The Optimized Selection of Base-Classifiers for Ensemble Classification using a Multi-Objective Genetic Algorithm

Sam Fletcher
Centre for Intelligent Systems
Central Queensland University
Brisbane, Australia
s.fletcher@cqu.edu.au

Brijesh Verma
Centre for Intelligent Systems
Central Queensland University
Brisbane, Australia
b.verma@cqu.edu.au

Zohaib M Jan
Centre for Intelligent Systems
Central Queensland University
Brisbane, Australia
z.jan@cqu.edu.au

Mengjie Zhang
Evolutionary Computation Research Group
Victoria University of Wellington
Wellington, New Zealand
mengjie.zhang@ecs.vuw.ac.nz

***Abstract* – We propose an ensemble classification algorithm that uses multi-objective optimization instead of relying on heuristics or fragile user-defined parameters. Only two user-defined parameters are included, with both being found to have large windows of values that produce statistically indistinguishable results, indicating the low level of expertise required from the user to achieve good results. Additionally, when given a large initial set of trained base-classifiers, we demonstrate that a multi-objective genetic algorithm aiming to optimize prediction accuracy and diversity will prefer particular types of classifiers over others. The total number of chosen classifiers is also surprisingly small – only 5.86 classifiers on average, out of a starting pool of 900. This occurs without any explicit preference for small ensembles of classifiers. Even with these small ensembles, significantly lower empirical classification error is achieved compared to the current state-of-the-art.**

***Keywords* – ensemble classification, multi-objective optimization, genetic algorithm, bagging, multiple classifiers, classifier selection, diversity, double-fault measure.**

I. Introduction

This paper proposes a novel ensemble classification algorithm, designed to be generalized to a large range of possible domains, not specialized to data with certain characteristics or needs. Rather than relying on user design decisions or heuristics, the ensemble uses multi-objective optimization via a genetic algorithm to allow the ensemble to search for its own best solution. The algorithm takes some $n \times m$ data X as input, with each of the n rows $x \in X$ representing an independently and identically distributed sample from some universe χ. Each x is made up of m values uniquely describing the characteristics of x according to a set of m features, where each feature is a set of either categorical (unordered) or numerical (ordered) values. Each x has a label y that represents the classification category that x belongs to in some output space Y. The proposed ensemble classifier $f(X)$ is trained and validated using X so that it can predict the label y of previously unseen (i.e. test) samples $z \in \mathrm{Z} \sim \chi$.

The proposed ensemble $f(X)$ is a collection of multiple base-classifiers $f_i(X_i)$, whose predictions for the labels of $z \in Z$ are combined (also known as "fused") together using majority voting:

$$\tilde{y}_z = f(X) = mode(f_i(X_i);\ \forall i) \quad . \tag{1}$$

The performance of $f(X)$ can then be measured as the fraction of records $z \in Z$ whose labels are correctly predicted:

$$Error_f = \frac{1}{|Z|} \sum_{z}^{Z} \mathbf{1}(\tilde{y}_z \neq y_z) \tag{2}$$

where $\mathbf{1}(\cdot)$ is the indicator function, returning 1 if the statement within the brackets is true, and 0 otherwise.

The paper makes the following novel contributions:

- An algorithm for building ensemble classifiers is proposed, using multi-objective optimization to find the best subset of a diverse collection of base-classifiers.
- The base-classifiers are generated using two levels of diversity-generation: input data diversity and classifier diversity.
- A third level of diversity is induced by having the genetic algorithm optimize for diversity, as well as accuracy.
- The types of base-classifiers that are preferred by the optimization process is explored.

Section II presents related work; Section III presents the novel algorithm; experiments are presented in Section IV; and the paper is concluded in Section V.

This research was supported under the Australian Research Council's Discovery Project funding scheme (Project Number DP160102639).

978-1-5090-6014-6/18/$31.00 ©2018 IEEE

II. Related Work

Ensemble classification is about a collection of classifiers being able to perform better than any individual classifier is capable of [1], [2]. The combination of the base-classifier outputs – known commonly as “fusing” [3], [4] – is often done with simplistic techniques such as majority voting [5], with some research investigating the theoretical properties of fusion techniques such as majority voting, weighted majority voting, recall combination and naïve Bayes combination [6]. Other research has instead customized the ensemble framework to better suit specific domains, such as using base-classifier pruning [7] or under-sampling [8] on imbalanced data.

Diversity is an important characteristic of ensemble classifiers – the idea that if all the base-classifiers make the same mistakes, nothing is gained, and instead it is preferable that the base-classifiers have diversified mistakes. Put another way, more can be learned from base-classifiers with uncorrelated errors, than can be learned from base-classifiers with correlated errors [9]–[11]. While a mathematical definition of “diversity” has remained elusive [12], [13], many empirical results have demonstrated the effectiveness of considering diversity in the design of ensemble classification algorithms [12], [14]–[17]. Some focused research on diversity includes tailoring diversity for imbalanced data [18], fusing the outputs of base-classifiers in a way that encourages diversity [19], and combining multiple explicit measures of diversity into a single measure [20].

