Evolving an optimal decision template for combining classifiers.

NGUYEN, T.T., LUONG, A.V., DANG, M.T., DAO, L.P., NGUYEN, T.T.T., LIEW, A.W.-C. and MCCALL, J.

2019

Evolving an Optimal Decision Template for Combining Classifiers

Tien Thanh Nguyen[1], Anh Vu Luong[2], Manh Truong Dang[1], Lan Phuong Dao[3], Thi Thu Thuy Nguyen[2], Alan Wee-Chung Liew[2], and John McCall[1]

[1] School of Computing Science and Digital Media, Robert Gordon University, Aberdeen, United Kingdom {t.nguyen11,t.dang1,j.mccall}@rgu.ac.uk
[2] School of Information and Communication Technology, Griffith University, Gold Coast, Australia {vu.luong, thithuthuy.nguyen}@griffithuni.edu.au a.liew@griffith.edu.au
[3] AN Company, Hanoi, Vietnam lan.pd@outlook.com

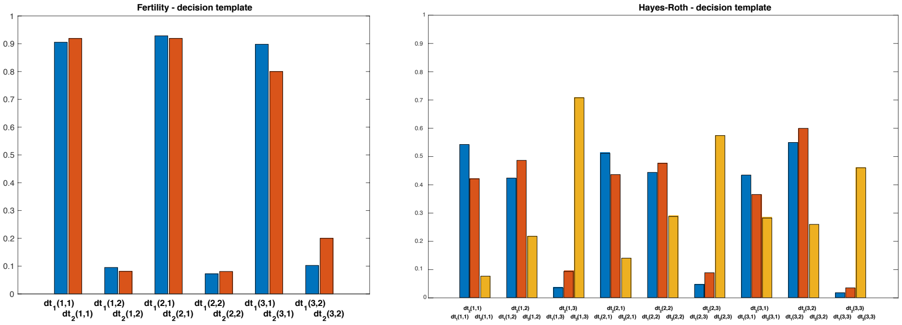
Abstract. In this paper, we aim to develop an effective combining algorithm for ensemble learning systems. The Decision Template method, one of the most popular combining algorithms for ensemble systems, does not perform well when working on certain datasets like those having imbalanced data. Moreover, point estimation by computing the average value on the outputs of base classifiers in the Decision Template method is sometimes not a good representation, especially for skewed datasets. Here we propose to search for an optimal decision template in the combining algorithm for a heterogeneous ensemble. To do this, we first generate the base classifier by training the pre-selected learning algorithms on the given training set. The meta-data of the training set is then generated via cross validation. Using the Artificial Bee Colony algorithm, we search for the optimal template that minimizes the empirical 0-1 loss function on the training set. The class label is assigned to the unlabeled sample based on the maximum of the similarity between the optimal decision template and the sample's meta-data. Experiments conducted on the UCI datasets demonstrated the superiority of the proposed method over several benchmark algorithms.

Keywords: Ensemble method · Combining classifiers · Multiple classifiers · Classifier fusion · Artificial Bee Colony.

1 Introduction

In recent years, there has been an intense research activity focusing on ensemble learning [13, 11]. The interest emerges from the fact that ensemble methods can achieve higher performance than using single learners in many learning tasks such as supervised (i.e. classification and prediction) and unsupervised learning (i.e. clustering). Until now, ensemble methods have been applied to many areas such as bioinformatics, computer vision, and software engineering [18].

In this paper, we focus on the heterogeneous ensemble in which several learning algorithms train base classifiers on a given training set. A combining algorithm is then used to aggregate the output of these base classifiers to obtain the



* $dt_j(k,m)$ is defined in Section 2.2

Fig. 1. Decision templates for class labels computed on Fertility and Hayes-Roth datasets

final prediction. The research here is to develop new combiners to obtain high accuracy [13, 11, 14, 12].

Among the combining algorithms developed for heterogeneous ensemble systems, Decision Template is one of the most popular methods [11]. In this method, we group the outputs of base classifiers on the training observations (called meta-data) based on their class labels. The decision template for each class label is computed as the mean i.e. average of the meta-data of the observations in the associated group. It is noted that Decision Template method may perform poorly if it does not provide a good enough representation for a class. When data distribution is skewed for example, the mean loses its ability to provide the best central location. Moreover, the base classifiers will tend to predict the dominant class on some imbalanced datasets. Fig. 1 shows the decision templates computed on the outputs of an example of ensemble with 3 base classifiers, i.e. Linear Discriminative Analysis (denoted by LDA), Naïve Bayes, and k Nearest Neighbor (k set to 5, denoted by KNN_5), on the two imbalanced datasets: 2-class Fertility and 3-class Hayes-Roth. In Fertility dataset, 80% of the observations belongs to the first class label. Clearly, the Decision Templates of the 2 classes are very similar and consequently have low discriminative ability. On Hayes-Roth dataset, the decision templates of the first two classes have the same values. This makes Decision Template method poor on these datasets.

