International Journal of Machine Learning and Cybernetics (2022) 13:551–558
https://doi.org/10.1007/s13042-021-01442-1

ORIGINAL ARTICLE

Building heterogeneous ensembles by pooling homogeneous ensembles

Maryam Sabzevari[1] · Gonzalo Martínez-Muñoz[2] · Alberto Suárez[2]

Received: 1 March 2020 / Accepted: 30 September 2021 / Published online: 13 October 2021

Abstract
Heterogeneous ensembles consist of predictors of different types, which are likely to have different biases. If these biases are complementary, the combination of their decisions is beneficial and could be superior to homogeneous ensembles. In this paper, a family of heterogeneous ensembles is built by pooling classifiers from M homogeneous ensembles of different types of size T. Depending on the fraction of base classifiers of each type, a particular heterogeneous combination in this family is represented by a point in a regular simplex in M dimensions. The M vertices of this simplex represent the different homogeneous ensembles. A displacement away from one of these vertices effects a smooth transformation of the corresponding homogeneous ensemble into a heterogeneous one. The optimal composition of such heterogeneous ensemble can be determined using cross-validation or, if bootstrap samples are used to build the individual classifiers, out-of-bag data. The proposed heterogeneous ensemble building strategy, composed of neural networks, SVMs, and random trees (i.e. from a standard random forest), is analyzed in a comprehensive empirical analysis and compared to a benchmark of other heterogeneous and homogeneous ensembles. The achieved results illustrate the gains that can be achieved by the proposed ensemble creation method with respect to both homogeneous ensembles and to the tested heterogeneous building strategy at a fraction of the training cost.

Keywords Ensembles · Homogeneous · Heterogeneous · Simplex · Optimal composition

1 Introduction

Building an effective classifier for a specific problem is a difficult task. To be successful, a variety of aspects need to be taken into account: the data structure, the information that can be used for prediction, the number of the labeled examples available for induction, the noise level, among others. Another crucial choice is the type of predictor to be used. The strategies implemented by the different classifiers are diverse. Another crucial choice is the type of predictor to be used. The strategies implemented by the different classifiers are diverse. For instance, decision trees [1] adopt a divide-and-conquer approach in which the original prediction task is recursively divided by partitioning the attribute space into disjoint regions. Within each of these regions, the prediction problem is simpler than the original. A neural network [2] provides a global sub-symbolic representation of the decision problem in terms of the set of synaptic weights. Another illustration is the strategy adopted in kernel methods, such as Suppor Vector Machines (SVM) [3]. In SVMs the original problem is embedded into an extended feature space. In this extended space, the discrimination problem is solved by finding the widest margin hyperplane that separates classes, except for, possibly, a few instances. In practice, combining the outputs of individual classifiers often leads to more accurate predictions, whence the popularity of ensemble methods [4–8]. A necessary condition to obtain such improvements is that the ensemble members be diverse. In addition, the individual predictors should be complementary, in the sense that each of them tends to make errors on different test instances [4, 9].

Ensembles can be categorized into two groups based on the homogeneity of their base learners. *Homogeneous* ensembles are composed of classifiers of the same type,

✉ Maryam Sabzevari
maryam.sabzevari@aalto.fi

[1] Computer Science Department, Aalto University, Konemiehentie 2, 02150 Espoo, Finland

[2] Escuela Politécnica Superior, Universidad Autónoma de Madrid, Ciudad Universitaria de Cantoblanco, 28049 Madrid, Spain

whereas ensembles composed of classifiers of different types are known as *heterogeneous*. The strategies to generate diversity among the base classifiers are different for homogeneous and heterogeneous ensembles. In homogeneous ensembles, the main difficulty is to generate diversity, in despite of using the same learning algorithm. To this end, one can use bootstrap techniques (e.g. bagging [10]), randomized steps in the base learning algorithm (e.g. the random subspace method used random forest [11]), noise injection in the class labels (e.g. class-swithcing [12]) or adaptive emphasis protocols (e.g. boosting [13]). These techniques, which have been exploited mainly in the context of homogeneous ensembles, can also be used to achieve further diversity in heterogeneous ensembles [14]. However, since different learning algorithms are used to generate the base learners, heterogeneous ensembles are intrinsically diverse. In this case, the main difficulty resides in determining the optimal way to combine the predictions of the different models in the ensemble.

