Neurocomputing 354 (2019) 20–26

Contents lists available at ScienceDirect

Neurocomputing

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

Hybrid methodology based on Bayesian optimization and GA-PARSIMONY to search for parsimony models by combining hyperparameter optimization and feature selection

F.J. Martinez-de-Pison*, R. Gonzalez-Sendino, A. Aldama, J. Ferreiro-Cabello, E. Fraile-Garcia

EDMANS Group, Department of Mechanical Engineering, University of La Rioja, Logroño, Spain

ARTICLE INFO

Article history:
Received 19 November 2017
Revised 24 April 2018
Accepted 1 May 2018
Available online 24 April 2019

Keywords:
GA-PARSIMONY
Bayesian optimization
Hyperparameter optimization
Parsimonious models
Genetic algorithms

ABSTRACT

This article presents a hybrid methodology that combines Bayesian optimization (BO) with a constrained version of the GA-PARSIMONY method to obtain parsimony models. The proposal is designed to reduce the sizeable computational effort associated with the use of GA-PARSIMONY alone. The method begins with BO to obtain favorable initial model parameters. Then, with these parameters, a constrained GA-PARSIMONY is implemented to generate accurate parsimony models by using feature reduction, data transformation and parsimonious model selection. Experiments with extreme gradient boosting machines (XGBoost) and ten UCI databases demonstrated that the hybrid methodology obtains models analogous to those of GA-PARSIMONY while achieving significant reductions in elapsed time in eight out of ten datasets.

1. Introduction

Hyperparameter optimization (HO) is extremely important for finding accurate models. In addition, feature selection (FS) is useful for determining the least complex models among solutions with similar accuracy. Thus, the least complex model (the most parsimonious) among several accurate models is usually more robust against perturbations or noise, and easier to maintain and understand [3,19].

In recent years, interest in reducing the human effort involved in HO and FS has grown due to the fact that these tasks are time-consuming and quite tedious [11,17]. The latest learning methods such as deep learning (DL) or gradient boosting machines (GBM) have up to a dozen tuning parameters, also known as hyperparameters, which hinder the use of traditional optimization methods such as a grid or random search. Therefore, companies are demanding new methodologies to automate these processes because they prefer to invest their time and energy in other critical tasks such as data transformation (DT) or feature engineering (FE) which are more difficult to automate [14]. This new paradigm is referred to as 'automated machine learning' (AML) and there is a growing research community [18] interested in it, along with companies such as Google with 'Cloud Automl' or DataRobot that are researching this field.

New libraries are emerging to perform HO with Bayesian optimization (BO) like *mlrMBO* [4] and *rBayesianOptimization* in R [30], or *bayesian-optimization* [24] in Python. In addition, there are other tools focused on the optimization of more machine learning (ML) stages such as DT, dimensional reduction (DR), FE, or model selection (MS). For example, the MATLAB *SUMO-Toolbox* [13] adopts different plugins for each of the different stages. They can be optimized with other 'meta' plugins available in the toolbox. The *Auto-WEKA* [38] suite also combines HO with BO and MS with classification and regression algorithms that are implemented in WEKA. Given a specific dataset, *Auto-WEKA* makes HO for many algorithms and offers the user a recommendation regarding which method will likely perform a better generalization. *TPOT* [25] is another library in Python that automatically searches thousands of machine learning pipelines created with genetic programming. These pipelines involve tasks such as FS, feature preprocessing (FP), FE, MS and HO. At the end of the process, *TPOT* provides the Python code with the best pipeline. Similar to *Auto-WEKA*, *Auto-sklearn* [11] automates with BO a ML framework which combines DT, FE and HO of many algorithms from the *scikit-learn* Python library. However, *Auto-sklearn* includes a previous meta-learning step to suggest some instantiations for the ML framework. The decision is based on 38 meta-features related to the performance achieved by many ML algorithms from 140 datasets with different sizes and

* Corresponding author.
E-mail address: fjmartin@unirioja.es (F.J. Martinez-de-Pison).

https://doi.org/10.1016/j.neucom.2018.05.136
0925-2312/

feature characteristics. Furthermore, the library constructs an automated ensemble with the best non-correlated base meta-models. *auto_ml* [26] provides a complete framework for applying AML to a dataset. It includes DL and GBM, and methods for DT, FE, and so on.

Among the different existing methods, BO is currently the most popular optimization method implemented in these tools. However, soft computing (SC) seems to be an effective approach for reducing computational costs and improving model accuracy [5,7,15,32,45]. Thus, there is an increasing number of studies reporting SC strategies that combine FS and HO applied to multiple fields. For example, Ma and Xia [20] use a tribe competition-based GA for FS in pattern classification. Perez-Rodriguez et al. [27] improve model accuracy using evolutionary computation with a simultaneous feature weighting, FS and instance selection. With GA, Huang and Chang [16] optimize FS and HO tasks for seeking accurate SVM in micro-array classification. Ding [9] applies PSO in the search for hyper-spectral remote sensing images classifiers. Wei et al. [44] present a binary particle swarm optimization (BPSO) for FS and HO with SVM. Vieira et al. [41] predict survived or deceased patients with septic shock through a wrapper SVM approach optimized with a binary PSO. Wan et al. [42] present a binary ant colony optimization algorithm combined with GA for FS. Ahila et al. [1] use PSO to search for the best classifier of power system disturbances. Dhiman et al. [8] detect epileptic seizures from background electroencephalogram signals with a GA-SVM scheme for FS and HO. Wang et al. [43] report a chaotic moth-flame HO and FS optimization strategy in medical diagnoses. Medjahed et al. [22] apply gray wolf optimization for hyper-spectral images classification.