Data diversity is one type of diversity – diversifying the input space, so that each base-classifier is seeing a different view of the original data. The original logic behind bagging [21] (aggregating predictions across multiple classifiers that, individually, are unstable to small fluctuations in the data [22]) fits naturally with the goal of diversity in ensemble classification. If the outputs of a classifier are changed by inputting different bootstrapped samples, the errors of those classifiers are likely to be less correlated than the errors of classifiers that use the same data. This reasoning holds up in practice [23]–[26], and is something taken advantage of in this paper.

Clustering has been suggested as an alternative to bagging [27]–[32]. The advantage of clustering is that local clusters of data are more diverse than global bags of data, however this diversity comes at the cost of having base-classifiers that are tasked with classifying records from the global data space, but only learned from a local space. This downside can be avoided by dynamically selecting classifiers that were trained on the same data region as the new record being classified [25], [28], [33]–[35].

Diversity can also be added to the ensemble by using multiple types of classification algorithms to build the base-classifiers [27], [36]. Some work has even explicitly compared the base-classifiers to each other to assess their similarities and differences [37]. This type of diversity can be called “classifier diversity”, and has been shown to be a good way of diversifying the outputted predictions [10].

It is also possible to explicitly maximize diversity, by defining diversity in terms of an objective function that is to be either minimized or maximized. Of course, the ultimate aim of an ensemble classifier is to have as low a prediction error as possible, so optimizing diversity is often framed as a second objective, in conjuncture with minimizing error. Extending the optimization problem to a multi-objective problem is an extension of the simpler problem of explicitly minimizing error [38]–[40]. Since this is a non-convex optimization problem, algorithms such as PSO (particle swarm optimization) [41], [39], [42], [43] and genetic algorithms (i.e. evolutionary algorithms) [38], [40] are often used. These algorithms offer ways to heuristically explore the output space to find the global minima in a tractable amount of time.

Multi-objective and single-objective optimization differ in that in the multi-objective case, there is no longer a single solution. Instead, there are multiple non-dominated solutions, where in order to improve the output of one objective function, the output of the second objective function (or another objective function, in cases with more than two) necessarily gets worse. These non-dominated solutions make up what is known as the Pareto front [44], [45], with every solution on the front being equally “best” (assuming each objective function is equally important). A wide variety of work has been done on how best to use multi-objective optimization in ensemble classifiers, such as optimizing feature selection [44], [41], optimizing weights for voting [36], tailoring it to class imbalance problems [45], and optimizing the selection of classifiers built from local cluster regions of the data space [46].

The ideal number of base-classifiers in an ensemble has also been researched. While there are obviously computational benefits in having less base-classifiers in an ensemble, there also appear to be improvements in prediction accuracy [47], [48]. Some work explicitly forces the ensemble to become smaller by pruning away bad base-classifiers [49], [47]. Other work achieves a similar effect by using sparse weights for classifier selection [48]. Genetic algorithms have also been used to reduce the number of base-classifiers [50], as has convex quadratic programming [51]. In this paper, we demonstrate that achieving a small ensemble does not require explicit or dedicated processes, but is instead a natural outcome of diversifying the base-classifiers.

III. The Proposed Ensemble Algorithm

The proposed algorithm is first summarized, and then outlined more formally in the steps below. Dataset X is separated into training data T and selection data U. b bootstrapped subsets of T are inputted into c different base-classifiers, for a total of bc trained classifiers. Selection data U is run through the bc classifiers, and the predicted labels are inputted into a genetic algorithm for multi-objective optimization. The two objectives being optimized are prediction error and diversity, which are formally defined later in the paper. From the resulting Pareto front of solutions, the solution with the lowest prediction error is selected as the final ensemble classifier. The full algorithm is outlined in the steps below, with pseudocode provided in Fig. 1.

Steps:

1) Separate X into one subset T containing θn of the records for training, and another subset U containing the other $(1-\theta)n$ records for classifier selection, $0 < \theta < 1$.

2) Create b bootstrap bags of size $|T|$ by sampling T with replacement.

3) For each bag B, train c classifiers.

4) Find the optimal subset of the bc classifiers that best classifies the records in U.

   a) Do this using a multi-objective genetic algorithm that aims to minimize:

      i. Classification error; and

      ii. The correlation between the erroneous votes made by the selected classifiers.

   b) If the Pareto front contains multiple data points, select the point with the minimum classification error, breaking ties based on which one has the smallest number of selected classifiers.

5) The chosen subset of classifiers is the final ensemble.

The guiding principle of the proposed algorithm is that a diverse ensemble has more potential than a homogeneous ensemble (as described in Section 2). Diversity is injected into the algorithm in three places:

Data diversity: bagging [21] provides a simple method for modifying the training data inputted into each base-classifier. By inputting n records that are sampled with replacement into each base-classifier, the internal decisions and final outputs of the base-classifiers are diversified. The ideal number of bags b to generate is empirically investigated in Section 4.

