A New Ensemble Pruning Method Based on Margin and Diversity

Zixiong Shen[1] and Xingcheng Liu[1,2,3(✉)]

[1] School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou 510006, China
isslxc@mail.sysu.edu.cn
[2] School of Information Science, Guangzhou Xinhua University, Guangzhou 510520, China
[3] Southern Marine Science and Engineering Guangdong Laboratory (Zhuhai), Zhuhai 519082, China

Abstract. Classification is one of the main tasks of machine learning, and ensemble learning has become a successful paradigm in the data classification field. This work aims to present a new method for pruning an ensemble classification model based on margin theory and ensemble diversity. Firstly, a new unsupervised form of instances margin metric is proposed, which does not need to consider the true class labels of the instances. This mechanism can improve the robustness of the algorithm against mislabeled noise instances. Then, the Jensen-Shannon (J-S) divergence between the classifiers is calculated based on the probability distribution of the class labels. Finally, all base classifiers are ordered with respect to a new criterion which combines the obtained margin values and the J-S divergence of base classifiers. Experiments show that the proposed method has a stable improvement on a significant proportion of benchmark datasets over existing ensemble pruning methods.

Keywords: Ensemble learning · Classification · Multiple classifier systems

1 Introduction

Ensemble learning is an important branch in the field of machine learning, which trains multiple base models explicitly or implicitly from data. As a mainstream machine learning paradigm, ensemble learning uses the strategy of "perturb and combine" to train multiple base classifiers, that is, randomly perturb the samples space or features space and randomly adjust the parameters of the base classifiers [1]. Individual classifiers may only focus on part of the information on

This work was supported by the Key Project of NSFC-Guangdong Province Joint Program (Grant No. U2001204), the National Natural Science Foundation of China (Grant Nos. 61873290 and 61972431), and the Science and Technology Program of Guangzhou, China (Grant No. 202002030470).

J. Xiong et al. (Eds.): MobiMedia 2021, LNICST 394, pp. 689–701, 2021.
https://doi.org/10.1007/978-3-030-89814-4_50

the decision boundary, which leads to certain limitations when making classification decisions. If the predicted information of multiple base classifiers can be integrated, a more reasonable classification result can be obtained. At present, many theoretical analyses and experiments have proved that integrating numerous base classifiers to train data can overcome the limitations of a single base classifier, improve the generalization performance of the original base classifiers, thus improve the classification accuracy [2].

However, training a large number of base classifiers requires additional storage resources, and the consumption of computing resources has also become a problem that can not be ignored in ensemble learning. Besides, it is not the fact that all classifiers used in the ensemble system can make the final classification results better. The probability of having highly similar base classifiers increases as the scale of ensemble model expands, and the accuracy of the entire ensemble classification system will also decrease with the increase of bad base classifiers. Zhou [3] has pruned the parallel ensemble methods in his research work and found that it can achieve better generalization performance with a smaller-scale ensemble. Some unnecessary base classifiers in the ensemble system are eliminated by a certain method, so that the generalization performance after pruning is better than the ensemble of all base classifiers before pruning. This is the so-called ensemble pruning, also called selective ensemble or ensemble selection [3]. The prerequisite for ensemble pruning is that all base classifiers have been generated, and no new base classifiers will be generated during the construction process. This is different from the classic serial ensemble learning method, which generates individual classifiers one by one during the training process, but ensemble pruning may discard any base classifier that has been generated [4].

The existing ensemble pruning methods can be mainly divided into three categories: sorting-based ensemble pruning, clustering-based ensemble pruning, and optimization-based ensemble pruning [5]. Sorting-based ensemble pruning method sorts individual classifiers in descending order based on a predetermined criterion, such as classification accuracy, diversity of the ensemble model. Next, the top-ranked base classifiers will be added to the final ensemble set [6,7]. The advantage of the sorting-based method is that it has lower computational complexity, but there is currently no unified sorting criterion. Clustering-based ensemble pruning method attempts to cluster similar base classifiers based on the generalization performance of the base classifiers [8]. Some representative base classifiers close to the cluster center can be used to fit the best decision boundary. Clustering-based methods are usually classified into two steps: firstly, divide all base classifiers into multiple clusters, which involves the problem, which clustering method is to be adopted. Secondly, select the appropriate base classifiers from the clusters, which involves the problem of the pruning strategy to be used [9]. Optimization-based ensemble pruning method transforms the ensemble pruning problem into an optimization problem, where, the final ensemble set is selected through optimizing the overall generalization capability [10]. Since searching for the optimal subset directly requires a lot of calculations, this method often

resorts to optimization algorithms such as genetic algorithm, multi-objective optimization algorithm, and hill-climbing algorithm [11].

