An Evolutionary Approach to Feature Selection and Classification

Rodica Ioana Lung[1] and Mihai-Alexandru Suciu[1,2(✉)]

[1] Centre for the Study of Complexity, Babeș-Bolyai University, Cluj-Napoca, Romania
{rodica.lung,mihai.suciu}@ubbcluj.ro
[2] Faculty of Mathematics and Computer Science, Babeș-Bolyai University, Cluj-Napoca, Romania

Abstract. The feature selection problem has become a key undertaking within machine learning. For classification problems, it is known to reduce the computational complexity of parameter estimation, but it also adds an important contribution to the explainability aspects of the results. An evolution strategy for feature selection is proposed in this paper. Feature weights are evolved with decision trees that use the Nash equilibrium concept to split node data. Trees are maintained until the variation in probabilities induced by feature weights stagnates. Predictions are made based on the information provided by all the trees. Numerical experiments illustrate the performance of the approach compared to other classification methods.

Keywords: feature selection · evolutionary strategy · decision tree · random forest · game theory

1 Introduction

Evolutionary algorithms (EAs) have been widely used for feature selection and classification purposes [6,23,27], as they are flexible and adaptable to different optimization environments. Genetic algorithms (GAs) have been the first natural choice, since the binary representation fits within this problem naturally [9]. Many examples of genetic algorithms are mentioned in [27], and in particular, the combination with Decision Trees (DT) has been appealing from the start [2], with many variants following, expanding to random forests [10] or multi-objective optimization [26]. Examples of EAs for feature selection and decision trees can be found in [13,14], and applications in network intrusion detection [21], chemistry [10], speech recognition [15], etc.

However, the efficiency of any evolutionary approach depends on proper parameter tuning and fitness evaluation mechanisms. Within EAs, the selection is mainly responsible for guiding the search, as the survival of newly created individuals ultimately relies on their fitness. When the fitness is associated with the results of classification tasks and is based on some performance indicator

G. Nicosia et al. (Eds.): LOD 2023, LNCS 14505, pp. 333–347, 2024.
https://doi.org/10.1007/978-3-031-53969-5_25

reported on data samples, we find a high variability between different samples, which makes comparisons of results irrelevant for selection purposes.

The fighting for survival paradigm is usually implemented within EAs by comparing individuals using their fitness values and deciding, depending on the selection mechanism used, which ones are preserved and which are discarded. It is considered that individuals compete for resources and the fittest will survive, to further access, exploit, and explore them.

In this paper, we introduce a feature selection-based classification method that evolves feature weights by using game-theoretic decision trees. Individuals represent vectors of feature importances, evolved with the purpose of identifying the most relevant features in the data set that may explain the classification problem. A game-theoretic decision tree is used for classification and evaluation purposes. However, there is no selection involved, trees are grown together, and they form an ecosystem in which all of them are involved in the prediction task. An individual stops evolving when there is no more variation in the probabilities that it provides for selecting features for inducting its tree. A practical application that analyses countries' income group classification based on world development indicators presents an interpretation of the feature selection approach.

2 Evolution Strategy Decision Forest (ESDF)

ESDF evolves individuals representing feature weights to identify those that most explain the data. The evolution strategy mechanism, as well as the decision trees used for classification, are presented in what follows.

Decision Trees and Random Forests. Decision trees were some of the most popular machine learning techniques [19,25] due to their efficiency and explainability. They recursively split the data space into separate regions aiming to find areas as pure as possible. Large trees tend to overfit the data, and small trees may not split it enough. One way to overcome these drawbacks is to use ensembles of trees, e.g. in the form of random forests [4], and aggregate their results in some form. Decision trees can also be used to assess feature importances based on the tree structure, and the purity of split data in each node [24].

In what follows, we consider the binary classification problem: given a data set $\mathcal{X} \subset \mathbb{R}^{N \times d}$ containing N instances $x_i \in \mathbb{R}^d$, $i = 1, \ldots, N$ and d and $\mathcal{Y}$ their corresponding labels, with $y_i \in \{0, 1\}$ the label of x_i, the goal is to find a rule that best predicts the labels $\hat{y}$ for instances x that come from the same distribution as $\mathcal{X}$.

Equilibrium-Based Decision Tree. Most decision trees are built top-down starting with the entire data set at the root level. Different trees split data in different manners, by using either axes parallel, oblique, or non-linear hyperplanes [1,12,16], computing their parameters by using some purity indicators that evaluate sub-nodes data, e.g. gini index, entropy, etc. [28]. At each node level, some optimisation process takes place involving either hyperplane parameters, the attributes to use for the split, or both.