In this paper, we aim to search for the optimal decision template for a heterogeneous ensemble. To do this, we first generate the base classifiers by training the pre-selected learning algorithms on the given training set. The meta-data of the training set is then generated via a cross validation procedure. By using the Artificial Bee Colony algorithm [4, 5], we search for the optimal decision template which minimizes the empirical 0-1 loss on the training set. In detail, for each candidate template, we measure the similarity between it and the meta-data of each training observations. The class label is assigned to the observations based on the maximum similarity. The optimal decision template is the candidate that minimizes the loss function. During classification, the class label that maximizes the similarity between the meta-data of the unlabeled sample and the optimal decision template of a class is returned. Experiment conducted on the 31 datasets demonstrated that the proposed method is better than the benchmark algorithms we compared.

2 BACKGROUND

2.1 Heterogeneous ensemble method

Let $\{y_m\}_{m=1,\ldots,M}$ denotes the set of M labels, N denotes the number of training observations, and K denotes the number of learning algorithms. For an observation $\mathbf{x}$, $P_k(y_m|\mathbf{x})$ is the probability that $\mathbf{x}$ belongs to the class with label y_m given by the k^{th} classifier. In this study, we focus on the soft label output: $P_k(y_m|\mathbf{x}) \in [0,1]$ and $\sum_m P_k(y_m|\mathbf{x}) = 1$. In heterogeneous ensemble learning, the soft labels output by the base classifiers for the training set become the *meta-data* of the training set, which is given by the $N \times KM$ matrix :

$$\mathbf{L} = \begin{bmatrix} P_1(y_1|\mathbf{x}_1) \ldots & P_K(y_1|\mathbf{x}_1) \ldots & P_K(y_M|\mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ P_1(y_1|\mathbf{x}_N) \ldots & P_K(y_1|\mathbf{x}_N) \ldots & P_K(y_M|\mathbf{x}_N) \end{bmatrix} \quad (1)$$

Meanwhile, the meta-data of an observation $\mathbf{x}_i$ is given by:

$$\mathbf{L}(\mathbf{x}_i) = \begin{bmatrix} P_1(y_1|\mathbf{x}_i) \ldots & P_1(y_M|\mathbf{x}_i) \\ \ldots & \\ P_K(y_1|\mathbf{x}_i) \ldots & P_K(y_M|\mathbf{x}_i) \end{bmatrix} \quad (2)$$

There are two combining approaches for the heterogeneous ensemble: fixed combining method and trainable combining method [13]. For fixed combining method, the combiner works directly on the meta-data of a test sample to assign the class label and does not exploit the meta-data of the training set. It therefore has fast training time. There are some popular fixed combining methods such as fixed combining rules [14, 6] and fixed combining based on Ordered Weighted Averaging operator (OWA) [7]. Among these methods, the Sum Rule is the most popular [6].

For the trainable combining method, two well-known strategies to obtain the discriminative decision model are weighted classifiers combining methods and methods based on meta-data representation. In weighted classifiers combining methods, each classifier is assumed to differently contribute to the combining result i.e. putting a different weight on each class. The combining algorithm is based on the M linear combinations of posterior probabilities and the associated weights for the M classes. There are several approaches in this category such as the Multi-Response Linear Regression (MLR) method [17] and MLR with hinge loss [15]. On the other hand, the meta-data representation approach aims to find the representation for the meta-data associated with each class label. The class label is assigned to a test sample based on the similarity between its meta-data and the representation. Some examples of meta-data representation methods are the Decision Template method [8], the Bayesian-based method [13], and granular-based prototype [12].

Heuristic search based approaches have also been proposed to enhance the heterogeneous ensemble. These approaches aim to search for the optimal subset of base classifiers, of meta-classifier, the input features, and the meta-data features, to boost the performance of the ensemble system. In detail, Nguyen et al. [10] encoded the base classifiers and the features in a single chromosome and used Genetic Algorithm to simultaneously search for the optimal set of classifiers and associated features. Nguyen et al. [9] also proposed a new encoding for meta-data feature and used Genetic Algorithm to search for the optimal set of meta-data features for the Decision Tree meta-classifiers. Shunmugapriya and Kanmani [16] used Artificial Bee Colony (ABC) to find the optimal set of base classifiers and the meta-classifiers. Chen et al. [2] used Ant Colony Optimization (ACO) to find the optimal set of base classifiers in an ensemble system with the Decision Tree as the meta-classifier.