The main challenge of ensemble pruning is to design a practical algorithm that can reduce the ensemble scale without reducing the generalization performance [12]. At present, relevant research [13] has proved that the ensemble pruning method based on sorting is superior to the enumeration searching method, which directly selects the best subset in terms of classification accuracy and computational performance. In this paper, the concepts of instances margin [14] are applied to ensemble pruning, and the fact is that when ensemble pruning is performed, the instances with small margin values should be the main concern [15]. The performance of a base classifier is evaluated by those instances with small margin values. Besides, because the diversity of the classifier set is also an issue in the process of constructing an ensemble model, a new measuring criterion for pairwise difference is constructed to measure the diversity of the classifiers set. All the individual base classifiers are sorted by a predefined criteria, and the top-ranked base classifiers are incorporated into the final ensemble classifiers subset, so this method has higher computational efficiency than other state-of-the-art methods.

The rest of this paper is organized as follows. Section 2 presents the proposed ensemble pruning methodology. Section 3 gives the details of the experimental setup, results and comparative analysis. Discussions and concluding remarks are given in Sect. 4.

2 Proposed Method

In this section, a new ensemble pruning method based on margin and diversity named EPMD is proposed, which eliminates the useless base classifiers while improving the final accuracy of the ensemble learning framework. In order to solve the data classification problem effectively, the sample points near the classification decision boundary are more inclined to be focused on. Part of the original dataset is used as a training dataset to generate the base classifiers pool, and then a validation dataset is used to evaluate the generated base classifiers based on the proposed heuristic metrics. After obtaining the simplified subset of ensemble classifiers, classification tests are performed on the unused test dataset to obtain the final classification results.

2.1 Generate Base Classifier Pool

Suppose the initial dataset is a matrix of dimension $N \times n$: $D = \{\boldsymbol{x}_i, y_i)|i = 1, 2 \cdots, N\}$, including N samples $\boldsymbol{x}_i$ and N true class labels y_i, $y_i \in \{1, 2, \cdots, L\}$, that is, there are L classes in the original dataset. Each sample point $\boldsymbol{x}_i$ is a d-dimensional feature vector; $H = \{h_t|t = 1, 2, \cdots, T\}$ is a classifiers pool containing totally T base classifiers, each of which is equivalent to a mapping function of $\boldsymbol{x}_i : y_i^{'} = h_t(\boldsymbol{x}_i)$, and $y_i^{'}$ is the predicted class label.

Firstly, leave-m-out cross-validation is utilized to divide the initial data set into three equal parts, which are used as training dataset $D_{tr} \in \mathbb{R}^{N^{'} \times n}$, validation dataset $D_{va} \in \mathbb{R}^{N^{'} \times n}$ and test dataset $D_{te} \in \mathbb{R}^{N^{'} \times n}$. The operation steps here are basically the same as the Bagging algorithm [16]. For the training dataset $D_{tr} \in \mathbb{R}^{N^{'} \times n}$, Bootstrap [17] method is used to perform m random sampling with replacement. This work will be repeated until the number of samples in each sample set is the same as in the initial training dataset before sampling. After repeating T rounds of operation to obtain T sample sets $D_{tr\_t}(1 \leqslant t \leqslant T)$, the sampled training subsets are different from each other, and $|D_{tr\_t}| = |D_{tr}|$.

Next, Classification and Regression Tree (CART) [18] is utilized to train each training dataset D_{tr_t}, then obtain the ensemble classifiers $ES = \{h_1, h_2, \cdots, h_T\}$, and are added to the base classifier pool. The type of base classifiers used here is not unique, CART tree is utilized here because it is more sensitive to the perturbation of the input data, it is easier to produce diversified base classifiers and the computational complexity is not high.

2.2 Base Classifier Evaluation

Each base classifier of the ensemble system ES is used to classify the validation dataset samples, and the majority voting is used here to obtain the prediction results matrix of the validation dataset:

$$Mat = [\boldsymbol{R}_1, \boldsymbol{R}_2, \cdots, \boldsymbol{R}_t, \cdots, \boldsymbol{R}_T] \in \mathbb{R}^{N^{'} \times T}, \tag{1}$$

where $\boldsymbol{R}_t = [C_t(\boldsymbol{x}_1), C_t(\boldsymbol{x}_2), \cdots, C_t(\boldsymbol{x}_i), \cdots, C_t(\boldsymbol{x}_{N'})]$ is the vector formed by the classification results of the t-th base classifier in the ensemble system ES.