Broadly speaking, the methods to build heterogeneous ensemble can be grouped into two categories. In the first family of methods a fixed number of different models are combined. A second strategy is to build a collection of models with different parametrizations and then select the best subset to include in the final ensemble. In ref. [15] a static heterogeneous ensemble is proposed. In this study 5 different base classifiers are combined: a Support Vector Machine (SVM), a multilayer perceptron (MLP), logistic regression, K nearest neighbors and decision tree. The parameters and architecture of the individual classifiers are determined using 10-fold cross-validation. The proposed approach shows good results in the specific application of lithofacies classification. In ref. [16], a combination of several carefully optimized strong learners, such as deep neural networks, SVM, adaboosts, and gaussian processes, is proposed. The study shows a good performance of the proposed combination over several image classification and UCI tasks with respect to any of its constituents. The authors generate an ensemble using a simple fusion by the sum rule of different classifiers. However, the problem of determining the number of classifiers of each type that need to be used has not been addressed. Furthermore, the optimal composition of the ensemble is problem-dependent. A possible way to overcome this difficulty is to create a library of classifiers and then select a subset for the final ensemble [17–19]. For instance in ref. [17] a library of 2000 different methods trained with a wide range of different parametrizations is build. From that library of models, an iterative greedy selection algorithm is applied to build the final ensemble. The procedure starts with empty ensemble. Then, at each iteration the model that maximizes a performance measure (such as AUC or accuracy on a validation set) is included into the ensemble until all models in the library have been aggregated. Finally, the ensemble with the best performance in the validation set is selected as the final combination. Tsoumakas et al. have made several interesting contribution in this line of research [18, 20]. For instance, in ref. [18] the authors propose a greedy selection method from a library composed of 200 classifiers: 40 neural networks, 60 nearest neighbor classifers, 80 SVMs and 20 decision trees). For each type of classifier, a parameter grid was defined and a single model was trained for each node in the grid. In their proposal, the ensemble is grown incrementally by selecting from the library one classifier at a time. At each step, the selection is made in terms of both individual accuracy and complementarity with the rest of the classifiers in the ensemble. In the problems investigated, such heterogeneous ensembles were found to be more accurate that their constituents. In ref. [19] a genetic algorithm has been proposed to select the optimum structure of a heterogeneous ensemble from 20 different base models. These selection techniques, also known as ensemble pruning, have been also extensively applied to homogeneous ensembles [21, 22]. In a recent study [23], to build an effective heterogeneous combination, the authors trim the base learners with poor performance so that only optimal classifiers will be preserved in the ensemble. The effectiveness of a classifier is identified by means of Area Under the ROC Curve (AUC) measurement.

In another study [24], which to the best of our knowledge is one of the most related works to ours, the authors used a differential evolution algorithm to optimize the weighting votes of diverse base learns in a heterogeneous ensemble. They used the average Matthews Correlation Coefficient (MCC), calculated over 10-fold cross-validation, to evaluate each combination and obtain the base learners' optimal voting weights. We compared this approach against (more details in Sect. 3) the proposed strategy in this study to assess the proposed approach's effectiveness.

In this work we propose to analyze heterogeneous ensembles in which the individual classifiers are selected from homogeneous ensembles. The goal is to build a family of heterogeneous ensembles that can be smoothly transformed into each other another. We aimed to show that carefully selecting the distribution of base classifiers can be beneficial to outperform both homogeneous and fixed heterogeneous ensembles. To this end, a family of heterogeneous ensembles of size T are built by pooling different fractions of base classifiers from M homogeneous ensembles of different types. Depending on the proportion of classifiers of each type, a particular heterogeneous combination in created. This family of heterogeneous ensembles can be represented in a regular simplex in M dimensions. The M vertices of this simplex represent the different homogeneous ensembles. The optimal fraction of each type of classifiers for the final ensemble is found by searching in this simplex.

The main contributions of this study can be summarized as follows:

- A systematic heterogeneous ensemble creation method is proposed that pools base classifiers from the already built homogeneous ensembles.
- A tool for the visualization of the search space is proposed that gives insights of the process of finding the optimal proportion of base classifiers.
- Experimental comparison of the proposed technique with homogeneous ensembles and state-of-the-art heterogeneous ensemble creation method is carried out. This comparison shows the benefits of the proposed heterogeneous ensemble creation method. In addition, the efficiency of partially optimize ensembles of MLPs and SVMs, against their fully optimized single base learners (i.e. MLP and SVM) is tested.

The paper is organized as follows: In Sect. 2, the design process to build optimal heterogeneous ensembles by pooling from homogeneous ensembles is described; Sect. 3, presents a comprehensive empirical evaluation of the proposed methodology and a comparison with the corresponding homogeneous ensembles, individual classifiers and a benchmark of other heterogeneous ensemble. Finally, the conclusions of the present work are summarized.

2 From homogeneous to heterogeneous ensembles

In this study we analyze in a systematic manner the construction of heterogeneous ensembles by pooling individuals from different homogeneous ensembles. This problem is related to the Matrix Cover problem (MC) [25], in which the rows and columns correspond to the decisions made by each base learners $h_i \in T$ for each data point n of a training set $\mathcal{D} = \{\boldsymbol{x}_n, y_n\}_{l=1}^{N_{train}}$, respectively. Specifically, the elements of MC are defined as:

$$MC_{i,l} = \begin{cases} 1, & \text{if } h_i(x_n) = y_n, \\ -1, & \text{otherwise.} \end{cases} \tag{1}$$

where it is assumed that MC satisfies the positive columnn-sum property, that is:

$$\sum_{i=1}^{T} MC_{i,n} > 0 \quad \forall n \in \mathcal{D}, \tag{2}$$

which corresponds to the scenario in which for every training instance, the number of base classifiers with correct decisions is greater than the wrong ones. That is, the ensemble achieves perfect classification on the training set, which is often the case for ensembles [13]. Assuming that minimizing training error minimizes the generalization error, the idea is to find the smallest subset of rows from the matrix cover MC, so that the positive column-sum property holds. It can be seen that this problem reduces to the Set Cover NP-complete problem and that its approximation is intractable [25].

