The Thirty-Second International Florida
Artificial Intelligence Research Society Conference (FLAIRS-32)

A Novel Combining-Based Method of Pool Generation for Ensemble Regression Problems

Robson D. A. Timoteo, Daniel C. Cunha, Paulo S. G. de Mattos Neto
Centro de Informática - UFPE
Av. Jornalista Anibal Fernandes, s/n
Recife, PE, Brazil 50740-560

Abstract

A crucial point for ensemble learning systems is the capacity of making different errors on any given sample, which highlights the importance of diversity for ensemble-based decision systems. A usual way of increasing diversity is to combine traditional ensemble methods. Based on this context, we propose a novel combining-based algorithm of pool generation using a merging of bagging, random patches, and boosting techniques for ensemble regression problems. Numerical results indicate that, depending on both the dataset and the diversity measurement, our proposal generates a pool of regressors with more diversity when compared to single ensemble generator approaches.

Introduction

An outstanding growth of data-oriented systems has been noticed in recent years. This evidence is motivated by the increasing of data volume generated by Internet users. In this context, machine learning (ML) techniques have played a central role, making it possible to solve complex problems such as face recognition (A. A. Mohammed and Sid-Ahmed 2012), stock price prediction (C. K. S. Leung and Wang 2014), consumer mood prediction (J. Bollen and Zeng 2011), and other applications. Over the last couple of decades, ensemble learning systems have experienced increasing attention within ML community. According to (J. Mendes-Moreira and de Sousa 2012), ensemble learning deals with methods that produce several models which are combined to obtain a prediction, either in classification or regression. The assumption of using ensemble-based decision systems is based on the fact that we examine other opinions before making a decision, trying to take advantage of not only the variability of the past history, but also the accuracy of the individual decision makers.

Making different errors on any given sample is a crucial point for ensemble learning systems. The combination of predictors will be useless if all ensemble components give the same output. So, we need diversity in the decisions of ensemble members, specifically when they are making an error. The importance of diversity for ensemble-based systems is well-established in the literature (Breiman 1996; Ho 1998; Avnimelech and Intrator 1999; Brown 2004; Brown et al. 2005).

A way of increasing diversity in ensemble-based systems is to combine traditional ensemble methods. In this context, several initiatives can be found in the literature. For example, concerning classification, a combination of bagging and random subspace, two single ensemble approaches, is proposed in (Panov and Džeroski 2007). In this work, the authors show that the combining has similar performance to that of random forests (Breiman 2001), with the advantage of being applicable to any base-level algorithm, without the need of randomize it.

Regarding regression problems, an ensemble of bagging, random subspace, and boosting is proposed in (Kotsiantis and Kanellopoulos 2012). In this case, eight regressors are used for each component method. Then, an averaging methodology is used to obtain the final prediction. It is shown that the proposed methodology gives better correlation coefficient (diversity gain) in most cases. In (Hadavandi, Shahrabi, and Shamshirband 2015), boosting and subspace projection are used to model multi-target regression problems with high-dimensional feature spaces and a small number of instances. It is demonstrated that the proposed method offers the capability to improve diversity and accuracy of the neural network ensembles.

In this work, we introduce a new combining-based method of pool generation for ensemble regression problems. Our motivation is based on the fact that single ensemble approaches usually present some weaknesses. For example, the boosting algorithm is sensitive to noise, while the bagging method reduces the variance, but has little effectiveness in bias reduction (Louppe and Geurts 2012). Concerning the random patches algorithm, it is hard to apply it in problems with a low number of features. With this in mind, we propose a combination approach focused on the merging of popular ensemble-based strategies to increase diversity. The objective is to create weak predictors, which are experts in different subspaces of the training set (instances and features).

The outline of the paper is as follows. In Section , we present some basic concepts about ensemble regression as well as the definitions of correlation coefficient and disagreement measure, two types of diversity measurements. In Section , we propose a new method of pool generation based on the combination of single ensemble learning algorithms. Numerical results comparing our proposal to some well-known ensemble learning approaches are shown in Section . Finally,

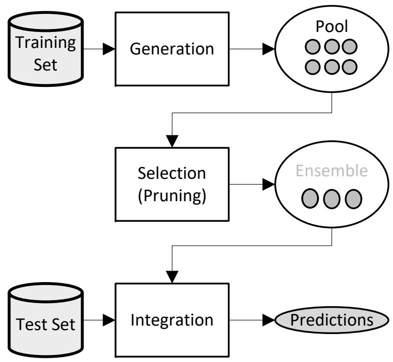

Figure 1: Diagram of the ensemble learning building process: generation, selection, and integration steps.

conclusions are drawn in Section .