In this context, we propose GA-PARSIMONY [33], a genetic algorithm (GA) methodology whose main objective is to obtain accurate parsimonious models. It optimizes HO, DT, and FS with a new parsimonious model selection (PMS) process based on a double criteria that considers accuracy and complexity in two steps. GA-PARSIMONY has successfully been applied to obtain accurate parsimonious models with the most popular machine learning techniques such as support vector regression (SVR), random forest (RF) or artificial neural networks (ANNs); and in different fields as such as mechanical design [10], solar radiation forecasting [2], industrial processes [34], and hotel room demand estimation [40]. Additionally, a preliminary evaluation of the methodology was performed with extreme gradient boosting machines (XGBoost), high-dimensional databases and different complexity metrics [29].

GA-PARSIMONY performs well only with HO, but prior experiments demonstrated that including the number of features in the model complexity measurement helped obtain better parsimonious solutions when HO, FS, DT and PMS were included in the GA optimization process. From these experiences, an updated GA-PARSIMONY methodology has been recently published in [39]. Also, the `GAparsimony` [21] package for R is available with a General Public License since July 2017.

Despite the fact that this methodology has been successfully applied in several practical fields, it might be too computationally expensive when it is implemented with large and high-dimensional databases, even using parallel computing techniques. Therefore, the contribution of the present study is to present a new hybrid methodology that combines BO and a constrained version of GA-PARSIMONY with the objective of reducing the computational effort but obtaining parsimonious models similar in accuracy to those obtained with GA-PARSIMONY. The main idea is based on initiating the GA process with an adequate approximation of the model parameters that have been obtained beforehand with BO. Thus, subsequent GA optimization process can converge faster.

The rest of the paper is organized as follows: Section 2 presents a brief description of BO, GA-PARSIMONY and the new hybrid proposal. Section 3 describes the experiments performed with the three methods to obtain parsimonious XGBoost models in ten UCI datasets. In Section 4 an analysis of the experiment results is provided. Finally, Section 5 presents the discussion, conclusions and suggestions for further research.

2. Materials and methods

2.1. Extreme gradient boosting machines

The *eXtreme Gradient Boosting* (XGBoost) [6] algorithm is one of the most popular machine learning methods. This powerful method is based on gradient boosting machines (GBM) [12]. GBM uses a gradient-descent based algorithm that optimizes a differentiable loss function to create a boosting ensemble of weak prediction models. The main idea is to construct each new additive base-learner to be maximally correlated with the negative gradient of the loss function of the ensemble. However, XGBoost with tree-based learners is computationally more efficient and scalable than GBM. It incorporates more regularization strategies to reduce over-fitting and control model complexity, such us the limitation of the minimum loss reduction at each tree partition, the sum of instances weight per leaf or the depth of each tree. It also incorporates Lasso (L1) and Ridge (L2) penalties, similar to other machine learning methods. Moreover, it integrates 'random subspaces' and 'random subsampling' parameters to shrink the variance.

The high number of model parameters increases the computational effort of the tuning process. Besides, despite the fact that tree-based ensemble methods perform well with high-dimensional data, the inclusion of irrelevant or noisy features can degrade the accuracy of these models [28]. Therefore, there is an increasing interest in developing new SC methods to efficiently optimize HO and FS and obtain models with strong generalization capabilities.

2.2. Bayesian optimization

Since mid of 2000s, *Bayesian optimization* (BO) has emerged as an interesting alternative among other classic HO alternatives like random search or grid search [31]. BO uses Bayesian models based on *Gaussian processes* (GP) to formalize the relationship between model error/accuracy (y_n) with its parameters by means of a sequential design strategy. According to GP, any finite set of N points, where $\{\mathbf{x}_n \in \emptyset\}_{n=1}^{N}$, induces a multivariate Gaussian distribution on $\Re^n$. Then, GP defines a powerful prior distribution on functions f: $\emptyset \rightarrow \Re$ where the nth model performance is obtained from $f(\mathbf{x}_n)$ and the marginals and conditionals are calculated by the marginalization properties of the Gaussian distribution. These properties are determined by a predefined mean function m: $\chi \rightarrow \Re$ and a positive-definitive kernel or covariance function k: $\chi \times \chi \rightarrow \Re$.

From a practical point of view [36], BO starts with the evaluation of a small number of N models with a random set of parameters $\mathbf{x}_n$ where $y_n \sim \mathcal{N}(f(\mathbf{x}_n, v))$ is the nth measured model performance and v is the variance of function noise. Thus, considering that $f(\mathbf{x})$ is obtained from a prior Gaussian process and with the pre-computed experiments, a posterior over function $a(\mathbf{x})$ is induced. This function, denoted acquisition function, depends on the model through its predictive mean function $\mu(\mathbf{x}; \{\mathbf{x}_n, y_n\}, \theta)$ and predictive variance function $\sigma^2(\mathbf{x}; \{\mathbf{x}_n, y_n\}, \theta)$. Therefore, next point is evaluated by $\mathbf{x}_{next} = argmax_{\mathbf{x}}\, a(\mathbf{x})$ balancing the search of places with high variance (exploration) and places with low mean (exploitation).