In this study we propose an alternative heuristic to solve this problem as the cost of selecting individual base classifiers is intractable with a cost of $\binom{M \times T}{T}$. We pose the problem of finding optimum heterogeneous ensemble out of homogeneous ones as the problem of choosing the right percentage of the base classifiers from each type. The heterogeneous ensemble of size T is created by pooling $(t_1, t_2, \dots, t_M)$ classifiers from the M ensembles, where t_j is the number of base classifiers pooled from the j^{th} homogeneous ensemble and $\sum_{j=1}^{M} t_j = T$. We are assuming that, on average, base learners of a given type are equivalent and in consequence we focus on selecting the right distribution of base classifiers rather than selecting specific base classifiers as in ref. [25]. This reduces the search space to $\binom{T+M-1}{M-1}$ different heterogeneous ensembles that can be built in this manner. The problem remains intractable on the number of types of ensembles, M. The optimum percentage of each type of base classifier can be obtained by cross-validation or out-of-bag error in a grid search in the space given by $(t_1, t_2, \dots, t_M)$. In any case, the search space can be rather large even for small values of M and T. For instance, for $M = 3$ and $T = 101$, 5253 different heterogeneous can be built. In order to additionally reduce the search space, the ensembles can be evaluated using intervals of i base classifiers of each type. For instance for $M = 3$, the followings configurations of the generated ensembles could be tested:

$$(0,0,T), (0,i,T-i), (i,0,T-i), (0,2*i,T \\ -(2*i)), (2*i,0,T-(2*i)), \cdots .$$

This reduces the search space to $\binom{T/i+M-1}{M-1}$ possible ensemble configurations. Finally, the ensemble composition with minimum validation error is determined as the optimal ensemble. In the case that more than one ensemble configuration has the same minimum validation error, the average ensemble compositions for all minima with the same validation error is selected as the optimal heterogeneous ensemble. For instance, if there are three ensemble compositions, $(a, b, c), (a', b', c'), (a'', b'', c'')$, with the same validation minimum, then the final distribution would be $((a + a' + a'')/3, (b + b' + b'')/3, (c + c' + c'')/3)$. The pseudocode of the method is detailed in Algorithm 1.

For this study, we have used three homogeneous ensembles: random forests (RF), ensembles of support vector machines

(SVMs) and of multilayer perceptrons (MLPs). All base classifiers of these ensembles are created using random samples from the training set to allow for a fast validation of the optimum heterogeneous ensemble by means of out-of-bag [26]. In order to generate ensembles of SVMs the following randomized procedure is used. First, B sets of partially optimized parameters for the SVMs, $\mathbf{\Theta}_b$ with $b = 1, \ldots, B$, are obtained. More details on how these sets of partially optimized parameters are obtained are given below. Then, the ensemble is built in B batches of T/B SVMs. Each batch uses a different set of parameters $\mathbf{\Theta}_b$ and each individual SVMs is trained on a different random bootstrap sample without replacement of size 50% (i.e. subbagging) from the original training set. In this way the variability among the SVMs can be increased. Using subbagging has the advantage with respect to using standard bootstrap samples that the base models can be trained faster. This speedup is approximately 4 times, considering the near quadratic training times of SVMs. In addition, the performance of both sampling strategies, bootstrapping and subbagging, has been demonstrated to be equivalent [27, 28]. To obtain the B sets of partially optimized parameters, we first define a parameter grid. Next, a subbagging sample is generated. One SVMs is trained for each combination of parameters and validated on the left-out set. Finally, the set of parameter with lower error is kept for building the ensemble. This process is repeated B times to obtain the $\mathbf{\Theta}_b$ with $b = 1, \ldots, B$ sets of parameters. The same procedure is used to generate the ensembles of MLPs. The training time complexity of the ensemble depends on the size of the parameter grid, B, on the sampling rate and on the complexity of the base classifier. In spite of creating an ensemble of SVMs (or MLPs), this procedure can be faster to train than training a single SVM by grid search and cross-validation, which is the most common way of training an SVM [29, 30]. This procedure for building ensembles can be over 2 times faster than training a single carefully tuned model SVM [31]. In the next section we will show the validity of this procedure to generate homogeneous ensembles of SVMs and MLPs, and also of the procedure to obtain heterogeneous ensembles from them.

Algorithm 1: Construction of heterogeneous ensemble by pooling from homogeneous ensembles

```
Input: $\mathcal{D}_{train} = \{(\mathbf{x}_n, y_n)\}_{n=1}^{N_{train}}$  % Training set
       $\mathcal{D}_{test} = \{(\mathbf{x}_m, y_m)\}_{m=1}^{N_{test}}$  % Test set
       $build\_homogeneous\_ensemble$   % obtain the votes of individuals in a
       homogeneous ensemble
       $T$                              % Ensemble size
       $M = 3$            % number of homogeneou ensemble type (
       [$"SVM", "MLP", "RF"$] in this study)
Output: $SIM$              % build hetergenous ensemble on test data, using the
        optimal configuration obtained on training set
for $type = 1 : M$ do
    $H_{type} \leftarrow build\_homogeneous\_ensemble(type, \mathcal{D}_{train}, T)$
$ss \leftarrow search\_space(T, M, i, )$ %generate $\binom{T/i+M-1}{M-1}$ possible combinations of $M$
  type of learners to form heterogeneous ensembles of size $T$ with intervals $i$ that is
  $(0,0,T)$, $(0,i,T-i)$, $(i,0,T-i)$, $(0,2*i,T-(2*i))$, $(2*i,0,T-(2*i))$, ...
$min\_error \leftarrow \infty$ % determine optimal heterogeneous configuration
foreach $conf$ in $ss$ do
    $error \leftarrow compute\_oob\_error(\mathcal{D}_{train}, conf, H_{1:M})$
    if $error < min\_error$ then
        $min\_error \leftarrow error$
        $opt\_conf \leftarrow conf$
$SIM = opt\_hetergenous\_ensemble(\mathcal{D}_{test}, opt\_conf, H_{1:M})$
```

3 Experimental results

In this section we present the empirical analysis of heterogeneous ensembles as the combination of homogeneous base classifiers. Furthermore, we validate the procedure to obtain SVM and MLP ensembles by partial optimization of their training parameters. We carried out the analysis on 21 datasets from the UCI repository [32]. In all tested datasets, except of the synthetic problems, the training and test sets were generated using random stratified sampling with sizes 2/3 and 1/3 of the original sets respectively. In the synthetic classification problems, which are *Ringnorm*, *Threenorm* and *Twonorm*, 300 examples are sampled at random for training and 2000 for testing using independent realizations. The results reported are averages over 100 executions.