According to the classification results matrix Mat, the votes number matrix $Vote \in \mathbb{R}^{N^{'} \times L}$ of each data sample belonging to each class in the validation dataset calculated (that is, the number of all base classifiers that classify the data samples into a certain class). Sort the row elements of the votes number matrix Vote in descending order, and get the sorted votes number vector $\boldsymbol{v}(x_i) = [v_{c_1}, v_{c_2}, \cdots, v_{c_L}] \in \mathbb{R}^{1 \times L}$ for each data sample $\boldsymbol{x}_i$ in the validation dataset.

A new unsupervised form of instances margin metric is proposed here to eliminate useless weak classifiers:

$$\begin{aligned} margin(\boldsymbol{x}_i, y_i) = & \frac{1}{N} \cdot \frac{1}{\sum_{l=1}^{L}(v_{c_l})} \cdot \\ & \sqrt{(v_{c_1} - v_{c_2})^2 + (v_{c_2} - v_{c_3})^2 + \cdots + (v_{c_{L-1}} - v_{c_L})^2}. \end{aligned} \tag{2}$$

For a sample point $(\boldsymbol{x}_i, y_i)$ in the validation dataset, v_{c_1} represents the number of votes of the class with the most votes, that is, the vast majority of base classifiers in the ensemble system classify and predict the sample $(\boldsymbol{x}_i, y_i)$ into class c_1, and v_{c_2} represents the number of votes for the class with the second most votes, and so on, v_{c_L} represents the number of votes for the class label with the least number of votes.

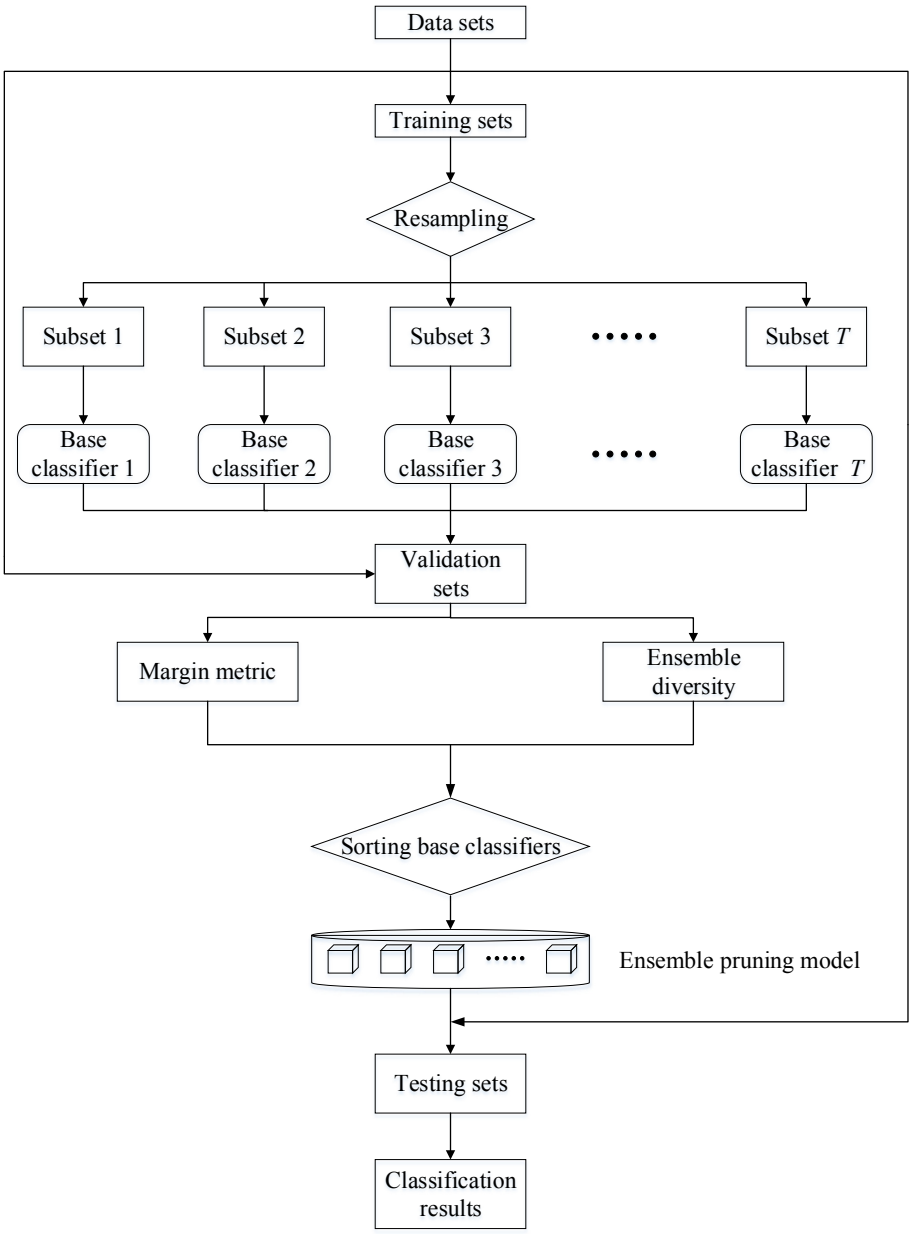

Fig. 1. Generation framework of the proposed method EPMD.

