Cognitive Computation (2019) 11:685–696
https://doi.org/10.1007/s12559-019-09652-0

Diversity-Based Random Forests with Sample Weight Learning

Chun Yang[1,2] · **Xu-Cheng Yin**[1,3]

Received: 13 January 2019 / Accepted: 10 May 2019 / Published online: 7 June 2019

Abstract

Given a variety of classifiers, one prevalent approach in classifier ensemble is to diversely combine classifier components, i.e., diversity-based ensembles, and a lot of previous works show that these ensembles can improve classification accuracy. Random forests are one of the most important ensembles. However, most random forests approaches with diversity-related aspects focus on maximizing tree diversity while producing and training component trees. Alternatively, a novel cognitive-inspired diversity-based random forests method, diversity-based random forests via sample weight learning (DRFS), is proposed. Given numerous component trees from the original random forests, DRFS selects and combines tree classifiers adaptively via diversity learning and sample weight learning. By designing a matrix for the data distribution creatively, a unified optimization model is formulated to learn and select diverse trees, where tree weights are learned through a convex quadratic programming problem with sample weights. Moreover, a self-training algorithm is proposed to solve the convex optimization iteratively and learn sample weights automatically. Comparative experiments on 39 typical UCI classification benchmarks and a variety of real-world text categorization benchmarks of our proposed method are conducted. Extensive experiments show that our method outperforms the traditional methods. Our proposed DRFS method can select and combine tree classifiers adaptively and improves the performance on a variety of classification tasks.

Keywords Diversity-based ensembles · Classifier ensemble · Random forests · Sample weight learning · Convex quadratic programming

Introduction

There are many famous ensembles (e.g., random forests [9], bagging [8], boosting [15, 16], stacking [47], and neural network ensembles [19, 54]) and also many recent ensemble methods [2, 4, 37, 38, 46], many of which have been widely applied in numerous real-world intelligent and cybernetics applications. Many research efforts have demonstrated that the random forests [9] are one of the most important ensemble methods [1, 3, 5, 22, 31, 39]. Some previous researches show that the performance of a classifier ensemble relies on not only the accuracy but also the diversity of base classifiers [19, 24, 36, 53]. Generally speaking, an ensemble of diverse classifiers can allow us to get higher accuracy, which is often not achievable by a single model. Consequently, how to diversely combine classifiers plays an important role and becomes a main topic in random forests, and other ensemble methods.

In general, the random forests method can exploit implicit and explicit diversities together. This method combines the "Bagging" idea for instance sampling with the implicit diversity and the random selection of variables for feature selection with the explicit diversity. As discussed before, the performance of random forests always relies on not only the accuracy but also the diversity of component classifiers (decision trees). In the literature, there are many research efforts conducted for improving random forests with diversity-related aspects, most of which focus on maximizing tree diversity while producing and training component trees. For example, Liu et al. [27, 28] proposed max-diverse ensembles with probability averaging to maximizing tree diversity in random forests. Later, they [29] proposed the coalescence

✉ Xu-Cheng Yin
xuchengyin@ustb.edu.cn

Chun Yang
chunyang@ustb.edu.cn

[1] Department of Computer Science and Technology, School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China

[2] Beijing Key Laboratory of Materials Science Knowledge Engineering, School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China

[3] Institute of Artificial Intelligence, University of Science and Technology Beijing, Beijing 100083, China

method, which coalesces a number of points in the random-half of the spectrum and was found to perform better than any single operating point in the spectrum, with no necessity of tuning to a specific level of randomness.

Obviously, on data with various noises, large categories and a huge number of features, there are a lot of diverse characteristics for trained component trees in random forests, i.e., some trees will be much diverse from each other when properly learned or selected. As we have known, there are a variety of cognitive-inspired computation techniques in the literature [21]. Alternatively, in this paper, we improve random forests with diversity via ensemble pruning, as ensemble of the partial available component trees may be better than that of the whole [54]. Specifically, given numerous component trees from the original random forests, we want to diversely and adaptively select and combine tree classifiers with cognitive-inspired diversity learning.

Moreover, in classifier ensemble, all existing diversity measures are calculated on the training set, which means the performance of optimization heavily relies on the samples of the training set besides the diversity learning itself [48, 50, 51]. In some related fields, researchers suggested that sample weight learning is needed, where imperfections in the samples might lead to bias and other departures between the samples and the reference population. For example, AdaBoost [15] is one of the most famous models for sample weight learning.

Consequently, given numerous component trees from the original random forests, we propose a novel method, diversity-based random forests with sample weight learning (DRFS), which is inspired from cognitive, and diversely and adaptively select and combine tree classifiers with diversity learning and sample weight learning. DRFS uses an iterative learning algorithm to solve the convex optimization and automatically learn the sample weights iteratively. Each iteration of this iterative learning algorithm consists of two main steps: (1) Calculate tree weights by solving an optimization problem with sample weights known and (2) update sample weights with tree weights known. Specifically, in the first step, diversity learning for tree weights with known sample weights is converted into a unified convex quadratic programming optimization model, by setting the sample distribution as a diagonal matrix. In the second step, sample weights are updated with a dynamically damped learning trick automatically. Moreover, a variety of experiments on UCI classification benchmarks and real-world text categorization benchmarks show that our proposed method has very promising performance.

The main contributions of this work can be summarized as follows. Firstly, different from conventional random forests with diversity-related aspects for producing and training diverse component trees, we propose diversity-based random forests to select more diverse tree classifiers from the original random forests in a totally new perspective, which adaptively combine component trees via sample weight learning. Secondly, different from many conventional diversity-based ensembles with optimization using heuristic rules, our proposed DRFS method (diversity-based random forests with sample weight learning) is formulated as a unified optimization model within an iterative learning framework for diversity learning and sample weight learning, where diversity learning for tree weights is conducted by a convex quadratic programming optimization procedure, and sample weight learning for sample weights is performed by a dynamically damped learning trick. Thirdly, different from conventional ensemble methods with experiments on some specific data sets, this paper presents numerous experiments with a variety of typical data sets (with totally more than 50 classification data sets), including UCI classification benchmarks and real-world text categorization benchmarks.

The rest of the paper is organized as follows.[1] Related work is presented in "Related Work." The DRFS model is presented in "Methods," and the (optimization) learning procedure of DRFS is described in "DRFS Algorithm." The "Results" section shows extensive experiments, results, and analyses. Final remarks are presented in "Conclusions."

Related Work

As described above, diversity should be a necessary condition for high generalization ability of classifier ensemble. Similarly, diversely generating and combining diverse classifiers play an important role in random forests. However, it is still a grand challenge to improve random forests by diversity explicitly. In this section, we will first review some typical diversity-based ensemble methods with ensemble pruning, and then survey a variety of approaches for random forests with diversity-related aspects.

Diversity-Based Ensembles

Numerous diversity-based ensembles have also been proposed, most of which are related to ensemble pruning.[2] Marginneantu et al. [32] proposed a Kappa pruning

[1] Parts of this work previously appeared in [48, 49]. Here, we focus on random forests, propose an improve random forests method (diversity-based random forests), clarify the optimization model, improve the learning procedure, and add a variety of experiments.