In this paper, we propose the use of a decision tree that computes hyperplane parameters by approximating the equilibrium of a non-cooperative game [22]. The equilibrium of the game aims to find parameters such that each sub-node 'receives' data as pure as possible by shifting instances with different labels to the left/right of the hyperplane. Thus, in order to split node data X, Y based on an attribute j, we use the following non-cooperative game $\Gamma(X, Y|j)$:

- the players, L and R correspond to the two sub-nodes and the two classes, respectively;
- the strategy of each player is to choose a hyperplane parameter: β_L and β_R, respectively;
- the payoff of each player is computed in the following manner:

$$u_L(\beta_L, \beta_R|j) = -n_0 \sum_{i=1}^{n} (\beta_{1|j} x_{ij} + \beta_{0|j})(1 - y_i),$$

and

$$u_R(\beta_L, \beta_R|j) = n_1 \sum_{i=1}^{n} (\beta_{1|j} x_{ij} + \beta_{0|j}) y_i,$$

where

$$\beta = \frac{1}{2}(\beta_L + \beta_R)$$

and n_0 and n_1 represent number of instances having labels 0 and 1, respectively.

The concept of Nash equilibrium for this game represents a solution such that none of the players can find a unilateral deviation that would improve its payoff, i.e., none of the players can shift data more to obtain a better payoff. An approximation of an equilibrium can be obtained by using a stylized version of fictitious play [5] in the following manner. For a number of iterations (η), the best response of each player against the strategy of the other player is computed using some optimization algorithm. As we only aim to approximate β values that reasonably split the data, the search stops after the number of iterations has elapsed. In each iteration, the best response to the average of the other player's strategies in the previous ones is considered the fixed one. The procedure is outlined in Algorithm 1.

For each attribute $j \in \{1, \ldots, d\}$, data is split using Algorithm 1; the attribute that is actually used to split the data is chosen based on the Gini index [28]. The game theoretic decision tree splits data in this manner, recursively, until node data becomes pure (all instances have the same label) or a maximum tree depth has been achieved.

Prediction. A DT provides a partition for the training data. To predict the label for a tested instance x, the corresponding region of the space, i.e., its leaf, is identified. The decision is made based on the proportion of labels in that leaf. Let DT be a decision tree based on a data set $\mathcal{X}$ and x a tested value. Then

Algorithm 1. Approximation of Nash equilibrium

Input: X, Y - data to be split by the node; j - attribute evaluated
Output: $X_{L|j}, y_{L|j}, X_{R|j}, y_{R|j}$, and β_j to define the split rule for the node based on attribute j;
Initialize β_L, β_R at random (standard normal distribution)
for η iterations: **do**
  Find $\beta_L = \underset{b}{\operatorname{argmin}}\, u_L(b, \beta_R)$;
  Find $\beta_R = \underset{b}{\operatorname{argmin}}\, u_R(\beta_L, b)$;
end for
$\beta_j = \frac{1}{2}(\beta_L + \beta_R)$
$X_{L|j} = \{x \in X | x_j^T \beta \leq 0\}$, $y_{L|j} = \{y_i \in y | x_i \in X_{L|j}\}$
$X_{R|j} = \{x \in X | x_j^T \beta > 0\}$, $y_R = \{y_i \in y | x_i \in X_{R|j}\}$

the decision tree DT has partitioned $\mathcal{X}$ into data found in its leaves, denoted by $DT_1, \ldots, DT_m$, where m is the number of leaves of DT. Let $DT(x)$ be the data set corresponding to the leaf region of x, $DT(x) \subset \mathcal{X}$. Typically, the model would assign to x label y with a probability equal to the proportion of elements with class y in $DT(x)$.

Feature Importance. The game-based splitting mechanism of the tree indicates for each node the attribute that 'best' splits the data. It is reasonable to assume that the position of the node in the tree indicates also the importance of the attribute in classifying the data and an importance measure can be derived based on the structure of the tree. Thus, for each feature $j \in \{1 \ldots d\}$ we denote by

$$\nu_j = \{\nu_{jl}\}_{l \in I_j},$$

the set containing the nodes that split data based on attribute j, with I_j the set of corresponding indexes in the tree and let $\delta(\nu_{jl})$ be the depth of the node ν_{jl} in the decision tree, with values starting at one at the root node. Then the importance $\phi(j)$ of attribute j can be computed as:

$$\phi(j) = \begin{cases} \sum\limits_{l \in I_j} \frac{1}{\delta(\nu_{jl})}, & I_j \neq \emptyset \\ 0, & I_j = \emptyset \end{cases}. \tag{1}$$

The formula (1) is based on the assumption that attributes that split data at the first levels of the tree may be more influential. Also, multiple appearances of an attribute in nodes with higher depths may indicate its importance and are counted in $\phi()$.

Evolution Strategy Decision Forest. The Evolution Strategy decision forest evolves a population of feature weights in order to identify their importance for classification. Individuals in the final population indicate feature importances while, overall, the evolution strategy performs classification using the evolved feature weights.

Encoding. Individuals w are encoded as real, positive valued vectors of length d, where w_j represents the importance of feature j, $j = 1, \ldots, d$.

Initialization. All individuals are initialized with equal weights of $1/d$. ESDF maintains a population of *pop_size* individuals.