Classifier diversity: by training a diverse range of classifier types, the algorithm is provided with more opportunities to discover new relationships and patterns in the data. Classifier "types" refers to different architectures of classifier, such as Decision Trees (DT) [52], [53], Support Vector Machines (SVM) [54], Naïve Bayes classifiers (NB) [55], Discriminant Analysis classifiers (DA) [56], k-Nearest Neighbor classifiers (kNN) [57], Randomly Under-Sampled Boosted classifiers (RUSBoost) [58], Adaptive Boosting classifiers (AdaBoost) [59], Random Forest (RF) [60] and Artificial Neural Networks (ANN) [61], [62].

While using multiple classifier types provides opportunities for new discoveries, it does not guarantee it, nor is there any guarantee that some of the classifiers will not be very similar (i.e. have low diversity). This risk is averted however by the multi-objective genetic algorithm, in which one of the two objectives is diversity; if two classifiers have low diversity, the genetic algorithm will aim to exclude one of them. Interestingly, the optimization algorithm prefers some types of classifiers over others, depending on the data being learned from. This phenomenon is investigated further in Section 4.

Output diversity: **t**he third place that diversity is added to the proposed algorithm is in the fusing of the base-classifiers' outputs. After the c classifiers have been trained on the b bags, the bc classifiers are pruned down to a smaller set. This smaller set of classifiers is optimized to not only have high accuracy, but also high diversity. This is achieved by using a multi-objective genetic algorithm, where:

- One objective is to find the set of classifiers that has the fewest incorrect predictions when using majority voting to predict the labels of selection data; and
- The other objective is to find the set of classifiers where each pair of classifiers both make incorrect predictions as rarely as possible.

```
procedure Build_Ensemble
input: Dataset $X$ of size $n$, labels $Y$, base-classifier algorithms $C$, training ratio $\theta$, number of bags $b$
Set of trained base-classifiers $S \leftarrow \emptyset$
Training Data $\{T, Y_T\} \leftarrow \theta n$ records randomly sampled from $\{X, Y\}$, without replacement
Selection Data $\{U, Y_U\} \leftarrow \{X \setminus T,\ Y \setminus Y_T\}$
for $b' =\ 1$ to b:
    Bag $\{B, Y_B\}\ \leftarrow |T|$ records randomly sampled from $\{T, Y_T\}$, with replacement.
    for $c$ in $C$:
        $S \leftarrow S \cup$ Train classifier $c$ with data $\{B, Y_B\}$.
    end for
end for
$W \leftarrow$ Optimize_Ensemble($S$, $\{U, Y_U\}$)
output: $W$


procedure Optimize_Ensemble
input: Ensemble $S$, data $\{U, Y_U\}$
Classification error objective function $f_E \leftarrow \sum_u^U \mathbf{1}(mode(s(u); \forall s \in S) \neq y_u)$
Diversity objective function $f_D \leftarrow |U| \times \frac{|S|(|S|-1)}{2} \times \sum_i^{|S|} \sum_{j \neq i}^{|S|} \sum_u^U \mathbf{1}\big(s_i(u) \neq y_u\ \wedge\ s_j(u) \neq y_u\big)$
$W \leftarrow$ Find $W \subseteq S$ that minimizes $\{f_E, f_D\}$ using a multi-objective genetic algorithm.
output: $W$
```

Fig. 1. Pseudocode of the proposed ensemble algorithm.

Formally, the first objective function $f_E(S,U)$ aims to minimize:

$$f_E(S,U) := \sum_{u}^{U} \mathbf{1}(mode(s(u); \forall s \in S) \neq y_u) \tag{3}$$

where S is the set of trained base-classifiers, U is the selection data, and $s(u)$ is classifier s's predicted label for record u. The second objective $f_D(S,U)$ aims to minimize the double-fault measure [11]:

$$f_D(S,U) := |U| \times \frac{|S|(|S|-1)}{2} \times \sum_{i}^{|S|} \sum_{j \neq i}^{|S|} \sum_{u}^{U} \mathbf{1}\left(s_i(u) \neq y_u \wedge s_j(u) \neq y_u\right) \tag{4}$$

which at its core is a summation of the number of pairs of classifier that both predict incorrect labels, for records $u \in U$. Minimizing the double-fault measure therefore aligns well with the goal of diversity: of having uncorrelated errors [11].

The genetic algorithm will then converge on a Pareto front of solutions that optimize both objective functions as much as possible. This front includes all non-dominated solutions; that is, all solutions in which it is not possible to decrease one objective function without increasing the other. This means that barring getting stuck in a local minima (which the genetic algorithm is designed to avoid using strategies such as random mutations [44], [46]), the solution with the lowest possible validation error is guaranteed to be included on the Pareto front, and if multiple solutions with equally low validation error were discovered, the solutions with the best diversity were the ones that were selected. An example of a Pareto front is provided in Fig. 2.

