Pattern Recognition Letters 74 (2016) 38–45

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Combining diversity measures for ensemble pruning☆

George D.C. Cavalcanti[a,*], Luiz S. Oliveira[a,b], Thiago J.M. Moura[a,c], Guilherme V. Carvalho[a]

[a] Centro de Informática (CIn), Universidade Federal de Pernambuco (UFPE), Av. Jornalista Anibal Fernandes s/n, Recife, Brazil
[b] Universidade Federal do Paraná (UFPR), Rua Cel. Francisco Heraclito dos Santos, 100, Curitiba, Brazil
[c] Instituto Federal de Educação, Ciência e Tecnologia da Paraíba (IFPB), Av. Primeiro de Maio, 720, João Pessoa, Brazil

ARTICLE INFO

Article history:
Received 17 August 2015
Available online 11 February 2016

Keywords:
Ensemble pruning
Diversity measure
Graphs
Multiple classifier systems

ABSTRACT

Multiple Classifier Systems (MCSs) have been widely used in the area of pattern recognition due to the difficult task that is to find a single classifier that has a good performance on a great variety of problems. Studies have shown that MCSs generate a large quantity of classifiers and that those classifiers have redundancy between each other. Various methods proposed to decrease the number of classifiers without worsening the performance of the ensemble succeeded when using diversity to drive the pruning process. In this work we propose a pruning method that combines different pairwise diversity matrices through a genetic algorithm. The combined diversity matrix is then used to group similar classifiers, i.e., those with low diversity, that should not belong to the same ensemble. In order to generate candidate ensembles, we transform the combined diversity matrix into one or more graphs and then apply a graph coloring method. The proposed method was assessed on 21 datasets from the UCI Machine Learning Repository and its results were compared with five state-of-the-art techniques in ensemble pruning. Results have shown that the proposed pruning method obtains smaller ensembles than the state-of-the-art techniques while improving the recognition rates.

1. Introduction

Ensemble methods began gathering attention of the pattern recognition community after Wolpert's no free lunch theorem [1] stated that given enough problems and two classifiers, the number of problems in which a classifier outperforms the other is roughly equal. This means that searching for a single classifier model that had good performance at a wide array of problems is unproductive. Multiple classifier systems, another name for ensembles of classifiers, avoid the problem stated by Wolpert by combining the output of various classifiers. The combination softens the differences between problems in which the classifiers of the ensemble have different performances. Besides this softening effect ensembles use weaker classifiers which are easier to train.

The main problem with ensemble methods, such as Bagging or AdaBoost, is that the final ensemble has a large number of classifiers. In the late 1990s it had been shown that some of the classifiers in the ensemble could be removed without impairing the ensembles ability to generalize [2,3]. These findings led to more research being done on the area of ensemble pruning since searching exhaustively for the best subset of an ensemble can become intractable for relatively small ensemble sizes.

The seminal work in this field was published by Margineantu and Dietterich [3] where the authors compared five different pruning algorithms on ten datasets and concluded that in most of the experiments the ensemble of decision trees produced by AdaBoost could be pruned substantially with no considerable impacts of the performance. Tamon and Xiang [4] proposed an improvement to one of the methods described by Margineantu and Dietterich [3], the Kappa pruning, and also addressed the boosting pruning problem from a theoretical perspective.

Zhou et al. [5] introduced the GASEN (Genetic Algorithm based Selective ENsemble) method, which selects the classifiers to constitute an ensemble according to some evolved weights that could characterize the fitness of including the classifiers in the ensemble. In their empirical study they used neural networks as classifiers, genetic algorithms, and 20 different datasets. They show that the pruned ensemble generated by the GASEN method was able to outperform the popular ensemble approaches such as Bagging and Boosting. Other examples of methods using global search to prune the ensembles can be found in [6,7].

A different approach, based on a greed local search, was proposed by Martínez-Muñoz and Suárez [8,9], Martínez-Muñoz et al. [10]. In these works they explored the idea that the order in which classifiers are aggregated in ensemble methods can be an

☆ This paper has been recommended for acceptance by Egon L. van den Broek.

* Corresponding author. Tel.: +55 81 2126 8430.
E-mail address: gdcc@cin.ufpe.br, darmiton@gmail.com (G.D.C. Cavalcanti).

http://dx.doi.org/10.1016/j.patrec.2016.01.029
0167-8655/

important tool to prune ensembles. Their algorithm is based on ordering the predictors in the ensemble according to a number of rules that exploit the complementariness of the individual classifiers. Experiments on several UCI repository datasets show that ordered ensembles produced a generalization error lower than the full ensembles created by Bagging.

An issue not to be neglected when building ensembles of classifiers is the diversity, which is the underpinning to successful deployment of classifiers ensemble. Empirical results have shown that there exists positive correlation between performance of the ensemble and diversity among the base classifiers [11,12]. On the other hand, the usefulness of diversity measures to build ensembles of classifiers is questioned by some authors. Kuncheva and Whitaker [13] performed a considerable amount of experiments but could not find a definitive connection between the diversity measures and the improvement of the ensemble accuracy. In other words, designing diverse classifiers is important but the problem of measuring this diversity and so using it effectively for building better ensembles is still an open problem. Ko et al. [14] investigated 10 diversity measures into a pairwise fusion matrix transformation to combine classifiers and concluded that the use of diversity might slightly improve the methods for classifier combination in some problems, but the effect is not significant. Tang et al. [15] evaluated six different measures of diversity and concluded that none of them is suitable for the task of building ensemble of classifiers. According to the authors, if one exploits diversity measures as criteria to select the base classifiers, then the diversity measure is required to be precise, since the choice of diversity measure will directly influence the final ensemble and subsequently the classification result.