Evaluation. There is no explicit fitness assignment mechanism within ESDF. Individuals are evolved regardless of their performance based on the information received from the environment. The motivation behind this approach can be expressed in two ways: on the one hand, the evaluation of feature weights may be performed using some classification algorithm based on its performance. However, there is no universally accepted performance indicator that can be used to compare results in a reliable manner. From a nature-inspired point of view, on the other hand, a forest paradigm does not require direct competition for resources. Trees grow in forests and adapt to each other. Some may stop growing due to a lack of resources, but they do not replace each other every generation. Thus, all trees are added to the forest and evaluation takes place on the entire forest at the end of the search.

Evolution. The evolution process takes place iteratively until a maximum number of generations is reached, or until all individuals have achieved maturity.

Updating Mechanism. In each iteration, a bootstrapped sample from the data is used to induct a game theoretic decision tree for each individual in the population. Attributes for each tree are selected with a probability proportional to their corresponding weights in the individual. Thus, for individual w representing feature weights, the probability to select feature j is:

$$P(j|w) = \frac{w_j}{\sum_{k=1}^{d} w_k}. \tag{2}$$

In the first iteration, probabilities are all equal. However, in subsequent generations, feature importances reported by each tree are used to update the corresponding individuals:

$$w_j \leftarrow w_j + \phi(j) \cdot \alpha, j = 1, \ldots, d, \tag{3}$$

where $\phi(j)$ represents the importance of feature j reported by the tree inducted by using individual w (Eq. (1)), and α is a parameter controlling the magnitude of the update.

In this manner, the weights of attributes that are deemed important by the tree are increased, also increasing the probability that they are selected in the next iterations. While apparently, this may lead to overfitting features, the fact that in each iteration, a different sample from the data is used for inducting trees, that the search of an individual stops when it reaches maturity, and also that there are several individuals maintained on the same data ensures diversity preservation.

Thus, the role of the tree is to assign feature importances for the updating mechanism. Apart from that, each tree is also preserved and further used in prediction for classification.

Maturity. Individuals are used to select attributes for training using probabilities in Eq. (2). The goal of the search is to find a distribution over the feature set: if several iterations of applying Eq. (3) feature importances reported by the tree do not change significantly, the standard deviation of the probabilities $P()$ will not vary. ESDF considers that an individual has reached maturity and stops evolving and inducting trees using it if there is no change in the variation of the corresponding probabilities. The following condition is used to compare the evolution of an individual from generation t to $t+1$:

$$\frac{\sigma(P(\cdot|w_t)}{\sigma(P(\cdot|w_{t+1})} < \epsilon \tag{4}$$

where σ denotes the standard deviation and $P(\cdot|w)$ the vector of probabilities in Eq. (2) taken for all attributes j. If condition 4 holds, the individual is considered to have reached maturity and is no longer updated,i.e., and no more trees are inducted based on him. Not all individuals reach maturity at the same time, which means that the size of the population decreases during the search, reducing the complexity of the method.

Classification. Each individual inducts several trees until reaching maturity. All these trees form a forest that can be used to make predictions for the classification problem. This is the last step of the algorithm, and it can be used to validate results. Trees are inducted by using different data samples and different attributes. Selecting attributes without any fitness measure may provide (or not) good classification trees. To avoid overfitting or using misleading trees, prediction is not made by considering labels in the trees' leaves but by aggregating data from the leaves corresponding to tested instances and further applying logistic regression (LR) to make predictions. Each tree offers a neighborhood for the tested instance. Aggregating all these regions will provide a set of relevant instances, allowing the algorithm to make an informed prediction.

Outline of ESDF. ESDF has two main steps: an evolution step (Algorithm 2, line 6) and a prediction step (Algorithm 2, line 17).

During the evolution step, a population of weights is updated several iterations until there is no variation in the probabilities they provide for selecting attributes for tree induction. Prediction is performed for each tested instance by aggregating data corresponding to its leaves in all inducted trees and applying logistic regression on the resulting data set.

The output of ESDF consists of prediction probabilities for the tested data that can be used to evaluate the entire approach and the average feature weights over the entire population.

Algorithm 2. ES-DF: Evolution Strategy Decision Forest

input: training set $\mathcal{X}, \mathcal{Y}$,
parameters:: - *pop_size* - population size;
- p - the proportion of attributes used for a tree;
- μ - maximum tree depth;
- $MaxGen$ - maximum number of generations;
output: predictions C for a (test) set T; Feature weights ω;
t = 0;
Initialize population W_0 with $w_{0,ij} = 1/d$, $i = 1, \ldots, pop\_size$, $j = 1, \ldots, d$;
Step 1: Evolution
while $t < MaxGen$ **or not** all trees have reached maturity **do**
  $X_t \leftarrow$ sample of size N with replacement from $\mathcal{X}$;
  for each individual w_t **do**
    $\overline{X}_{t,w} \leftarrow$ sample proportion p of attributes from X_t using probabilities P in Eq. (2);
    $DT_{t,w} \leftarrow$ game based decision tree based on $\overline{X}_{t,w}, \mu$;
    Update w_{t+1} using Eq. (3);
    Check maturity using condition (4); if (4) holds, mark individual as mature and stop its update;
  end for
  $t \leftarrow t + 1$;
end while
Step 2: Prediction
for each $x_t \in T$ **do**
  $RF(x_t) = \cup_{w,t} DT(x_t)$;
  Fit LR on $RF(x_t)$;
  Assign c_t to x_t - probability that x_t has class 1, based on LR;
end for
return
- $C = (c_1, c_2, \ldots, c(x_{|T|^1}))$
- ω - average feature weights over the entire population.