2.2 Decision Template method

In this section, we briefly introduce the Decision Template method [8] which is the basis for our approach. In this method, after obtaining the base classifiers by learning the learning algorithms on the training set, the meta-data $\mathbf{L}$ is obtained via a cross validation procedure. The decision template $\boldsymbol{\mathcal{DT}} = \{\mathbf{DT}_j\}$ where $\mathbf{DT}_j$ is the decision template for j^{th} class, computed on the meta-data is given by:

$$\mathbf{DT}_j = \begin{bmatrix} dt_j(1,1) & \dots & dt_j(1,M) \\ \dots & \ddots & \dots \\ dt_j(K,1) & \dots & dt_j(K,M) \end{bmatrix} \tag{3}$$

where each element is computed by:

$$dt_j(k,m) = \frac{\sum_{i=1}^{N} \mathbb{I}[y_j = \hat{y}_i] P_k(y_m|\mathbf{x}_i)}{\sum_{i=1}^{N} \mathbb{I}[y_j = \hat{y}_i]} \tag{4}$$

for $k = 1, ..., K;\ m = 1, ..., M;\ j = 1, ..., M$
in which $\hat{y}_i$ is the true class label of $\mathbf{x}_i$, $\mathbb{I}[\cdot]$ is the indicator function which returns 1 if the condition is true and 0 otherwise. In (4), the $dt_j(k,m)$ is the average value of the meta-data of the observations belonging to the j^{th} class (the condition $y_j = \hat{y}_i$ is true for observations that belong to class y_j) associated with the k^{th} classifier and class label y_m.

In the classification stage, the distance between the meta-data of a test sample $\mathbf{x}$ and $\mathbf{DT}_j (j = 1, ..., M)$ are computed. The class label is assigned to $\mathbf{x}$ based on the maximum similarity or the minimum dissimilarity between $\mathbf{L}(\mathbf{x})$ and $\mathbf{DT}_j$.

As mentioned in Section 1, the Decision Template method has some limitations when modelling skewed data or working with imbalance datasets. We address these disadvantages in our proposed method.

3 PROPOSED METHOD

3.1 Problem formulation

In this study, we focus on searching for the optimal decision template on the meta-data of the training observations for the combining classifiers. For an observation $\mathbf{x}$ and an arbitrary decision template $\boldsymbol{\mathcal{DT}} = \{\mathbf{DT}_j\}$, we compute the similarity between its meta-data $\mathbf{L}(\mathbf{x})$ and $\mathbf{DT}_j$ as:

$$S(\mathbf{L}(\mathbf{x}), \mathbf{DT}_j) = \frac{C\{\mathbf{L}(\mathbf{x}) \cap \mathbf{DT}_j\}}{C\{\mathbf{L}(\mathbf{x}) \cup \mathbf{DT}_j\}} \tag{5}$$

where the relative cardinality $C\{\cdot\}$ is given by:

$$C\{\mathbf{L}(\mathbf{x}) \cap \mathbf{DT}_j\} = \frac{1}{MK} \sum_{k=1}^{K} \sum_{m=1}^{M} \min\Big(P_k(y_m|\mathbf{x}), dt_j(k,m)\Big) \tag{6}$$

and

$$C\{\mathbf{L}(\mathbf{x}) \cup \mathbf{DT}_j\} = \frac{1}{MK} \sum_{k=1}^{K} \sum_{m=1}^{M} \max\Big(P_k(y_m|\mathbf{x}), dt_j(k,m)\Big) \tag{7}$$

The class label is assigned to $\mathbf{x}$ by selecting the one that has the maximum similarity among the M decision templates:

$$\mathbf{x} \in y_t \ \textit{if} \ y_t = \text{argmax}_{y_j, j=1,\ldots,m} S\{\mathbf{L}(\mathbf{x}), \mathbf{DT}_j\} \tag{8}$$

The empirical loss function $\mathcal{L}_{0-1}$ computed on the training set is given by:

$$\mathcal{L}_{0-1}(\boldsymbol{\mathcal{DT}}) = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}\Big[\underset{y_j, j=1,\ldots,M}{\text{argmax}} S\{\mathbf{L}(\mathbf{x}_i), \mathbf{DT}_j\} \neq \hat{y}_i \Big] \tag{9}$$

where $\mathbf{x}_i$ is the training observation with true label $\hat{y}_i$.
We can simply show that $0 \leq dt_j(k,m) \leq 1$. It is straightforward that: $0 \leq P_k(y_m|\mathbf{x}_i) \leq 1$.. For each $\mathbf{x}_i \in \mathcal{D}$, we have: $0 \leq \mathbb{I}[y_j = \hat{y}_i] P_k(y_m|\mathbf{x}_i) \leq \mathbb{I}[y_j = \hat{y}_i]$. Hence:

$$0 \leq \sum_{i=1}^{N} \mathbb{I}[y_i = \hat{y}_i] P_k(y_m|\mathbf{x}_i) \leq \sum_{i=1}^{N} \mathbb{I}[y_j = \hat{y}_i]$$

Therefore:

$$0 \leq dt_j(k,m) \leq 1 \qquad \square \tag{10}$$

To find the optimal decision template for the ensemble, we minimize the loss function (9) subject to the constraints (10).