As one may notice, understanding how diversity can be used to build ensembles remains an open problem. In spite of that, the literature shows us several cases where the diversity has been successfully applied to build ensembles of classifiers. Tsymbal et al. [16] point out the importance of the diversity measures during the search problem for ensemble feature selection. Oliveira et al. [17] show that diversity is quite useful to build ensembles of classifiers through feature selection since it helps preventing overfitting during the search. Li et al. [18] presented a theoretical study on the effect of diversity in voting. They concluded that by enforcing large diversity, the hypothesis space complexity of voting can be reduced, and then better generalization performance can be expected. These findings were used to build a method called DREP (Diversity Regularized Ensemble Pruning) which explicitly exploit diversity regularization. Experimental results show that with the help of diversity regularization, DREP is able to achieve significantly better generalization performance with smaller ensemble size than the compared methods.

Motivated by the success of Li et al. [18] and also by the findings of Kuncheva [19], which suggests that a single measure of diversity might not be accurate enough to capture all the relevant diversities in the ensemble, in this study we argue that the combination of several diversity measures can be an useful tool to prune an ensemble of classifiers. To support this idea, we propose an ensemble pruning method where the undermining concept is the combination of different pairwise diversity matrices. The weights of this combination are provided by a genetic algorithm. From the combined diversity matrix we are able to group similar classifiers, i.e., those with low diversity, that should not belong to the same ensemble. In order to generate the candidate ensembles, we transform the combined diversity matrix into one or more graphs and then apply a graph coloring method. The fitness of the genetic algorithm is provided by the ensemble that minimizes the error on a validation set.

Through a set of comprehensive experiments on 21 datasets of the UCI repository we show that the proposed method is able to considerably reduce the original size of the ensemble while improving the recognition rates. The results reached by our method compare favorably to other published methods.

The rest of this article is organized as follows: Section 2 reviews the diversity measures used in this work; Section 3 describes the proposed method for pruning a pool of classifiers; Section 4 reviews the methodology and experiments run to validate the proposed method; Section 5 lists the conclusions that can be taken from the experiments.

Table 1
Contingency table for two classifiers d_i and d_j.

	$d_i = +$	$d_i = -$
$d_j = +$	a	c
$d_j = -$	b	d

2. Diversity measures

There is not a widely accepted definition of diversity between classifiers. For that reason there are many definitions used throughout the literature. In the proposed method five pairwise diversity measures are combined to reach a broader definition of diversity. This section describes these five measures and how to calculate them.

The diversity measures are calculated using a contingency table [20] that summarizes the behavior of two classifiers d_i and d_j across a dataset. Table 1 shows an example of a contingency table. The values on the table have the following meaning: a is the number of examples in the dataset correctly classified by both d_i and d_j; b is the number of examples correctly classified by d_i and incorrectly classified by d_j; c is the number of examples incorrectly classified by d_i and correctly classified by d_j; and d is the number of examples incorrectly classified by both classifiers.

Disagreement is the proportion of examples differently classified by d_i and d_j. Its value is calculated by Eq. (1), where $m = a + b + c + d$. Its value ranges from 0 to 1, with higher values indicating more diversity.

$$dis_{ij} = \frac{b + c}{m} \tag{1}$$

The Q-statistic is defined by Eq. (2). Q_{ij} ranges from -1 to 1, where 0 means the two classifiers are independent, 1 both classifiers make similar predictions, and -1 the classifiers make different predictions.

$$Q_{ij} = \frac{ad - bc}{ad + bc} \tag{2}$$

The Correlation Coefficient of two classifiers is calculated by Eq. (3) and the meaning of its value is similar to that of the Q-statistic.

$$\rho_{ij} = \frac{ad - bc}{\sqrt{(a+b)(a+c)(c+d)(b+d)}} \tag{3}$$

The Kappa-statistic is widely used in statistics and was used to analyze the diversity between classifiers for the first time by Margineantu and Dietterich [3]. κ_p (Eq. (4)) is equal to 1 if the classifiers completely agree, 0 if they randomly agree, and less than 0 is a rare case that happens when they agree less than what is expected by chance.

$$\kappa_p = \frac{\Theta_1 - \Theta_2}{1 - \Theta_2} \tag{4}$$

where

$$\Theta_1 = \frac{a + d}{m}, \tag{5}$$

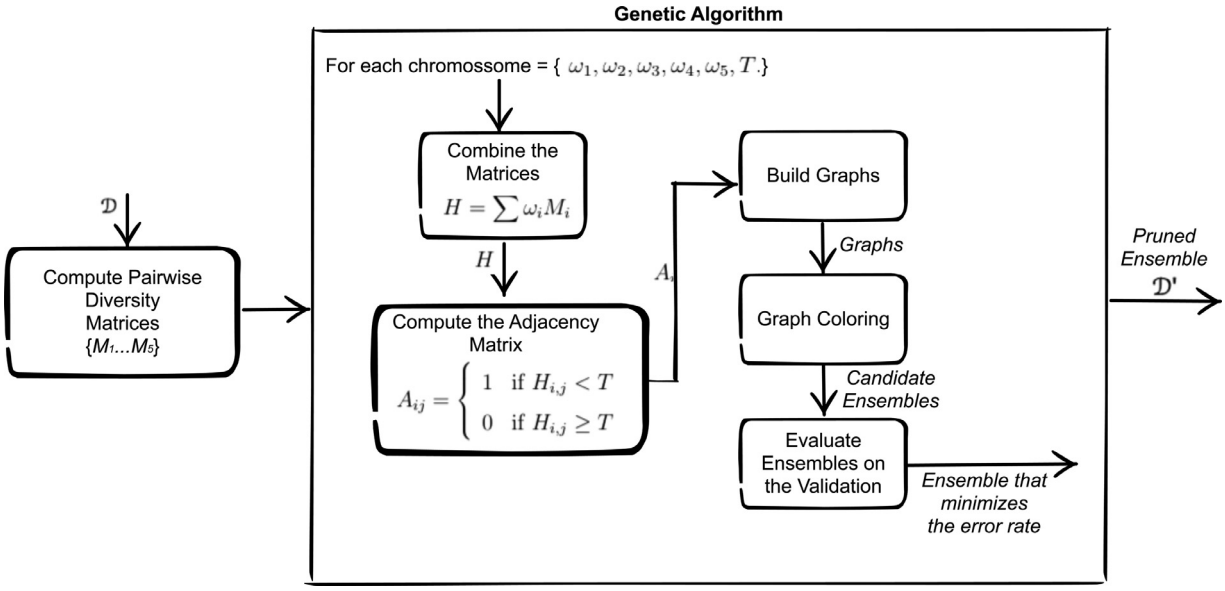

Fig. 1. Outline of the proposed method (DivP).