Primarily, we trained $M = 3$ homogeneous ensembles of size $T = 1001$. Specifically, the ensembles used are: standard random forest [11], partially optimized ensemble of support vector machines [33] and of multi layer perceptrons [34]. We have used e1071, RSNNS and randomForest R packages for creating SVMs, MLPs and RF respectively. Under these setting the possible configurations of the heterogeneous ensemble are $1003 \times 1002/2$. To reduce the computational burden to identify the optimum combination of base classifiers, we evaluated the heterogeneous ensembles in intervals of $i = 13$ base learners, which reduces the optimization to $78 \times 77/2$ evaluations. Given that all three ensembles were generated using random subsamples from the training set to train each base classifier, the optimum heterogeneous configuration is obtained by out-of-bag validation. The values of the hyperparameters for SVM with a RBF kernel are selected from a grid with $C = 2^q$ with $q = -5, \ldots, 15$ and $\boldsymbol{\gamma} = 2^p$ with $p = -15, \ldots, 3$. For MLP, the number of neurons in the hidden layer was optimized from the values $\{3, 4, 5, 6, 7, 8, 9, 10\}$. For building the partially optimized ensemble, $B = 10$ sets of hyperparameter were obtained using out-of-bag. For random forest, the default parameters were used.

3.1 Homogeneous ensemble of SVMs and MLPs

In order to validate the procedure to generate the partially optimized ensembles, a comprehensive comparison with respect to an optimized single base learner was carried out. For this purpose, a single SVM and a single MLP were trained using within-train 10-fold cross-validation and grid search over the same sets of parameters given above. The average errors for this experiments are shown in Table 1 for a single SVM and MLP, and for the homogeneous ensembles composed of SVMs (shown as E-SVM in the table) and of MLPs (shown as E-MLP). In addition, an overall comparison of the methods is shown in Fig. 1 by mean of the procedure proposed by Demšar in ref. [35]. In this diagram, the average ranks for each method are shown. Methods connected by a horizontal solid line indicate that their differences in average rank are not statistically significant according to a Nemenyi test (p-value < 0.05).

From Table 1, it can be observed that the ensemble of MLPs clearly outperforms the single MLP. The ensemble of MLPs outperforms a single MLP in all tested datasets except for *Ionosphere* and *Parkinsons*. The differences between the single SVM and its ensemble counterpart are not so pronounced as the ones observed for MLPs. The ensemble of SVM obtains a better result than a single SVM in 11 out of 19 datasets. This same result can be observed in Fig. 1 where the average rank of E-SVM is slightly better than a single SVM. However, the difference is not statistically significant. Even thought the differences are not statistically significant, this analysis shows that this procedure to build ensembles of SVMs is not detrimental. When using MLP as base classifiers, we observe that the differences are statistically significant with respect to a single MLP. In addition, with these setting, we have observed that the training time for E-SVM is over 2 times faster than training a single SVM using grid search and 10-fold cross-validation. For ensembles of MLP, the speedup is over 1.5 with respect to the single MLP.

Table 1 Test errors for a single optimized SVM and MLP, also their homogeneous ensembles as it is proposed in Sect. 3.1

Dataset	SVM	E-SVM	MLP	E-MLP
Australian	14.4 ± 2.3	13.7 ± 2.1	15.6 ± 2.2	14.2 ± 1.9
Boston	12.2 ± 2.4	12.2 ± 2.3	12.7 ± 2.1	12.3 ± 2.0
Breast	3.5 ± 1.1	3.4 ± 1.1	9.1 ± 12.1	3.2 ± 1.1
Bupa	29.1 ± 3.7	27.9 ± 3.4	30.3 ± 4.0	28.3 ± 3.7
Chess	0.8 ± 0.4	0.8 ± 0.3	1.0 ± 0.2	0.9 ± 0.3
Colic	31.8 ± 3.4	33.2 ± 1.4	32.4 ± 3.4	31.3 ± 3.3
German	25.1 ± 1.8	24.6 ± 1.6	28.0 ± 2.0	24.7 ± 1.9
Heart	16.0 ± 3.5	15.4 ± 3.0	18.2 ± 3.7	16.3 ± 3.1
Hepatitis	16.6 ± 3.6	15.8 ± 3.0	17.5 ± 4.3	15.4 ± 4.3
Ionosphere	6.3 ± 1.8	5.7 ± 1.7	10.6 ± 2.9	11.3 ± 2.5
Ozone	5.6 ± 0.4	5.6 ± 0.3	6.8 ± 0.5	5.5 ± 0.5
Parkinsons	8.7 ± 4.1	10.7 ± 3.7	11.3 ± 4.1	13.7 ± 3.9
Pima	23.1 ± 2.0	22.7 ± 1.8	24.7 ± 2.4	23.1 ± 2.1
Ringnorm	1.7 ± 0.6	1.6 ± 0.4	17.0 ± 1.5	16.4 ± 1.5
Spambase	6.4 ± 0.4	6.6 ± 0.4	7.0 ± 0.4	5.9 ± 0.4
Sonar	15.0 ± 4.3	17.8 ± 4.9	21.0 ± 4.2	20.7 ± 4.6
Threenorm	14.5 ± 1.3	14.1 ± 0.7	17.7 ± 2.0	16.9 ± 0.9
Tictactoe	1.0 ± 1.3	1.8 ± 0.7	4.4 ± 7.4	1.8 ± 0.7
Twonorm	2.6 ± 0.5	2.4 ± 0.3	2.4 ± 0.9	2.9 ± 0.4
Magic	14.2 ± 1.1	14.5 ± 0.9	16.0 ± 1.6	15.2 ± 0.8
Adult	14.6 ± 1.5	13.7 ± 1.3	14.8 ± 1.8	13.1 ± 1.1