[1] $|\cdot|$ denotes the cardinality of a set.

3 Numerical Experiments

Numerical experiments are used to test and illustrate the performance of ESDF and compare its results to other state-of-the-art classification models. This section is divided into two main parts: the first one presents results obtained on synthetic and real-world benchmarks with various degrees of difficulty used for classification, and the second part is a real data application involving the classification of countries' income groups.

Synthetic and Real-World Benchmarks
For synthetically generated test data, to ensure reproducibility and control the difficulty of the resulting data set, we use the `make_classification` function from the `scikit-learn`[1] Python library [18]. The degree of difficulty is con-

[1] version `1.1.1`.

trolled by the generating function parameters: number of instances, number of attributes/features, degree of overlap between instances of different classes, the seed used to generate the test data, and class imbalance. For our experiments, we use the following: number of instances $(100, 200, 500, 1000, 2500)$, number of attributes $(20, 50)$, the seed used to generate the data (500), degree of overlap between instances of different classes $(0.1, 0.5)$, and all data sets generated are balanced. We generate test data sets for all combinations of the above parameters. In order to evaluate the feature selection mechanism, only half of the features in each data set are generated using the `make_classification` function, and the other half at random following a uniform distribution.

For real-world benchmarks, we use the following data sets from the UCI Machine Learning Repository [7]: iris data set (R1) from which we removed the *setosa* instances to obtain a linear non-separable binary classification problem, Pima Indians Diabetes (R2), Connectionist Bench (Sonar, Mines vs. Rocks) (R3), acute inflamations (R4), heart disease (R5), Somerville Happiness Survey (R6), appendicitis (R7), blogger (R8), bupa (R9), monks (R10), thoracic-surgery(R11), vertebra-column-2c (R12), wholesale-channel (R13), and the wdbc (R14) data set.

Experimental Set-Up. A *Stratified k-fold Cross-Validation* strategy [11] is used to estimate the expected prediction error. The data set is divided into $k = 10$ balanced folds, of which nine are used to train the model, and the tenth fold (the test fold) is used to evaluate the model. The train and test part are repeated $k = 10$ times, each time a different fold is used as a test fold. We repeat the k-fold cross-validation four times, each time a different seed is used to split the data (we use as seed the values $1, 2, 3, 4$), resulting in 40 indicator values that are compared.

For each test fold of a data set, we report performance metrics based on which we compare the performance of ESDF with other state-of-the-art classifiers. We train each compared classifier on the same train data as ES-DF for each fold and compare the results of ES-DF to those reported by the compared models on the test fold.

The performance metrics used for comparison are: AUC (area under the ROC curve) [8,20], the F_1 score [29], the accuracy ACC and the log-loss score [11].

ESDF results are compared to other decision tree-based classifiers, and because it uses Logistic Regression in the prediction step, we also compare results with this method. We also compare the performance of ES-DF to other well-known classifiers. The list of compared classifiers is: M0 - Support Vector Machine with a linear kernel, M1 - Support Vector Machine with a radial kernel, M2 - $k-$nearest-neighbour classifier with $k = 3$, M3 - AdaBoost classifier, M4 - Gaussian Naive Bayes, M5 - stochastic gradient descent, M6 - Gaussian process classification, M7 - decision tree classifier which splits nodes until its leaves contain only instances of one class, M8 - a decision tree with maximum depth equal to that of ESDF, M9 - a random forest classifier for which each estimator splits nodes until its leaves are pure, M10 - a random forest classifier with 10 estimators, M11 - a random forest classifier with 50 estimators, M12 - a random

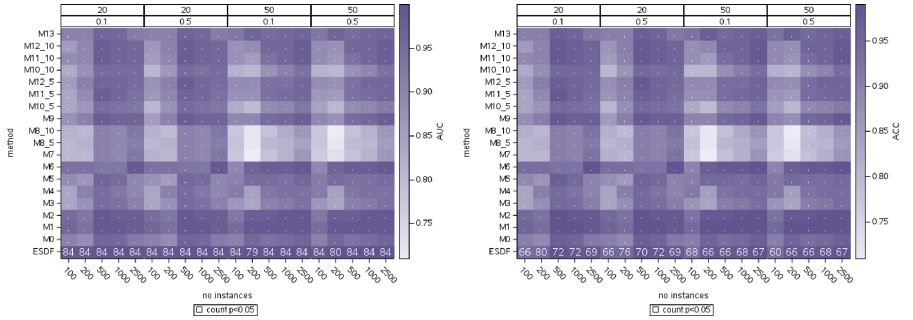

Fig. 1. Heatmaps for the AUC and ACC indicators for the synthetic data sets. A higher (darker) value is desirable. The last line also presents the number of times ESDF results were significantly better than the other method based on the p values of the t test comparing results reported for all folds (out of 120). The first line of the heading indicates the number of attributes and the second line the class separator for the data sets.

forest classifier with 100 estimators (for M10, M11 and M12 each estimator has a maximum depth equal to ESDF), and M13 - logistic regression classifier. For reproducibility and control, we use their implementation from the *scikit-learn* software library [18].