Next, the classification results matrix Mat is compared with the true class labels vector to find all the data points D_{va_t} that are correctly classified. For each classifier in the base classifiers pool, the number of validation dataset samples that are correctly classified and predicted is counted by formula (3):

$$N_R(h_t) = \sum_{i=1}^{N'} \Psi(C_t(\boldsymbol{x}_i), y_i), \tag{3}$$

where $I(true) = 1, I(false) = 0$.

Based on each classifier h_t in the base classifiers pool, the average margin value is calculated by formula (4):

$$\Phi(h_t) = \frac{1}{N_R(h_t)} \cdot \sum_{(\boldsymbol{x}_i, y_i) \in D_{va\_t}} margin(\boldsymbol{x}_i, y_i). \tag{4}$$

Considering that the existing selective ensemble learning algorithms based on margin values rarely involve the diversity between base classifiers, the Jensen-Shannon (J-S) divergence [19] is employed here from the perspective of information theory. For the classification results of each base classifier in the base classifiers pool, its probability distribution is calculated with respect to the class labels, and thereby the J-S divergence is obtained. The J-S divergence measures the degree of difference between the probability distributions of the classification results of different classifiers, eliminates base classifiers with low diversity, and improves the overall diversity of the ensemble system.

Let $p = \{p_1, p_2, \ldots, p_K\}$ and $q = \{q_1, q_2, \ldots, q_K\}$ be the two probability distributions on the random variable $\boldsymbol{X}$, where K is the number of discrete random variables. Then the J-S divergence between the probability distributions P and Q is defined as:

$$JS(\boldsymbol{p}, \boldsymbol{q}) = \frac{1}{2}\Big[S\Big((p), \frac{\boldsymbol{p}+\boldsymbol{q}}{2}\Big) + S\Big((q), \frac{\boldsymbol{p}+\boldsymbol{q}}{2}\Big)\Big], \tag{5}$$

where S is the Kullback-Leibler divergence (K-L) divergence between the two probability distributions:

$$S(\boldsymbol{p}, \boldsymbol{q}) = \sum_k p_k \cdot \log \frac{p_k}{q_k} \quad , \ (k = 1, 2, \cdots, K). \tag{6}$$

The J-S divergence can be obtained by the formulas (5) and (6), as shown in formula (7):

$$\begin{aligned} JS(\boldsymbol{p}, \boldsymbol{q}) &= H\Big(\frac{\boldsymbol{p}+\boldsymbol{q}}{2}\Big) - \frac{1}{2}H(\boldsymbol{p}) - \frac{1}{2}H(\boldsymbol{q}) \\ &= \frac{1}{2}\Big[\sum_k p_k \cdot \log\Big(\frac{2p_k}{p_k+q_k}\Big) + \sum_k q_k \cdot \log\Big(\frac{2q_k}{p_k+q_k}\Big)\Big]. \end{aligned} \tag{7}$$

The classification prediction result of a certain classifier h_t in the base classifier pool:

$$\boldsymbol{R}_t = [C_t(\boldsymbol{x}_1), C_t(\boldsymbol{x}_2), \cdots, C_t(\boldsymbol{x}_i), \cdots, C_t(\boldsymbol{x}_{N'})]. \tag{8}$$

Calculate the probability distribution of the class labels of the original dataset:

$$\boldsymbol{p}_t = (p_1, p_2, \cdots, p_l, \cdots, p_L)^T, \tag{9}$$

where p_l is the probability distribution about class label l:

$$p_l = \sum_{i=1}^{N'} I(C_t(\boldsymbol{x}_i) = l)/N'. \tag{10}$$

The J-S divergence between two different classifiers (Classifiers Jensen-Shannon divergence) is obtained by formula (7) as follows:

$$CJS(\boldsymbol{P_1}, \boldsymbol{P_2}) = \frac{1}{2}\Big[\sum_{l=1}^{L} p_l \cdot \log\Big(\frac{2p_l}{p_l + q_l}\Big) + \sum_{l=1}^{L} q_l \cdot \log\Big(\frac{2q_l}{p_l + q_l}\Big)\Big], \tag{11}$$

$$\boldsymbol{CJS} = \begin{bmatrix} CJS_{11} & CJS_{12} & \cdots & CJS_{1T} \\ CJS_{21} & CJS_{22} & \cdots & CJS_{2T} \\ \vdots & \vdots & \ddots & \vdots \\ CJS_{T1} & CJS_{T2} & \cdots & CJS_{TT} \end{bmatrix}. \tag{12}$$