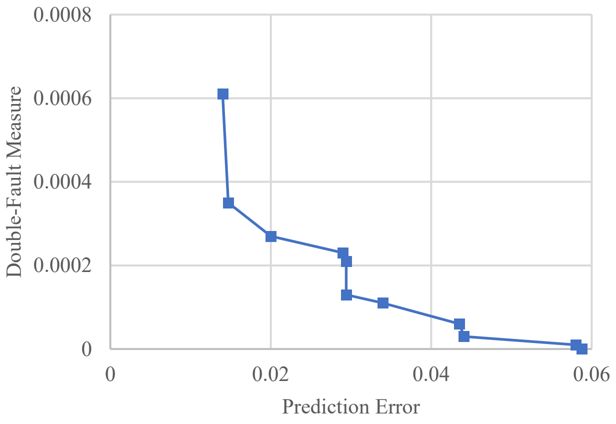

Fig. 2. An example of a Pareto front, using the two objective functions used in this paper.

Ultimately, the goal of an ensemble classifier is to minimize classification error. The proposed algorithm therefore selects the solution on the Pareto front with the minimum classification error, knowing that it is not possible for a chosen subset of classifiers to be any more diverse, without sacrificing classification performance. In the case of a tie (i.e. multiple solutions with the same error and diversity), the solution with the fewest selected classifiers is used. If there is still a tie, a solution is chosen randomly.

IV. Experiments

This section aims to answer the following questions:

- For ensemble classification, how does multi-objective optimization compare to single-objective optimization?
- What types of base-classifiers does the ensemble prefer?
- How does the proposed ensemble algorithm compare to state-of-the-art ensemble algorithms?

To answer these questions, empirical experiments are performed. Eleven datasets from the UCI Machine Learning Repository [63] are used, with the average results of five repetitions of ten-fold stratified cross-validation being used for all experiments. Summaries of the datasets used are provided in Table I.

For the experiment involving different versions of the algorithm proposed in this paper, statistical significance testing is conducted using the Mann-Whitney U test (also known as the Wilcoxon Rank-Sum test) at the $\alpha = 0.05$ level. To correct for multiple tests (i.e. having multiple datasets), the Holm-Bonferroni correction is used. The statistically-significantly best result for each dataset is presented in bold.

For experiments involving algorithms proposed in other papers (where raw results are not available), testing for statistically significant differences between pairs of algorithms is done across all datasets. The Wilcoxon Signed Rank test is used at the $\alpha = 0.05$ level, with the Holm-Bonferroni correction [64].

TABLE I

DETAILS OF THE DATASETS USED IN THE EXPERIMENTS.

Dataset	n	m	|Y|
Sonar	208	60	2
Heart	270	13	2
Ionosphere	351	34	2
Balance	625	4	3
WBC	683	9	2
AusCredit	690	14	2
PimaDiabetes	768	8	2
Vehicle	846	18	4
Segmentation	2310	19	7
Waveform	5000	40	3
PageBlocks	5473	10	5

For these experiments we use $c = 9$, with the following classification algorithms being used: SVM, DT, kNN, DA, NB, RUSBoost, ANN, RandomForest, AdaBoost. Matlab's implementation of the genetic algorithm and the base-classifiers are used in these experiments [65]. In order to make our experiments reproducible, the default Matlab settings are used in all cases, except for the following:

- Regularization is turned off for discriminant analysis models, to prevent the software throwing an error if a feature with zero variance is inputted.
- Kernel smoothing density estimation is used when building naive Bayes models, instead of using Gaussian distributions, to avoid the software throwing an error if a feature with zero variance is inputted.
- The population type is changed from continuous to binary, due to the selection of a classifier being a yes/no question, not a parameter in continuous space. Thus each "chromosome" is a bit-string of length bc, with each bit representing the selection status of each base-classifier.

A. Single-Objective vs. Multi-Objective Optimization

Table II shows the classification error of the proposed algorithm when using multi-objective optimization with the two proposed objective functions, compared to using the single classification error objective function. In Table II, three results are statistically significant (presented in bold), with no statistically significant difference between the results of the other datasets. For the three statistically significant differences, all of them show multi-objective optimization performing better than single-objective optimization. In no cases is single-objective optimization better.

B. Types of Base-Classifiers Selected

Table III presents the average number of times each of the nine types of base-classifier are selected by the genetic algorithm, across the 50 experiment runs. The final column provides the total number of base-classifiers selected, and the final row provides the average number of times each classifier type was selected across all datasets.

The first observation is that both the total number of selected base-classifiers, and the particular types of selected base-classifiers, depends heavily on the dataset. For some datasets, such as Segmentation and WBC, less than four base-classifiers are needed to make a highly accurate ensemble classifier. Meanwhile the Waveform dataset requires 13.4 to achieve the same high performance. The types of base-classifiers preferred by each dataset varies substantially as well. ANN's are the most preferred classifier type on average, but for several datasets there are close to zero ANN's. On the other hand, DA and AdaBoost only appear in small numbers, but they still appear in a non-zero number of ensembles, indicating that even rarely-occurring classifier types are worth including in the algorithm.

TABLE II

CLASSIFICATION ERROR WHEN USING SINGLE-OBJECTIVE OPTIMIZATION COMPARED TO MULTI-OBJECTIVE OPTIMIZATION IN THE PROPOSED ALGORITHM. STATISTICALLY SIGNIFICANT RESULTS ARE PRESENTED IN BOLD.