Among the available acquisition functions [35], *GP Upper Confidence Bound* (GP-UCB) has demonstrated a strong performance in *hyperparameter tuning* [37]. This acquisition function can be expressed as:

$$a_{LCB} = \mu(\mathbf{x}) - \kappa\sigma(\mathbf{x}) \quad (1)$$

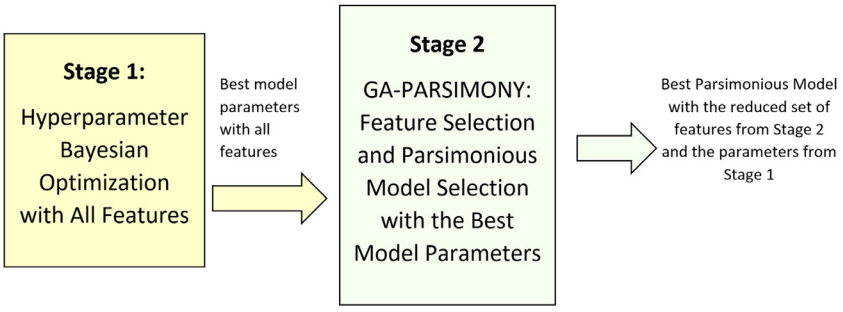

Fig. 1. Description of the hybrid methodology that combines BO and GA-PARSIMONY.

Table 1
Results obtained with BO, GA-PARSIMONY and the hybrid proposal. *SF* stands for the number of features of the best model, $RMSE_{tst}^{mean}$ is the mean testing error and *Time* the elapsed time in minutes. Best results for each database are depicted in bold.

Database			*Bayesian optim.*		*GA-PARSIMONY*				*Hybrid methodology*			
Name	# Inst	#FT	Time	$RMSE_{tst}^{mean}$	#Gen	#SF	Time	$RMSE_{tst}^{mean}$	#Gen	#SF	Time	$RMSE_{tst}^{mean}$
Ailerons	13750	40	295	0.0428	23	**13**	7949	0.0425	**14**	14	4221	**0.0420**
Bank	8192	32	104	0.0995	35	**18**	4036	**0.0980**	**13**	20	1533	0.0991
Blog	52397	276	1186	0.0155	13	**100**	5097	0.0148	**10**	108	3930	**0.0147**
Concrete	1030	8	152	0.0532	100	**7**	308	0.0521	**20**	8	272	**0.0519**
Cpu	8192	21	189	0.0232	**20**	**16**	4121	**0.0220**	26	**16**	4194	0.0231
Crime	2215	127	206	0.0612	100	**38**	1037	**0.0576**	**22**	40	626	**0.0576**
Elevators	16599	18	343	0.0322	39	**9**	16554	**0.0314**	**10**	12	2466	0.0319
Housing	506	13	136	0.0737	100	10	167	**0.0586**	**16**	**9**	191	0.0589
Pol	15000	26	176	0.0476	66	**16**	13203	**0.0400**	**17**	20	3231	0.0465
Puma	8192	32	209	0.0433	25	**4**	6168	0.0337	**13**	**4**	3337	**0.0336**

where κ balances exploration and exploitation. Also, *squared exponential kernel* (Eq. (2)) is often a default choice as covariance function for Gaussian process regression.

$$K_{SE}(\mathbf{x},\mathbf{x}') = \theta_0 \, exp\left\{\frac{1}{2}r^2(\mathbf{x},\mathbf{x}')\right\} \qquad r^2(\mathbf{x},\mathbf{x}') = \sum_{d=1}^{D}(x_d - x'_d)^2/\theta_d^2 \tag{2}$$

2.3. GA-PARSIMONY methodology

GA-PARSIMONY is an SC methodology based on genetic algorithms (GA) and designed to obtain precise overall parsimonious models automatically [33]. It includes HO, FS, and DT in the GA optimization process and has a flowchart similar to other classical GA methods. The main novelty of this methodology is the design of a *parsimonious model selection* (PMS) process arranged in two stages. First, the best models are sorted by their fitness function (J), which is an error or accuracy metric, and next, individuals with similar Js are rearranged based on their complexities. Models with less complexity are therefore promoted to the top positions of each generation. This choice of less complex solutions among those with similar accuracy fosters the generation of robust solutions with better generalization capabilities.

2.4. Hybrid method based on Bayesian optimization and GA-PARSIMONY

A hybrid method that combines BO and GA-PARSIMONY is presented here to reduce the computational costs associated with GA-PARSIMONY. The main idea is to use BO in the initial stage with all features to obtain the best model parameters. Next, GA-PARSIMONY with FS and PMS is used to find the best features of the parsimonious model with the fixed parameters obtained in the first step (Fig. 1).

3. Experiments

3.1. Datasets and validation process

The hybrid methodology with XGBoost was evaluated versus the use of either BO or GA- PARSIMONY alone. The experiments were conducted with ten UCI datasets (Table 1) that were divided into a validation set of 80% and a testing set of 20%, which was used to check the generalization capability of each model. The validation was based on the root mean squared error (*RMSE*) calculated as the mean of 5 runs of a 4-fold CV ($RMSE_{val}^{mean}$). This configuration showed, among different CV configurations, a beneficial trade-off between computational effort and generalization error estimation. All the datasets were normalized between 0 and 1.

3.2. GA-PARSIMONY settings

The GA settings were the following:

- The fitness function: $J = RMSE_{val}^{mean}$.
- $\alpha = 0.01$ was the maximum difference of J to consider individuals as having similar accuracy. Of these models, GA-PARSIMONY promotes those parsimonious solutions to top positions within the GA selection process.
- The population size was set to $P = 64$ with an elitism percentage of 25%.
- The selection method was *random uniform*, and crossing was performed with *heuristic blending* [23].
- A mutation percentage of 10% was used except for the best two elitists of each generation that were not mutated.
- The maximum number of generations was $G = 100$. However, an early stopping strategy was implemented when the J of the best individual did not decrease more than α in 10 generations, $G_{early} = 10$.

The search of the best XGBoost parameters were within the following ranges:

- Number of trees: $nrounds = [10,\ 2000]$.