3.2 The algorithm

The training phase of the proposed method is given in Algorithm 1 and 2. We first generate the base classifiers $\{BC_k\}$ by learning K learning algorithms on the training set $\mathcal{D}$. The meta-data from the training set is then obtained by the cross validation procedure in the form of matrix $\mathbf{L}$ (1) [13]. Meanwhile, the meta-data of $\mathbf{x}_i \in \mathcal{D}, \mathbf{L}(\mathbf{x}_i)$ is obtained from $\mathbf{L}$ in form of (2).

In this study, we applied the Artificial Bee Colony (ABC) algorithm [4] to find the optimal decision template $\boldsymbol{\mathcal{ODT}}$ that minimize $\mathcal{L}_{0-1}$ on the training set $\mathcal{D}$. The ABC algorithm, proposed by Karaboga [4], is a meta-heuristic search algorithm inspired by the intelligent foraging behavior of honey bee swarms. This algorithm provides a simple but powerful tool to search for the optimal solution with fewer control parameters [16]. In ABC, there are three types of bees in the swarm: employed bee, onlooker bee, and scout. The number of employed bee and onlooker bee is equal to the number of solution in the swarm (denoted by $nPop$). Employed bees exploit the food sources and share the information of nectar amount (the fitness of the solutions) to the onlooker bees. The onlooker bees tend to select good food sources. A food source becomes exhausted if it does not improve through a predetermined number of cycles (denoted by $maxC$). The employed bees of exhausted food sources then become scouts, which start to search for new food sources.

For the candidate generated by the ABC algorithm, $\boldsymbol{\mathcal{DT}} = \big\{\mathbf{DT}_j = \{dt_j(1,1), dt_j(1,2), ..., dt_j(K,M)\}\big\}$, we compute the fitness associated with $\boldsymbol{\mathcal{DT}}$ and the probabilistic selection for the candidate by (11) and (12)

$$fitness(\boldsymbol{\mathcal{DT}}) = exp\Big(\frac{-\mathcal{L}_{0-1}(\boldsymbol{\mathcal{DT}})}{(\sum \mathcal{L}_{0-1}(\boldsymbol{\mathcal{DT}}))/nPop}\Big) \tag{11}$$

$$\mathrm{P}(\boldsymbol{\mathcal{DT}}) = \frac{fitness}{\sum\limits_{j} fitness_j} \tag{12}$$

The value of the loss function $\mathcal{L}_{0-1}(\boldsymbol{\mathcal{DT}})$ associated with the candidate $\boldsymbol{\mathcal{DT}}$ is computed in Algorithm 2. It is the average of the 0-1 loss function of all training observations $\mathbf{x}_i \in \mathcal{D}$

$$\mathcal{L}_{0-1}(\boldsymbol{\mathcal{DT}}) = \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}_{0-1}(\mathbf{x}_i) \tag{13}$$

In the ABC algorithm, the new candidate solution is generated from $\boldsymbol{\mathcal{DT}}$ by searching for its neighborhood. If the solution cannot be improved over the predefined number of cycles $maxC$, the food source is abandoned and the employed bee of the abandoned food source becomes a scout. We also follow the original ABC algorithm [4, 5] to find the new food source with the note that $dt_j(k,m)$ is bounded in [0,1].

Algorithm 1 Training phase

```
Input: Training set $\mathcal{D}$, $K$ learning algorithms $\{\mathcal{K}_k\}$, maximum number of iteration: $maxT$, population size: $nPop$, abandonment limit parameter: $maxC$
Output: The optimal Decision Template of $\boldsymbol{\mathcal{ODT}}$ and $\{BC_k\}$

(Generate the base classifier)
Learn $K$ classifiers $\{BC_k\}$ on $\mathcal{D}$ using $\mathcal{K}_k,\ k = 1, ..., K$
(Generate the meta-data)
Meta-data $\mathbf{L} = \emptyset$
for each $\mathcal{D}_i$ do
    $\mathcal{D}^{-i} = \mathcal{D} - \mathcal{D}_i$
    Learn ensemble of classifiers on $\mathcal{D}^{-i}$ using $\{\mathcal{K}_k\}$
    Classify samples of $\mathcal{D}_i$ by these classifiers
    Add outputs on samples in $\mathcal{D}_i$ to $\mathbf{L}$ (1)
end for
Use ABC method: for each $\boldsymbol{\mathcal{DT}}$, compute the loss value using Algorithm 2
Select the optimal $\boldsymbol{\mathcal{ODT}}$ with the smallest loss at the end of ABC
Return $\boldsymbol{\mathcal{ODT}}$ and $\{BC_k\}$
```