Dataset	**Single-Objective**	**Multi-Objective**
Sonar	0.193	0.167
Heart	0.164	0.173
Ionosphere	0.077	0.068
Balance	0.100	**0.070**
WBC	0.034	0.029
AusCredit	0.138	0.136
PimaDiabetes	0.235	0.240
Vehicle	0.246	**0.222**
Segmentation	0.032	**0.025**
Waveform	0.139	0.137
PageBlocks	0.029	0.027

TABLE III

THE NUMBER OF EACH TYPE OF CLASSIFIER CHOSEN BY THE GENETIC ALGORITHM, FROM A STARTING POOL OF 100 BAGS (I.E. 100 OF EACH CLASSIFIER TYPE). NOTE THAT THE AVERAGE TOTAL NUMBER OF BASE-CLASSIFIERS SELECTED IS 5.86, OUT OF THE ORIGINAL 900.

Dataset	SVM	DT	kNN	DA	NB	RUSBoost	ANN	Random Forest	AdaBoost	Total
Sonar	0.3	0.9	1.7	0.1	0.5	0.8	0.8	0.8	0.1	5.9
Heart	0.6	0.5	0.4	0.2	0.5	0.8	0.7	0.3	0.5	4.5
Ionosphere	0.4	0.7	0.0	0.0	1.1	0.7	0.4	0.8	0.0	4.1
Balance	0.0	0.2	0.0	0.0	0.2	0.0	4.2	0.0	0.0	4.6
WBC	0.0	0.4	0.8	0.0	0.2	0.4	0.8	0.4	0.2	3.2
AusCredit	0.6	1.0	0.6	0.2	0.4	1.2	0.6	0.2	0.6	5.4
PimaDiabetes	0.8	1.8	1.0	0.6	0.4	0.6	1.0	0.4	0.2	6.8
Vehicle	0.6	0.6	0.4	0.0	0.0	0.2	5.2	0.4	0.0	7.4
Segmentation	0.2	0.4	1.4	0.0	0.0	0.4	0.4	1.0	0.0	3.8
Waveform	1.6	1.6	2.2	0.6	0.4	0.4	4.8	1.8	0.0	13.4
PageBlocks	0.2	0.8	0.6	0.0	0.2	2.2	0.6	0.8	0.0	5.4
Average	**0.48**	**0.81**	**0.83**	**0.15**	**0.35**	**0.70**	**1.77**	**0.63**	**0.15**	**5.86**

TABLE IV

AN ERROR COMPARISON OF THE PROPOSED ALGORITHM (WITH $b = 100, c = 9, \theta = 0.5$) TO THREE OTHER ALGORITHMS. STATISTICALLY SIGNIFICANT DIFFERENCES BETWEEN THE PROPOSED ALGORITHM AND EACH OTHER ALGORITHM ARE TESTED, WITH THE P-VALUE PROVIDED AT THE TOP OF EACH COLUMN. FOR THE TWO STATISTICALLY DIFFERENT ALGORITHMS, THE BEST-PERFORMING RESULTS ARE PRESENTED IN BOLD.

	Proposed Algorithm	Kuncheva & Rodriguez 2014 (NB)	Fletcher & Verma 2017	Zhang & Suganthan 2015 (MPRRoF-P)
p-value	**-**	**0.006**	0.288	0.342
Sonar	**0.167**	0.238	0.1295	0.192
Heart	**0.173**	0.195	0.1778	0.176
Ionosphere	**0.068**	0.070	0.079	0.053
Balance	**0.070**	0.180	0.099	0.113
WBC	0.040	0.040	0.034	0.033
AusCredit	**0.138**	0.142	0.136	0.134
PimaDiabetes	0.249	0.245	0.228	0.247
Vehicle	**0.210**	0.275	0.241	0.222
Segmentation	**0.023**	0.036	0.030	0.020
Waveform	**0.140**	0.170	0.140	0.140
PageBlocks	**0.027**	0.032	0.030	0.030

The second thing to note in Table III is that the average total number of selected base-classifiers is 5.86. This is out of original pool of 900 base-classifiers ($b \times c = 100 \times 9$). This demonstrates two things: that only a very small number of base-classifiers is needed in order to make a powerful ensemble classifier; and that a genetic algorithm is all that is needed to heavily reduce an initially large pool of classifiers. It also indicates that optimizing for both accuracy and diversity may naturally lead to a small number of base-classifiers being best.

C. Performance Analysis

Using $b = 100$ and $\theta = 0.5$, the proposed algorithm is compared to the current state-of-the-art in ensemble classification. In Table IV, the error scores achieved by the algorithm proposed in this paper are compared to the error scores reported by three recent papers: Zhang & Suganthan, 2015 [52], Fletcher & Verma, 2017 [27] and Kuncheva & Rodriguez [6]. In cases where multiple versions of an algorithm were presented in a paper, the results of the version that the authors recommend (i.e. the highest-performing version) are used. Using the Wilcoxon Signed Rank test (with the Holm-Bonferroni correction) with a significance threshold of $\alpha = 0.05$, it is found that the proposed algorithm is statistically significantly different from one of the three other algorithms (Kuncheva & Rodriguez [6]), and statistically indistinguishable from the other two algorithms. In the case of Kuncheva & Rodriguez, the proposed algorithm achieves lower classification error by an average of 2.9 percentage points.

