.10±4.79
D14	51.00±3.53	45.70±2.67	48.70±3.62
D15	72.00±15.30	49.00±2.62	52.50±3.72
D16	54.90±7.98	43.10±2.33	46.50±2.95
D17	61.90±12.16	42.00±4.22	46.00±3.23
D18	58.90±11.47	45.90±3.38	48.10±2.85
D19	56.60±9.75	45.40±3.20	47.70±3.09
D20	48.00±3.53	45.40±2.50	47.10±3.35
D21	66.90±12.69	48.30±3.68	48.80±3.08
D22	77.00±7.36	50.40±3.53	53.80±3.22
D23	70.60±10.29	47.70±2.58	50.80±2.62
D24	54.00±8.00	50.60±2.07	50.50±4.14
D25	56.00±10.97	50.70±2.21	51.70±5.33
D26	71.50±11.57	49.90±2.85	50.20±3.74
D27	83.70±8.38	46.20±5.20	59.30±4.40
D28	76.00±11.09	51.70±2.87	53.40±3.44
D29	81.60±15.03	49.30±4.64	55.40±2.22
D30	54.20±7.07	43.60±3.24	45.70±3.62
D31	52.20±4.13	43.90±2.02	49.00±1.83
D32	73.40±9.13	49.10±3.54	52.00±5.16
D33	79.90±14.00	50.50±3.92	51.80±4.29
D34	76.60±10.01	46.60±4.81	50.50±3.10
D35	56.40±8.63	46.00±3.83	46.10±3.07
Mean	**68.37±10.49**	**47.84±3.66**	**51.13±3.61**

Table 4. Average pruning time (in milliseconds).

GA	BB	OO	BESTN	CC	MDSQ	RE	SDP	ISGEP
56560	33.8	3.08	1.70	106	37.2	6420	223	19.5

5.5. *Effect of the number of selected classifiers on the classification accuracy*

This section is devoted to investigate how the size of the pruned ensemble L influences the performance of ISGEP, OO, BB, and RE. We carried out the following experiment: We varied L from 5 to 100, and plotted the error curves for six



(a) Heart disease cleveland

(b) Musk2

(c) Multi-feature karhunen-love

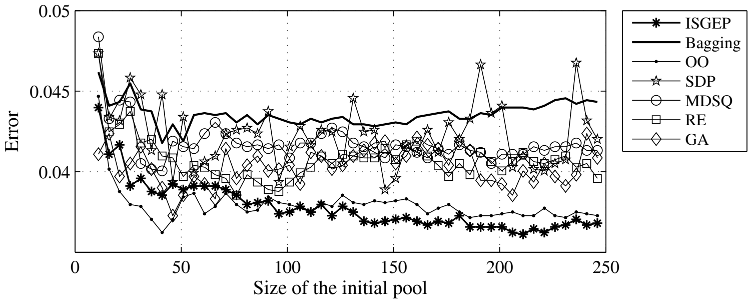
(d) Optical recognition

(e) Nursery

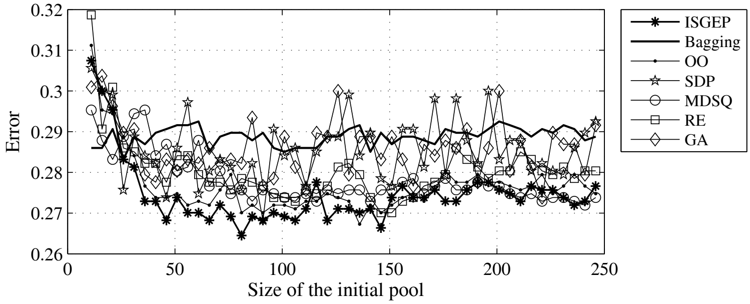
(f) Wisconsin diagnostic

Fig. 4. Test error curves of various ensemble approaches.

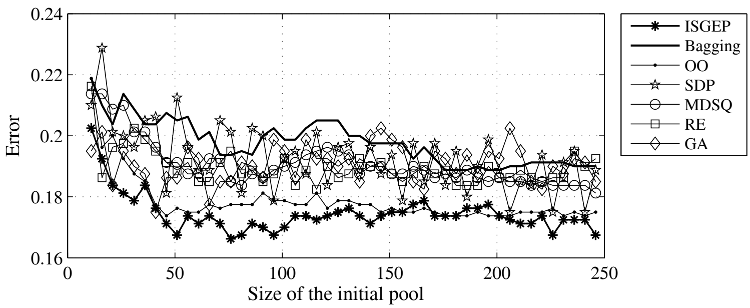
datasets: Heart disease cleveland, Musk2, Multi-feature karhunen-love, Optical recognition, Nursery, and Wisconsin diagnostic breast cancer. We also reported the error curves of Bagging. To this end, we aggregated the individual classifiers in the same order as they were included in the initial ensemble. We generated the ensemble members using the training fold and estimated the error rates on the test fold. The reported results are averaged measurements of the 10 partitions of the six datasets. The results of this experiment are illustrated by Figure 4.