Algorithm 2 Compute the loss value for each candidate generated in ABC algorithm

```
Input: Candidate $\boldsymbol{\mathcal{DT}}$
Output: The loss value for $\boldsymbol{\mathcal{DT}}$

for each $\mathbf{x}_i \in \mathcal{D}$ do
    for each $\mathbf{DT}_j$ in $\boldsymbol{\mathcal{DT}}$ do
        Compute cardinality between $\mathbf{L}(\mathbf{x}_i)$ and $\mathbf{DT}_j$ (6) (7)
        Compute the similarity $S(\mathbf{L}(\mathbf{x}_i), \mathbf{DT}_j)$ (5)
    end for
    Assign the class label $y$ for $\mathbf{x}_i$ by using (8)
    $\mathcal{L}_{0-1}(\mathbf{x}_i) = \mathbb{I}[y \neq \hat{y}_i]$
end for
Compute $\mathcal{L}_{0-1}(\boldsymbol{\mathcal{DT}})$ by (13)
Return $\mathcal{L}_{0-1}(\boldsymbol{\mathcal{DT}})$
```

4 EXPERIMENTAL STUDIES

4.1 Datasets and Experimental Settings

We conducted experiments on 31 datasets selected from the UCI data depository to compare the performance of the proposed method and the benchmark algorithms (Table 1). We chose 3 learning algorithms: LDA, Naïve Bayes, and KNN$_5$ to construct the ensemble system [13]. These algorithms were chosen because they perform significantly different strategies to train the base classifier therefore they ensure the generation of diverse outputs. For the ABC algorithm, we set the maximum number of iterations $maxT$ to 100, the number of food source $nPop$ to 50, and the abandonment limit parameter $maxC$ to $round(0.6 \times K \times nPop)$.

The benchmark algorithms we used are:

- Decision Template [8] and Sum Rule [6]: We used similar settings as in the proposed method.

Algorithm 3 Classification phase

```
Input: Unlabeled sample $\mathbf{x}$, the optimal Decision Template $\{\boldsymbol{\mathcal{ODT}}_j\}$ and
    $\{BC_k\}$
Output: Predicted class label for $\mathbf{x}$
Obtain the meta-data $\mathbf{L}(\mathbf{x})$ by using $BC_k$ (2)
for each $\boldsymbol{\mathcal{ODT}}_j$ do
    Compute cardinality between $\mathbf{L}(\mathbf{x})$ and $\boldsymbol{\mathcal{ODT}}_j$ (6) (7)
    Compute the similarity $S(\mathbf{L}(\mathbf{x}), \boldsymbol{\mathcal{ODT}}_j)$ (5)
end for
Assign the class label by using (8)
```

Table 1. Datasets in the experimental studies

Dataset name	# of observations	# of classes	# of dimensions
Abalone	4174	3	8
Appendicitis	106	2	7
Australian	690	2	14
Balance	625	3	4
Banana	5300	2	2
Biodeg	1055	2	41
Breast-Cancer	683	2	9
Bupa	345	2	6
Cleveland	297	5	13
Fertility	100	2	9
Haberman	306	2	3
Hayes-Roth	160	3	4
Heart	270	2	13
Iris	150	3	4
Isolet	7797	26	617
Madelon	2000	2	500
Magic	19020	2	10
Musk1	476	2	166
Musk2	6598	2	166
Newthyroid	215	3	5
Page-Blocks	5472	5	10
Phoneme	5404	2	5
Pima	768	2	8
Ring	7400	2	20
Skin_NonSkin	245057	2	3
Spambase	4601	2	57
Vehicle	846	4	18
Vertebral	310	3	6
Waveform_w_Noise	5000	3	40
Waveform_wo_Noise	5000	3	21
Wdbc	569	2	30

– ACO1 and ACO2 [2]: The methods aim to search for optimal subset of base classifiers with Decision Tree as meta-classifier (ACO1) or with the optimal meta-classifier (ACO2) for the heterogeneous ensemble. We used the same three learning algorithms as in the proposed method. For ACO2, one of the three learning algorithms was randomly chosen to train the meta-classifier according to the uniform distribution like in [2].