$$\Theta_2 = \frac{(a+b)(a+c)+(c+d)(b+d)}{m^2} \tag{6}$$

The Double-fault measure [21] is the proportion of examples misclassified by both classifiers and its value is calculated using Eq. (7).

$$DF_{ij} = \frac{d}{m} \tag{7}$$

3. Proposed method

In this section we introduce the proposed method, which combines diversity measures for ensemble pruning. We call it DivP. It is composed of two main modules: computation of the diversity matrices based on the initial pool of classifiers and the pruning method that is performed inside a genetic algorithm. Fig. 1 depicts all the modules of the proposed method, which are described in the following sections.

3.1. Initial pool of classifiers

In this work the initial pool of classifiers $\mathcal{D}$ of size L was created using Bagging [22]. To take advantage of this method, the base classifier must be unstable, i.e., minor changes in the training set can lead to major changes in the classifier output. The unstable classifier used in our experiments was the Perceptron with threshold activation function.

3.2. Diversity measures

The five pairwise diversity measures reviewed in Section 2 were considered in this work: Disagreement (M_1), Q-Statistics (M_2), Correlation Coefficient (M_3), Kappa-statistic (M_4), and Double-fault measure (M_5). Here it is worth mentioning that the method is not limited to these five measures. They were selected because, to the best of our knowledge, they are the most commonly pairwise diversity measures used in the literature.

The output of this module consists in five $L \times L$ matrices. The diversity values for each pair of classifiers are calculated using the validation set $\mathcal{V}_1$. The weights used to combine these matrices are found by the genetic algorithm as discussed below.

3.3. Genetic algorithm

The pruning mechanism is based on a genetic algorithm with real representation, crossover intermediate function, adaptive feasible mutation, stochastic uniform selection, and elitism which is implemented using a generational procedure. The following parameter settings were employed: population size: 22, number of generations: 600, and probability of crossover: 0.8. All these parameters were defined empirically.

The chromosome encodes six real values, five weights $(\omega_1, \ldots, \omega_5)$ to combine the diversity matrices and the threshold T used to compute the adjacency matrix. The fitness function consists in finding the candidate ensemble that minimizes the error rate on the validation set $\mathcal{V}_2$. In the end, the fittest individual represents the pruned ensemble that minimizes the error rate on an independent validation set.

3.3.1. Computing the fitness

In this section we explain in details how do compute the fitness function based on the pairwise diversity matrices. The first step is to calculate the combined diversity matrix (H) considering the weights produced by the genetic algorithm. This is done by Eq. (8).

$$H = \sum_{i=1}^{5} \omega_i M_i \tag{8}$$

Thereafter we create the adjacency matrix, which is defined according to Eq. (9). This rule specifies that if the combined diversity between classifiers d_i and d_j is smaller than the threshold T there will be an edge between the vertices i and j.

$$A_{ij} = \begin{cases} 1 & \text{if } H_{i,j} < T \\ 0 & \text{if } H_{i,j} \geq T \end{cases} \tag{9}$$

Eqs. (10) and (11) show examples of H and A matrices, respectively. These matrices were computed using a pool of five classifiers; in this example, the threshold $T = 1.1$. So, applying the matrix H to Eq. (9) we obtain the matrix A showed in (11). Notice that the main diagonal is set to zero to avoid linking a vertice to itself.

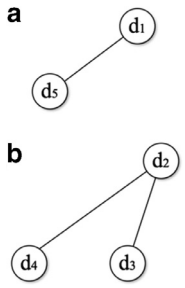

Fig. 2. Two graphs created from the adjacency matrix A grouping classifiers with low diversity among them.

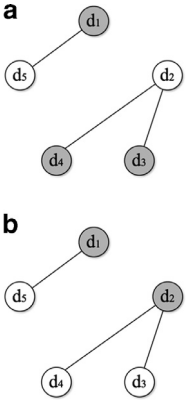

Fig. 3. Two possible color configurations for the graphs depicted in Fig. 2.

$$H = \begin{array}{c} \\ d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \end{array} \begin{array}{c} \begin{array}{ccccc} d_1 & d_2 & d_3 & d_4 & d_5 \end{array} \\ \begin{pmatrix} 0.0 & 1.5 & 1.3 & 1.2 & 1.0 \\ 1.5 & 0.0 & 0.7 & 0.9 & 1.2 \\ 1.3 & 0.7 & 0.0 & 1.1 & 1.7 \\ 1.2 & 0.9 & 1.1 & 0.0 & 1.4 \\ 1.0 & 1.2 & 1.7 & 1.4 & 0.0 \end{pmatrix} \end{array} \tag{10}$$

$$A = \begin{array}{c} \\ d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \end{array} \begin{array}{c} \begin{array}{ccccc} d_1 & d_2 & d_3 & d_4 & d_5 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array} \tag{11}$$

Building a graph from the adjacency matrix is straightforward. From matrix A two independent graphs are build as presented in Fig. 2. The first one (Fig. 2(a)) indicates that classifiers d_1 and d_5 should belong to the same graph since they have a low combined diversity. In other words, d_1 and d_5 should not belong to the same ensemble. The same happens in the second graph (Fig. 2(b)). In this case, the diversity between the pairs of classifiers d_2 and d_4 and d_2 and d_3 is low.