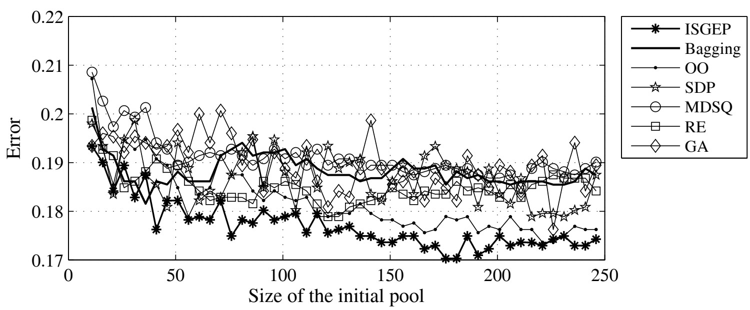
(a) Car evaluation



(b) Glass identification



(c) Hayes-Roth



(d) Heart disease cleveland

Fig. 5. Variations of accuracy with respect to the size of the initial pool of base learners.

The curves shown by Figure 4 indicate that, in case of unordered Bagging, the test error rate decreases as the number of selected classifiers increases; then it settles at a certain rate and keeps it with little variations. On the other hand, the error curves obtained by the pruning techniques drop rapidly and attain lower rates than Bagging; after that, they increase until reaching the error rate of the entire ensemble. Particularly, as reported by Figures 4(c)–(e), we notice that Isgep and OO exhibit comparable performance. Furthermore, overall Isgep achieves better error rates than the other alternatives.

5.6. *Effect of the size of the initial pool of classifiers*

Figure 5 shows the influence of the initial ensemble size on the performance of seven pruning approaches namely Isgep, Bagging, Sdp, OO, Mdsq, Re, and Ga for several problem sets.[a] Due to the high computational cost required to perform this experiment with all datasets, we selected only four classification problems: Car evaluation, Glass identification, Hayes-Roth, and Heart disease cleveland. The curves were produced by evaluating the error rate for different sizes of the initial pool of classifiers between 10 and 250. For each ensemble size L, we first built a committee composed of L Cart trees using Bagging on the training fold; then, we recorded the error rates estimated on the test fold. At the end, we reported only averaged measurements of the 10 partitions of these datasets. It is worth noting that we set the pruned ensemble size of Sdp, Mdsq, and Re to the same value found by our approach.

From Figure 5, we can generally note that the pruning approaches Isgep, OO, Mdsq, and Re outperform the unordered Bagging (except for the performance of Mdsq on the Heart disease cleveland data set). In particular, the error curves of orientation ordering and the proposed technique fall below the other alternatives. Additionally, as pointed out in Section 5.5, Isgep and OO show comparable performance, with Isgep being consistently superior on all problem sets.

6. Conclusion

This paper introduced Isgep, a novel ensemble pruning methodology using Shapley value. The proposed framework measures the contribution of the ensemble members by considering the tradeoff between the individual accuracies and the diversity of the group. The focus is on accurate members that contribute considerably to the overall ensemble diversity. The experimental results revealed that Isgep significantly improves the original ensemble and performs better than some major state-of-the-art selection techniques.

[a]We would like to thank the anonymous reviewer for suggesting us to carry out this experiment.

Acknowledgment

The authors would like to thank the action editor and the anonymous reviewers for their constructive comments and valuable suggestions which have helped us to improve the quality of this manuscript.