V. CONCLUSION

An ensemble classification algorithm using multi-objective optimization has been proposed in this paper. Three types of diversity were injected into the algorithm at different stages, resulting in a small, diverse subset of base-classifiers being selected. This small collection of base-classifiers was shown to rival the predictive performance of state-of-the-art ensemble algorithms.

An analysis of the types of base-classifiers preferred in the optimal ensemble was also performed. While some classifier types (such as kNN and ANN) are more commonly selected than others (such as DA and AdaBoost), it ultimately depends on the characteristics of the data. Even rarely-selected classifier types are still sometimes selected, and the optimizer is capable of pruning away all copies of a classifier type if they are unnecessary. Interestingly, only a very small number of base-classifiers are selected by the optimizer for the final ensemble; out of an initial pool of 900 classifiers, an average of only 5.86 are needed to make an ensemble classifier capable of out-performing multiple state-of-the-art algorithms.

Future research will investigate diversifying the base-classifiers even more, by using a range of values for the parameters of the base-classifiers. Using the out-of-bag data to validate the performance of the base-classifiers trained on each bag is another promising direction to explore.

References

[1] D. Gopika and B. Azhagusundari, “An analysis on ensemble methods in classification tasks,” *Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 3, no. 7, pp. 7423–7427, 2014.

[2] L. Rokach, “Ensemble-based classifiers,” *Artif. Intell. Rev.*, vol. 33, no. 1, pp. 1–39, 2010.

[3] A. H. R. Ko, R. Sabourin, A. de S. Britto, and L. Oliveira, “Pairwise fusion matrix for combining classifiers,” *Pattern Recognit.*, vol. 40, no. 8, pp. 2198–2210, 2007.

[4] L. I. Kuncheva, “Switching Between Selection and Fusion in Combining Classifiers: An Experiment,” *IEEE Trans. Syst. Man. Cybern.*, vol. 32, no. 2, pp. 146–156, 2002.

[5] J. Kittler and M. Hatef, “On combining classifiers,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 3, pp. 226–239, 1998.

[6] L. I. Kuncheva and J. J. Rodriguez, “A weighted voting framework for classifiers ensembles,” *Knowl. Inf. Syst.*, vol. 38, no. 2, pp. 259–275, 2014.

[7] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera, “Ordering-based pruning for improving the performance of ensembles of classifiers in the framework of imbalanced datasets,” *Inf. Sci. (Ny).*, vol. 354, no. 1, pp. 178–196, 2016.

[8] J. F. Diez-Pastor, J. J. Rodriguez, C. Garcia-Osorio, and L. I. Kuncheva, “Random Balance: Ensembles of variable priors classifiers for imbalanced data,” *Knowledge-Based Syst.*, vol. 85, no. 1, pp. 96–111, 2015.

[9] L. I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*, 2nd ed. John Wiley & Sons, Inc., 2014.

[10] A. S. Britto, R. Sabourin, and L. E. S. Oliveira, “Dynamic selection of classifiers - A comprehensive review,” *Pattern Recognit.*, vol. 47, no. 11, pp. 3665–3680, 2014.

[11] E. K. Tang, P. N. Suganthan, and X. Yao, “An analysis of diversity measures,” *Mach. Learn.*, vol. 65, no. 1, pp. 247–271, 2006.

[12] L. Didaci, G. Fumera, and F. Roli, “Diversity in classifier ensembles: Fertile concept or dead end?,” in *International Workshop on Multiple Classifier Systems*, 2013, pp. 37–48.

[13] L. I. Kuncheva and J. J. Rodriguez, “Classifier ensembles with a random linear oracle,” *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 4, pp. 500–508, 2007.

[14] M. Asafuddoula, B. Verma, and M. Zhang, “An Incremental Ensemble Classifier Leaning by Means of a Rule-Based Accuracy and Diversity Comparison,” in *International Joint Conference on Neural Networks*, 2017, p. 8.

[15] L. I. Kuncheva and C. J. Whitaker, “Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy,” *Mach. Learn.*, vol. 51, no. 2, pp. 181–207, 2003.

[16] Y. Ren, L. Zhang, and P. N. Suganthan, “Ensemble Classification and Regression - Recent Developments, Applications and Future Directions,” *IEEE Comput. Intell. Mag.*, vol. 11, no. 1, pp. 41–53, 2016.

[17] B. Sluban and N. Lavrac, “Relating ensemble diversity and performance: A study in class noise detection,” *Neurocomputing*, vol. 160, no. 1, pp. 120–131, 2015.