[2] From the view of ensemble pruning, Tsoumakas et al. presented a taxonomy of ensemble pruning methods, i.e., ranking based, clustering based, optimization based, and other categories [45]. Zhou divided related methods into three categories: ordering-based pruning, clustering-based pruning, and optimization-based pruning approaches [53]. More specifically, optimization-based pruning methods formulate the ensemble pruning problem as an optimization problem that aims to find the subset of available component classifiers which maximizes or minimizes an objective related to the generalization ability of the final ensemble, which is also the focus of our paper.

approach for maximizing the pairwise diversity among the selected component classifiers. Martinez et al. presented an experimental comparison and analysis of ensemble pruning methods based on ordered aggregation, one of which sorts component classifiers by diversity and selects the top 20–40% classifiers for combining [33]. Li et al. [26] tried to present a theoretical study on the effectiveness of diversity on the generalization performance of voting in the PAC-learning framework for classifier ensemble. Following this analysis, they also proposed the diversity regularized ensemble pruning method. Trawinski et al. [44] used genetic algorithm (GA) to search the best classifier subset by a linear combination of accuracy and diversity. Lu et al. [30] proposed an ensemble pruning method with a heuristic optimization by considering both accuracy and diversity. Yin et al. [49, 50] also proposed a heuristic learning approach with diversity and sparsity to learn classifiers' weights and combine multiple classifiers. The major problem with the methods above is that when it comes to optimizing some criteria of the selected subset, they all resort to greedy or heuristic search and may get stuck in a local optima.

To get a solution more closer to the global optima, several ensemble pruning methods with mathematical programming optimization have been proposed. One typical work is conducted by Zhang et al. [52]. They proposed a semi-definite programming (SDP) approach, which formulates the ensemble pruning problem as a quadratic integer programming problem and solves it by a semi-definite programming solution technique. In their method, the size of the selected subset after pruning should be known in advance. Yin et al. [51] proposed a convex ensemble learning method which learns classifier weights that have the optimal accuracy-diversity trade-off.

However, most of diversity-based ensembles are related to ensemble pruning with "Bagging," but without "random forests."

Diversity-Based Random Forests

Random forests [9] are a typical method for ensemble learning. They generate a multitude of decision trees based on bootstrapped samples of the training data and output the class that is the mode of the classes output by individual trees. For each node of a component tree, m variables are randomly chosen and the best split based on these m variables is calculated based on the bootstrapped data. Conventionally, m is set to $\sqrt{u}$, where u stands for the number of variables. Each decision tree results in a classification label and is set to cast a weighted vote for classification, and random forests finally return the class that received the most votes.

As various theoretical and empirical studies show [1, 6], random forests are fast and easy to be implemented, produce highly accurate predictions, and can handle a very large number of input variables without overfitting. In fact, they are considered to be one of the most accurate general-purpose learning techniques available in the literature.

For diversity-based random forests, most efforts have been conducted for maximizing the tree diversity while producing and training component trees. For example, rotation forests [42] improve the diversity and accuracy within component classifiers by using a combination of covariate sub-sampling and PCA-based rotation of covariate axes. Conditional inference forests [20] use conditional inference trees instead of regular classification trees, where a conditional inference tree firstly picks the best variable based on statistical testing and then picks the best split within this variable, while a regular classification tree picks the best split among all available variables which can lead to biased selection for variables with more categories. Conditional inference forests were found to be less biased than the original random forests in assessing the variable importance. Oblique random forests [35] use oblique splits instead of orthogonal splits during the test process. The main difference between these two splits is that the oblique split is based on a combination of variables while an orthogonal split is based on a single variable. Oblique random forests empirically outperform the original random forests in numerical and spectral data [35].

Similarly, Robnik-Sikonja investigated some possibilities to increase strength or decrease correlation of individual trees in the forest and used several attribute evaluation measures to improve random forests [41]. Bernard et al. proposed dynamic random forests with an adaptive tree induction procedure [5]. Their method utilizes resampling of the training data and wants to guide the tree induction so that each tree will complement as much as possible the existing trees in the ensemble.

However, as described before, ensemble pruning is a common strategy for diversity-based ensembles. Consequently, pruning the original random forests can also improve the diversity and accuracy. Additionally, some previous works showed that beyond certain number of trees, adding more trees in random forests does not improve accuracy [53]. Research efforts should be carried out in this direction to limit the number of trees in random forests and to find the optimal subset of random forests, while very few works of ensemble pruning for random forests have been conducted in the literature.

Moreover, most existing diversity measures are evaluated on the validation set without considering data noise of given samples. Here, we focus on random forests with diversity learning and sample weighting and propose an improved random forests method, i.e., diversity-based random forests (DRFS). The DRFS model learns to combine decision trees in random forests by considering both accuracy and diversity adaptively.

Methods

As mentioned before, by designing a diagonal matrix for the sample distribution, the DRFS method is formulated as a unified optimization model within an iterative learning framework, where diversity learning for tree weights is optimized with a convex quadratic programming problem, and sample weight learning for sample weights is performed by a dynamically damped learning trick. In the following, the DRFS model with a unified optimization formulation is described in detail in this section ("Methods"), and the iterative learning framework (a self-training algorithm) is presented in the next section ("DRFS Algorithm").

Notations

Let the training data set be $Tr = \{(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)\}$, where y_i is the output of the sample x_i, and all the outputs are in C classes $\{\Omega_1, \Omega_2, ..., \Omega_C\}$. The base classifiers $H = \{h_1, h_2, ..., h_L\}$ of ensemble are trained on the training set, and an output of a base classifier h_j on sample x_i is $h_j(x_i)$. Given each base classifier h_j, together with its weight w_j, we define the vector of classifier weights as $\mathbf{w} = [\mathrm{w}_1, \mathrm{w}_2, ..., \mathrm{w}_L]$, where $\sum_{j=1}^{L} \mathrm{w}_j = 1, \mathrm{w}_j \geq 0$. In this paper, we focus on the linear combination of classifiers. By taking a weighted vote among the base classifiers and choosing the class label receives the largest weighted vote, the ensemble H classifies sample x_i as $H(x_i)$.

Moreover, instead of the original output, the oracle output O of random forests is used for the optimization. Let the number of sample sets be N and the number of component trees L. O is a $N \times L$ matrix, and element

$$O_{ij} = \begin{cases} 1 & h_j(x_i) = y_i \\ -1 & h_j(x_i) \neq y_i \end{cases} \tag{1}$$

Optimization Model

As an ensemble approach, random forests can be improved by pruning component trees. Specially, for weighted-vote random forests, the improvement is equivalent to a mathematical optimization problem with tree weights. Define tree weights vector $\mathbf{w} = [\mathrm{w}_1, \mathrm{w}_2, ..., \mathrm{w}_L]$, where $\sum_j^L \mathrm{w}_j = 1, \mathrm{w}_j \geq 0$. Traditionally, in diversity-based random forests, $\mathbf{w}$ is learned by

$$\begin{aligned} \mathbf{w_{opt}} &= argmin_{\mathbf{w}} loss(O, \mathbf{w}) - \lambda div(O, \mathbf{w}) \\ s.t.\ \ \mathbf{w_{opt}} &\succeq 0, \mathbf{1}_{L\times 1}^T \mathbf{w_{opt}} = 1, \lambda \geq 0. \end{aligned} \tag{2}$$

where $loss()$ is the loss function of the classification error, $div()$ is the diversity of the ensemble, and λ is the balance parameter.