Table 2
Testing RMSE obtained with the three methodologies. The last columns regarding BO and the hybrid method show the *p-value* obtained with the Wilcoxon test when comparing each method with GA-PARISMONY.

Database	GA-PARSIMONY		Bayesian optim.			Hybrid methodology		
Name	$RMSE_{tst}^{mean}$	$RMSE_{tst}^{sd}$	$RMSE_{tst}^{mean}$	$RMSE_{tst}^{sd}$	*p*-value	$RMSE_{tst}^{mean}$	$RMSE_{tst}^{sd}$	*p*-value
Ailerons	0.0425	0.042429	0.0428	0.000947	=(0.700)	**0.0420**	0.000784	=(1.000)
Bank	**0.0980**	0.097594	0.0995	0.001253	=(0.100)	0.0991	0.001149	=(0.200)
Blog	0.0148	0.014595	0.0155	0.010170	**+(0.039)**	**0.0147**	0.000994	=(1.000)
Concrete	0.0521	0.052261	0.0532	0.013800	=(0.100)	**0.0519**	0.013542	=(0.750)
Cpu	**0.0220**	0.021727	0.0232	0.002806	=(0.100)	0.0231	0.002863	=(0.100)
Crime	**0.0576**	0.058036	0.0612	0.004623	=(0.300)	**0.0576**	0.003234	=(0.834)
Elevators	**0.0314**	0.031355	0.0322	0.000641	=(0.100)	0.0319	0.000679	=(0.400)
Housing	**0.0586**	0.057918	0.0737	0.005727	**+(0.000)**	0.0589	0.005402	=(0.757)
Pol	**0.0400**	0.040358	0.0476	0.002647	**+(0.008)**	0.0465	0.001483	**+(0.030)**
Puma	0.0337	0.000420	0.0433	0.001411	**+(0.008)**	**0.0336**	0.000648	=(0.200)

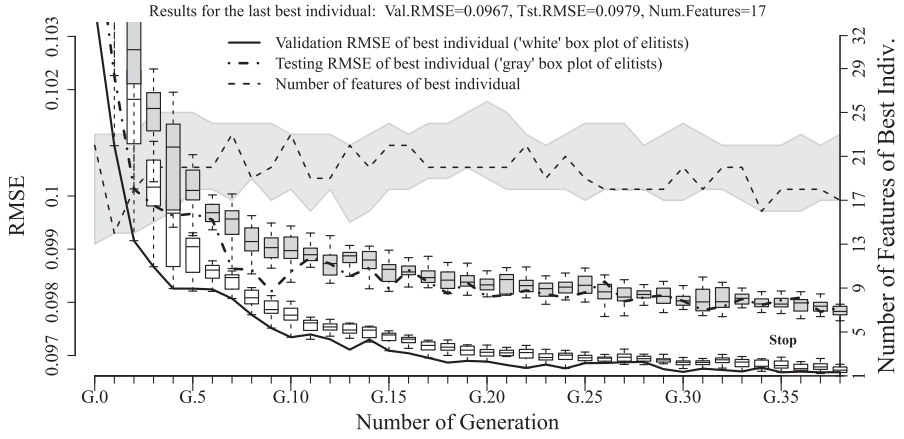

Fig. 2. Evolution of elitist individuals in *bank* database using GA-PARSIMONY for HO, FS, DT and PMS. White and gray box-plots represent $RMSE_{val}$ and $RMSE_{tst}$ evolution respectively. Discontinuous lines represent the best individual. The shaded area delimits the maximum and minimum N_{FS}.

- Maximum depth of a tree: $max\_depth = [2,\ 20]$.
- Minimum sum of instance weight needed in a child: $min\_child\_weight = [1,\ 20]$.
- *lasso* regularization term on weights: $alpha = [0.0,\ 1.00]$.
- *ridge* regularization term on weights: $lambda = [0.0,\ 1.00]$.
- Subsample ratio of the training instances: $subsample = [0.60,\ 1.00]$
- Subsample ratio of columns when constructing each tree: $colsample\_bytree = [0.80,\ 1.00]$.
- Random seed and learning rate were fixed to the following values: $seed = 1234$ and $eta = 0.01$.

Also, to transform the dependent variable, k exponent was used in the following way $y^* = y^k$. In this case, the range set for this parameter was $k = [0.20,\ 1.79]$.

Hence, each individual (i) of generation (g) was represented with a chromosome defined by:

$$\lambda_g^i = [nrounds,\ max\_depth,\ min\_child\_weight,\ alpha,\ lambda,\ subsample,\ colsample\_bytree,\ k,\ Q] \quad (3)$$

where the first seven values corresponded to the XGBoost parameters, k is the exponent to transform the dependent variable and Q is a binary-coded array with the selected features defined with ones.

3.3. Bayesian optimization settings

XGBoost parameter bounds of BO were identical to GA-PARSIMONY settings. The acquisition function selected was the GP-UCB while the covariance function was the squared exponential kernel with $\kappa = 2.576$. The number of initial points was set to 10, and the number of sequential iterations for the optimization process to 50.

3.4. Hybrid method settings

The first stage of the hybrid method was based on the same BO settings as those described in Section 3.3. In the second stage, GA-PARSIMONY performed FS and PMS with the best model parameters obtained during the first stage. Chromosomes at each generation were only defined by the binary-coded array $\lambda_g^i = Q$ because HO was disabled. Except λ_g^i, the rest of GA settings were similar to those described in Section 3.2.

3.5. Computational resources

All the experiments were implemented with nine 28-core servers (Intel ®Xeon ®E5-2670 @ 2.30 GHz) of the *Beronia* HPC cluster at the Universidad de La Rioja. Statistical software R [30] was used with `XGBoost` [6] and `GAparsimony` [21] packages.