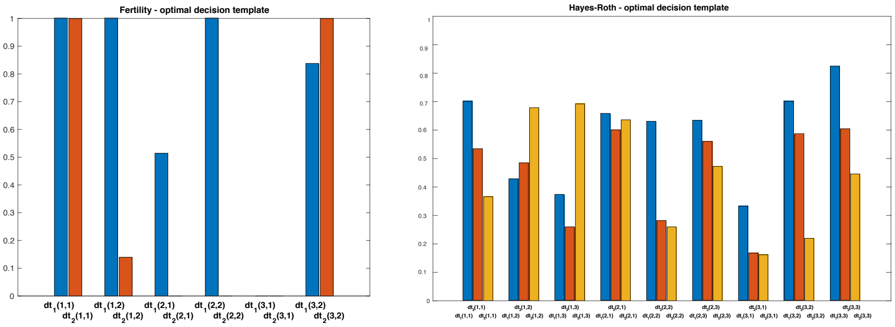


Fig. 2. The Optimal Decision Templates of Fertility and Hayes-Roth datasets

- Random Subspace [1]: We used Decision Tree as the learning algorithm to train 200 base classifiers.
- GA Meta-data [9]: The method searches for the optimal subset of meta-data for heterogeneous ensemble.

We performed 10-fold cross validation and run the test 3 times to obtain 30 test results for each dataset. We used the Wilcoxon signed rank test [3] to compare the classification results of the proposed method and each benchmark algorithm on each dataset. The performance scores of two methods are treated as significantly different if the p-value of the test is smaller than a given significance level. For all tests, the level of significance was set to 0.05. The details of these tests can be found in [3].

4.2 Results and Discussions

Comparison of benchmark algorithms: Table 2 shows the mean and standard deviation of classification error rate computed on 30 runs on each experimental dataset. Because of the singularity property of the covariance matrix of the meta-data [13], some learning methods such as LDA cannot be used as a meta-classifier. This explains why ACO2 cannot be run on the Iris dataset in the experiment. Therefore ACO2 and the proposed method were compared on 30 datasets. The proposed method is developed to work on diverse datasets, not only handling imbalanced ones. Therefore we did not use an appropriate metric to evaluate imbalanced data.

From the Wilcoxon test results in Table 2, we can see that the proposed method is better than the benchmark algorithms on the experimental datasets. Compared to Random Subspace, the proposed method wins on 14 datasets and loses on 6 datasets. Our method also performs better than Sum Rule (the proposed method wins in 21 cases and loses in only 1 case) and Decision Template (the proposed method wins in 21 cases).

In comparison to GA Meta-data, the proposed method wins on 14 datasets while does not lose on any dataset. The pattern is nearly similar when comparing

Table 2. Classification error rates of the benchmark algorithms and the proposed method