When the J-S divergence between two different classifiers in the base classifier pool is larger, it indicates that the information difference between the probability distributions of the corresponding classification results is greater. Then the average degree of difference between the t-th base classifier and other base classifiers is defined as:

$$\widetilde{CJS_t} = \frac{\sum_{s=1, s\neq t}^{T} CJS_{st}}{T-1}. \tag{13}$$

The greater the average degree of difference between a certain base classifier and other base classifiers, the greater the diversity contribution that the base classifier makes to the ensemble system. In order to take average margin and diversity contribution of the base classifier into consideration simultaneously, a trade-off between margin and diversity (TMD) is defined as an objective function:

$$TMD(h_t) = \lambda \cdot \Big(\frac{1}{\Phi(h_t)}\Big) + (1-\lambda) \cdot \log(1 + e^{-\widetilde{CJS_t}}), \tag{14}$$

where $\lambda \in [0, 1]$ is a regularization factor, which is used to balance the importance of these two classifiers metrics $\Phi(h_t)$ and $\widetilde{CJS_t}$. By sorting all the classifiers in the base classifier pool in descending order according to the obtained value, a new base classifier sequence can be obtained: $ES' = \{h_1', h_2', \cdots, h_t', \cdots, h_T'\}$, which satisfies $TMD(h_{t-1}') \geqslant TMD(h_t')$, $0 \leqslant t \leqslant T$. The higher the ranking of the base classifier, the larger the value, and it is considered to have better generalization performance.

By selecting the first S base classifiers that can maximize the average classification accuracy of the ensemble system on the validation dataset, a selective ensemble classifiers subset is obtained as follows:

$$ES_{new} = \underset{S,(1 \leqslant S \leqslant T)}{arg\ max}\ \ accuracy(ES^{'}). \tag{15}$$

Finally, the ensemble pruning task has been modeled as an objective function value ordering problem as shown in (15), the pruned classifiers subset is used to predict the test dataset and then the final classification results are obtained. The flowchart of novel method for ensemble pruning proposed in this paper is shown in Fig. 1.

3 Experiments and Results

In this section, several experiments are presented to evaluate the performance of the proposed ensemble pruning method.

3.1 Datasets and Experimental Setup

Table 1. Summary of UCI repository datasets used in experiments

Datasets	# of samples	# of features	# of classes	Division ratio
Glass	214	9	6	70:70:70
Zoo	101	16	7	33:33:33
Air	359	64	3	120:120:119
Hayesroth	160	4	3	53:53:53
Appendicitis	106	7	2	35:35:35
M-of-N	1000	13	2	333:333:333
Car	1728	6	4	576:576:576
Ecoli	336	7	8	112:112:112
DNA test	1186	12	3	395:395:395
Tic-tac-toe	958	9	2	319:319:319
Seeds	210	7	3	70:70:70
Segment	2310	18	7	770:770:770
Tae	151	5	3	50:50:50
Vowel	528	10	11	176:176:176
Wdbc	569	30	2	189:189:189
Wpbc	198	25	2	66:66:66
Breast-w	699	10	2	233:233:233
X8D5K	1000	8	5	333:333:333
Penbased	10992	16	10	3600:3600:3600
Phoneme	5404	5	2	1800:1800:1800
Ringnorm	7400	20	2	2400:2400:2400
Spambase	4597	24	2	1532:1532:1532

Table 2. Average accuracy by the ensemble pruning methods and by complete bagging on test datasets