References

1. C.-X. Zhang, J.-S. Zhang, N.-N. Ji and G. Guo, Learning ensemble classifiers via restricted Boltzmann machines, *Pattern Recognition Letters* **36** (2014) 161–170.
2. 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* **31**(2) (2009) 245–259.
3. L. Rokach, Collective-agreement-based pruning of ensembles, *Computational Statistics and Data Analysis* **53**(4) (2009) 1015–1026.
4. Z.-H. Zhou, *Ensemble Methods: Foundations and Algorithms*, 1st edn. (Taylor & Francis, Boca Raton, FL, 2012).
5. M. Han and B. Liu, Ensemble of extreme learning machine for remote sensing image classification, *Neurocomputing* **149** (2015) 65–70.
6. A. Mashhoori, Block-wise two-directional 2DPCA with ensemble learning for face recognition, *Neurocomputing* **108** (2013) 111–117.
7. B. Kavitha, S. Karthikeyan and P. S. Maybell, An ensemble design of intrusion detection system for handling uncertainty using neutrosophic logic classifier, *Knowledge-Based Systems* **28** (2012) 88–96.
8. L. Rokach, *Pattern Classification Using Ensemble Methods*, 1st edn. (World Scientific Publishing Company, Singapore, 2010).
9. N. García-Pedrajas and D. Ortiz-Boyer, Boosting random subspace method, *Neural Networks* **21**(9) (2008) 1344–1362.
10. S. González, F. Herrera and S. García, Monotonic random forest with an ensemble pruning mechanism based on the degree of monotonicity, *New Generation Computing* **33**(4) (2015) 367–388.
11. J. Meynet and J.-P. Thiran, Information theoretic combination of pattern classifiers, *Pattern Recognition* **43**(10) (2010) 3412–3421.
12. Z. Lu, X. Wu, X. Zhu and J. Bongard, Ensemble pruning via individual contribution ordering, in *Int. Conf. on Knowledge Discovery and Data Mining* (2010), pp. 871–880.
13. G. Giacinto, F. Roli and G. Fumera, Design of effective multiple classifier systems by clustering of classifiers, in *Int. Conf. on Pattern Recognition* (2000), pp. 160–163.
14. Z.-H. Zhou, J.-X. Wu, Y. Jiang and S.-F. Chen, Genetic algorithm based selective neural network ensemble, in *Int. Joint Conf. on Artificial Intelligence* (2001), pp. 797–802.
15. Y. Zhang, S. Burer and W. N. Street, Ensemble pruning via semi-definite programming, *Journal of Machine Learning Research* **7** (2006) 1315–1338.
16. X. Sun, Y. Liu, J. Li, J. Zhu, H. Chen and X. Liu, Feature evaluation and selection with cooperative game theory, *Pattern Recognition* **45**(8) (2012) 2992–3002.
17. V. K. Garg, Y. Narahari and M. Narasimha Murty, Novel biobjective clustering (BiGC) based on cooperative game theory, *IEEE Transactions on Knowledge and Data Engineering* **25**(5) (2013) 1070–1082.
18. H. Ykhlef, D. Bouchaffra and F. Ykhlef, Coalitional game-based adaboost, in *IEEE Int. Conf. on Systems, Man and Cybernetics* (2014), pp. 194–199.

19. H. Ykhlef and D. Bouchaffra, Induced subgraph game for ensemble selection, in *IEEE Int. Conf. on Tools with Artificial Intelligence* (2015), pp. 636–643.
20. D. D. Margineantu and T. G. Dietterich, Pruning adaptive boosting, in *Int. Conf. on Machine Learning* (1997), pp. 211–218.
21. G. Martínez-Muñoz and A. Suárez, Aggregation ordering in bagging, in *Int. Conf. on Artificial Intelligence and Applications* (2004), pp. 258–263.
22. G. Martínez-Muñoz and A. Suárez, Pruning in ordered bagging ensembles, in *Int. Conf. in Machine Learning* (2006), pp. 609–616.
23. L. Xu, B. Li and E. Chen, Ensemble pruning via constrained eigen-optimization, in *IEEE Int. Conf. on Data Mining* (2012), pp. 715–724.
24. A. Lazarevic and Z. Obradovic, Effective pruning of neural network classifier ensembles, in *Int. Joint Conf. on Neural Networks* (2001), pp. 796–801.
25. G. Martínez-Muñoz and A. Suárez, Using boosting to prune bagging ensembles, *Pattern Recognition Letters* **28**(1) (2007) 156–165.
26. G. Tsoumakas, L. Angelis and I. Vlahavas, Selective fusion of heterogeneous classifiers, *Intelligent Data Analysis* **9**(6) (2005) 511–525.
27. H. Ykhlef and D. Bouchaffra, An efficient ensemble pruning approach based on simple coalitional games, *Information Fusion* **34** (2017) 28–42.
28. M. J. Osborne and A. Rubinstein, *A Course in Game Theory* (MIT Press, Cambridge, 1994).
29. G. Chalkiadakis, E. Elkind and M. Wooldridge, *Computational Aspects of Cooperative Game Theory* (Morgan & Claypool Publishers, California, 2011).
30. D. B. Gillies, Solutions to general non-zero-sum games, *Contributions to the Theory of Games* **4** (1959) 47–85.
31. L. S. Shapley, A value for n-person games, *Annals of Mathematical Studies* **2** (1953) 307–317.
32. S. Ieong and Y. Shoham, Marginal contribution nets: A compact representation scheme for coalitional games, in *ACM Conference on Electronic Commerce* (2005), pp. 193–202.
33. E. Kalai and E. Zemel, Totally balanced games and games of flow, *Mathematics of Operations Research* **7**(3) (1982) 476–478.
34. X. Deng and C. H. Papadimitriou, On the complexity of cooperative solution concepts, *Mathematics of Operations Research* **19**(2) (1994) 257–266.
35. V. Conitzer and T. Sandholm, Complexity of constructing solutions in the core based on synergies among coalitions, *Artificial Intelligence* **170**(6-7) (2006) 607–619.
36. K. Bache and M. Lichman, UCI Machine Learning Repository (2015), http://archive.ics.uci.edu/ml.
37. I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, 3rd edn. (Morgan Kaufmann Publishers, California, 2011).
38. M. Yamashita, K. Fujisawa, M. Fukuda, K. Kobayashi, K. Nakta and M. Nakata, Latest developments in the SDPA family for solving large-scale SDPs, in *Handbook on Semidefinite, Cone and Polynomial Optimization: Theory, Algorithms, Software and Applications* (Springer, New York, USA, 2011) pp. 687–714.