Simply, given the average accuracy $\mathbf{P} = \frac{1}{N}\mathbf{1}_{N\times 1}^T O$ and the pairwise diversity matrix D, (e.g., disagreement diversity measure [25]), $loss()$ and $div()$ can be calculated by

$$\begin{aligned} loss(O, \mathbf{w}) &= -\mathbf{P}\mathbf{w} \\ div(O, \mathbf{w}) &= \mathbf{w}^T D \mathbf{w} \end{aligned} \tag{3}$$

Instead of the traditional average accuracy $\overline{\mathbf{P}} = \frac{1}{2N}\left(\mathbf{1}_{N\times 1}^T O + \mathbf{1}_{L\times 1}\right)$. Note that there is a linear relationship between $\mathbf{P}$ and $\overline{\mathbf{P}}$, i.e., $\mathbf{P} = 2\overline{\mathbf{P}} - \frac{1}{N}\mathbf{1}_{L\times 1}$. Here, to simplify calculation, $\mathbf{P}$ is chosen. Consequently, by combining the above formulations, the criterion (in Eq. 2) changes to

$$\begin{aligned} w_{opt} &= argmin_{\mathbf{w}} - \lambda \mathbf{w}^T D \mathbf{w} - \mathbf{P}\mathbf{w} \\ s.t.\ \mathbf{w_{opt}} &\succeq 0, \mathbf{1}_{L\times 1}^T \mathbf{w_{opt}} = 1, \lambda \geq 0. \end{aligned} \tag{4}$$

The ensemble can be optimized by solving Eq. 4. Such convex optimization is called as quadratic programming problem with diversity (QPD). As we know, the performance of QPD is always different along with different validation sets. Considering the influence of the training set, this diversity-based random forests method is changed to DRFS by expanding Eq. 4 as

$$\begin{aligned} w_{opt} &= argmin_{\mathbf{w}} - \lambda \mathbf{w}^T D_{\Omega} \mathbf{w} - \mathbf{P}_{\Omega}\mathbf{w} \\ s.t.\ \mathbf{w_{opt}} &\succeq 0, \mathbf{1}_{L\times 1}^T \mathbf{w_{opt}} = 1, \lambda \geq 0. \end{aligned} \tag{5}$$

where Ω is a parameter of the data distribution (sample weights), and actually these sample weights are used to select the validation set from the training set automatically.

To simplify calculation and keep the optimization as a convex problem, we set Ω as a $N \times N$ diagonal matrix, and $diag(\Omega)_i = \Omega_{ii}$ stands for the sample weight x_i, where $diag(\Omega)_i \geq 0, \mathbf{1}_{N\times 1}^T diag(\Omega) = 1$. Thus, $\mathbf{P}_{\Omega}$ and $D_{dis,\Omega}$ can be calculated by

$$\begin{aligned} \mathbf{P}_{\Omega} &= \mathbf{1}_{N\times 1}^T \Omega O \\ D_{dis,\Omega} &= \frac{1}{2}\left(1_{L\times L} - O^T \Omega O\right) \end{aligned} \tag{6}$$

Consequently,the optimization function (in Eq. 5) can be simplified to a convex quadratic programming problem with a given Ω.

DRFS Algorithm

In general, it is difficult to find the solution for the optimization in Eq. 5 when both $\mathbf{w}$ and Ω are unknown. However, with known Ω, the optimization is simplified to a quadratic programming problem. That is to say, given sample weights (with known Ω), tree weights in random forests can be easily learned by this convex quadratic

programming optimization procedure. Given tree weights (with known **w**), sample weights can be conveniently learned by a dynamically damped trick. Thus, we propose an iterative learning algorithm (a self-training algorithm) for DRFS which is shown in Algorithm 1.[3]

In summary, there are three important steps in this self-training algorithm, i.e., initialization for parameters and weights, weight updating for classifiers, and weight updating for samples, which are sequentially presented in the following subsections in detail. At last, we also empirically analyze the convergence performance of this iterative learning algorithm.

Note that in order to avoid overfitting, in our iterative learning algorithm, the final learned results will be selected as the sample weights and the classifier weights at the t^{th} iteration where the learned classifier ensemble has the minimal classification error on the validation set. Basically speaking, it may be more rational to use a new validation set in this training algorithm if there are additional data available.

Algorithm 1 Iterative learning algorithm for DRFS.

Input:

Tr: the training set. $|Tr| = N$

$H = \{h_1, h_2, ..., h_L\}$: the base classifier set, $|H| = L$.

M: pairwise diversity method.

Output:

w: the classifier weights.

Parameter:

T: the max epoch.

Ω_t: a diagonal matrix, and $diag(\Omega_t)_i$ is the weight of sample x_i used to calculate **w** on the t^{th} iteration.

Ω_t^*: a diagonal matrix, and $diag(\Omega_t^*)_i$ is the updated weight of sample x_i on the t^{th} iteration.

ϵ_t: the ensemble error rate on the t^{th} iteration.

β_t: a parameter that $\beta_t \in [0, 1]$, and $\beta_t \geq \beta_{t+1}$.

Procedure:

1: Set $diag(\Omega_1)_i = 1/N$.
2: **For** $t = 1, 2, ..., T$;
3: Use Eqs. 5 and 6 to calculate **w**.
4: Calculate ϵ_t by **w** and Tr.
5: Use ϵ_t to calculate updated weight Ω_t^*.
6: $\Omega_{t+1} = \beta_t \Omega_t^* + (1 - \beta_t)\Omega_t$
7: **End**

[3]Here, the optimization problem with **w** and Ω in an iterative learning framework is similar to an EM (expectation-maximization) procedure. We also initially used the EM algorithm in our method, and the results were not encouraged compared with this iterative learning algorithm. However, how to adaptively design, improve, and use a variant of EM algorithm for our DRFS method is a near future topic.

Initialization for Parameters and Weights

In the iterative learning algorithm, firstly given the validation set[4] Tr and base classifiers H for optimization, assign the pairwise diversity **D**. Then, set the max epoch T as a stop constraint, and $\Omega_i = 1/N$ for each sample x_i, where N is the number of samples in Tr.

Weight Updating for Classifiers

In each iteration, given samples with known weights Ω, we first use Eqs. 5 and 6 to calculate and update the base classifier weights vector **w**. As described above, this optimization can be solved as a typical convex quadratic programming problem. Then, samples of training set Tr are classified with **w**, and the ensemble classification error rate ϵ_t is calculated.

Weight Updating for Samples

Updating the sample weights Ω_{t+1} is a key process in DRFS. In the iterative learning algorithm, we assume the sample weights Ω_{t+1} have a relationship with Ω_t and use a dynamically damped trick, i.e., the damped factor $\beta_t \in [0, 1]$ and $\beta_t \leq \beta_{t+1}$. We set β_t at the t^{th} iteration as

$$\beta_t = \frac{1}{t}. \tag{7}$$

Then, we use the following equation to update sample weights for the $(t+1)^{th}$ iteration,

$$\Omega_{t+1} = \beta_t \Omega_t^* + (1 - \beta_t)\Omega_t \tag{8}$$

where Ω_t and Ω_t^* are the original and new sample weights at the t^{th} iteration respectively.

Here, we use hinge loss to update Ω_t^*. The hinge loss is a loss function used to train classifiers in conventional machine learning methods. It is used for "maximum-margin" classification, especially for support vector machines. Given the weights w_j of each base classifier h_j, the margin m_i of the sample x_i is calculated by

$$m_i = \sum_{h_j(x_i)=y_i} w_j - \sum_{h_j(x_i)\neq y_i} w_j = \sum_{j=1}^{L} w_j O_{ij} \tag{9}$$

Thus, DRFS updates Ω_t^* by

$$diag(\Omega_t^*)_i = \begin{cases} \frac{1}{N\epsilon_t} & m_i \leq 0 \\ 0 & otherwise \end{cases} \tag{10}$$

where $N\epsilon_t$ is the number of samples which are wrongly classified by ensemble with weight **w** in iteration t.