4. Results

Table 1 summarizes the results obtained with ten UCI high-dimensional datasets. The first three columns show the number of instances (*#Inst*) and the number of input features (*#FT*) corresponding to each dataset. And for each method, the elapsed time (in minutes), the $RMSE_{tst}^{mean}$, and the number of selected features (*#SF*) are included in the Table. Also, the number of GA generations is depicted in *#Gen* columns. *#SF* of BO is not included because is similar to *#FT*.

Table 3
Summary of the hybrid method stages.

Database	*Stage 1 (BO)*			*Stage 2 (GA-PARSIMONY without HO)*				*Stage 2* vs *GA-PARSIMONY*
Name	#FT	Time	$RMSE_{tst}^{mean}$	#Gen	#FT	Time	$RMSE_{tst}^{mean}$	Diff. time (%)
Ailerons	40	295	0.0428	14	14	3926	0.0420	3568 min (50.61%)
Bank	32	104	0.0995	13	20	1429	0.0991	2607 min (64.59%)
Blog	276	1186	0.0155	10	108	2744	0.0147	2353 min (46.16%)
Concrete	8	152	0.0532	20	8	120	0.0519	188 min (61.03%)
Cpu	21	189	0.0232	26	16	4005	0.0231	116 min (02.81%)
Crime	127	206	0.0612	22	40	420	0.0576	617 min (59.50%)
Elevators	18	343	0.0322	10	12	2123	0.0319	14431 min (87.18%)
Housing	13	136	0.0737	16	9	55	0.0589	112 min (67.07%)
Pol	26	176	0.0476	17	20	3055	0.0465	9972 min (75.53%)
Puma	32	209	0.0433	13	4	3128	0.0336	3040 min (49.29%)

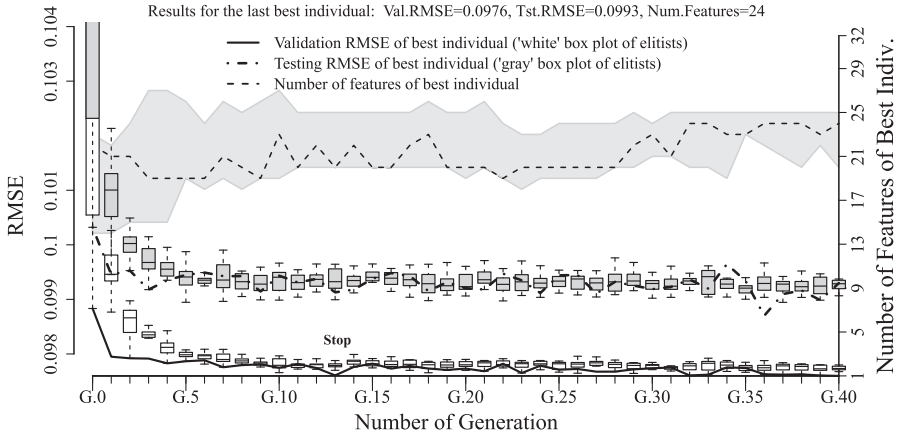

Fig. 3. Evolution of elitist individuals in *bank* database of Stage 2 of hybrid proposal which uses GA-PARSIMONY with XGBoost parameters set to the best ones obtained with BO. White and gray box-plots represent $RMSE_{val}$ and $RMSE_{tst}$ evolution respectively. The shaded area delimits the maximum and minimum N_{FS}.

Among the three methods, the hybrid methodology obtained models with the best $RMSE_{tst}^{mean}$ in five out of ten datasets, while it has errors similar to GA-PARSIMONY in the other five datasets. Both methods improved on the $RMSE_{tst}^{mean}$ achieved with BO. In addition, it can be observed that with GA-PARSIMONY there is an important reduction of *#SF* for all solutions, leading to parsimonious models with similar or better accuracy. However, the hybrid method considerably reduced the elapsed time in large datasets with only a slight increase in *#SF* versus GA-PARSIMONY parsimonious models.

Table 2 shows *p-values* obtained with the Wilcoxon test of GA-PARSIMONY versus BO and the hybrid proposal. Despite the fact that GA-PARSIMONY obtained a smaller $RMSE_{tst}^{mean}$ than BO in all datasets, the differences are only statistically significant in the 40% of the databases: *blog, housing, pol* and *puma*. With respect to the hybrid methodology, errors are similar to those of GA-PARSIMONY. The only exception appears in *pol* dataset, although the *p-value* is close to 95% of confidence level (*p-value* $\simeq 0.05$).

Results of the hybrid methodology stages are summarized in Table 3. The last column includes the time reduction in Stage 2 compared to GA-PARSIMONY. All calculations were made using 28-core servers (Intel ®Xeon ®E5-2670 @ 2.30 GHz).

In the first step, BO was applied to extract the best model parameters with all the features of the database. In the second stage, and with these parameters, FS was performed with GA- PARSIMONY but without HO.

In nine of the ten databases, *#Gen* of stage 2 reduced substantially as compared to GA-PARSIMONY. Therefore, the most important reduction in elapsed time was obtained in this stage, with a reduction of over 46% in the execution time in 90% of cases and with a sizeable contraction for large databases such as *elevators* or *pol*. However, with small datasets like *housing* and *concrete*, the execution time of stage 1 was greater than stage 2 because BO is more time-consuming than GA optimization.

It is also interesting to observe that although XGBoost is a trees ensemble method which performs quite well without FS, the combination of FS and PMS in stage 2 of the hybrid method improved the $RMSE_{tst}^{mean}$ of stage 1 in all datasets.