For each data set, each method reports 40 performance indicator values corresponding to the ten folds generated four times with different random number seed generators. To assess the difference in results, these values are compared using a paired t-test, with the null hypothesis that results provided by ESDF are worse than those reported by the other method. Rejecting this hypothesis, with a p-value smaller than 0.05, indicates that we can consider ESDF results significantly better than the others.

ESDF tested parameters are: population size 5, maximum number of generations 20, maximum tree depth 5 and 10, and α values 0.3, 0.8, and 1. All experiments are run with combination of these parameters.

Results. Figures 1 and 2 present the performance indicators obtained by all methods on the synthetic data sets. Figure 1 presents the AUC and accuracy indicators, and Fig. 2 the F1 and Log-loss indicators. A square in the heatmap shows the value obtained by a classifier for a specific set of parameters used to generate the data set (no instances - $100, 200, 500, 1000, 2500$, no attributes - $20, 50$, and overlap degree - $0.1, 0.5$). Each row presents results obtained by a different classifier. For our approach, we also report the number of cases in which ESDF obtains significantly better results than the compared methods according to a t-test. In the case of AUC, ESDF obtains the best results. for all data sets. For the accuracy indicator, ESDF consistently gives better results, and when more instances are available in the data set, the ensemble and logistic regression classifiers report, for a few data sets, results as good as the ones reported by ESDF. This is also the case for the F1 and Log-loss indicators (Fig. 2).

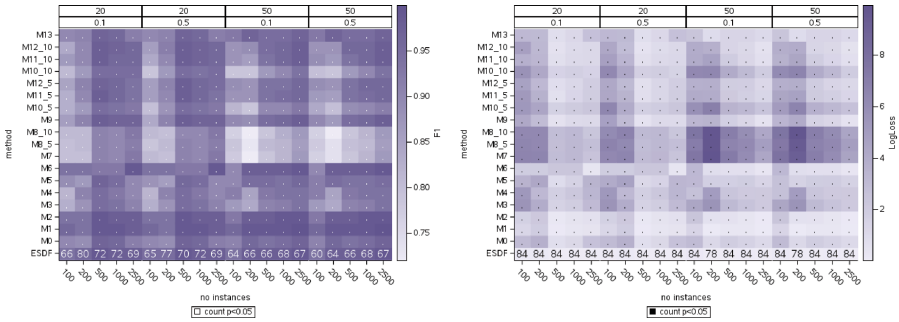

Fig. 2. Heatmaps for the F1 and LogLoss indicators for the synthetic data sets. For F1, a higher (darker) value is desirable, while for the LogLoss a smaller (lighter) value is better. The last line also presents the number of times ESDF results were significantly better than the other method based on the p values of the t test comparing results reported for all folds (out of 120). The first line in the heading indicates the number of attributes and the second line the class separator for the data sets.

Feature Selection. One possible manner to evaluate the efficiency of the feature selection mechanism is to compute the stability indicator SC [3,17] over the ten folds. Values of the stability indicator are based on average correlations: an SC close to 1 indicates that the feature selection method selects the same features in several runs on different samples of the data set, indicating stability. The values of the SC score when selecting half of the features based on their weights for the synthetic data sets vary between 0.7 and 1, indicating the stability of the approach. For data sets with 20 attributes the confidence interval for SC is (0.97, 0.99), and for those with 50 attributes is (0.88, 0.93).

Table 1 presents the results obtained by ESDF against the best result reported by the compared methods on the real-world data sets. The mean and standard deviation for the AUC and Log-loss indicators are presented. The statistically better results are highlighted. It can be seen that ESDF consistently gives better results. When comparing AUC values it can be seen that ESDF either gives statistically better results or is indifferent when compared to the best performing compared classifier.

Regarding different ESDF parameter settings, we found no significant differences among different values, either for synthetic or real-world benchmarks. The value of α does not influence the results because it is not directly used for the induction of trees.

Classification of low-income countries based on world development indicators: an application

The World Bank classifies countries into five income groups: high income, upper middle income, lower middle income, and low income yearly, based on gross national income (GNI) per capita in USD values, using the Atlas methodology[2]. The classification list for 2022 is based on 2021 data. In 2022, the GNI

[2] https://datahelpdesk.worldbank.org/knowledgebase/articles/378832-what-is-the-world-bank-atlas-method, last accessed January 2023.