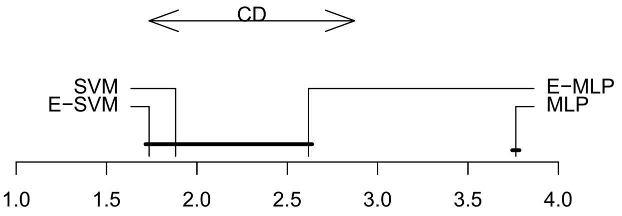

Fig. 1 Average ranks for SVM, E-SVM, MLP and E-MLP (more details in the text)

3.2 Heterogeneous ensemble pooled from homogeneous ensembles

In this section the performance of the proposed procedure to built heterogeneous ensembles by pooling from homogeneous ensembles is analyzed. The objective is to find the optimum proportion of each of the possible base classifiers to build the final heterogeneous ensemble. Each of the possible selected proportions, which correspond to a different heterogeneous ensemble, can be represented by a point in a regular simplex in M dimensions. This is shown in Fig. 2 for three representative datasets: *Heart*, *Colic* and *Tic-tac-toe*. Each plot in Fig. 2 shows in a 3 dimensional simplex, the average test error for the different combinations of base classifiers in intervals of $i = 13$ classifiers using a grey scale scheme. Darker colors indicate higher average error as indicated by the color legend at the right of each plot. The three vertices in the plots correspond to the three tested homogeneous ensembles. The vertices in the upper left, right and bottom left of the plot correspond to E-SVM, E-MLP and random forest respectively. A displacement away from one of these vertices smoothly transforms the corresponding homogeneous ensembles into a heterogeneous one. The horizontal axis shows the number of selected MLPs in the heterogeneous ensemble, while the vertical axis indicates the number of SVMs minus the number of random trees. In addition, all plots show the average selected position using out-of-bag validation (marked with a ’o’ sign) and the average position for the best test errors (marked with a ’T’ sign).

In the plots of Fig. 2 different behaviours of the combination of base classifiers can be observed. In *Heart* (left plot), the best position is observed quite centered, showing that a heterogeneous ensemble composed of base classifiers from different types is beneficial to improve the generalization performance of the ensemble. However, this is not a general trend as it can be observed in the center plot (*Colic*). In this case, the best result is clearly located at one of the vertices of the simplex that correspond to a homogeneous ensemble—random forest in this case. Finally, it is important to note that the optimum location need not be close to the best homogeneous ensemble. For instance, in *Tic-tac-toe*, the location of the minimum error is very close to the random forest

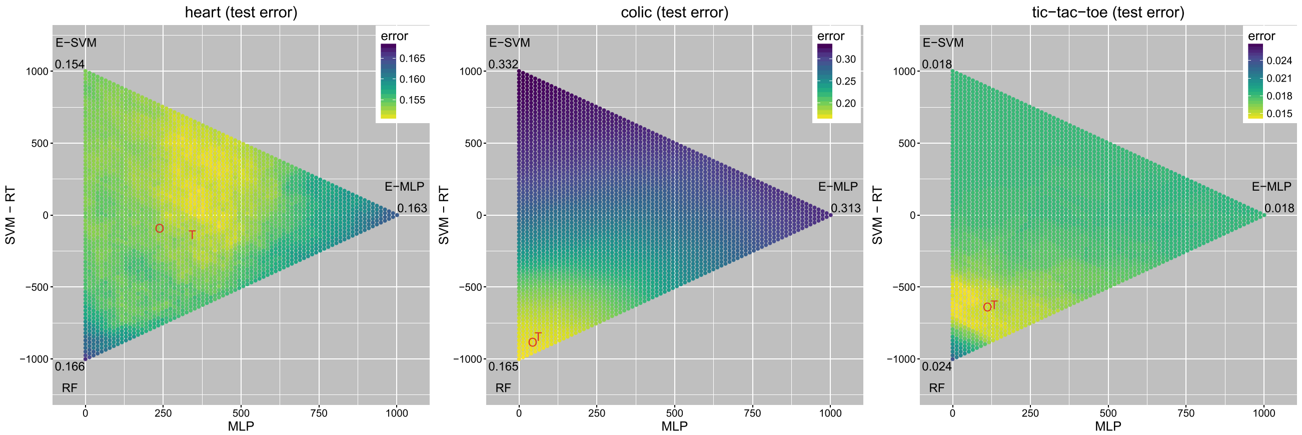

Fig. 2 Test error rate of the heterogenous ensembles in the simplex for different classification problems. Darker colors correspond to higher errors

vertex in spite of the fact that this homogeneous ensemble presents the worst average performance. Finally, we can observe that the average location of the minima identified using out-of-bag is quite close to the location in test. We have also observed, however, that for the smaller datasets the identification of the optimum point is less accurate.