[18] J. F. Diez-Pastor, J. J. Rodriguez, C. I. Garcia-Osorio, and L. I. Kuncheva, “Diversity techniques improve the performance of the best imbalance learning ensembles,” *Inf. Sci. (Ny).*, vol. 325, no. 1, pp. 98–117, 2015.

[19] Q. Hu, L. Li, X. Wu, G. Schaefer, and D. Yu, “Exploiting diversity for optimizing margin distribution in ensemble learning,” *Knowledge-Based Syst.*, vol. 67, no. 1, pp. 90–104, 2014.

[20] G. D. C. Cavalcanti, L. S. Oliveira, T. J. M. Moura, and G. V. Carvalho, “Combining diversity measures for ensemble pruning,” *Pattern Recognit. Lett.*, vol. 74, no. 1, pp. 38–45, 2016.

[21] L. Breiman, “Bagging predictors,” *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996.

[22] J. Abellan and C. J. Mantas, “Improving experimental studies about ensembles of classifiers for bankruptcy prediction and credit scoring,” *Expert Syst. Appl.*, vol. 41, no. 8, pp. 3825–3830, 2014.

[23] K. H. Chang and D. S. Parker, “Complementary prioritized ensemble selection,” in *International Joint Conference on Neural Networks*, 2016, pp. 863–872.

[24] S. Mao, L. Jiao, L. Xiong, S. Gou, B. Chen, and S. K. Yeung, “Weighted classifier ensemble based on quadratic form,” *Pattern Recognit.*, vol. 48, no. 5, pp. 1688–1706, 2015.

[25] D. V. R. Oliveira, G. D. C. Cavalcanti, and R. Sabourin, “Online Pruning of Base Classifiers for Dynamic Ensemble Selection,” *Pattern Recognit.*, vol. 72, no. 1, pp. 44–58, 2017.

[26] E. Santucci, L. Didaci, G. Fumera, and F. Roli, “A parameter randomization approach for constructing classifier ensembles,” *Pattern Recognit.*, vol. 69, no. 1, pp. 1–13, 2017.

[27] S. Fletcher and B. Verma, “Removing Bias from Diverse Data Clusters for Ensemble Classification,” in *International Conference on Neural Information Processing*, 2017, pp. 1–10.

[28] G. Nasierding, G. Tsoumakas, and A. Kouzani, “Clustering Based Multi-Label Classification for Image Annotation and Retrieval,” in *IEEE International Conference on Systems, Man, and Cybernetics*, 2009, pp. 4514–4519.

[29] A. Rahman and B. Verma, “Ensemble classifier generation using non-uniform layered clustering and Genetic Algorithm,” *Knowledge-Based Syst.*, vol. 43, pp. 30–42, 2013.

[30] A. Strehl and J. Ghosh, “Cluster Ensembles - A Knowledge Reuse Framework for Combining Multiple Partitions,” *J. Mach. Learn. Res.*, vol. 3, no. 1, pp. 583–617, 2002.

[31] B. Verma and A. Rahman, “Cluster Oriented Ensemble Classifier: Impact of Multi-cluster Characterisation on Ensemble Classifier Learning,” *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 4, pp. 605–618, 2012.

[32] Y. Yang and J. Jiang, “Hybrid Sampling-Based Clustering Ensemble with Global and Local Constitutions,” *IEEE Trans. Neural Networks Learn. Syst.*, vol. 27, no. 5, pp. 952–965, 2016.

[33] R. Cruz, R. Sabourin, G. Cavalcanti, and T. I. Ren, “META-DES: A dynamic ensemble selection framework using meta-learning,” *Pattern Recognit.*, vol. 48, no. 5, pp. 1925–1935, 2015.

[34] R. Lysiak, M. Kurzynski, and T. Woloszynski, “Optimal selection of ensemble classifiers using measures of competence and diversity of base classifiers,” *Neurocomputing*, vol. 126, no. 1, pp. 29–35, 2014.

[35] J. Xiao, L. Xie, C. He, and X. Jiang, “Dynamic classifier ensemble model for customer classification with imbalanced class distribution,” *Expert Syst. Appl.*, vol. 39, no. 3, pp. 3668–3675, 2012.

[36] A. Onan, S. Koruko, and H. Bulut, “A multiobjective weighted voting ensemble classifier based on differential evolution algorithm for text sentiment classification,” *Expert Syst. Appl.*, vol. 62, no. 1, pp. 1–16, 2016.

[37] J. Abellan and J. G. Castellano, “A comparative study on base classifiers in ensemble methods for credit scoring,” *Expert Syst. Appl.*, vol. 73, no. 1, pp. 1–10, 2017.

[38] H. Chen and X. Yao, “Evolutionary random neural ensembles based on negative correlation learning,” in *2007 IEEE Congress on Evolutionary Computation*, 2007, pp. 1468–1474.

[39] M. R. Bonyadi, “Particle Swarm Optimization for Single Objective Continuous Space Problems: A Review,” *Evol. Comput.*, vol. 25, no. 1, pp. 1–54, 2017.

[40] I. Mendialdua, A. Arruti, E. Jauregi, E. Lazkano, and B. Sierra, “Classifier Subset Selection to construct multi-classifiers by means of estimation of distribution algorithms,” *Neurocomputing*, vol. 157, no. 1, pp. 46–60, 2015.