Table 1. Mean and standard deviation for the AUC and Log-loss indicators in the case of real-world data sets for ESDF and the best performing compared classifier (best M). A ($\star$) symbol highlights the ESDF results that can be considered statistically better than the other method.

data set	AUC (ESDF)	AUC (best M)	Log-loss (ESDF)	Log-loss (best M)
R1	$0.99 \pm 0.03\star$	M0: 0.95 ± 0.06	$0.15 \pm 0.14\star$	M3: 0.89 ± 0.08
R2	$0.83 \pm 0.05\star$	M6: 0.73 ± 0.05	0.65 ± 0.17	M5: 0.69 ± 0.06
R3	$0.92 \pm 0.06\star$	M2: 0.86 ± 0.08	$0.56 \pm 0.35\star$	M8: 0.72 ± 0.10
R4	1.00 ± 0.00	M0: 1.00 ± 0.00	$0.00 \pm 0.00\star$	M4: 0.83 ± 0.08
R5	$0.88 \pm 0.07\star$	M4: 0.84 ± 0.07	0.73 ± 0.53	M7: 0.72 ± 0.07
R6	$0.66 \pm 0.13\star$	M11: 0.62 ± 0.12	0.94 ± 0.30	M5: $0.53 \pm 0.10\star$
R7	$0.83 \pm 0.16\star$	M3: 0.79 ± 0.16	1.34 ± 1.64	M8: $0.69 \pm 0.18\star$
R8	$0.92 \pm 0.11\star$	M1: 0.82 ± 0.14	$0.62 \pm 1.05\star$	M3: 0.64 ± 0.15
R9	$0.77 \pm 0.08\star$	M9: 0.72 ± 0.08	$0.66 \pm 0.14\star$	M3: 0.70 ± 0.07
R10	1.00 ± 0.01	M8: 0.99 ± 0.02	$0.13 \pm 0.05\star$	M13: 0.76 ± 0.05
R11	$0.63 \pm 0.09\star$	M7: 0.56 ± 0.09	0.64 ± 0.22	M7: $0.56 \pm 0.09\star$
R12	$0.94 \pm 0.04\star$	M6: 0.84 ± 0.07	$0.39 \pm 0.25\star$	M4: 0.80 ± 0.06
R13	$0.96 \pm 0.03\star$	M9: 0.91 ± 0.05	$0.36 \pm 0.28\star$	M8: 0.85 ± 0.05
R14	$1.00 \pm 0.01\star$	M1: 0.98 ± 0.02	$0.12 \pm 0.20\star$	M7: 0.92 ± 0.04

per capita is influenced by factors such as economic growth, inflation, exchange rates, and population growth. The classification is based on GNI intervals[3]. The World Bank also offers data related to a variety of other indicators. The World Development Indicator data-set contains information regarding various financial indicators that may be used to explain a country's income group classification. To test this assumption, as well as the efficiency of ESDF on a real-world application, we used these data to classify low (low and low-middle) income countries and identify features in the world development indicators list that most explain the classification.

Data Processing. The world development indicators data set (for the year 2021) contains 108 indicators for 218 countries for which an income category is also assigned. However, not all indicators have values for all countries. All indicators with values for less than half the number of countries were removed, resulting in a data set with 218 countries and 40 indicators. Further, removing all countries with less than half indicator values resulted in a data set containing 138 countries and 40 indicators. In this data set, we found 13.35% missing values that were replaced, for each indicator, with the average value of its country's region, which is part of the data set. Countries with lower and lower middle income were assigned the label 1, and the others 0, resulting in a slightly imbalanced data

[3] https://blogs.worldbank.org/opendata/new-world-bank-country-classifications-income-level-2022-2023, accessed Jan. 2023.

set with 37% instances having class 1. In what follows, we will call this data set the World Bank Income indicators (WBII) data set.

Experimental Set-Up. The same methodology used to test ESDF on the synthetic and real-world benchmarks was also used for the WBII data set. 10-fold cross-validation was applied four times with different seeds for the random number generator, and the four indicators were used to evaluate the results. ESDF parameters were $\alpha = 0.8$, the maximum tree depth used was 10, and the population size was 5.

Results - Classification. Numerical results for classification reported by all methods for the WBII data set are presented in Table 2. We find that results reported by ESDF are as good as or even better than those reported by the other methods. Particularly the Log Loss values are significantly better than all the other methods.

Table 2. WBII data set: mean and standard deviation values for the four indicators reported by all methods. We find results reported by ESDF better or as good as the others for all indicator values.