The problem now consists in building the candidate ensemble from these graphs, i.e., putting together those classifiers with a high degree of diversity. From the graph perspective, this means that adjacent vertices should not be in the same ensemble. To perform this operation, we have employed a graph coloring algorithm based on greedy search so that adjacent vertices have different colors. Fig. 3 shows the graphs depicted in Fig. 2 after the coloring process.

The candidate ensembles are created by grouping the vertices (classifiers) with the same color. This is done for all possible color configurations. Therefore, for the graphs presented in Fig. 3(a) and (b) the candidate ensembles would be $\{d_1, d_3, d_4\}$, $\{d_2, d_5\}$ and $\{d_3, d_4, d_5\}$, $\{d_2, d_3\}$, respectively.

Each candidate ensemble is evaluated using the second validation set $\mathcal{V}_2$. The candidate ensemble with the best performance (smallest error on the set) will be used as the output ensemble for this individual.

Once one of the termination conditions of the genetic algorithm is satisfied, usually reaching the maximum number of generations, the fittest individual within the final population is selected. The candidate ensemble that resulted in that individual's fitness is then chosen to be the resulting ensemble $\mathcal{D}'$ of the pruning process.

The proposed technique does not impose any restrictions on the method used for combining the classifiers on the ensemble. Restriction can be placed depending on the type of classifier used as base classifier. Plurality voting is chosen for having good results and being easy to understand and use.

Table 2
Datasets used in this work sorted by the number of instances.

Dataset	# Instances	# Attributes	# Classes
Wine	178	13	3
Parkinsons	195	22	2
Ecoli	336	7	8
Ionosphere	351	34	2
Musk	476	166	2
Balance-scale	625	4	3
Transfusion	748	4	2
Pima	768	8	2
CMC	1473	9	3
Wineq-red	1599	11	6
Segment	2310	19	7
Spambase	4601	57	2
Wineq-white	4898	11	7
Waveform	5000	21	3
Phoneme	5404	5	2
Wall-following	5456	24	4
Page-blocks	5473	10	5
Satimage	6435	36	6
Pen-digits	10992	16	10
Magic04	19020	10	2
Shuttle	58000	9	7

4. Experiments

In order show how DivP performs, we have carried out experiments on 21 classification problems from the UCI Machine Learning Repository [23]. These databases, described in Table 2, contain different types of problems with different number of instances, attributes, and classes.

The pool of classifiers was created using Bagging and the Perceptron was the unstable classifier used in our experiments. In our experiments, the size of the pool assumes five different values, $L = \{50, 100, 150, 200, 250\}$. In this case, the goal is to assess the impacts of the pruning methods on pools of different sizes.

Experiments were run using k-fold cross validation with $k = 6$ divided as: three folds for training, two for validation, and one for test.

Majority vote was the fusion rule employed because (i) it does not assume prior knowledge about the classifiers, (ii) it is a non-trainable rule, (iii) it can be used with any classifier since it is a hard level combination rule – soft level combination rules require classifiers probabilities estimation. Besides, the literature pruning methods we have selected to assess the performance of the proposed method also used majority vote. In spite of that, we also evaluated other combination rules, such as: average, product, median, minimum, and maximum. However, majority vote was slightly better when compared with these combination rules. So, for the sake of clarity, we decided to report only the results of the proposed approach with majority vote. In order to speed up the convergence of the genetic algorithm the first two individuals of the population are initialized with the single best classifier of the pool and the ensemble composed of all classifiers of the pool,

Table 3
Accuracy of the ensembles for different pool sizes (50, 100, 150, 200, and 250) on the test set. $\vee$, $\tilde{x}$, and $\wedge$ are the performances of the best, median, and worst classifier observing the whole pool.