Fig. 2 depicts the evolution of the $RMSE_{val}$ and $RMSE_{tst}$ for the elitist individuals using the GA-PARSIMONY and *bank* database, without using early stopping to observe the optimization convergence errors. Fig. 3 shows the same evolution for the second stage of the hybrid method where GA-PARSIMONY is used without HO. In this second optimization, XGBoost parameters are obtained from the previous BO process (stage 1) computed with all the database features. By comparing both figures, one can observe than the optimization process converges faster with the hybrid methodology than with GA-PARSIMONY. With this database and using an early stopping criteria of 10 generations ($G_{early} = 10$), GA-PARSIMONY stops at the 35th generation, whereas the hybrid solution does so at the 13th, leading to the observed reduction in elapsed time.

Fig. 4 compares execution times of the hybrid methodology and GA-PARSIMONY. A significant reduction was achieved by the hybrid proposal in eight out of ten databases. The exceptions are *cpu* and *housing*. In the first case, GA-PARSIMONY stops 6 generations earlier than the hybrid method. In the hybrid method and with the smallest database in the table, *housing*, sequential search with BO is more computational expensive than stage 2. Thus, the reduction of execution time in stage 2 is lower than the BO process (stage 1). However, it can be observed that the hybrid methodology clearly achieves significant reductions in time for large databases such as *ailerons, bank, elevators, pol*, and *puma*.

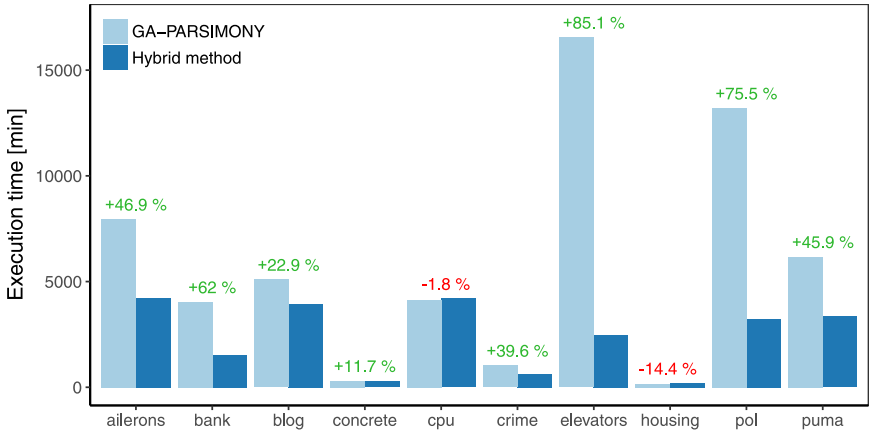

Fig. 4. Execution times of the GA-PARSIMONY and the hybrid methodology. Percentages correspond to the relative reduction obtained by the new hybrid proposal versus GA-PARSIMONY.

5. Discussion and conclusions

Although it is well known that tree ensemble algorithms like XGBoost perform well without FS, experiments with ten UCI databases have demonstrated that original GA-PARSIMONY, which combines HO, PMS, FS, and DT, improves testing errors in all datasets as compared to BO. Furthermore, GA-PARSIMONY reduces the number of select features to seek parsimonious solutions that are more robust against noise and easier to interpret and maintain. However, although GA- PARSIMONY obtains better models than BO, the computational effort needed for large and high dimensional databases was much greater. Due to this fact, the main objective of this study was to evaluate a new hybrid method which combined BO and a constrained version of GA-PARSIMONY to obtain similar parsimonious solutions but with a significant reduction in elapsed time.

In the first stage, the hybrid proposal used BO to determine the best model parameters. With these parameters, GA-PARSIMONY optimization without HO was performed in a second step. Compared with GA-PARSIMONY, a significant reduction in elapsed time was obtained in this stage because it converged faster. The results demonstrate an important reduction in the largest datasets such as 46.9%, 22.9%, 85.1% and 75.5%, for *ailerons, blog, elevators* and *pol*, respectively. Testing errors were similar to GA-PARSIMONY with a number of features slightly higher than GA-PARSIMONY in all datasets except *housing*. In 80% of the databases the hybrid methodology undoubtedly obtained parsimonious solutions similar to those of GA-PARSIMONY but with a significant reduction in elapsed time, especially in large databases. However, there were not significant improvements in two databases. With *cpu*, GA-PARSIMONY converged faster than the hybrid method. And with *housing*, the smallest dataset, BO of stage 1 of the hybrid proposal was more computationally expensive than second stage: the elapsed time increased by 14.4% as compared to GA-PARSIMONY.

Nevertheless, future research could develop many improvements. Firstly, new configurations of the constrained GA-PARSIMONY stage 2 could be investigated. And furthermore, new bio-inspired metaheuristics like PSO or artificial bee colony (ABC) could substitute GA optimization to achieve better convergence times. Further experiments are still necessary with additional high-dimensional databases to formulate more detailed conclusions.

Acknowledgments

We are greatly indebted to Banco Santander for the APPI16/05, APPI17/04 and REGI2018/43 fellowships, and to the University of La Rioja for the EGI16/19 fellowship. This study used the Beronia cluster (Universidad de La Rioja), which is supported by FEDER-MINECO grant number UNLR-094E-2C-225.