We implemented the Differential Evolution (DE) approach proposed in ref. [24], as a comparison method against the proposed strategy in this study. This method leverages on a specific evolutionary algorithm called differential evolution to optimize 20 heterogeneous base classifiers' voting weights in an ensemble. The procedure starts with a randomly generated population of N vector-valued individuals, where each one corresponds to a particular weighing vote configuration of an ensemble. The randomly initialized population will be evolved through some mutation, crossover, and selection procedures. Namely, for mutation a DE/*rand*/1 procedure is used for which 3 exclusive individuals $\boldsymbol{x}_1$, $\boldsymbol{x}_2$ and $\boldsymbol{x}_3$ are taken at random from the population and the first is perturbed using the weighted difference of the other two individuals $(\boldsymbol{x}_1 + F \cdot (\boldsymbol{x}_2 - \boldsymbol{x}_3))$. In the crossover stage, the resulting member from the mutation stage plays a donor role. A certain percentage of the original elements in $\boldsymbol{x}_1$ will be substituted with the donor elements based on a user-defined crossover probability. The newly built $\boldsymbol{x}_1$ will be replaced with the original one when it has a higher fitness value. The fitness is evaluated using the average Matthews Correlation Coefficient (MCC), calculated over 10-fold cross-validation, as the ensemble's quality measure. The described procedure continues until the stopping condition is satisfied. As a comparison approach to the proposed strategy in this study, instead of using 20 base classifiers mentioned in the original publication [24], we have used the three homogeneous ensembles (E-SVM, E-MLP, and RF). Hence, each vector in the population has three weights that translate into a number of base classifiers that are selected from the mentioned homogeneous ensembles. Moreover, for evaluating a member in the population, instead of using 10-folds cross-validation, we have used the related out-of-bag set.

In the Table 2, the average test errors for the homogeneous ensembles of SVMs (E-SVM) and MLPs (E-MLP), random forest (RF), Differential Evolution (DE) and the proposed strategy (SIM) over the investigated problems are reported. The best and second best results for each dataset are highlighted in boldface and underlined respectively. In addition, the best result is marked with an asterisk (*) if the improvement over to the second best is statistically significant, at a significance level $\alpha = 0.05$. The significance is determined using a paired resampled t-test for synthetic problems, and a corrected resampled paired *t*-test [36] when train/test partitions are randomly taken. In addition, the table shows the average percentage of classifiers of each type selected by out-of-bag validation for the heterogeneous ensembles. The percentages are shown in the same order that the ensembles are shown, that is, % of SVMs, % of MLP and % of random trees.

As shown in Table 2, the proposed method is the best or the second best method for all datasets except in Adult. DE and E-SVM also achieve rather good results but it is somehow less consistent. This results are summarized using a Demšar plot [35] in Fig. 3. From this diagram, it can be observed that the proposed procedure is significantly better than random forest and E-MLP (as given by a Nemenyi test with p-value < 0.05). The proposed methodology has an average rank better than DE and E-SVM but their difference is not statistically significant.

In terms of time complexity, the problem of selecting a set of base classifiers whose combination is best, is an NP problem [25]. In our case, we are selecting T models from a pool

Table 2 Test errors of single classifiers, homogeneous ensembles and optimal heterogeneous ensemble