Dataset	50				100				150				200				250			
	DivP	$\vee$	$\tilde{x}$	$\wedge$	DivP	$\vee$	$\tilde{x}$	$\wedge$	DivP	$\vee$	$\tilde{x}$	$\wedge$	DivP	$\vee$	$\tilde{x}$	$\wedge$	DivP	$\vee$	$\tilde{x}$	$\wedge$
Wine	95.0	98.3	93.3	84.8	93.9	98.3	92.7	82.0	97.2	99.4	94.3	84.2	97.2	99.4	94.9	82.0	94.4	100.0	94.9	82.6
Parkinsons	84.6	90.2	80.2	66.2	82.1	90.8	80.5	63.6	84.6	91.8	79.5	61.1	80.5	92.3	80.5	55.9	83.0	91.3	80.0	60.5
Ecoli	74.9	80.9	72.9	62.2	76.9	79.6	72.8	59.8	78.3	83.9	74.4	54.3	77.2	80.5	72.8	53.9	74.6	80.6	72.2	49.1
Ionosphere	88.9	94.3	88.1	81.8	89.7	94.9	88.0	80.1	88.6	94.6	88.1	80.4	88.9	95.2	88.6	81.2	89.8	94.0	88.0	79.8
Musk	75.0	83.4	73.9	60.5	76.9	84.2	75.5	57.8	79.2	85.5	74.5	56.1	78.8	85.9	75.8	56.7	78.0	85.9	76.3	59.0
Balance-scale	88.8	91.4	87.7	76.3	88.5	92.3	87.7	73.0	90.1	92.5	87.4	71.9	89.3	93.6	87.5	71.2	87.7	93.0	87.5	72.8
Transfusion	75.8	79.3	70.7	48.4	76.8	78.5	70.1	47.7	77.1	78.9	70.3	47.5	75.9	79.8	70.1	45.6	76.6	79.3	70.6	48.1
Pima	74.6	77.5	70.0	52.2	74.7	77.6	69.9	57.4	74.1	78.3	69.4	53.4	74.2	79.6	70.2	51.6	75.5	78.8	69.8	51.8
CMC	49.6	51.5	45.6	36.7	49.8	52.6	45.2	32.5	49.0	53.0	45.5	31.0	51.5	53.0	45.0	31.9	50.0	53.4	44.9	32.1
Wineq-red	56.0	54.4	44.7	32.8	55.9	55.6	44.6	32.5	57.2	56.9	45.8	32.2	55.4	55.9	45.6	32.3	57.7	55.6	45.0	31.4
Segment	91.0	92.0	86.0	69.2	91.3	91.6	86.7	67.3	91.3	91.8	86.3	68.1	90.5	92.1	85.9	62.0	91.5	92.2	86.2	60.6
Spambase	92.1	92.0	87.7	82.0	92.0	92.5	87.9	81.1	92.7	92.5	88.2	81.5	92.1	92.3	88.0	80.4	92.2	92.4	87.9	80.2
Wineq-white	49.9	48.6	40.9	24.7	50.4	48.5	39.6	22.9	50.2	49.1	39.6	22.4	51.2	48.8	39.6	22.4	50.9	49.8	39.5	22.7
Waveform	84.6	84.6	77.3	67.3	84.6	85.3	76.5	66.9	85.2	85.2	77.0	67.2	84.6	85.0	77.3	65.9	85.6	85.5	77.0	65.7
Phoneme	76.5	76.0	69.4	58.5	75.2	76.9	70.0	54.3	76.7	76.6	69.6	56.2	77.4	76.7	69.2	56.5	76.9	77.3	69.8	49.8
Wall-following	67.3	65.0	56.6	43.1	66.9	64.9	56.6	40.9	66.5	65.4	56.6	40.2	66.5	65.3	56.8	38.8	65.9	65.3	56.8	39.4
Page-blocks	95.6	95.6	92.3	87.9	95.5	95.7	93.0	88.1	95.4	95.9	92.9	81.6	96.0	96.2	92.7	85.6	95.7	95.8	92.2	84.2
Satimage	66.3	63.4	54.4	36.2	66.3	63.8	54.5	35.6	66.9	64.3	54.5	34.1	67.2	64.2	54.5	35.1	66.7	64.8	54.5	33.7
Pen-digits	90.6	89.7	87.0	73.0	90.7	90.6	87.4	73.5	90.8	90.2	87.2	69.6	90.8	90.7	87.3	72.1	90.9	90.5	87.2	68.7
Magic04	75.4	72.1	69.9	67.9	75.6	72.7	70.1	67.9	75.7	72.2	70.3	67.9	75.6	73.0	70.1	67.4	75.7	72.5	70.0	67.6
Shuttle	97.0	96.9	86.1	46.4	97.1	96.9	86.6	33.0	97.0	96.9	86.2	24.8	97.2	97.0	86.3	30.3	97.2	96.9	86.4	23.9
Average	78.5	79.9	73.1	59.9	78.6	80.2	73.1	58.0	79.2	80.7	73.2	56.5	79.0	80.8	73.3	56.1	78.9	80.8	73.2	55.4

Table 4
Accuracy of the ensembles for different pool sizes (50, 100, 150, 200, and 250) on the test set. Best results per database are in bold. B: Bagging; DivP: Pruned ensemble using majority vote; #: Number of classifiers after pruning.

Dataset	50			100			150			200			250		
	B	DivP	#	B	DivP	#	B	DivP	#	B	DivP	#	B	DivP	#
Wine	94.9	**95.0**	1	**95.0**	93.8	1	96.0	**97.2**	1	96.6	**97.2**	1	**96.1**	94.4	1
Parkinsons	82.5	**84.6**	1	**83.1**	82.1	1	82.6	**84.6**	1	**84.1**	80.5	1	**84.1**	83.0	1
Ecoli	**76.4**	74.9	5	**78.7**	76.9	4	**79.4**	78.3	2	**77.3**	77.2	3	**78.5**	74.6	2
Ionosphere	**91.2**	88.9	1	89.2	**89.7**	1	88.3	**88.6**	2	**89.7**	88.9	2	89.2	**89.7**	1
Musk	**77.3**	75.0	3	**80.6**	76.9	3	78.6	**79.2**	2	**79.8**	78.8	3	**80.7**	78.0	1
Balance-scale	87.5	**88.8**	1	88.2	**88.5**	2	87.4	**90.1**	1	87.2	**89.3**	1	**88.2**	87.7	2
Transfusion	75.3	**75.8**	3	74.9	**76.7**	5	74.9	**77.1**	3	75.1	**75.9**	3	75.1	**76.6**	3
Pima	74.4	**74.6**	3	**75.1**	74.7	3	**74.1**	**74.1**	3	72.3	**74.2**	3	75.1	**75.5**	3
CMC	**50.2**	49.6	4	**50.4**	49.8	5	**50.5**	48.9	5	**51.6**	51.5	6	**50.0**	**50.0**	4
Wineq-red	**57.3**	56.0	5	55.4	**55.9**	6	56.3	**57.2**	6	**58.6**	55.4	6	56.8	**57.7**	9
Segment	**91.5**	91.0	4	91.2	**91.3**	5	91.2	**91.3**	4	**91.1**	90.5	6	**91.6**	91.5	12
Spambase	92.0	**92.1**	7	**92.3**	92.0	4	92.3	**92.7**	7	**92.5**	92.1	7	**92.3**	92.2	6
Wineq-white	48.3	**49.9**	4	48.7	**50.3**	4	48.6	**50.2**	2	48.6	**51.2**	4	48.6	**50.9**	4
Waveform	83.7	**84.6**	3	83.5	**84.6**	3	**85.2**	**85.2**	3	84.5	**84.6**	6	85.1	**85.6**	3
Phoneme	71.7	**76.5**	5	72.4	**75.2**	3	72.1	**76.7**	2	72.0	**77.4**	3	72.2	**76.9**	3
Wall-following	65.2	**67.3**	7	66.2	**66.9**	9	65.7	**66.5**	10	66.2	**66.5**	5	**66.6**	65.9	9
Page-blocks	94.5	**95.6**	3	95.3	**95.5**	4	95.0	**95.4**	3	95.2	**96.0**	2	94.4	**95.7**	3
Satimage	64.2	**66.2**	5	64.4	**66.3**	6	64.2	**66.9**	4	64.5	**67.1**	6	64.3	**66.7**	4
Pen-digits	90.1	**90.6**	8	90.6	**90.7**	6	90.6	**90.8**	10	90.5	**90.8**	7	90.4	**90.9**	5
Magic04	71.7	**75.4**	2	72.0	**75.6**	3	71.8	**75.7**	2	72.0	**75.6**	2	71.8	**75.7**	3
Shuttle	91.2	**97.0**	2	91.2	**97.1**	3	91.5	**97.0**	5	91.4	**97.2**	2	91.6	**97.2**	3
Average			3.7			3.9			3.7			3.8			3.9
Win/Tie/Loss	15/0/6			14/0/7			17/2/2			13/0/8			12/1/8		