[4]In our experiments, this validation set is bootstrapped from the initial training set, and is used as the new training set for learning sample weights and classifier weights in the iterative learning algorithm for DRFS.

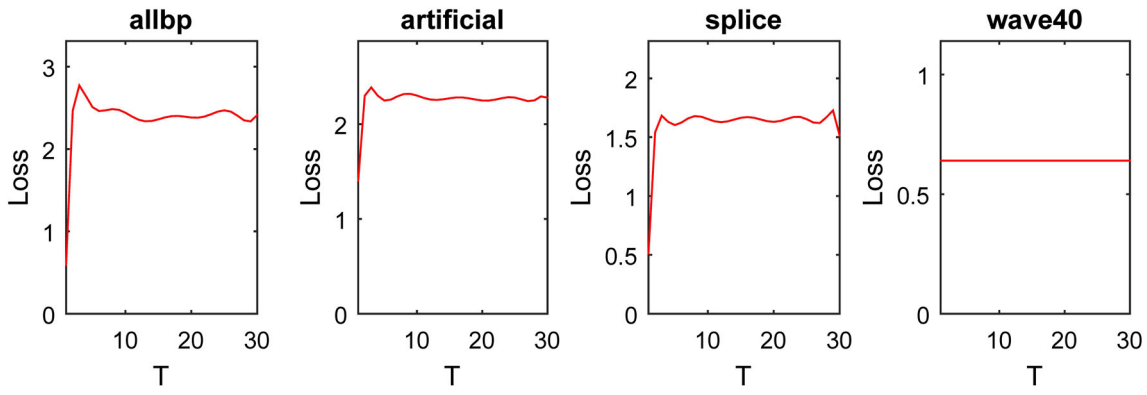

Fig. 1 Loss of DRFS with different T

Empirical Analysis for Convergence

At each iteration, with L parameters (component classifiers), this iterative learning algorithm calculates the diversity of each classifier pair (totally with $L(L-1)/2$ times), solves a quadratic programming problem ($O(L^2)$), and updates sample weights ($O(N)$). Consequently, with N data samples, the time complexity for each iteration is $O(NL^2 + L^2)$. The space complexity for the whole iterative learning algorithm is just $O(NL + L^2)$, where $O(NL)$ is for storing classifiers' outputs and $O(L^2)$ is for ensemble diversity.

Similar to many iterative learning algorithms, convergence of the proposed algorithm is not guaranteed because the objective function is not guaranteed to decrease in stage 3 (Weight Updating for Samples) in Algorithm 1. However, we find our algorithm can usually achieve good performance after a number of iterations. We empirically investigate the convergence performance of this algorithm with the parameter T for DRFS. Results of DRFS on some typical UCI classification data sets are shown in Fig. 1. Experimental data and setting can be referred to the descriptions in "Experimental Setup."

As shown in Fig. 1, the ensemble performance (loss) tends to be converge (e.g., "artificial") or exhibits slight fluctuations (e.g., "allbp" and "splice") after several iterations. As a result, we just set $T = 20$ in the following

Table 1 Information of UCI classification data sets

Data set	Instances	Attributes	Classes	Data set	Instances	Attributes	Classes
Allbp	3772	29	3	iris	150	4	3
Anneal	858	38	4	kr-vs-kp	3196	17	2
Artificial	5109	7	10	labor	57	16	2
Audiology	226	69	24	led24	3200	24	10
Auto-mpg	398	7	3	led7	3200	7	10
Autos	205	25	6	machine	209	7	7
Balloons	76	4	2	page-blocks	5473	10	5
Breast-w	699	9	2	sick	3772	29	2
Bridges2	108	11	7	sickeuthyroid	3163	24	2
Chess	3196	36	2	sonar	208	60	2
Colic	368	22	2	splice	3190	60	3
Credit-a	690	15	2	tic-tac-toe	958	9	2
Diabetes	768	8	2	vehicle	846	18	4
Echocardiogram	132	8	3	vote	435	16	2
German	1000	24	2	vowel	990	12	11
Heart-c	303	13	2	wave21	5000	21	3
Hepatitis	155	19	2	wave40	5000	40	3
House-votes-84	435	16	2	wine	178	13	3
Hypothyroid	3163	24	2				
Imageseg	2310	19	7				
Ionosphere	351	34	2				

experiments. Note that the iterative learning algorithm could be terminated when the error rate $\epsilon_t = 0$ (e.g., "wave40"). In such a case, DRFS is degenerated into QDP.

Results

In this section, DRFS is firstly compared with some existing ensemble methods on the UCI data sets for general classification tasks. Moreover, experiments with the proposed method are performed on a specific topic, i.e., text categorization. Extensive experimental results show that our proposed method outperforms several state-of-the-art ensemble methods.

Experiments on UCI Data Sets

To verify the effectiveness of the proposed algorithms, 39 typical data sets from UCI machine learning repository [14] are used. More information for the data sets is presented in Table 1.

Experimental Setup

In our method, there are two algorithms that are used, i.e., quadratic programming problem with diversity (QPD, Eq. 4) and DRFS (Eq. 5). Our proposed method is compared with some conventional ensemble methods (including diversity-based ensembles), e.g., random forests (RF) [9], Gasen [54],[5] forward ordering (FO) [33], backward ordering (BO) [33], genetic algorithm with diversity (GA-div) [44], and semi-definite programming (SDP) [52]. In QPD, DRFS, the ordering algorithms, GA-div, and SDP, the disagreement (*dis*) is chosen to measure the ensemble diversity. In the ordering algorithms and SDP, about 21% of classifiers are selected. In genetic algorithm–based methods (Gasen and GA-div), the selecting threshold is set to the default value $1/L$, where L stands for the number of classifiers. In QPD and DRFS, $\lambda = 1.25$. All ensemble methods select a subset of component classifiers (trees) on the validation set, which is bootstrapped from the training set. In the experiments, 10-fold cross validation is performed on each data set, and each ensemble contains 301 tree base classifiers.

Experimental Results

The experimental results are shown in Table 2. The last rows present the counts of wins, ties, and losses [13], where W/T/L means that DRFS wins, ties, and loses on #W, #T, and #L data sets, respectively. Moreover, we assign ranks to evaluate the methods' performance on each data set [7]. Here, mark the best method as rank 1, and the worse the larger. The average rank is calculated for each method. In addition, the highest classification accuracy rate for each data set is highlighted in italics in Table 2.

As shown in Table 2, firstly, for ensembles with genetic algorithms, Gasen (rank 3.5) performs a little worse than random forests (rank 3.4), while by adding a regularization term about diversity to expand the fitness function, GA-div (rank 3.3) ranks the same as RF. These results show that diversity-based ensembles by optimizing both accuracy and diversity are much more effective than the ones by optimizing with accuracy only. Secondly, QPD ranks 4.7 and performs a little worse than RF. This pure-diversity–based selection method has a weak connection between the diversity measure and the ensemble performance. Our proposed method, DRFS, has a more impressive performance considering both ensemble diversity and sample weights. Specifically, DRFS ranks 3.0, and wins random forests on 12 data sets. It performs the best on "allbp," "anneal," "audiology," "auto-mpg," "chess," "german," "heart-c," "imageseg," "kr-vs-kp," "sickeuthyroid," "sonar," "tic-tac-toe," "vote," and "wine" data sets. By selecting the validation set (calculating the sample weights) carefully and adaptively, DRFS has the minimum rank and outperforms Gasen and SDP.