Datasets	Bagging (%)	SEMD (proposed) (%)	MDEP (%)	UMEP (%)	COMEP (%)
Glass	0.6595 (0.0625)	**0.6824** (0.0276)	0.6567 (0.0684)	0.6614 (0.0598)	0.6678 (0.0527)
Zoo	0.8220 (0.0936)	**0.8716** (0.0709)	0.8347 (0.0796)	0.8512 (0.0710)	0.8486 (0.0762)
Air	0.8282 (0.0490)	**0.8453** (0.0504)	0.8218 (0.0486)	0.8310 (0.0476)	0.8247 (0.0471)
Hayesroth	0.7294 (0.0881)	0.7709 (0.0707)	0.7377 (0.0783)	0.7666 (0.0712)	**0.7879** (0.0614)
Appendicitis	0.8314 (0.0625)	**0.8377** (0.0580)	0.8303 (0.0641)	0.8337 (0.0643)	0.8354 (0.0638)
M-of-N	0.9157 (0.0277)	**0.9484** (0.0218)	0.9230 (0.0265)	0.9274 (0.0263)	0.9421 (0.0227)
Car	0.9373 (0.0148)	**0.9442** (0.0139)	0.9392 (0.0139)	0.9401 (0.0152)	0.9420 (0.0131)
Ecoli	0.7945 (0.0410)	**0.7994** (0.0379)	0.7950 (0.0420)	0.7963 (0.0406)	0.7956 (0.0379)
DNA test	0.9057 (0.0168)	**0.9122** (0.0188)	0.9040 (0.0167)	0.9074 (0.0182)	0.9065 (0.0180)
Tic-tac-toe	0.8655 (0.0227)	**0.8948** (0.0193)	0.8696 (0.0245)	0.8663 (0.0253)	0.8867 (0.0209)
Seeds	0.8840 (0.0471)	**0.8907** (0.0428)	0.8831 (0.0475)	0.8840 (0.0464)	0.8859 (0.0471)
Segment	0.9524 (0.0100)	**0.9557** (0.0093)	0.9519 (0.0102)	0.9537 (0.0100)	0.9553 (0.0096)
Tae	0.4814 (0.0689)	0.4912 (0.0766)	0.4778 (0.0599)	0.4904 (0.0716)	**0.4914** (0.0706)
Vowel	0.7197 (0.0410)	**0.7362** (0.0373)	0.7124 (0.0435)	0.7165 (0.0412)	0.7150 (0.0390)
Wdbc	0.9385 (0.0192)	0.9379 (0.0214)	**0.9386** (0.0175)	0.9371 (0.0209)	0.9366 (0.0196)
Wpbc	0.7239 (0.0523)	**0.7345** (0.0459)	0.7244 (0.0482)	0.7279 (0.0500)	0.7274 (0.0542)
Breast-w	**0.9594** (0.0125)	0.9559 (0.0127)	0.9582 (0.0125)	0.9554 (0.0144)	0.9545 (0.0127)
X8D5K	0.9917 (0.0071)	**0.9937** (0.0066)	0.9932 (0.0059)	0.9931 (0.0068)	0.9929 (0.0063)
Penbased	0.9732 (0.0039)	**0.9755** (0.0035)	0.9728 (0.0041)	0.9729 (0.0042)	0.9736 (0.0036)
Phoneme	0.8695 (0.0081)	**0.8709** (0.0083)	0.8687 (0.0079)	0.8685 (0.0079)	0.8697 (0.0091)
Ringnorm	0.9461 (0.0067)	**0.9506** (0.0055)	0.9457 (0.0067)	0.9456 (0.0065)	0.9453 (0.0067)
Spambase	0.9288 (0.0073)	**0.9310** (0.0062)	0.9287 (0.0078)	0.9282 (0.0074)	0.9290 (0.0071)

The experiments are run on 22 randomly selected datasets from UCI Machine Learning repository [20] and Keel repository [21], these datasets are selected for comparative analysis since they are usually widely used in the similar ensemble learning methods. Table 1 gives a summary of these datasets. In the experiments, a dataset is randomly divided into three equal parts by cross-validation, which are training dataset, validation dataset and test dataset. CART tree [18] is utilized as the base classifier, which comes from the Classregtree classifier implemented in MATLAB 2016a. The initial ensemble classifiers scale is set to 200 base classifiers; the results of 100 repeated experiments are averaged to obtain the final classification accuracy. In order to ensure the fairness of the experiments, the division ratio of each experimental dataset remains the same when different classification methods are performed on the same dataset, which ensures that the training dataset, validation dataset, and test dataset are the same for each classification method.

3.2 Comparative Analysis of the Proposed Method to the State-of-the-Art Ensemble Learning Methods

Four algorithms are selected for comparison with the proposed method EPMD in the experiments, all of which are ensemble learning classification algorithms. Among them, Bagging [16] is the most classic ensemble learning algorithm without ensemble pruning; both UMEP [6] and MDEP [22] are ensemble pruning algorithms that use margin theory for selective ensemble; COMEP [23] is a selective ensemble algorithm that uses the normalized variation of information and the normalized mutual information to sort and select ensemble classifiers. In the experiments, for convenience, α in MDEP is set to 0.2 according to the original paper, and λ in the algorithm proposed by this paper and COMEP are both set to 0.2. For different data sets, different λ values will have slightly different results.