References

[1] R. Ahila, V. Sadasivam, K. Manimala, An integrated PSO for parameter determination and feature selection of ELM and its application in classification of power system disturbances, Appl. Soft Comput. 32 (2015) 23–37.
[2] F. Antonanzas-Torres, R. Urraca, J. Antonanzas, J. Fernandez-Ceniceros, F.M. de Pison, Generation of daily global solar irradiation with support vector machines for regression, Energy Convers. Manag. 96 (2015) 277–286.
[3] M. Avalos, Y. Grandvalet, C. Ambroise, Parsimonious additive models, Comput. Stat. Data Anal. 51 (6) (2007) 2851–2870.
[4] B. Bischl, J. Richter, J. Bossek, D. Horn, J. Thomas, M. Lang, mlrMBO: A Modular Framework for Model-Based Optimization of Expensive Black-Box Functions. http://arxiv.org/abs/1703.03373.
[5] P. Caamaño, F. Bellas, J.A. Becerra, R.J. Duro, Evolutionary algorithm characterization in real parameter optimization problems, Appl. Soft Comput. 13 (4) (2013) 1902–1921.
[6] T. Chen, T. He, M. Benesty, XGBoost: Extreme Gradient Boosting Machines., 2015. R package version 0.4–3. https://github.com/dmlc/xgboost
[7] E. Corchado, M. Wozniak, A. Abraham, A.C.P.L.F. de Carvalho, V. Snásel, Recent trends in intelligent data analysis, Neurocomputing 126 (2014) 1–2.
[8] R. Dhiman, J. Saini, Priyanka, Genetic algorithms tuned expert model for detection of epileptic seizures from EEG signatures, Appl. Soft Comput. 19 (2014) 8–17.
[9] S. Ding, Spectral and wavelet-based feature selection with particle swarm optimization for hyperspectral classification, J. Softw. 6 (7) (2011) 1248–1256.
[10] J. Fernandez-Ceniceros, A. Sanz-Garcia, F. Antonanzas-Torres, F.M. de Pison, A numerical-informational approach for characterising the ductile behaviour of the T-stub component. Part 2: parsimonious soft-computing-based metamodel, Eng. Struct. 82 (2015) 249–260.
[11] M. Feurer, A. Klein, K. Eggensperger, J. Springenberg, M. Blum, F. Hutter, Efficient and robust automated machine learning, in: C. Cortes, N.D. Lawrence, D.D. Lee, M. Sugiyama, R. Garnett (Eds.), Advances in Neural Information Processing Systems, 28, Curran Associates, Inc., 2015, pp. 2962–2970.
[12] J.H. Friedman, Greedy function approximation: a gradient boosting machine, Ann. Stat. 29 (5) (2001) 1189–1232.
[13] D. Gorissen, I. Couckuyt, P. Demeester, T. Dhaene, K. Crombecq, A surrogate modeling and adaptive sampling toolbox for computer based design, J. Mach. Learn. Res. 11 (2010) 2051–2055.
[14] I.A. Hashem, I. Yaqoob, N.B. Anuar, S. Mokhtar, A. Gani, S. Ullah Khan, The rise of big data on cloud computing: review and open research issues, Inf. Syst. 47 (2015) 98–115.
[15] C.-L. Huang, J.-F. Dun, A distributed PSO-SVM hybrid system with feature selection and parameter optimization, Appl. Soft Comput. 8 (4) (2008) 1381–1391.
[16] H.-L. Huang, F.-L. Chang, ESVM: Evolutionary support vector machine for automatic feature selection and classification of microarray data, Biosystems 90 (2) (2007) 516–528.
[17] H. Husain, N. Handel, Automated Machine Learning. A Paradigm Shift that Accelerates Data Scientist Productivity, 2017. https://medium.com/airbnb-engineering/.
[18] Andrea Brown, International Conference on Machine Learning, AutoML Workshops at ICML, 2019. http://www.ml4aad.org/automl/workshops/
[19] H. Li, D. Shu, Y. Zhang, G.Y. Yi, Simultaneous variable selection and estimation for multivariate multilevel longitudinal data with both continuous and binary responses, Comput. Stat. Data Anal. 118 (2018) 126–137.
[20] B. Ma, Y. Xia, A tribe competition-based genetic algorithm for feature selection in pattern classification, Appl. Soft Comput. 58 (2017) 328–338.