In summary, our methods achieve the best performance in most cases with enough training data for learning base classifiers, classifier weights, and sample weights. Consequentially, our methods obtain the best rank (3.0) and good win-tie-lose performance among all compared ensembles.

Experiments with Parameters λ

In order to illustrate how to select the control parameter λ of diversity regularization (in Eq. 5) in DRFS, we also analyze experimental results on the validation set.[6] When setting and testing different values of λ, DRFS is performed with 301 base classifiers generated by random forests.

To observe the influence of parameter λ, DRFS is performed with different values of λ.[7] The results of the average ranks of random forests and 20 DRFS ensembles with different λ on all 39 data sets are shown in Fig. 2. When λ is about equal to 1.25, DRFS gets the minimum average rank. DRFS has a similarly competitive performance for $\lambda \in [1, 10]$ and ranks higher (1.76–2.26) than random forests

[5] http://lamda.nju.edu.cn/code_GASEN.ashx

[6] Note that in these experiments with parameters, the used validation set is the same as the validation set in Algorithm 1, and all other experimental conditions are the same as the ones in "Experimental Setup."

[7] Totally, there are 20 DRFS ensembles with 20 different values λ by $1/\lambda = \{0, 0.1, ..., 1, 2, ..., 10\}$.

Table 2 Comparison of the classification accuracy rates (average ± standard deviation) of DRFS with state-of-the-art ensemble methods (%), where the highest classification accuracy rate for each data set is highlighted in italics, and the W/T/L row summarizes the comparison of DRFS against other ensembles, and the rank row presents the average rank over all data sets

Data sets	RF	GASEN	GA-DIV	FO	BO	QDP	SDP	DRFS
Allbp	97.69±0.31	97.67±0.30	97.67±0.30	97.46±0.35	97.67±0.45	97.83±0.66	97.56±0.48	*97.83±0.70*
Anneal	*99.65±0.78*	*99.65±0.78*	*99.65±0.78*	99.42±0.82	99.30±0.98	99.53±0.81	99.42±0.82	*99.65±0.78*
Artific.	61.44±1.46	61.40±1.65	61.34±1.62	61.07±1.80	60.56±1.50	60.78±1.46	*61.46±1.18*	60.89±2.08
Audio.	75.38±19.84	74.51±19.03	75.81±18.66	74.07±19.74	72.33±18.35	75.72±19.04	75.81±18.99	*77.12±18.63*
Auto-m.	86.68±5.65	86.42±5.94	86.93±5.24	84.67±4.16	85.92±6.49	86.17±4.32	85.92±5.57	*87.67±5.12*
Autos	80.95±10.29	*82.38±9.27*	81.90±9.73	80.48±9.90	*82.38±8.71*	80.95±9.78	80.95±10.53	81.90±9.47
Balloons	78.75±10.29	78.75±10.29	78.75±10.29	78.75±10.29	76.25±13.76	67.50±20.58	*80.00±10.54*	78.75±10.29
Breast-w	96.85±2.41	*97.00±2.18*	96.71±2.61	96.71±2.61	96.71±2.34	96.57±2.63	96.85±2.41	96.99±2.47
Bridges2	70.30±7.26	69.39±7.48	*72.12±6.36*	63.94±7.48	68.48±4.56	65.76±7.15	67.58±6.23	71.06±6.08
Chess	98.75±0.36	98.81±0.29	98.81±0.53	98.40±0.40	98.81±0.44	99.12±0.35	98.72±0.43	*99.25±0.34*
Colic	85.64±7.39	*86.19±6.38*	85.92±7.58	85.09±6.48	*86.19±6.63*	85.64±8.23	85.66±7.83	84.82±7.83
Credit-a	86.96±5.34	86.81±5.53	87.10±5.44	87.54±4.94	86.96±4.58	*87.83±5.43*	87.10±5.44	87.54±5.17
Diabetes	76.45±3.06	76.05±2.52	*76.57±2.78*	75.53±3.41	76.45±3.41	76.05±3.35	76.45±3.50	76.06±3.16
Echoca.	71.17±19.28	71.17±19.28	69.74±19.78	69.74±19.78	69.74±17.34	69.74±17.34	69.74±19.78	*71.17±16.76*
German	76.40±5.32	76.50±5.10	76.60±4.99	75.30±6.52	75.90±6.21	76.60±6.92	76.50±5.34	*77.10±5.26*
Heart-c	80.97±6.91	80.97±7.26	80.97±6.91	80.30±5.61	81.64±7.13	79.33±6.63	80.33±6.75	*81.67±7.07*
Hepatit.	80.68±8.57	81.93±7.48	81.31±7.80	*83.81±9.85*	*83.81±8.92*	81.93±13.64	80.68±9.98	82.56±11.71
House-v.	96.08±3.24	95.85±3.19	96.08±3.24	95.59±3.02	*96.82±3.07*	95.88±2.97	95.85±3.19	96.56±3.07
Hypoth.	99.97±0.10	99.97±0.10	99.97±0.10	99.56±0.22	*100.00±0.00*	99.78±0.21	99.78±0.26	99.84±0.17
Images.	97.92±0.81	97.79±0.85	97.92±0.81	97.97±0.79	97.84±0.79	97.84±0.76	97.88±0.97	*98.05±0.68*
Ionosph.	93.72±5.69	93.72±5.69	93.44±5.57	92.87±6.21	*94.01±5.30*	*94.01±4.76*	93.44±5.57	92.87±6.36
Iris	95.33±5.49	95.33±5.49	95.33±5.49	94.67±5.26	*96.67±3.51*	94.67±5.26	94.67±5.26	94.67±5.26
kr-vs-kp	82.10±1.85	82.10±1.85	82.04±1.78	82.13±1.96	81.85±1.95	82.01±1.48	82.19±1.98	*82.29±1.73*
Labor	*95.00±8.05*	*95.00±8.05*	*95.00±8.05*	93.33±11.65	90.00±14.05	93.33±8.61	*95.00±8.05*	93.33±8.61
Led24	72.78±2.09	*73.03±2.13*	72.97±1.96	71.53±1.93	72.91±2.16	72.09±2.47	72.00±1.97	72.16±2.51
Led7	73.91±2.10	73.63±2.02	73.84±2.12	73.69±1.90	72.16±1.88	73.09±2.14	*73.94±1.99*	73.56±1.81
Machine	91.38±4.93	91.38±4.93	91.38±4.93	90.40±5.57	90.40±6.01	*91.86±5.54*	*91.86±4.54*	91.38±3.77
Pageblo.	97.50±0.71	97.48±0.73	97.50±0.71	97.46±0.76	97.44±0.83	97.39±0.74	*97.52±0.68*	97.37±0.67
Sick	98.41±0.56	98.46±0.60	98.44±0.59	97.69±0.88	*98.54±0.50*	98.52±0.59	98.28±0.42	98.49±0.53
Sickeut.	97.88±0.76	97.88±0.77	97.88±0.77	96.87±0.95	97.91±0.82	97.91±0.87	97.53±0.81	*97.98±0.60*
Sonar	84.14±7.45	84.61±7.02	83.66±7.49	82.23±5.44	82.71±7.12	83.66±8.43	84.14±6.35	*85.09±5.71*
Splice	96.14±0.99	96.21±0.93	96.18±0.83	95.55±1.05	95.27±0.82	96.24±0.75	*96.43±0.90*	96.27±0.89
Tic-tac.	95.30±3.21	94.57±3.16	95.19±3.13	91.76±3.51	95.40±2.84	97.07±2.47	93.32±2.62	*97.49±1.99*
Vehicle	75.62±4.68	75.28±4.17	75.38±4.71	*77.06±6.16*	75.39±4.41	75.39±5.72	75.97±5.92	75.40±4.85
Vote	*96.56±2.44*	*96.56±2.44*	*96.56±2.44*	95.88±2.56	96.31±2.70	96.08±2.18	96.33±2.65	*96.56±2.44*
Vowel	97.58±2.14	97.47±1.86	97.47±2.34	97.17±2.60	96.67±2.48	97.27±2.24	*97.68±1.72*	97.37±2.34
Wave21	85.42±1.75	85.44±1.68	85.44±1.71	84.66±1.84	*85.50±2.02*	85.24±1.68	85.40±1.43	85.30±1.68
Wave40	*85.88±1.89*	85.76±2.37	85.72±1.81	85.08±2.24	84.58±1.86	85.52±2.02	85.48±2.08	85.52±1.48
Wine	97.78±2.87	97.78±2.87	97.78±2.87	*98.33±2.68*	97.78±2.87	*98.33±2.68*	*98.33±2.68*	*98.33±2.68*
W/T/L	12/21/6	10/23/6	9/23/7	19/18/2	16/18/5	11/26/2	12/21/6	–
Rank	3.4	3.5	3.3	6.0	4.7	4.7	4.1	*3.0*