Dataset	E-SVM	E-MLP	RF	DE	SIM	[% SVM, % MLP, % trees]
Australian	13.7 ± 2.1	14.2 ± 1.9	**13**.0 ± **2**.1*	13.7 ± 2.1	13.5 ± 2.0	[24.5 , 16.7 , 58.8]
Boston	12.2 ± 2.3	12.3 ± 2.0	12.9 ± 2.1	**11**.9 ± **2**.1	12.2 ± 2.0	[39.2 , 23.1 , 37.7]
Breast	3.4 ± 1.1	**3**.2 ± **1**.1	3.3 ± 1.1	3.3 ± 1.1	3.3 ± 1.0	[27.6 , 29.0 , 43.4]
Bupa	27.9 ± 3.4	28.3 ± 3.7	**27**.2 ± **3**.6	27.5 ± 3.4	27.3 ± 3.5	[21.4 , 15.6 , 63.0]
Chess	0.8 ± 0.3	0.9 ± 0.3	1.7 ± 0.4	**0**.8 ± **0**.2	**0**.8 ± **0**.2	[34.7 , 22.7 , 42.6]
Colic	33.2 ± 1.4	31.3 ± 3.3	**16**.5 ± **2**.9*	27.6 ± 3.4	17.2 ± 3.0	[3.7 , 4.4 , 91.9]
German	24.6 ± 1.6	24.7 ± 1.9	**23**.9 ± **1**.8	24.0 ± 1.9	24.3 ± 1.9	[15.9 , 28.3 , 55.7]
Heart	**15**.4 ± **3**.0	16.3 ± 3.1	16.6 ± 2.9	15.5 ± 3.1	15.5 ± 3.1	[33.5 , 23.8 , 42.7]
Hepatitis	15.8 ± 3.0	15.4 ± 4.3	**15**.1 ± **3**.6	15.1 ± 3.6	15.2 ± 3.6	[25.6 , 28.7 , 45.7]
Ionosphere	**5**.7 ± **1**.7	11.3 ± 2.5	6.7 ± 1.7	6.3 ± 2.1	5.8 ± 1.7	[64.4 , 13.7 , 21.9]
Ozone	5.6 ± 0.3	5.5 ± 0.5	5.7 ± 0.3	**5**.4 ± **0**.3	**5**.4 ± **0**.4	[16.7 , 53.6 , 29.7]
Parkinsons	**10**.7 ± **3**.7	13.7 ± 3.9	11.1 ± 4.0	10.8 ± 3.7	**10**.7 ± **3**.9	[44.1 , 12.9 , 43.0]
Pima	**22**.7 ± **1**.8*	23.1 ± 2.1	23.1 ± 2.0	23.0 ± 1.9	22.9 ± 1.8	[44.5 , 18.1 , 37.4]
Ringnorm	**1**.6 ± **0**.4*	16.4 ± 1.5	5.9 ± 1.0	2.6 ± 1.9	1.7 ± 0.5	[62.2 , 11.2 , 26.6]
Spambase	6.6 ± 0.4	5.9 ± 0.4	5.1 ± 0.4	5.8 ± 0.4	**5**.0 ± **0**.3	[12.2 , 11.1 , 76.8]
Sonar	**17**.8 ± **4**.9	20.7 ± 4.6	18.9 ± 4.8	18.5 ± 4.6	18.0 ± 4.4	[39.1 , 16.9 , 44.0]
Threenorm	**14**.1 ± **0**.7*	16.9 ± 0.9	16.7 ± 1.0	14.7 ± 1.0	14.4 ± 0.8	[52.1 , 10.8 , 37.1]
Tictactoe	1.8 ± 0.7	1.8 ± 0.7	2.4 ± 1.1	1.8 ± 0.6	**1**.5 ± **0**.7*	[12.5 , 11.2 , 76.3]
Twonorm	**2**.4 ± **0**.3*	2.9 ± 0.4	3.9 ± 0.5	2.6 ± 0.3	2.5 ± 0.4	[42.5 , 22.7 , 34.8]
Magic	14.5 ± 0.9	15.2 ± 0.8	**13**.5 ± **1**.0	13.8 ± 0.4	13.6 ± 0.5	[25.2 , 23.8 , 50.9]
Adult	13.7 ± 1.3	13.1 ± 1.1	**12**.7 ± **1**.6	13.3 ± 1.2	13.4 ± 0.9	[32.9 , 28.2 , 38.8]

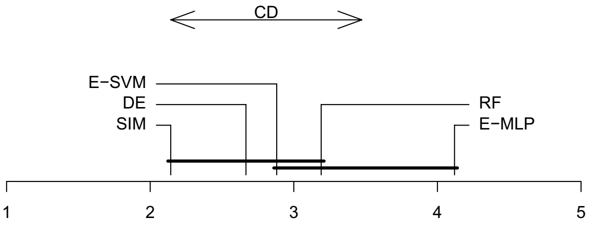

Fig. 3 Average ranks for E-SVM, E-MLP, RF, the optimal estimated heterogeneous ensemble (SIM) and Differential Evolution (DE)

of $M \times T$ heterogeneous models. This problem involves $\binom{MT}{T}$ possible subsets. Our assumption is that, on average, individual models of the same type are equivalent on average and the proposed method focuses on selecting the right distribution of base classifiers rather than selecting specific base classifiers. Hence, our approach to find the optimal heterogeneous ensemble searches in a space of size $\binom{T/i+M-1}{M-1}$, which is polynomic in M. In the studied setup, we have considered $T = 1001$, size of the ensemble, $M = 3$, learners types and $i = 13$ for the grid interval. In total, for each out-of-bag set we built 3081 heterogeneous ensembles. On the other hand, for DE approach, within each generation of size 100 (random configurations of different heterogeneous ensembles), we performed mutation, cross-over and fitness evaluation for each member. We repeat this process 100 times for the evolution of the generations. In total, 10000 heterogeneous ensembles are built using this approach. The proposed approach, despite computing three times less ensemble evaluations than DE, obtains favorable results.

4 Conclusions

In this study, we propose a systematic method for creating heterogeneous ensembles optimizing the propositions of base classifiers of different types. To this end, we first generate M different homogeneous ensembles. Diversification in these ensembles is obtained by using both subsampling and randomization techniques. Then a heterogenous ensemble is built by pooling classifiers from these homogeneous ensembles. The proportions of classifiers of different types in the heterogeneous combination can be represented with a point in a simplex in M dimensions. Each of the M vertices in this simplex corresponds to one of the homogeneous ensembles. We have observed that the optimal proportion of base classifiers in the final ensemble is strongly problem-dependent.

A comprehensive empirical evaluation is carried out to compare the proposed creation strategy with respect to homogeneous ensembles and other state-of-the-art technique for creating heterogeneous ensembles. This analysis shows that the proposed strategy exhibits excellent performance. In all the problems investigated except one, it is either the first or second most accurate method. The results show that the proposed combination is better than the implemented benchmark heterogeneous ensemble (DE) and also any of the homogeneous ensembles; i.e. random forest, ensembles of MLPs and ensembles of SVMs.

Acknowledgements The authors acknowledge financial support from PID2019-106827GB-I00/AEI/10.13039/501100011033.

Funding Open access funding provided by Aalto University.