Method	AUC	ACC	F1	Log-loss
ESDF	0.90 ± 0.08	0.85 ± 0.09	0.79 ± 0.12	0.99 ± 1.28
M0	0.78 ± 0.11	0.80 ± 0.11	0.72 ± 0.16	7.02 ± 3.67
M1	0.85 ± 0.09	0.86 ± 0.08	0.81 ± 0.12	4.77 ± 2.93
M2	0.81 ± 0.09	0.84 ± 0.08	0.75 ± 0.13	5.40 ± 2.71
M3	0.81 ± 0.10	0.83 ± 0.10	0.76 ± 0.16	5.71 ± 3.32
M4	0.78 ± 0.11	0.76 ± 0.11	0.72 ± 0.13	8.19 ± 3.75
M5	0.75 ± 0.12	0.76 ± 0.11	0.67 ± 0.17	8.20 ± 3.97
M6	0.84 ± 0.10	0.85 ± 0.10	0.79 ± 0.13	5.20 ± 3.30
M7	0.81 ± 0.08	0.83 ± 0.08	0.76 ± 0.12	6.06 ± 2.73
M8_5	0.80 ± 0.08	0.82 ± 0.07	0.74 ± 0.11	6.38 ± 2.62
M8_10	0.79 ± 0.09	0.81 ± 0.08	0.72 ± 0.13	6.90 ± 2.91
M9	0.86 ± 0.09	0.87 ± 0.08	0.82 ± 0.11	4.58 ± 2.81
M10_5	0.85 ± 0.10	0.87 ± 0.09	0.81 ± 0.13	4.76 ± 3.12
M11_5	0.87 ± 0.09	0.88 ± 0.08	0.83 ± 0.11	4.37 ± 2.93
M12_5	0.87 ± 0.09	0.88 ± 0.08	0.83 ± 0.11	4.44 ± 2.80
M10_10	0.84 ± 0.09	0.86 ± 0.08	0.79 ± 0.12	5.16 ± 2.82
M11_10	0.86 ± 0.08	0.88 ± 0.08	0.83 ± 0.11	4.44 ± 2.80
M12_10	0.86 ± 0.08	0.88 ± 0.08	0.83 ± 0.11	4.44 ± 2.75
M13	0.81 ± 0.10	0.82 ± 0.10	0.75 ± 0.14	6.41 ± 3.54

Results - Feature Selection. To illustrate a possible practical interpretation of the selected features, Fig. 3 represents feature weights reported by ESDF on

the 10 folds used for cross-validation. The corresponding stability score is 0.74 indicating a strong correlation between selected features (when half of them are chosen based on the value of their weights). The features with the highest weights are:

1. GFDD.AI.11: Received wages: into a financial institution account (% age 15+)
2. GFDD.AI.05: Financial institution account (% age 15+)
3. GFDD.AI.21: Debit card ownership (% age 15+) and GFDD.AI.20: Credit card ownership (% age 15+)
4. GFDD.EI.01: Bank net interest margin (%)
5. GFDD.AI.06: Saved at a financial institution (% age 15+)
6. GFDD.AI.10: Received domestic remittances: through a financial institution (% age 15+)

This list indicates that individual banking activities may be considered as indicators for a country's income group. While there is no causation involved here, results indicate a relationship between these indicators and the income group.

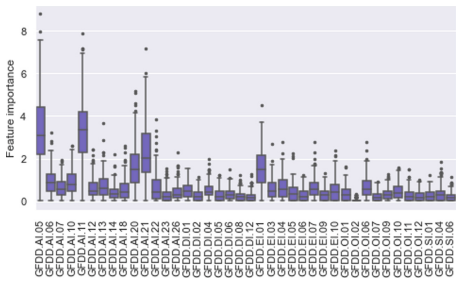

Fig. 3. Example of distribution of feature weights for one run on the WBII data set.

4 Conclusions and Further Work

The evolution strategy random forest for feature selection and classification proposed in this paper presents several original aspects: individuals are evolved without an explicit fitness; an updating mechanism, imitating mutation, always increases each component; converting values to probabilities when necessary decreases the additive effect; the search stops for each individual when there is no more variation in the probability values; the evaluation takes place at the end of the search, during the prediction phase for classification when data is gathered from the leaves in which tested instances are found from all trees, and logistic regression (but any classification method can be used) is applied for prediction.

Numerical experiments performed on synthetic and real-world benchmarks illustrate the efficiency of the approach for classification compared to other standard methods. A stability measure for feature selection indicates the potential of the approach to identify relevant feature sets. Furthermore, the method is used to analyse the classification of low-income countries based on several world development indicators. Classification results and most popular features are discussed.

Acknowledgements. This work was supported by a grant of the Ministry of Research, Innovation and Digitization, CNCS - UEFISCDI, project number PN-III-P1-1.1-TE-2021-1374, within PNCDI III.