(4.73). DRFS ranks 3.70 for $1/\lambda = 0$, and it performs poorly (7.23–13.03) when $\lambda \in [0.1, 0.5]$. It seems that the optimization based on trade-off between diversity and accuracy performs better than the one based on single, and the diversity regularization is a little more important than the classification accuracy and in learning classifiers' weights for ensembles. As a result, we set $\lambda = 1.25$ in the above comparison experiments.

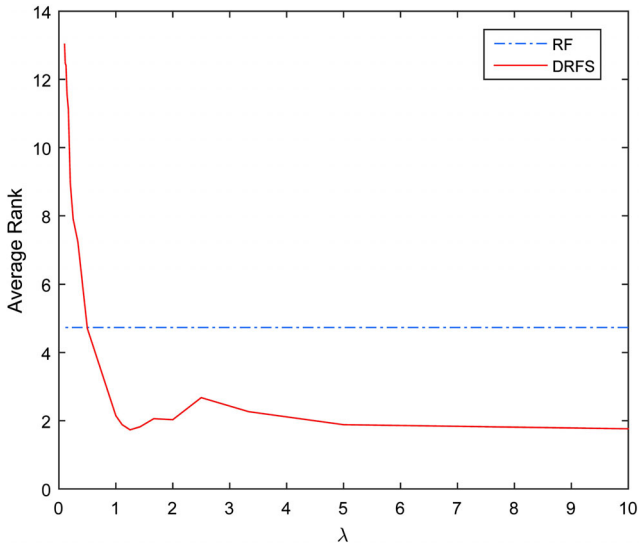

Fig. 2 Comparison of the average ranks of DRFS with different λ

Failed Case Analysis

As shown in Table 2, there is a small part of data sets on which DRFS's performance is slightly worse than the original random forests.

Data sets in the failed cases always have very few samples and/or attributes, e.g., "iris" with 150 samples and 4 attributes and "labor" with 57 samples and 16 attributes (see Table 1). It is essentially difficult to train diverse and efficient base classifiers on such data sets where the existing data itself cannot reflect some diversity, and it is also challenging to learn and select diverse classifiers with few samples. We argue that our DRFS methods should improve the performance on such data sets, which is also a near future issue of our work.

Experiments with Text Categorization

We also evaluate the performance of DRFS on another real application—text categorization. There are a variety of techniques for text categorization [23, 43]. In the experiments, DRFS is compared against some representative methods, such as multinomial naive Bayesian, J48, support vector machines, and random forests.

The detailed characteristics of the various document collections used in our experiments are available in [18].[8] Moreover, to demonstrate the effectiveness of DRFS on the larger data set, we also compare all methods on the Reuters-21578 data set. The original Reuters-21578 data set with the standard "Mod Apte" train/test split is available in [12].[9] As the goal of our work is to consider single-label data sets, we do the experiments on two general subsets (i.e., r8 and r52) [10] of the Reuters-21578 data set. Here, r8 has 8 categories and is split to 6487 training and 2545 test documents. r52 has 52 categories, and is split to 7715 training and 2995 test documents. More information for the data sets is presented in Table 3.

Table 3 Text categorization benchmark data sets

Data set	Source	Documents	Words	Classes
fbis	TREC	2463	2000	17
la1s	TREC	3204	13472	6
la2s	TREC	3075	13472	6
oh0	OHSUMED	1003	3182	10
oh10	OHSUMED	918	3012	10
oh15	OHSUMED	1050	3238	10
oh5	OHSUMED	913	3100	10
ohscal	OHSUMED	11162	11465	10
re0	Reuters	1504	2886	13
re1	Reuters	1657	3758	25
tr11	TREC	414	6429	9
tr12	TREC	313	5804	8
tr21	TREC	336	7902	6
tr23	TREC	204	5832	6
tr31	TREC	927	10128	7
tr41	TREC	878	7454	10
tr45	TREC	690	8261	10
wap	WebACE	1560	8460	20
r8	Reuters	9032 (6487 trn, 2545 tst)	24329	8
r52	Reuters	10710 (7715 trn, 2995 tst)	24329	52

Experimental Setup

The experiments compare DRFS with some typical methods, e.g., multinomial naive Bayes (MNB) [34], J48 [40], support vector machines (SVM) [11], and random forests (RF). Both multinomial naive Bayes and J48 classifier are generated by WEKA [17],[10] and random forests classifier is generated by Matlab toolbox.[11] For each method, all parameters are set by default. In SVM, the Gaussian kernel is used, and the best c and g parameters are selected by cross validation from $c = 2^{-5}, 2^{-4}, ..., 2^{5}$, $g = 2^{-5}, 2^{-4}, ..., 2^{5}$. In DRFS, $\lambda = 1.25$.

In the experiments, 10-fold cross validation is performed on each data set except r8 and r10. For r8 and r10, we use the

[8] http://sourceforge.net/projects/weka/files/datasets/text-datasets/19MclassTextWc.zip

[9] http://prdownloads.sourceforge.net/weka/reuters21578-ModApte.tar.bz2?download

[10] http://www.cs.waikato.ac.nz/ml/weka/

[11] https://code.google.com/p/randomforest-matlab/

standard "Mod Apte" train/test split, and just run 10 times to evaluate the performance. Moreover, we also assign win-tie-lose and ranks to evaluate the methods' performance on each data set.

Experimental Results

The experimental results are shown in Table 4, where the highest classification accuracy rate for each data set is highlighted in italics.

As shown in Table 4, firstly, among four typical methods (J48, MNB, SVM, and RF), the best rank corresponds to RF (rank 3.2), followed by SVM (rank 3.4), MNB (rank 4.3), and J48 (rank 5.5). On most data sets, RF achieves the best classification accuracy rate, and is slightly worse than SVM on "fbis," "re1," "tr11," "tr21," "tr41," "wap," "r8," and "r52" data sets. These results show that RF is a powerful technique for text categorization. Moreover, DRFS ranks 1.8, and is 1.4 higher than random forests (3.2) for the average classification precision. On most data sets, DRFS achieves a 1–4% higher classification accuracy rate than RF, except on "la1s," "la2s," "wap," "r8," and "r52." That is to say, in TC, our proposed method, DRFS, can utilize diversity in component trees and select a proper subset of trees in RF for ensemble. Specifically, by selecting training sets (calculating the sample weights) carefully, DRFS has the minimum rank and largest average classification accuracy rate and outperforms J48, MNB, SVM, and RF. In most cases, DRFS achieves the best performance when there are enough training data for learning component trees, tree weights, and sample weights. Consequently, our methods obtain the best rank (1.8) in all experimental approaches.