Average classification accuracies of five ensemble learning methods on 22 test datasets are given in Table 2, the results with better performance of the proposed method are highlighted in bold. For most datasets, the method proposed by this paper can show better classification performance compared with Bagging and other three ensemble pruning methods. In addition, this experiment also calculated the size of the classifiers subset after ensemble pruning, and compared the four selective ensemble classification algorithms. The running speed of the ensemble learning method mainly depends on the complexity and number of the base classifiers in the ensemble system; for algorithms that uniformly use the same base classifiers, minimizing the scale of the ensemble system can reduce the running time and storage overhead. After selective ensemble, the average number of classifiers in the ensemble classifier subset obtained by the four ensemble pruning methods based on sorting is shown in Table 3. Our proposed method is slightly higher than the COMEP method in the classifier scale after selective ensemble, but the overall gap is not big, and the ensemble scale is significantly smaller than the other two classification algorithms UMEP and MDEP. It reveals that using the method of our proposed method can significantly reduce the number of classifiers in the ensemble system and then reduce the computational cost.

The time complexity of the proposed method can be simply expressed as: $\mathcal{O}(T \times m \times log(m) \times n) + \mathcal{O}(T \times N^{'}) + \mathcal{O}(T \times log(T)) + \mathcal{O}(T)$, where m is the number of samples in each sampling subset, and the other symbols here have the same meaning as before. Since the number of samples, $N^{'}$, and the number of features, n, are both fixed values, the final time complexity can be approximately expressed as: $\mathcal{O}(T \times log(T))$. It can be seen that for the same data set, the running time of the algorithm depends to a large extent on the number of base classifiers. As the number of base classifiers in the ensemble model continues to increase, the running time consumed will also become longer.

Table 3. Average number of base classifiers selected by the ensemble pruning methods on test datasets

Datasets	SEMD (proposed)	MDEP	UMEP	COMEP
Glass	20.600	30.390	24.790	**11.980**
Zoo	15.810	19.120	14.080	**8.920**
Air	20.230	38.770	42.650	**19.390**
Hayesroth	16.810	23.600	16.410	**9.040**
Appendicitis	**10.460**	18.540	13.670	18.930
M-of-N	18.650	37.800	45.400	**13.930**
Car	**14.960**	40.040	37.540	16.020
Ecoli	12.050	21.260	16.170	**10.980**
DNA test	**16.950**	40.250	37.470	19.090
Tic-tac-toe	18.200	42.090	50.220	**15.430**
Seeds	**7.170**	16.200	12.660	7.710
Segment	**14.430**	41.320	31.560	15.420
Tae	21.670	22.180	18.120	**10.950**
Vowel	**23.590**	55.230	51.010	27.650
Wdbc	**10.940**	23.280	15.680	11.410
Wpbc	12.260	20.030	17.270	**9.400**
Breast-w	**8.180**	26.040	16.620	9.020
X8D5K	**6.990**	13.140	13.480	7.380
Penbased	**28.510**	66.820	67.790	35.350
Phoneme	**24.060**	65.750	62.640	25.250
Ringnorm	**28.540**	61.190	58.960	42.080
Spambase	**23.120**	50.830	54.380	23.650

4 Conclusion

In this paper, we proposed a novel ensemble pruning algorithm or selective ensemble algorithm based on margin theory and ensemble diversity (EPMD).

This method presents a new unsupervised form of average samples margin measurement criterion, it considers the margin distance from the unknown data samples to the classification decision boundary, and then characterize the overall performance of the classifiers in the ensemble system according to the samples margin values. In addition, the J-S divergence of the classification results is calculated with respect to the probability distribution of the class labels, and used to evaluate the diversity of the base classifiers in the ensemble system. All base classifiers are sorted according to the proposed measurement criteria, and a simplified ensemble classifiers subset is generated by selecting the subset of classifiers that can maximize the overall accuracy. Our idea comes from the fact that the combination of a large number of base classifiers is not always a perfect ensemble, and relatively few excellent classifiers with diversity are sufficient to obtain the best generalization performance. Different from the ensemble pruning methods that simply pursue the maximization of accuracy, the proposed method also obtains the complementarity between the base classifiers to some extent.

To evaluate the performance of the proposed ensemble pruning algorithm, we compare it with the classic Bagging algorithm and three state-of-the-art ordering-based methods. The experiments are performed on the 22 benchmark datasets from UCI repository and KEEL repository. The results show that our proposed method has varying degrees of advantages on most datasets. The proposed method has achieved higher classification accuracy than the comparison methods on 18 benchmark datasets. In addition, the size of the pruned classifiers set is smaller than that of any other comparison methods on 13 benchmark datasets. Therefore, the proposed method can achieve relatively good generalization performance with a relatively small ensemble scale, thereby achieving the purpose of ensemble pruning. Since the experiments are only performed on a single-type base classifier, more different types of base classifiers will be further explored along with considering the impact of hyperparameters on the classification results.