respectively. The algorithm stops if there is no improvement in the fitness function for 15 consecutive generations or when it reaches the maximum number of generations, which is set to 600. In our experiments the average number of iterations was 25.

Table 3 shows the results of the proposed approach (DivP) against the results extracted from the whole initial pool of classifiers. The columns labeled with $\vee$ and $\wedge$ show the accuracies of the best and the worst classifier in the pool on the test set, respectively; and, the columns labeled with $\tilde{x}$ show the median accuracy of the classifiers in the pool on the test set. DivP attains a classification accuracy that is always better than the average classifier.

The results of the proposed pruning method on the test set for all the databases are reported in Table 4. It compares the performance of the original ensemble (B) and the proposed pruned ensemble (DivP). The average number of classifiers after pruning the ensemble is also available.

Table 4 also shows the win-tie-loss count where "win"/"tie"/"loss" means the number of times the pruned ensemble scores better/neutral/inferior than the original ensemble. The number of "wins" is greater than "tie" and "loss", but it is worth remarking that the proposed method has a more homogeneous performance (i.e., more wins) on bigger databases. This can be explained by the fact that bigger databases allow bigger

Table 5
Evaluation of the proposal with and without the combination of diversity measures. Original method (DivP), Disagreement (M_1), Q-Statistics (M_2), Correlation Coefficient (M_3), Kappa-statistic (M_4), and Double-fault (M_5).

Dataset	*DivP*	M_1	M_2	M_3	M_4	M_5
Wine	**97.2**	94.9	96.0	94.9	95.5	94.3
Parkinsons	**84.6**	72.4	74.4	72.4	72.4	81.6
Ecoli	**78.3**	75.8	77.6	77.0	77.6	77.6
Ionosphere	88.6	90.0	89.8	89.5	**90.1**	88.6
Musk	**79.2**	67.4	70.4	70.2	70.2	78.6
Balance-scale	**90.1**	85.9	86.7	87.8	88.2	87.2
Transfusion	**77.1**	54.7	54.3	53.5	53.9	73.8
Pima	**74.1**	67.6	67.6	67.1	67.1	72.0
CMC	**48.9**	40.9	40.9	40.5	40.5	47.3
Wineq-red	**57.2**	46.3	46.2	46.5	46.5	47.8
Segment	**91.3**	77.9	86.8	87.8	85.4	90.3
Spambase	**92.7**	87.3	86.8	86.0	85.9	92.5
Wineq-white	**50.2**	40.4	40.3	40.5	40.5	46.1
Waveform	**85.2**	69.8	71.1	70.8	70.8	84.0
Phoneme	**76.7**	65.1	65.8	65.1	65.1	72.3
Wall-following	**66.5**	44.8	45.4	44.8	44.8	63.1
Page-blocks	**95.4**	92.5	93.9	94.0	94.0	93.7
Satimage	**66.9**	49.7	53.2	53.8	53.9	57.9
Pen-digits	**90.8**	84.7	86.3	85.6	84.7	89.6
Magic04	**75.7**	68.6	68.6	68.6	68.6	71.2
Shuttle	**97.0**	57.9	58.5	63.2	63.2	91.3
Average	**79.2**	68.3	69.5	69.5	69.5	76.2

validation sets, which are always useful for the optimization process.

Besides the good performance, it is important to highlight the capacity of the proposed method in pruning ensemble independent of its original size. From Table 4 we can see that the ensembles were considerably reduced to less than 10 classifiers. The average size of the pruned ensemble is four classifiers for all the experiments. One may observe that sometimes the pruned ensemble produced by the proposed method contains only one classifier. This means that the method was not able to outperform the single best on the validation set, hence, the pruned ensemble contains only the single best or one classifier that achieved the same performance of the single best. In these cases, the single best and the pruned ensemble reach the same performance on the validation set, but they may be different on the test set.

4.1. Analyzing the diversity measures

The proposed method takes into account an optimization approach to search for the best diversity measure weights ω_i and the threshold T, which are used in Eqs. (8) and (9). Based on this strategy, one question arises: do we need to combine different diversity measures? In other words, is only one diversity measure enough to achieve high accuracy rates? In order to address this question, we evaluated a modified version of the proposed approach in which just one diversity measure is used at a time. So, in this modified version, Eq. (8) does not combine the diversity measures and H is replaced by one diversity matrix $H = M_i$.

Since no optimization is used in this modified version of the proposed approach, we need to define a procedure to set the threshold T (Eq. (9)). This procedure has three steps: (i) H is equal to one diversity matrix M_i ($H = M_i$), (ii) θ_{d_i} is the sum of the diversities values per classifier d_i given by $\theta_{d_i} = \sum_{k=1,k\neq i}^{L} H(i,k)$, where L is the number of classifiers, and (iii) the classifiers d_i associated with the highest values of θ_{d_i} are selected to compose the final ensemble.