References

1. Aich, S., Younga, K., Hui, K.L., Al-Absi, A.A., Sain, M.: A nonlinear decision tree based classification approach to predict the Parkinson's disease using different feature sets of voice data. In: 2018 20th International Conference on Advanced Communication Technology (ICACT), pp. 638–642 (2018)
2. Bala, J., Huang, J., Vafaie, H., Dejong, K., Wechsler, H.: Hybrid learning using genetic algorithms and decision trees for pattern classification. In: Proceedings of the 14th International Joint Conference on Artificial Intelligence, IJCAI 1995, vol. 1, p. 719–724. Morgan Kaufmann Publishers Inc., San Francisco (1995)
3. Bommert, A., Sun, X., Bischl, B., Rahnenführer, J., Lang, M.: Benchmark for filter methods for feature selection in high-dimensional classification data. Comput. Stat. Data Anal. **143**, 106839 (2020). https://doi.org/10.1016/j.csda.2019.106839
4. Breiman, L.: Random Forests. Mach. Learn. **45**(1), 5–32 (2001)
5. Brown, G.W.: Iterative solution of games by fictitious play. Act. Anal. Prod. Allocation **13**(1), 374–376 (1951)
6. Cai, J., Luo, J., Wang, S., Yang, S.: Feature selection in machine learning: a new perspective. Neurocomputing **300**, 70–79 (2018)
7. Dua, D., Graff, C.: UCI machine learning repository (2017)
8. Fawcett, T.: An introduction to ROC analysis. Pattern Recogn. Lett. **27**(8), 861–874 (2006). https://doi.org/10.1016/j.patrec.2005.10.010
9. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning, 1st edn. Addison-Wesley Longman Publishing Co., Inc., USA (1989)
10. Hansen, L., Lee, E.A., Hestir, K., Williams, L.T., Farrelly, D.: Controlling feature selection in random forests of decision trees using a genetic algorithm: classification of class I MHC peptides. Combin. Chem. High Throughput Screen. **12**(5), 514–519 (2009). https://doi.org/10.2174/138620709788488984
11. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference and Prediction, 2nd edn. Springer, Heidelberg (2009). https://doi.org/10.1007/978-0-387-84858-7
12. Irsoy, O., Yıldız, O.T., Alpaydın, E.: Soft decision trees. In: Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012), pp. 1819–1822. IEEE (2012)
13. Jovanovic, M., Delibasic, B., Vukicevic, M., Suknović, M., Martic, M.: Evolutionary approach for automated component-based decision tree algorithm design. Intell. Data Anal. (2014). https://doi.org/10.3233/ida-130628

14. Krętowski, M., Grześ, M.: Evolutionary learning of linear trees with embedded feature selection. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Żurada, J.M. (eds.) ICAISC 2006. LNCS (LNAI), vol. 4029, pp. 400–409. Springer, Heidelberg (2006). https://doi.org/10.1007/11785231_43
15. Mao, Q., Wang, X., Zhan, Y.: Speech emotion recognition method based on improved decision tree and layered feature selection. Int. J. Humanoid Rob. (2010). https://doi.org/10.1142/s0219843610002088
16. Murthy, S.K., Kasif, S., Salzberg, S.: A system for induction of oblique decision trees. J. Artif. Intell. Res. **2**, 1–32 (1994)
17. Nogueira, S., Brown, G.: Measuring the stability of feature selection. In: Frasconi, P., Landwehr, N., Manco, G., Vreeken, J. (eds.) ECML PKDD 2016. LNCS (LNAI), vol. 9852, pp. 442–457. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46227-1_28
18. Pedregosa, F., et al.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
19. Quinlan, J.R.: Induction of decision trees. Mach. Learn. (1986). https://doi.org/10.1007/bf00116251
20. Rosset, S.: Model selection via the AUC. In: Proceedings of the Twenty-First International Conference on Machine Learning, ICML 2004, p. 89. Association for Computing Machinery, New York (2004). https://doi.org/10.1145/1015330.1015400
21. Stein, G., Chen, B., Wu, A.S., Hua, K.A.: Decision tree classifier for network intrusion detection with GA-based feature selection. In: Proceedings of the 43rd Annual Southeast Regional Conference, vol. 2, p. 136–141. ACM-SE 43, Association for Computing Machinery, New York (2005). https://doi.org/10.1145/1167253.1167288
22. Suciu, M.A., Lung, R.: A new filter feature selection method based on a game theoretic decision tree. In: Abraham, A., Hong, T.P., Kotecha, K., Ma, K., Manghirmalani Mishra, P., Gandhi, N. (eds.) HIS 2022. LNNS, vol. 647, pp. 556–565. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-27409-1_50
23. Vafaie, H., De Jong, K.: Genetic algorithms as a tool for feature selection in machine learning. In: Proceedings Fourth International Conference on Tools with Artificial Intelligence, TAI 1992, pp. 200–203 (1992). https://doi.org/10.1109/TAI.1992.246402
24. Wang, S., Tang, J., Liu, H.: Embedded unsupervised feature selection. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 29, no. 1 (2015)
25. Wu, X., et al.: Top 10 algorithms in data mining. Knowl. Inf. Syst. **14**(1), 1–37 (2008). https://doi.org/10.1007/s10115-007-0114-2
26. Xue, B., Cervante, L., Shang, L., Browne, W.N., Zhang, M.: Multi-objective evolutionary algorithms for filter based feature selection in classification. Int. J. Artif. Intell. Tools **22**(04), 1350024 (2013)
27. Xue, B., Zhang, M., Browne, W.N., Yao, X.: A survey on evolutionary computation approaches to feature selection. IEEE Trans. Evol. Comput. **20**(4), 606–626 (2016). https://doi.org/10.1109/TEVC.2015.2504420
28. Zaki, M.J., Meira, W., Jr.: Data Mining and Machine Learning: Fundamental Concepts and Algorithms, 2nd edn. Cambridge University Press, Cambridge (2020)
29. Zijdenbos, A., Dawant, B., Margolin, R., Palmer, A.: Morphometric analysis of white matter lesions in MR images: method and validation. IEEE Trans. Med. Imaging **13**(4), 716–724 (1994). https://doi.org/10.1109/42.363096