References

1. Jan, Z., Verma, B.: Multicluster class-balanced ensemble. IEEE Trans. Neural Netw. Learn. Syst. **32**(3), 1014–1025 (2021)
2. Zhu, Z., Wang, Z., Li, D., Zhu, Y., Wenli, D.: Geometric structural ensemble learning for imbalanced problems. IEEE Trans. Cybern. **50**(4), 1617–1629 (2020)
3. Zhou, Z.-H., Jianxin, W., Tang, W.: Ensembling neural networks: many could be better than all. Artif. Intell. **137**(1–2), 239–263 (2002)
4. Ali, M.A., Üçüncü, D., Ataş, P.K., Akyüz, S.Ö.: Classification of motor imagery task by using novel ensemble pruning approach. IEEE Trans. Fuzzy Syst. **28**(1), 85–91 (2020)
5. Tsoumakas, G., Partalas, I., Vlahavas, I.: An ensemble pruning primer. In: Okun, O., Valentini, G. (eds.) Applications of Supervised and Unsupervised Ensemble Methods. SCI, vol. 245, pp. 1–13. Springer, Heidelberg (2009) . https://doi.org/10.1007/978-3-642-03999-7_1
6. Guo, L., Boukir, S.: Margin-based ordered aggregation for ensemble pruning. Pattern Recognit. Lett. **34**(6), 603–609 (2013)

7. Zhang, C.-X., Zhang, J.-S., Yin, Q.-Y.: A ranking-based strategy to prune variable selection ensembles. Knowl. Based Syst. **125**, 13–25 (2017)
8. Lazarevic, A., Obradovic, Z.: Effective pruning of neural network classifier ensembles. In: International Joint Conference on Neural Networks. Proceedings (Cat. No. 01CH37222), IJCNN 2001, vol. 2, pp. 796–801 (2001)
9. Onan, A., Korukoğlu, S., Bulut, H.: A hybrid ensemble pruning approach based on consensus clustering and multi-objective evolutionary algorithm for sentiment classification. Inform. Process. Manag. **53**(4), 814–833 (2017)
10. Ykhlef, H., Bouchaffra, D.: An efficient ensemble pruning approach based on simple coalitional games. Inf. Fus. **34**, 28–42 (2017)
11. Qian, C., Yu, Y., Zhou, Z.-H.: Pareto ensemble pruning. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, AAAI 2015, pp. 2935–2941. AAAI Press (2015)
12. Zhu, X., Ni, Z., Ni, L., Jin, F., Cheng, M., Li, J.: Improved discrete artificial fish swarm algorithm combined with margin distance minimization for ensemble pruning. Comput. Ind. Eng. **128**, 32–46 (2019)
13. Martínez-Muñoz, G., Hernández-Lobato, D., Suárez, A.: An analysis of ensemble pruning techniques based on ordered aggregation. IEEE Trans. Pattern Anal. Mach. Intell. **31**(2), 245–259 (2009)
14. Schapire, R.E., Freund, Y., Barlett, P., Lee, W.S.: Boosting the margin: a new explanation for the effectiveness of voting methods. In: Proceedings of the Fourteenth International Conference on Machine Learning, ICML 1997, San Francisco, CA, USA, pp. 322–330 (1997). Morgan Kaufmann Publishers Inc
15. Feng, W., Dauphin, G., Huang, W., Quan, Y., Liao, W.: New margin-based subsampling iterative technique in modified random forests for classification. Knowl. Based Syst. **182**, 104845 (2019)
16. Breiman, L.: Bagging predictors. Mach. Learn. **24**(2), 123–140 (1996)
17. Efron, B.: Bootstrap methods: another look at the Jackknife. Ann. Stat. **7**(1), 1–26 (1979)
18. Breiman, L., Friedman, J., Stone, C.J., Olshen, R.A.: Classification and Regression Trees. CRC Press, Boca Raton (1984)
19. Lin, J.: Divergence measures based on the Shannon entropy. IEEE Trans. Inf. Theory **37**(1), 145–151 (1991)
20. Bache, K., Lichman, M.: UCI machine learning repository. UCI Machine Learning Repository University of California, Irvine, School of Information and Computer Sciences, December 2013
21. Alcala-fdez, J., et al.: Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. J. Mult. Valued Log. Soft Comput. **17**(2–3), 255–287 (2011)
22. Guo, H., Liu, H., Li, R., Changan, W., Guo, Y., Mingliang, X.: Margin & diversity based ordering ensemble pruning. Neurocomputing **275**, 237–246 (2018)
23. Bian, Y., Wang, Y., Yao, Y., Chen, H.: Ensemble pruning based on objection maximization with a general distributed framework. IEEE Trans. Neural Netw. Learn. Syst. **31**(9), 3766–3774 (2020)