References

1. Breiman L, Friedman J, Stone CJ, Olshen RA (1984) Classification and regression trees. CRC Press, Boca Raton
2. Anthony M, Bartlett PL (2009) Neural network learning: theoretical foundations. Cambridge University Press, Cambridge
3. Cristianini N, Shawe-Taylor J (2000) An introduction to support vector machines and other kernel-based learning methods. Cambridge University Press, Cambridge
4. Dietterich TG (2000) Ensemble methods in machine learning. International workshop on multiple classifier systems. Springer, Berlin, pp 1–15
5. Banfield RE, Hall LO, Bowyer KW, Kegelmeyer WP (2007) A comparison of decision tree ensemble creation techniques. IEEE Trans Pattern Anal Mach Intell 29(1):173–180
6. Bauer E, Kohavi R (1999) An empirical comparison of voting classification algorithms: bagging, boosting, and variants. Mach Learn 36(1):105–139
7. Hoch T (2015) An ensemble learning approach for the kaggle taxi travel time prediction challenge. In: DC@ PKDD/ECML
8. Zikeba M, Tomczak SK, Tomczak JM (2016) Ensemble boosted trees with synthetic features generation in application to bankruptcy prediction. Expert Syst Appl 58:93–101
9. Hansen LK, Salamon P (1990) Neural network ensembles. IEEE Trans Pattern Anal Mach Intell 12(10):993–1001
10. Breiman L (1996a) Bagging predictors. Mach Learn 24(2):123–140
11. Breiman L (2001) Random forests. Mach Learn 45(1):5–32
12. Martínez-Muñoz G, Suárez A (2005) Switching class labels to generate classification ensembles. Pattern Recognit 38:1483–1494
13. Freund Y, Schapire R (1999) A short introduction to boosting. J Jpn Soci Artif Intell 14(771–780):1612
14. Lu Z, Wu X, Bongard JC (2015) Active learning through adaptive heterogeneous ensembling. IEEE Trans Knowl Data Eng 27(2):368–381
15. de Oliveira JM, dos Santos EM, Carvalho JRH, de Vasconcelos Marques LA (2013) Ensemble of heterogeneous classifiers applied to lithofacies classification using logs from different wells. International joint conference on neural networks. IJCNN, Dallas, TX, USA, pp 1–6
16. Nanni L, Brahnam S, Ghidoni S, Lumini A (2015) Toward a general-purpose heterogeneous ensemble for pattern classification. Comput Intell Neurosci 2015:85
17. Caruana R, Niculescu-Mizil A, Crew G, Ksikes A (2004) Ensemble selection from libraries of models, In: Proceedings of the twenty-first international conference on machine learning, vol ICML 04. New York, NY, USA, pp 18
18. Partalas I, Tsoumakas G, Vlahavas I (2010) An ensemble uncertainty aware measure for directed hill climbing ensemble pruning. Mach Learn 81(3):257–282
19. Haque MN, Noman N, Berretta R, Moscato P (2016b) Heterogeneous ensemble combination search using genetic algorithm for class imbalanced data classification. PloS One 11(1):e0146116
20. Tsoumakas G, Katakis I, Vlahavas I (2004) Effective voting of heterogeneous classifiers. European conference on machine learning. Springer, Berlin, pp 465–476
21. Tsoumakas G, Partalas I, Vlahavas I (2009) An ensemble pruning primer. Applications of supervised and unsupervised ensemble methods. Springer, Berlin, pp 1–13
22. Martınez-Munoz G, Hernández-Lobato D, Suárez A (2009) An analysis of ensemble pruning techniques based on ordered aggregation. IEEE Trans Pattern Anal Mach Intell 31(2):245–259
23. Alshdaifat E, Al-hassan M, Aloqaily A (2020) Effective heterogeneous ensemble classification: an alternative approach for selecting base classifiers. ICT Express, South Korea
24. Haque MN, Noman MN, Berretta R, Moscato P (2016a) Optimising weights for heterogeneous ensemble of classifiers with differential evolution. In: 2016 IEEE congress on evolutionary computation (CEC), pp 233–240
25. Tamon C, Xiang J (2000) On the boosting pruning problem. European conference on machine learning. Springer, Berlin, pp 404–412
26. Breiman L (1996b) Out-of-bag estimation. Statistics Department, University of California, Tech. rep
27. Friedman JH, Hall P (2007) On bagging and nonlinear estimation. J Stat Plan Inference 137(3):669–683
28. Martínez-Muñoz G, Suárez A (2010) Out-of-bag estimation of the optimal sample size in bagging. Pattern Recognit 43(1):143–152
29. Ben-Hur A, Weston J (2010) A user's guide to support vector machines. Data mining techniques for the life sciences. Springer, Berlin, pp 223–239
30. Hsu CW, Chang CC, Lin CJ (2003) A practical guide to support vector classification. Department of Computer Science, National Taiwan University, Tech. rep
31. Sabzevari M, Martínez-Muñoz G, Suárez A (2018) Randomization vs optimization in svm ensembles. In: International conference on artificial neural networks, pp 415–421
32. Bache K, Lichman M (2013) Uci machine learning repository
33. Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20(3):273–297
34. Haykin S (1999) Neural networks: a comprehensive foundation. Prentice Hall, Hoboken
35. Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7:1–30
36. Bouckaert RR, Frank E (2004) Evaluating the replicability of significance tests for comparing learning algorithms. In: Pacific-Asia conference on knowledge discovery and data mining, pp 3–12