Dataset	**Sum Rule**	**Random Subspace**	**Decision Template**	**GA Meta-data**	**ACO1**	**ACO2**	**Proposed Method**
Abalone	0.4681±0.0183	0.4679±0.0255	0.4867±0.0197	0.4736±0.0233	0.4720±0.0283	0.4531±0.0178	0.4553±0.0198
Appendicitis	0.1197±0.0940	0.1385±0.0861	0.1173±0.0846	0.1600±0.1018	0.1827±0.1229	0.1488±0.0940	0.1358±0.0934
Australian	0.1411±0.0355	0.1522±0.0330	0.1382±0.0317	0.1807±0.0360	0.1816±0.0488	0.1280±0.0367	0.1314±0.0339
Balance	0.1099±0.0221	0.2129±0.0355	0.0976±0.0312	0.0852±0.0331	0.0960±0.0285	0.0959±0.0577	0.0853±0.0325
Banana	0.1098±0.0097	0.3746±0.0469	0.1117±0.0117	0.1116±0.0111	0.1129±0.0152	0.1122±0.0117	0.1093±0.0109
Biodeg	0.1425±0.0313	0.1352±0.0251	0.1387±0.0303	0.1836±0.0350	0.1800±0.0348	0.1368±0.0386	0.1286±0.0230
Breast-Cancer	0.0464±0.0260	0.0283±0.0220	0.0459±0.0267	0.0420±0.0285	0.0405±0.0264	0.0347±0.0240	0.0361±0.0248
Bupa	0.2899±0.0532	0.3420±0.0762	0.3324±0.0790	0.3804±0.0925	0.3548±0.0737	0.3209±0.0743	0.2967±0.0625
Cleveland	0.4218±0.0561	0.4184±0.0380	0.4454±0.0615	0.4433±0.0637	0.4642±0.0781	0.4041±0.0706	0.4141±0.0688
Fertility	0.1333±0.0537	0.1200±0.0400	0.4167±0.1655	0.1900±0.1136	0.1467±0.0562	0.1300±0.0526	0.1533±0.0670
Haberman	0.2595±0.0470	0.2961±0.0548	0.3103±0.0574	0.2964±0.0476	0.2984±0.0415	0.2801±0.0507	0.2822±0.0518
Hayes-Roth	0.3417±0.1450	0.3458±0.1395	0.4292±0.1278	0.2917±0.0959	0.2708±0.1145	0.2833±0.1079	0.3000±0.0919
Heart	0.1704±0.0609	0.1877±0.0765	0.1741±0.0559	0.2395±0.0832	0.2185±0.0909	0.1815±0.0692	0.1691±0.0639
Iris	0.0333±0.0375	0.0511±0.0536	0.0333±0.0375	0.0333±0.0413	0.0400±0.0442	-	0.0289±0.0330
Isolet	0.0656±0.0097	0.0588±0.0078	0.0583±0.0093	0.0623±0.0081	0.0650±0.0081	0.0491±0.0076	0.0561±0.0085
Madelon	0.3680±0.0331	0.3800±0.0325	0.2868±0.0290	0.2870±0.0264	0.2870±0.0264	0.2882±0.0254	0.2832±0.0232
Magic	0.1909±0.0061	0.1730±0.0068	0.1901±0.0079	0.1920±0.0117	0.1902±0.0069	0.1918±0.0302	0.1852±0.0068
Musk1	0.1471±0.0426	0.0805±0.0356	0.1308±0.0452	0.1344±0.0401	0.1245±0.0420	0.1205±0.0470	0.1092±0.0406
Musk2	0.0497±0.0078	0.0209±0.0035	0.0463±0.0068	0.0350±0.0059	0.0355±0.0059	0.0399±0.0226	0.0349±0.0055
Newthyroid	0.0948±0.0507	0.0420±0.0390	0.0684±0.0492	0.0371±0.0350	0.0418±0.0369	0.0668±0.0465	0.0573±0.0444
Page-Blocks	0.0497±0.0069	0.0319±0.0055	0.0504±0.0082	0.0420±0.0066	0.0462±0.0085	0.0422±0.0077	0.0431±0.0071
Phoneme	0.1747±0.0173	0.1627±0.0183	0.1780±0.0278	0.1149±0.0145	0.1149±0.0145	0.1163±0.0161	0.1153±0.0137
Pima	0.2465±0.0437	0.2570±0.0477	0.2465±0.0427	0.3056±0.0484	0.3078±0.0485	0.2309±0.0490	0.2288±0.0477
Ring	0.2088±0.0110	0.0298±0.0051	0.1930±0.0128	0.1231±0.0135	0.1211±0.0114	0.1140±0.0126	0.1068±0.0125
Skin_NonSkin	4.12E-02±1.10E-03	2.62E-03±2.74E-04	3.30E-02±1.07E-03	4.31E-04±1.19E-04	4.34E-04±1.13E-04	4.39E-04±1.09E-04	4.15E-04±1.20E-04
Spambase	0.0969±0.0116	0.0960±0.0135	0.0920±0.0109	0.1185±0.0140	0.1224±0.0172	0.0942±0.0131	0.0909±0.0116
Vehicle	0.2605±0.0427	0.2600±0.0361	0.2151±0.0337	0.2627±0.0435	0.2597±0.0379	0.2187±0.0362	0.2186±0.0360
Vertebral	0.2054±0.0603	0.2893±0.0568	0.1925±0.0648	0.1893±0.0581	0.1527±0.0589	0.1516±0.0501	0.1785±0.0605
Waveform_w_Noise	0.1671±0.0127	0.1755±0.0171	0.1634±0.0140	0.1787±0.0143	0.1770±0.0149	0.1457±0.0136	0.1459±0.0163
Waveform_wo_Noise	0.1646±0.0186	0.1498±0.0191	0.1562±0.0181	0.1738±0.0211	0.1705±0.0166	0.1389±0.0188	0.1389±0.0168
Wdbc	0.0352±0.0188	0.0381±0.0187	0.0346±0.0190	0.0352±0.0249	0.0457±0.0292	0.0305±0.0197	0.0305±0.0175
	Win: 21	Win: 14	Win: 21	Win: 14	Win: 18	Win: 5	
	Equal: 9	Equal: 11	Equal: 10	Equal: 17	Equal: 13	Equal: 23	
	Loss: 1	Loss: 6	Loss: 0	Loss: 0	Loss: 0	Loss: 2	

to ACO1 as ours wins on 18 datasets. The proposed method meanwhile performs slightly better compared to ACO2 as ours wins in 5 cases and loses in 2 cases.

Discussion: We explain some reasons why the proposed method is better than the benchmark algorithms. Random Subspace generates the new training sets by choosing observations with a random subset of features from the original feature set. On datasets with high dimension like Musk1 and Musk2 (166 features), Random Subspace can generate the new diverse training sets, resulting in high classification accuracy. Obviously, the proposed method is significantly better than Sum Rule because Sum Rule do not train the combiner on the meta-data of the training set.

GA Meta-data meanwhile uses GA to learn the optimal subset of meta-data from the training set. Since the dimension of the meta-data depends on the number of class labels and the number of learning algorithms, for datasets with a small number of class labels, the subset of meta-data is not diverse enough to enhance the ensemble performance. ACO1 searches for the optimal subset of base classifiers for the training set. The limitation of not searching for meta-classifier makes ACO1 ineffective on many datasets. ACO2 meanwhile performs well since it searches for not only the base classifiers but also the meta-classifier for the optimal solution.