To perform these experiments we have selected the pool size that produced the best results in the previous experiments ($L = 150$). In the cases where each diversity measure was assessed independently, 5% of the classifiers associated with the highest values of θ_{d_i} were selected to compose the final ensemble. Table 5 shows the results of the proposed approach with and without the combination of the diversity measures. The combination of five diversity measures obtained the best accuracy rates in all except one dataset. These results corroborate to our initial hypothesis that the combination of diversity measures can be an useful tool to prune an ensemble of classifier.

4.2. Comparing with the state-of-the-art

To better assess the proposed method, we have implement the following state-of-the-art pruning techniques: Aggregation Ordering in Bagging (AGOB) [8], Pruning in Ordered Bagging Ensemble (POBE) [9], Genetic Algorithm based Selective ENsemble (GASEN) [5], Diversity Regularized Ensemble Pruning [18], and Kappa Pruning [3]. AGOB and POBE methods explore the idea that the order in which the classifiers are aggregated in the ensemble is important.

Table 6
Recognition rates (%) and size (#) of the pruned ensembles achieved by comparative methods.

Dataset	Bagging	DivP	#	AGOB	#	DREP	#	GASEN	#	Kappa	#	POBE	#
Wine	96.0	**97.2**	1	95.5	19	96.6	75	96.0	74	96.1	30	95.5	22
Parkinsons	82.6	**84.6**	1	83.1	26	83.1	75	83.1	71	81.1	30	82.1	34
Ecoli	79.4	78.3	2	**79.7**	28	**79.7**	75	79.4	75	78.2	30	78.5	31
Ionosphere	88.3	88.6	2	88.6	33	88.6	75	88.1	66	88.9	30	**90.0**	31
Musk	78.6	79.2	2	78.4	27	79.0	75	79.6	79	78.8	30	**79.8**	30
Balance-scale	87.4	**90.1**	1	88.3	23	88.3	75	87.2	78	88.2	30	89.6	27
Transfusion	74.9	**77.1**	3	68.9	21	**77.1**	75	74.7	74	65.0	30	74.7	34
Pima	74.1	74.1	3	74.1	19	**75.9**	75	74.5	113	69.7	30	71.2	34
CMC	50.5	48.9	5	**51.0**	41	50.4	75	50.5	141	50.4	30	49.6	35
Wineq-red	56.3	**57.2**	6	56.0	41	57.0	75	56.5	142	54.5	30	54.5	35
Segment	91.2	91.3	4	91.2	36	91.0	75	91.3	114	**91.4**	30	91.1	39
Spambase	92.3	**92.7**	7	91.7	37	92.6	75	92.4	114	90.4	30	92.1	33
Wineq-white	48.6	**50.2**	2	**50.2**	32	49.4	75	48.6	142	48.7	30	47.0	40
Waveform	**85.2**	**85.2**	3	82.9	30	**85.2**	75	85.1	110	80.0	30	82.6	36
Phoneme	72.1	**76.7**	2	71.3	27	72.5	75	71.3	141	66.8	30	71.9	32
Wall-following	65.7	**66.5**	10	64.6	40	65.8	75	65.8	143	61.5	30	66.4	36
Page-blocks	95.0	95.4	3	95.5	24	**95.7**	75	95.0	141	94.8	30	95.0	40
Satimage	64.2	**66.9**	4	64.2	36	64.3	75	64.3	142	63.5	30	65.1	38
Pen-digits	90.6	90.8	10	90.7	23	91.1	75	90.6	143	**91.2**	30	90.9	35
Magic04	71.8	**75.7**	2	74.9	2	72.6	75	71.7	144	75.5	30	69.8	30
Shuttle	91.5	**97.0**	5	84.2	15	91.8	75	91.5	143	89.4	30	94.8	24
Win/Tie/Loss	0/1/20	10/3/8		1/2/18		2/3/16		0/0/21		2/0/19		2/0/19	

Table 7
Results of the Wilcoxon signed-rank test. R^+ corresponds to the sum of the ranks for DivP and R^- for the literature methods. Cases where the difference is significant are marked with bullet "•", otherwise are marked with circle "○".

Comparison	R^+	R^-	p-value
DivP vs Bagging	29.43	2.71	0.0010 •
DivP vs AGOB	41.45	3.65	0.0038 •
DivP vs DREP	20.98	5.79	0.0542 ○
DivP vs GASEN	29.30	3.53	0.0053 •
DivP vs Kappa	61.25	2.22	0.0005 •
DivP vs POBE	33.96	3.12	0.0022 •

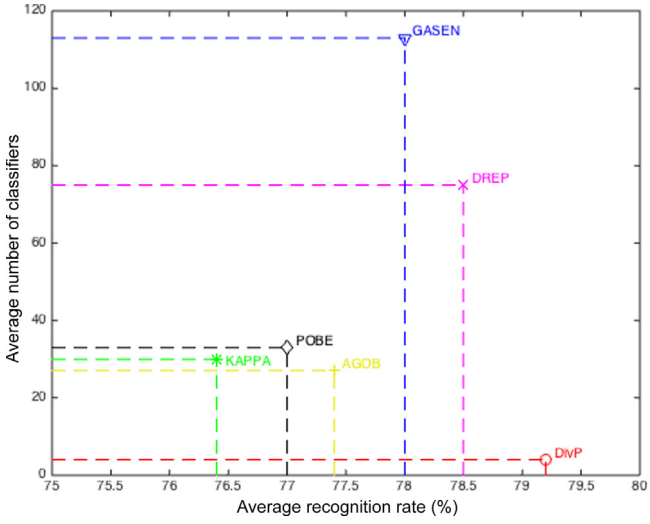

Fig. 4. Ensemble size vs average performance.

In Kappa Pruning, the final size of the ensemble is a parameter that must be informed by the user. The GASEN method uses a genetic algorithm to assign weights to the classifiers of the ensemble and prune those classifiers with a weight below a certain threshold that is set by the user. Finally, DREP starts the pool with just one classifier (the one in the pool that minimizes the error rate on the validation set) and grows the ensemble by adding new classifiers taking into account the diversity and the performance of the ensemble. These ensemble pruning methods have their own stop criterion and we implemented as described in their papers.