[41] L. E. Santana and A. M. de P. Canuto, “Filter-based optimization techniques for selection of feature subsets in ensemble systems,” *Expert Syst. Appl.*, vol. 41, no. 1, pp. 1622–1631, 2014.

[42] A. A. Aburomman and M. B. I. Reaz, “A novel SVM-kNN-PSO ensemble method for intrusion detection system,” *Appl. Soft Comput.*, vol. 38, no. 1, pp. 360–372, 2016.

[43] E. S. Correa, A. A. Freitas, and C. G. Johnson, “Particle Swarm for Attribute Selection in Bayesian Classification: An Application to Protein Function Prediction,” *J. Artif. Evol. Appl.*, vol. 2008, no. 1, p. 12, 2008.

[44] L. S. Oliveira, M. Morita, R. Sabourin, and F. Bortolozzi, “Multi-objective Genetic Algorithms to Create Ensemble of Classifiers,” *Lect. Notes Comput. Sci.*, vol. 3410, no. 1, pp. 592–606, 2005.

[45] U. Bhowan, M. Johnston, M. Zhang, and X. Yao, “Evolving Diverse Ensembles Using Genetic Programming for Classification With Unbalanced Data,” *IEEE Trans. Evol. Comput.*, vol. 17, no. 3, pp. 368–386, 2013.

[46] A. Rahman and B. Verma, “Cluster oriented ensemble classifiers using multi-objective evolutionary algorithm,” in *The 2013 International Joint Conference on Neural Networks*, 2013, pp. 1–6.

[47] M. Bhardwaj, V. Bhatnagar, and K. Sharma, “Cost-effectiveness of classification ensembles,” *Pattern Recognit.*, vol. 57, no. 1, pp. 84–

96, 2016.

[48] H. Chen, P. Tino, and X. Yao, “Predictive ensemble pruning by expectation propagation,” *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 7, pp. 999–1013, 2009.

[49] Q. Dai, R. Ye, and Z. Liu, “Considering diversity and accuracy simultaneously for ensemble pruning,” *Appl. Soft Comput.*, vol. 58, no. 1, pp. 75–91, 2017.

[50] X. C. Yin, K. Huang, H. W. Hao, K. Iqbal, and Z. Bin Wang, “A novel classifier ensemble method with sparsity and diversity,” *Neurocomputing*, vol. 134, no. 1, pp. 214–221, 2014.

[51] X. C. Yin, K. Huang, C. Yang, and H. W. Hao, “Convex ensemble learning with sparsity and diversity,” *Inf. Fusion*, vol. 20, no. 1, pp. 49–59, 2014.

[52] L. Zhang and P. N. Suganthan, “Oblique Decision Tree Ensemble via Multisurface Proximal Support Vector Machine,” *IEEE Trans. Cybern.*, vol. 45, no. 10, pp. 2165–2176, 2015.

[53] J. R. Quinlan, *C4.5: Programs for Machine Learning*, 1st ed. Morgan kaufmann, 1993.

[54] J. A. K. Suykens and J. Vandewalle, “Least Squares Support Vector Machine Classifiers,” *Neural Process. Lett.*, vol. 9, no. 3, pp. 293–300, 1999.

[55] A. Y. Ng and M. I. Jordan., “On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes,” in *Advances in neural information processing systems*, 2002, pp. 841–848.

[56] S. Mika, G. Ratsch, J. Weston, B. Schölkopf, and K.-R. Muller, “Fisher discriminant analysis with kernels,” in *IEEE Signal Processing Society Workshop*, 1999, pp. 41–48.

[57] K. Weinberger, J. Blitzer, and L. Saul, “Distance metric learning for large margin nearest neighbor classification,” in *Advances in Neural Information Processing Systems*, 2006, pp. 1473–1480.

[58] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, “RUSBoost: A Hybrid Approach to Alleviating Class Imbalance,” *IEEE Trans. Syst. Man, Cybern. Part A Syst. Humans*, vol. 40, no. 1, pp. 185–197, 2010.

[59] Y. Freund and R. Schapire, “A desicion-theoretic generalization of on-line learning and an application to boosting,” *Comput. Learn. theory*, vol. 55, pp. 119–139, 1995.

[60] L. Breiman, “Random forests,” *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[61] H. B. Demuth, M. H. Beale, O. De Jess, and M. T. Hagan, *Neural network design*. Martin Hagan, 2014.

[62] R. Hecht-Nielsen, “Theory of the backpropagation neural network,” in *International Joint Conference on Neural Networks*, 1989, pp. 445–448.

[63] M. Lichman, “UCI Machine Learning Repository,” 2013. [Online]. Available: http://archive.ics.uci.edu/ml/.

[64] J. Demšar, “Statistical comparisons of classifiers over multiple data sets,” *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, 2006.

[65] MathWorks, “MATLAB and Statistics and Machine Learning Toolbox.” The MathWorks, Inc., Natick, Massachusetts, United States.