Finally, the proposed method is significantly better than the Decision Template method. Fig. 2 illustrates the optimal decision templates on the Fertility and Hayes-Roth datasets. It is clear that for imbalanced datasets like Fertility, while the decision template is nearly identical among the two class labels, the optimal decision template from our algorithm can clearly distinguish between the two class labels. In proposed method, we search for the optimal decision template that maximizes the discrimination between the different classes and that strategy does not take into account by the Decision Template method.

5 CONCLUSIONS

In summary, we proposed a combining algorithm for heterogeneous ensemble systems. Our method is motivated by the observation that Decision Template method, a popular combining algorithm for heterogeneous ensemble, underperforms on imbalanced datasets because of the similar representations for the class labels. In addition, the average value-based meta-data representation in this method is not good for data with a skewed distribution. To overcome these limitations, we proposed the method to search for a decision template yielding an optimal representation for the meta-data. We used ABC algorithm to minimize the empirical 0-1 loss function on the training set to obtain the optimal solution. For the classification process, we assigned the label for a sample based on the maximization of similarity between the optimal templates and the sample's meta-data. Experiments on 31 UCI datasets showed that the proposed method is better than the selected benchmark algorithms.

References

1. Barandiaran, I.: The random subspace method for constructing decision forests. IEEE Transactions on Pattern Analysis and Machine Intelligence **20**(8) (1998)
2. Chen, Y., Wong, M.L., Li, H.: Applying ant colony optimization to configuring stacking ensembles for data mining. Expert Systems with Applications **41**(6), 2688–2702 (2014)
3. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research **7**(Jan), 1–30 (2006)
4. Karaboga, D.: An idea based on honey bee swarm for numerical optimization. Tech. rep., Technical report-tr06, Erciyes university, engineering faculty, computer engineering department (2005)
5. Karaboga, D., Akay, B.: A comparative study of artificial bee colony algorithm. Applied Mathematics and Computation **214**(1), 108–132 (2009)
6. Kittler, J., Hatef, M., Duin, R.P., Matas, J.: On combining classifiers. IEEE Transactions on Pattern Analysis and Machine Intelligence **20**(3), 226–239 (1998)
7. Kuncheva, L.I.: Combining pattern classifiers: methods and algorithms. John Wiley & Sons (2004)
8. Kuncheva, L.I., Bezdek, J.C., Duin, R.P.: Decision templates for multiple classifier fusion: an experimental comparison. Pattern Recognition **34**(2), 299–314 (2001)
9. Nguyen, T.T., Liew, A.W.C., Pham, X.C., Nguyen, M.P.: A novel 2-stage combining classifier model with stacking and genetic algorithm based feature selection. In: International Conference on Intelligent Computing. pp. 33–43. Springer (2014)
10. Nguyen, T.T., Liew, A.W.C., Tran, M.T., Pham, X.C., Nguyen, M.P.: A novel genetic algorithm approach for simultaneous feature and classifier selection in multi classifier system. In: Evolutionary Computation (CEC), 2014 IEEE Congress on. pp. 1698–1705. IEEE (2014)
11. Nguyen, T.T., Nguyen, M.P., Pham, X.C., Liew, A.W.C.: Heterogeneous classifier ensemble with fuzzy rule-based meta learner. Information Sciences **422**, 144–160 (2018)
12. Nguyen, T.T., Nguyen, M.P., Pham, X.C., Liew, A.W.C., Pedrycz, W.: Combining heterogeneous classifiers via granular prototypes. Applied Soft Computing **73**, 795–815 (2018)
13. Nguyen, T.T., Nguyen, T.T.T., Pham, X.C., Liew, A.W.C.: A novel combining classifier method based on variational inference. Pattern Recognition **49**, 198–212 (2016)
14. Nguyen, T.T., Pham, X.C., Liew, A.W.C., Pedrycz, W.: Aggregation of classifiers: a justifiable information granularity approach. IEEE Transactions on Cybernetics (2018)
15. ŞEn, M.U., Erdogan, H.: Linear classifier combination and selection using group sparse regularization and hinge loss. Pattern Recognition Letters **34**(3), 265–274 (2013)
16. Shunmugapriya, P., Kanmani, S.: Optimization of stacking ensemble configurations through artificial bee colony algorithm. Swarm and Evolutionary Computation **12**, 24–32 (2013)
17. Ting, K.M., Witten, I.H.: Issues in stacked generalization. Journal of Artificial Intelligence Research **10**, 271–289 (1999)
18. Zhou, Z.H.: Ensemble methods: foundations and algorithms. Chapman and Hall/CRC (2012)