The recognition rates achieved by all methods using an initial pool with 150 classifiers are reported in Table 6. To test the significance of difference between the recognition rates of the proposed method and that of Bagging and five other pruning algorithms, we performed the Wilcoxon signed-rank test at significance level of 5% ($\alpha = 0.05$). The p-values produced in the test are reported in Table 7, which shows that the proposed method produces better results with statistical difference in five out of six cases. The only case where the difference was not significant was DivP vs DREP, but even then the ranks are in favor of DivP. These results put out the superiority of the proposed approach with respect to the literature ones.

Besides comparing favorably to the literature in terms of recognition rate, Table 6 also shows that the combination of diversity measures allows the pruning method to generate considerably small ensembles. This trade-off can also be visualized in Fig. 4, where the best results should be located in the right lower quadrant of the plot.

5. Conclusions

In this paper we introduced an ensemble pruning method that combines multiple diversity measures by using a genetic algorithm and a graph coloring algorithm. By comparing the proposed method against Bagging and five other ensemble pruning methods available in the literature, we show its efficiency both in terms of generalization and capacity of pruning the original Bagging ensemble.

Regarding the performance, the experiments show that the proposed approach achieves better performance in 10 out of the 21 datasets used in our tests when compared to Bagging and five other pruning methods. With respect the size of the pruned ensembles, our experiments show that in average the final ensemble is composed of four classifiers, independently of the number of classifiers available in the original pool generated by Bagging. This is considerably smaller than the other methods available in the literature.

Combining diversity measures also brought advantages to the proposed method, since there is no widely accepted definition of diversity between classifiers. The measures often have a different perspective as to what defines diversity, which is why we believe that its combination enhances the pruning results. Researchers in this area, [13], found that there is no clear correlation between diversity and performance, but that heuristics such as the one used here can enhance the pruning results.

Acknowledgment

This research has been supported by the following Brazilian agencies: CNPq (#446831/2014-0, #151145/2014-8) and FACEPE (APQ-0192-1.03/14).

References

[1] D.H. Wolpert, The lack of a priori distinctions between learning algorithms, Neural Comput. 8 (7) (1996) 1341–1390.
[2] D. Partridge, W.B. Yates, Engineering multiversion neural-net systems, Neural Comput. 8 (4) (1996) 869–893.
[3] D.D. Margineantu, T.G. Dietterich, Pruning adaptive boosting, in: Proceedings of the Fourteenth International Conference on Machine Learning, vol. 97, 1997, pp. 211–218.
[4] C. Tamon, J. Xiang, On the boosting pruning problem, in: Proceedings of the Eleventh European Conference on Machine Learning, 2000, pp. 404–412.
[5] Z.-H. Zhou, J. Wu, W. Tang, Ensembling neural networks: Many could be better than all, Artif. Intell. 137 (1) (2002) 239–263.
[6] R. Caruana, A.N. Mizil, G. Crew, A. Ksikes, Ensemble selection from libraries of models, in: Proceedings of the Twenty-first International Conference on Machine Learning, 2004, p. 18.
[7] G. Giacinto, F. Roli, G. Fumera, Design of effective multiple classifier systems by clustering classifiers, in: Proceedings of the Fifteenth International Conference on Pattern Recognition, 2000, pp. 160–163.
[8] G. Martínez-Muñoz, A. Suárez, Aggregation ordering in bagging, in: Proceedings of the IASTED International Conference on Artificial Intelligence and Applications, 2004, pp. 258–263.
[9] G. Martínez-Muñoz, A. Suárez, Pruning in ordered bagging ensembles, in: Proceedings of the Twenty-third International Conference on Machine learning, 2006, pp. 609–616.
[10] G. Martínez-Muñoz, D. Hernández-Lobato, A. Suárez, An analysis of ensemble pruning techniques based on ordered aggregation, IEEE Trans. Pattern Anal. Mach. Intell. 31 (2) (2009) 245–259.
[11] T. Dietterich, An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization., Mach. Learn. 40 (2) (2000) 139–157.
[12] L. Kuncheva, C. Whitaker, C. Shipp, R. Duin, Limits on the majority vote accuracy in classifier fusion, Pattern Anal. Appl. 6 (1) (2003) 22–31.
[13] L.I. Kuncheva, C.J. Whitaker, Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy, Mach. Learn. 51 (2) (2003) 181–207.
[14] A. Ko, A.S. Britto-Jr, R. Sabourin, L.S. Oliveira, Pairwise fusion matrix for combining classifiers, Pattern Recognit. 40 (8) (2007) 2198–2210.
[15] E.K. Tang, P. Suganthan, X. Yao, An analysis of diversity measures, Mach. Learn. 65 (1) (2006) 247–271.
[16] A. Tsymbal, M. Pechenizkiy, P. Cunningham, Diversity in search strategies for ensemble feature selection, Inf. Fusion 6 (1) (2006) 83–98.
[17] L.S. Oliveira, M. Morita, R. Sabourin, Feature selection for ensembles applied to handwriting recognition, Int. J. Doc. Anal. Recognit. 8 (4) (2006) 262–279.

[18] N. Li, Y. Yu, Z.-H. Zhou, Diversity regularized ensemble pruning, in: Machine Learning and Knowledge Discovery in Databases, Springer, 2012, pp. 330–345.
[19] L.I. Kuncheva, Combining Pattern Classifiers: Methods and Algorithms, Wiley, 2004.
[20] Z.-H. Zhou, Ensemble Methods: Foundations and Algorithms, Chapman & Hall, 2012.
[21] G. Giacinto, F. Roli, Design of effective neural network ensembles for image classification purposes, Image Vis. Comput. 19 (9) (2001) 699–707.
[22] L. Breiman, Bagging predictors, Mach. Learn. 24 (2) (1996) 123–140.
[23] M. Lichman, UCI machine learning repository, [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science, 2013.