Background

Ensemble regression

Regression is one of the most important and widely used task in machine learning whose goal is to construct a hypothesis function using known data. This constructed function is used to assign a continuous value to an unknown test pattern, that is, the use of the hypothesis function allows the regressor to make a generalization based on already known data.

The generalization capability of a regression model is a key point in ML design (Kuhn and Johnson 2013). ML systems based on aggregated predictors, also named ensemble learning, have a better generalization capacity when compared to single predictors (Kuncheva 2004; Rooney et al. 2004). In recent years, the increasing processing power has enabled the usage of ensemble learning, which has performed better than individual predictors (Kuncheva 2004). A detailed description of the reasons that justify the best performance of ensemble learning is presented in (Dietterich 2000).

Fig. 1 illustrates the ensemble building process, that can be divided into three steps named generation, selection, and integration. The generation step creates the pool, which can be defined as the set of all predictors. These predictors can be regressors or classifiers, depending on the task to be performed. Considering that this work focuses in regression problems, we will refer to the predictors as regressors from now on. As can be seen in Fig. 1, the training data is used to train the regressors by using several pool generation strategies, such as bagging, random subspace or boosting, that can be adopted to guarantee a better generalization capacity (J. Mendes-Moreira and de Sousa 2012). In other words, these strategies are employed to increase diversity between the regressors.

After the generation stage, the next step is the selection (pruning), which consists of selecting a subset of the models generated in the previous step. The goal of the selection is to decrease the number of regressors without significantly reducing the accuracy of the predictions (Hernández-Lobato, Martínez-Muñoz, and Suárez 2006). The reduction of computational cost can be considered as an important argument that justifies the selection (J. Mendes-Moreira and de Sousa 2012). It is worth to stress that, in some cases, an appropriate selection of regressors may considerably outperform the full set of regressors (Bakker and Heskes 2003).

After the selection, the output of the models are integrated to generate the final prediction of the system. This last phase can be seen as a combination stage, where the strategies can vary depending on the task at hand. For example, voting can be used for classification, while averaging of the models outputs is an alternative for regression problems.

Ensemble diversity

Diversity can be seen as a disagreement measurement, which is directly correlated to the generalization capacity (Kuncheva 2004). Making different errors is essential for ensemble generalization capacity, because if all ensemble members provide the same output, their combination will be meaningless. Therefore, we want the regressors outputs to have diversity. Ideally, regressors outputs should be independent or negatively correlated (Zhang and Ma 2012).

There is no formal definition to measure diversity. Regarding to ensemble regression, different types of pairwise diversity measurements are presented in (Dutta 2009), which are correlation coefficient, covariance, disagreement measure, and mutual information. In this paper, we assume the correlation coefficient and the disagreement measure as diversity measurements. While the former is a more traditional approach, the latter is one of the most recent ways to evaluate diversity (Dutta 2009).

To define the diversity measurements considered in this work, let us assume a training dataset D with N instances. For the i-th instance, we have a continuous valued target y_i, that depends on both a regressor and a vector of features as input. In other words, the regressor output assume one continuous value for each instance of the set D.

Given that, let us consider a regressor R_a whose output Y^a can assume N values in the set $\mathbf{Y}^a = [y_1^a, \ldots, y_i^a, \ldots, y_N^a]$. In a similar way, consider also a regressor R_b whose output Y^b is one of the N values in the set $\mathbf{Y}^b = [y_1^b, \ldots, y_i^b, \ldots, y_N^b]$.

Correlation coefficient The linear correlation coefficient ρ_{ab} between the regressors outputs Y^a and Y^b is given by

$$\rho_{ab} = \sum_{i=1}^{N} \left(\frac{(y_i^a - \mu_a)(y_i^b - \mu_b)}{\sqrt{\sum_{i=1}^{N}(y_i^a - \mu_a)^2 \sum_{i=1}^{N}(y_i^b - \mu_b)^2}} \right) \quad (1)$$

where μ_a and μ_b are the averages of the values of $\mathbf{Y}^a$ and $\mathbf{Y}^b$, respectively. The diversity between Y^a and Y^b is inversely proportional to ρ_{ab} .Thereby, a pair of regressors with a low correlation coefficient is better for the pool regarding ones with high correlation (Dutta 2009).