Similarly, on text categorization data sets, our proposed approach has more advantages than several typical methods, and our method achieves the best performance (ranks and win-tie-lose).

Conclusions

Random forests are widely used as an effective method to improve accuracy of various component trees, which have a variety of applications in pattern recognition, data mining, and intelligent cybernetics. By improving random forests via ensemble pruning, we propose a generic optimization framework for ensembling diverse components in random

Table 4 Comparison of classification accuracy rates (average ± standard deviation) of DRFS with typical methods for text categorization (%), where the highest classification accuracy rate for each data set is highlighted in italics, the W/T/L row summarizes the comparison of DRFS against other algorithm, and the rank row presents the average rank over all data sets

Data sets	J48	MNB	SVM	RF	GA-Div	DRFS
Fbis	72.03 ± 2.07	77.30 ± 1.84	82.79 ± 1.07	82.74 ± 1.17	82.62 ± 0.72	*83.35 ± 1.20*
la1s	75.56 ± 1.93	87.45 ± 0.51	87.83 ± 1.11	88.08 ± 1.61	*88.11 ± 2.88*	88.05 ± 1.55
la2s	76.33 ± 1.66	88.78 ± 1.03	88.85 ± 1.17	88.93 ± 1.60	*89.00 ± 1.10*	88.80 ± 1.60
oh0	81.05 ± 4.99	*88.43 ± 3.09*	85.14 ± 2.85	88.03 ± 2.66	88.24 ± 2.95	88.03 ± 2.66
oh10	68.38 ± 3.06	78.00 ± 3.80	76.29 ± 4.54	80.95 ± 6.79	*81.24 ± 4.50*	81.14 ± 6.79
oh15	72.39 ± 5.08	*82.04 ± 1.81*	76.88 ± 3.74	80.49 ± 5.08	80.85 ± 4.40	81.04 ± 5.10
oh5	80.71 ± 5.13	87.47 ± 3.01	85.84 ± 4.68	87.58 ± 2.74	87.69 ± 4.54	*89.32 ± 2.74*
ohscal	70.23 ± 5.10	73.99 ± 1.14	76.63 ± 1.49	80.87 ± 1.21	80.83 ± 1.61	*80.93 ± 3.21*
re0	70.68 ± 1.96	76.87 ± 4.32	81.25 ± 4.30	81.32 ± 5.30	81.39 ± 5.23	*81.52 ± 5.26*
re1	77.43 ± 4.43	79.05 ± 6.16	81.83 ± 4.23	81.81 ± 5.86	82.29 ± 3.76	*82.35 ± 5.86*
tr11	77.06 ± 3.24	84.07 ± 3.07	87.20 ± 1.58	84.53 ± 2.87	84.53 ± 1.35	*88.41 ± 2.87*
tr12	79.21 ± 4.05	81.76 ± 7.43	85.93 ± 4.02	87.19 ± 5.33	87.81 ± 2.31	*87.84 ± 5.33*
tr21	77.95 ± 7.25	60.09 ± 6.01	86.00 ± 4.21	85.31 ± 4.53	85.87 ± 5.12	*86.28 ± 4.46*
tr23	*92.68 ± 5.17*	69.07 ± 9.13	83.34 ± 4.66	83.89 ± 8.54	83.89 ± 8.89	86.30 ± 8.54
tr31	93.53 ± 1.37	95.04 ± 1.35	97.09 ± 0.82	97.19 ± 2.52	97.19 ± 1.17	*97.52 ± 2.52*
tr41	92.03 ± 2.67	93.97 ± 2.94	*94.76 ± 1.69*	92.94 ± 2.35	92.94 ± 1.85	93.96 ± 2.35
tr45	91.01 ± 1.50	82.46 ± 3.78	89.28 ± 4.36	90.29 ± 4.51	90.29 ± 1.45	*92.75 ± 4.51*
wap	65.38 ± 2.58	79.94 ± 3.94	*84.49 ± 1.98*	82.71 ± 2.15	82.86 ± 2.12	81.23 ± 2.15
r8	88.84 ± 0.00	94.38 ± 0.00	93.59 ± 0.00	93.40 ± 0.11	93.34 ± 0.25	*94.53 ± 0.33*
r52	80.87 ± 0.00	83.74 ± 0.00	*87.15 ± 0.00*	82.70 ± 0.12	82.74 ± 0.21	84.51 ± 0.25
W/T/L(DRFS)	19/1/1	16/3/2	16/1/4	16/3/2	14/2/5	–
Rank	5.5	4.3	3.4	3.2	2.6	*1.8*

forests, which takes into account both ensemble diversity and sample weights. We also design an iterative learning algorithm for DRFS, where the optimization problem is simplified as a convex quadratic programming problem at each iteration.

In the experiments, DRFS is compared with not only conventional ensemble methods, e.g., original random forests and SDP, on a variety of 39 UCI benchmark data sets but also some typical methods for a real classification application (text categorization), e.g., J48, multinomial naive Bayes, support vector machines, and random forests. All experimental results show that our proposed method achieves very encouraging results.

Funding Information The research was partly supported by the National Natural Science Foundation of China (61473036), China Postdoctoral Science Foundation (2018M641199), and Beijing Natural Science Foundation (4194084).

Compliance with Ethical Standards

Conflict of interest The authors declare that they have no conflict of interest.