[21] F.J. Martínez-De-Pisón, GAparsimony: GA-based Optimization R Package for Searching Accurate Parsimonious Models, 2017. R package version 0.9–1. https://github.com/jpison/GAparsimony.
[22] S. Medjahed, T.A. Saadi, A. Benyettou, M. Ouali, Gray wolf optimizer for hyperspectral band selection, Appl. Soft Comput. 40 (2016) 178–186.
[23] Z. Michalewicz, C.Z. Janikow, Handling constraints in genetic algorithms, in: Proceedings of International Conference on Genetic Algorithms, ICGA, 1991, pp. 151–157.
[24] F. Nogueira, Bayesian Optimization for Python, 2019. https://github.com/fmfn/BayesianOptimization.
[25] R.S. Olson, N. Bartley, R.J. Urbanowicz, J.H. Moore, Evaluation of a tree-based pipeline optimization tool for automating data science, in: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '16, ACM, New York, NY, USA, 2016, pp. 485–492.
[26] P. Parry, auto_ml. Automatic Machine Learning for Python, 2018. https://github.com/ClimbsRocks/auto_ml.
[27] J. Perez-Rodriguez, A.G. Arroyo-Penj, N. Garcia-Pedrajas, Simultaneous instance and feature selection and weighting using evolutionary computation: Proposal and study, Appl. Soft Comput. 37 (2015) 416–443.
[28] P. Perner, Improving the accuracy of decision tree induction by feature preselection, Appl. Artif. Intell. 15 (8) (2001) 747–760.
[29] F.J. Martinez-de Pison, E. Fraile-Garcia, J. Ferreiro-Cabello, R. Gonzalez, A. Pernia, Searching Parsimonious Solutions with GA-PARSIMONY and XGBoost in High-Dimensional Databases, Springer International Publishing, Cham, pp. 201–210.
[30] R Core Team, R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria, 2013.
[31] C.E. Rasmussen, C.K.I. Williams, Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning), The MIT Press, 2005.
[32] M. Reif, F. Shafait, A. Dengel, Meta-learning for evolutionary parameter optimization of classifiers, Mach. Learn. 87 (3) (2012) 357–380.
[33] A. Sanz-Garcia, J. Fernandez-Ceniceros, F. Antonanzas-Torres, A. Pernia-Espinoza, F.J. Martinez-de Pison, GA-PARSIMONY: a GA-SVR approach with feature selection and parameter optimization to obtain parsimonious solutions for predicting temperature settings in a continuous annealing furnace., Appl. Soft Comput. 35 (2015) 13–28.
[34] A. Sanz-García, J. Fernández-Ceniceros, F. Antoñanzas-Torres, F.J. Martínez-de Pisón, Parsimonious support vector machines modelling for set points in industrial processes based on genetic algorithm optimization, in: Proceedings of International Joint Conference SOCO13-CISIS13-ICEUTE13, in: Advances in Intelligent Systems and Computing, 239, Springer International Publishing, 2014, pp. 1–10.
[35] B. Shahriari, K. Swersky, Z. Wang, R.P. Adams, N. de Freitas, Taking the Human Out of the Loop: A Review of Bayesian Optimization, Technical Report, Universities of Harvard, Oxford, Toronto, and Google DeepMind, 2015.
[36] J. Snoek, H. Larochelle, R.P. Adams, Practical Bayesian optimization of machine learning algorithms, in: F. Pereira, C.J.C. Burges, L. Bottou, K.Q. Weinberger (Eds.), Advances in Neural Information Processing Systems, 25, Curran Associates, Inc., 2012, pp. 2951–2959.
[37] N. Srinivas, A. Krause, S.M. Kakade, M.W. Seeger, Gaussian Process Bandits without Regret: An Experimental Design Approach, CoRR abs/0912.3995(2009).
[38] C. Thornton, F. Hutter, H.H. Hoos, K. Leyton-Brown, Auto-weka: combined selection and hyperparameter optimization of classification algorithms, in: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '13, ACM, New York, NY, USA, 2013, pp. 847–855.
[39] R. Urraca, E. Sodupe-Ortega, J. Antonanzas, F. Antonanzas-Torres, F.M. de Pison, Evaluation of a novel GA-based methodology for model structure selection: The GA-PARSIMONY, Neurocomputing 271 (Supplement C) (2018) 9–17.
[40] R. Urraca-Valle, A. Sanz-García, J. Fernández-Ceniceros, E. Sodupe-Ortega, F.J.M. de Pisón Ascacibar, Improving hotel room demand forecasting with a hybrid GA-SVR methodology based on skewed data transformation, feature selection and parsimony tuning, in: E. Onieva, I. Santos, E. Osaba, H. Quintián, E. Corchado (Eds.), Proceedings of the 10th International Conference on Hybrid Artificial Intelligent Systems, HAIS 2015, Bilbao, Spain, June 22–24, 2015, Proceedings, Lecture Notes in Computer Science, 9121, Springer, 2015, pp. 632–643.
[41] S.M. Vieira, L.F. Mendonza, G.J. Farinha, J.M. Sousa, Modified binary PSO for feature selection using SVM applied to mortality prediction of septic patients, Appl. Soft Comput. 13 (8) (2013) 3494–3504.
[42] Y. Wan, M. Wang, Z. Ye, X. Lai, A feature selection method based on modified binary coded ant colony optimization algorithm, Appl. Soft Comput. 49 (2016) 248–258.
[43] M. Wang, H. Chen, B. Yang, X. Zhao, L. Hu, Z. Cai, H. Huang, C. Tong, Toward an optimal kernel extreme learning machine using a chaotic moth-flame optimization strategy with applications in medical diagnoses, Neurocomputing 267 (2017) 69–84.
[44] J. Wei, R. Zhang, Z. Yu, R. Hu, J. Tang, C. Gui, Y. Yuan, A BPSO-SVM algorithm based on memory renewal and enhanced mutation mechanisms for feature selection, Appl. Soft Comput. 58 (2017) 176–192.
[45] B. Xue, M. Zhang, W.N. Browne, Particle swarm optimisation for feature selection in classification: novel initialisation and updating mechanisms, Appl. Soft Comput. 18 (0) (2014) 261–276.

F.J. Martinez-de-Pison was born in Spain in 1970. He is the head of the EDMANS group and professor at the University of La Rioja. His research activities focus on the use of soft computing, data mining and machine learning methods to solve real problems in various fields such as industry, energy, agriculture and business.

R. Gonzalez-Sendino was born in Spain in 1994. He has a Computer Engineering Degree from the University of La Rioja and a Master's in Artificial Intelligence from the Polytechnic University of Madrid. He has been working at CRIDA, a research, innovation and development center since 2017.

A. Aldama was born in Spain in 1992. He has a bachelor's degree in Industrial Electronics and Automation and a Master's in Industrial Engineering from the University of La Rioja. He works as an assistant professor in the Electronic Technology Department of the University of La Rioja. His research interests are data analysis and machine learning.

J. Ferreiro-Cabello is Ph.D. in Industrial Engineering and adjunct professor at the University of La Rioja. His research focuses on modeling, simulation and optimization of processes involving numerical analysis of cost, and environmental and social impact.

E. Fraile-Garcia was born in Spain in 1970. He has a Ph.D. in Industrial Engineering and associate professor at the University of La Rioja. His research activities focus on optimization methods for structural solutions in residential building. He is currently working on predictive models to obtain structural solutions based on economic, environment and social concerns.