Ethical Approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. Amasyali MF, Ersoy OK. Classifier ensembles with the extended space forest. IEEE Trans Knowl Data Eng. 2014;26(3):549–62.
2. Amozegar M, Khorasani K. An ensemble of dynamic neural network identifiers for fault detection and isolation of gas turbine engines. Neural Netw. 2016;76:106–21.
3. Ayerdi B, Graña M. Hybrid extreme rotation forest. Neural Netw. 2014;52:33–42.
4. Ball K, Grant C, Mundy WR, Shafer TJ. A multivariate extension of mutual information for growing neural networks. Neural Netw. 2017;95:29–43.
5. Bernard S, Adam S, Heutte L. Dynamic random forests. Pattern Recogn Lett. 2012;33(12):1580–6.
6. Biau G. Analysis of a random forests model. J Mach Learn Res. 2012;13:1063–95.
7. Brazdil P, Soares C. A comparison of ranking methods for classification algorithm selection. In: Proceedings of the 11th European Conference on Machine Learning, pp 63–74; 2000.
8. Breiman L. Bagging predictors. Mach Learn. 1996;24(1):123–40.
9. Breiman L. Random forests. Mach Learn. 2001;45:5–32.
10. Cardoso-Cachopo A. Improving methods for single-label text categorization. PdD Thesis. Instituto Superior Tecnico: Universidade Tecnica de Lisboa; 2007.
11. Chang CC, Lin CJ. Libsvm: a library for support vector machines. ACM Trans Intell Syst Technol. 2011;2(3):1–27. http://www.csie.ntu.edu.tw/cjlin/libsvm.
12. Debole F, Sebastiani F. An analysis of the relative hardness of Reuters-21578 subsets. JASIST. 2005;56(6):584–96. https://doi.org/10.1002/asi.20147.
13. Demsar J. Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res. 2006;7:1–30.
14. Frank A, Asuncion A. UCI machine learning repository. University of California, Irvine, School of Information and Computer Sciences. 2010. http://archive.ics.uci.edu/ml.
15. Freund Y, Schapire R. Experiments with a new boosting algorithm. In: Proceedings of the 13th International Conference on Machine Learning, pp 148–156; 1996.
16. Freund Y, Schapire R. A decision-theoretic generalization of on-line learning and an application to boosting. J Comput Syst Sci. 1997;55(1):119–39.
17. Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH. The WEKA data mining software: an update. SIGKDD Explor Newsl. 2009;11(1):10–8. https://doi.org/10.1145/1656274.1656278.
18. Han EH, Karypis G. Centroid-based document classification: analysis and experimental results. In: Principles of Data Mining and Knowledge Discovery, 4th European Conference, PKDD 2000, Lyon, France, September 13-16, 2000, Proceedings, pp 424–431; 2000.
19. Hansen LK, Salamon P. Neural network ensembles. IEEE Trans Pattern Anal Mach Intell. 1990;12(10):993–1001.
20. Hothorn T, Hornik K, Zeileis A. Unbiased recursive partitioning: a conditional inference framework. J Comput Graph Stat. 2006;15(3):651–74.
21. Huang K, Zhang R, Jin X, Hussain A. Special issue editorial: cognitively-inspired computing for knowledge discovery. Cogn Comput. 2018;10(1):1–2.
22. Jiang L. Learning random forests for ranking. Frontiers of Computer Science in China. 2011;5(1):79–86.
23. Jiang L, Wang S, Li C, Zhang L. Structure extended multinomial naive Bayes. Inf Sci. 2016;329:346–56.
24. Krogh A, Sollich P. Statistical mechanics of ensemble learning. Phys Rev E. 1997;55(1):811–25.
25. Kuncheva LI, Whitaker CJ. Measures of diversity in classifier ensembles. Mach Learn. 2003;51(2):181–207.
26. Li N, Yu Y, Zhou ZH. Diversity regularized ensemble pruning. In: Proceedings of European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases; 2012.
27. Liu FT, Ting KM. Variable randomness in decision tree ensembles. In: Advances in Knowledge Discovery and Data Mining, 10th Pacific-Asia Conference, PAKDD 2006, Singapore, April 9-12, 2006, Proceedings, pp 81–90; 2006.
28. Liu FT, Ting KM, Fan W. Maximizing tree diversity by building complete-random decision trees. In: Advances in Knowledge Discovery and Data Mining, 9th Pacific-Asia Conference, PAKDD 2005, Hanoi, Vietnam, May 18-20, 2005, Proceedings, pp 605–610; 2005.
29. Liu FT, Ting KM, Yu Y, Zhou ZH. Spectrum of variable-random trees. J Artif Intell Res (JAIR). 2008;32:355–84.
30. Lu Z, Wu X, Zhu X, Bongard J. Ensemble pruning via individual contribution ordering. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, July 25-28, 2010, pp 871–880; 2010.
31. Lulli A, OnetoEmail L, Anguita D. Mining big data with random forests. Cognitive Computation pp. 1–23. 2019. Published online.
32. Margineantu D, Dietterich T. Pruning adaptive boosting. In: Proceedings of International Conference on Machine Learning, pp 211–218; 1997.
33. Martinez-Munoz G, Hernandez-Lobato D, Suarez A. An analysis of ensemble pruning techniques based on ordered aggregation. IEEE Trans Pattern Anal Mach Intell. 2009;31(2):245–59.
34. McCallum A, Nigam K. A comparison of event models for naive Bayes text classification. In: Learning for text categorization: papers from the 1998 AAAI Workshop, pp 41–48. 1998. http://www.kamalnigam.com/papers/multinomial-aaaiws98.pdf.

35. Menze BH, Kelm BM, Splitthoff DN, Kothe U, Hamprecht FA. On oblique random forests. In: Proceedings of European Conference on Machine Learning and Knowledge Discovery in Databases, (ECML-PKDD'11), pp 453–469; 2011.
36. Opitz DW, Shavlik JW. Generating accurate and diverse members of a neural network ensemble. In: Advances in Neural Information Processing Systems (NIPS'96), pp 535–541. MIT Press; 1996.
37. Osadchy M, Keren D, Raviv D. Recognition using hybrid classifiers. IEEE Trans Pattern Anal Mach Intell. 2016;38(4):759–71.
38. Perera AG, Law YW, Chahl JS. Human pose and path estimation from aerial video using dynamic classifier selection. Cogn Comput. 2018;10(6):1019–41.
39. Qiu C, Jiang L, Li C. Randomly selected decision tree for test-cost sensitive learning. Appl Soft Comput. 2017;53:27–33.
40. Quinlan JR. C4.5: Programs for machine learning. Morgan Kaufmann. 1993.
41. Robnik-Sikonja M. Improving random forests. In: Proceedings of 15th European Conference on Machine Learning (ECML'04), pp 359–370; 2004.
42. Rodriguez JJ, Kuncheva LI, Alonso CJ. Rotation forest: a new classifier ensemble method. IEEE Trans Pattern Anal Mach Intell. 2006;28(10):1619–30.
43. Tang B, He H, Baggenstoss PM, Kay S. A Bayesian classification approach using class-specific features for text categorization. IEEE Trans Knowl Data Eng. 2016;28(6):1602–06.
44. Trawinski K, Quirin A, Cordon O. On the combination of accuracy and diversity measures for genetic selection of bagging fuzzy rule-based multiclassification systems. In: Proceedings of the 9th Intelligent Systems Design and Applications, pp 121–127; 2009.
45. Tsoumakas G, Partalas I, Vlahavas I. An ensemble pruning primer. In: Applications of Supervised and Unsupervised Ensemble Methods, pp 1–13; 2009.
46. Wen G, Hou Z, Li H, Li D, Jiang L, Xun E. Ensemble of deep neural networks with probability-based fusion for facial expression recognition. Cogn Comput. 2017;9(5):597–610.
47. Wolpert D. Stacked generalization. Neural Netw. 1992;5(2):241–60.
48. Yang C, Yin XC, Hao HW. Diversity-based ensemble with sample weight learning. In: 22nd International Conference on Pattern Recognition, ICPR 2014, Stockholm, Sweden, August 24-28, 2014, pp 1236–1241; 2014.
49. Yang C, Yin XC, Huang K. Text categorization with diversity random forests. In: Neural Information Processing - 21st International Conference, ICONIP 2014, Kuching, Malaysia, November 3-6, 2014. Proceedings, Part III, pp 317–324; 2014.
50. Yin XC, Huang K, Hao HW, Iqbal K, Wang ZB. A novel classifier ensemble method with sparsity and diversity. Neurocomputing. 2014;134:214–21.
51. Yin XC, Huang K, Yang C, Hao HW. Convex ensemble learning with sparsity and diversity. Inf Fusion. 2014;20:49–59.
52. Zhang Y, Burer A, Street WN, Bennett K, Parrado-hern E. Ensemble pruning via semi-definite programming. J Mach Learn Res. 2006;7:1315–38.
53. Zhou ZH. Ensemble methods: foundations and algorithms. Boca Raton: Chamman & Hall/CRC; 2012.
54. Zhou ZH, Wu J, Tang W. Ensembling neural networks: many could be better than all. Artif Intell. 2002;137:239–63.