Disagreement measure This measure was originally used to represent the diversity between base and complementary classifiers (Skalak and others 1996), and posteriorly for measuring diversity in random subspace method (Ho 1998). It is defined as the ratio between the number of observations on which one classifier is correct and the other is incorrect to the total number of observations (Kuncheva 2004).

Assume two classifiers denoted as C_a and C_b. Consider also that "0" represents a correct classification and that "1" represents an incorrect classification. The total number of instances predicted by C_a and C_b, denoted as N_{ab}, is given by

$$N_{ab} = N_{00} + N_{01} + N_{10} + N_{11}\ , \tag{2}$$

where N_{00} is the number of instances classified correctly by C_a and C_b, while N_{01} (N_{10}) depicts the number of predictions in which only C_a (C_b) hit. Finally, N_{11} is the number of instances predicted incorrectly by both classifiers.

For regression problems, the disagreement measure is extended as follows (Dutta 2009): for each instance, we obtain the standard deviation σ of the estimated target variable by all predictors. If the true value of the target is α then a prediction β is considered to be correct. If $\beta < \alpha + \sigma$ and $\beta > \alpha - \sigma$, the prediction has to fall within a margin of one standard deviation of the value of the target variable. Otherwise, the prediction is taken to be incorrect. Thus, we can define disagreement measure as follows:

$$d_{ab} = \frac{N_{01} + N_{10}}{N_{ab}}\ . \tag{3}$$

Since the diversity measurements previously mentioned are pairwise measures, we consider the average of all possible pair of regressors to calculate the pool diversity.

Proposed Method

In this section, we present an innovative combining-based approach to generate a pool of regressors, the first stage of the ensemble building process illustrated in Figure 1. New ensemble elements are created based on bagging (BAGG), boosting (BOOS), and random patches (RAPT), a derivation of the random subspace method, to increase the ensemble diversity. The main difference between random subspace and RAPT is that while only instances are sampled in the former, both instances and features are sampled in the latter (Louppe and Geurts 2012).

The proposed method is depicted in Algorithm 1 and it has the following input variables: M, the number of regressors in the pool; D, the training dataset; S_b, the number of instance samples for BAGG method; F_{min} and F_{max}, minimum and maximum numbers of features for RAPT approach, respectively. The main idea of the proposed method is to increase diversity by using strategies employed in BAGG, RAPT, and BOOS approaches. Our combining-based method returns a pool Ψ of M regressors, each one trained with a different training set.

Considering R^i and D^i as the i-th trained regressor and its training set, respectively, different pool generation strategies are used, depending on the amount of previously generated regressors. Every two regressors generated by using BAGG and RAPT methods, a third regressor is produced by means of a combination of the BOOS algorithm with the two previous methods. To put it differently, when the index i is not multiple of 3, BAGG and RAPT methods are used to construct the i-th regressor. Otherwise (when i is multiple of 3),

Algorithm 1 Proposed combining-based method for pool generation.

Inputs:
D, training set
M, the number of pool regressors
S_b, number of samples for BAGG
F_{min}, minimum number of features for RAPT
F_{max}, maximum number of features for RAPT
Returns:
Ψ = Pool of regressors
Initialize:
1: $\Psi \leftarrow$ empty list of regressors
2: $\mathbf{w}_u \leftarrow$ Vector of unit probability weights
3: **for** $i = 1$ **to** M **do**
4: **if** i is not multiple of 3 **then**
5: $D^i \leftarrow$SAMPLE (D, S_b, F_{min}, F_{max}, $\mathbf{w}_u$);
6: **else**
7: $\Gamma \leftarrow$ Set of instances that were not used to train
8: the previous regressors R^{i-1} and R^{i-2};
9: $\mathbf{w}_e \leftarrow$ Sum of the prediction errors obtained by
10: R^{i-1} and R^{i-2} for each instance of Γ;
11: $D^i \leftarrow$SAMPLE (Γ, S_b, F_{min}, F_{max}, $\mathbf{w}_e$);
12: **end if**
13: Train the regressor R^i using D^i;
14: Include R^i in Ψ;
15: **end for**
16: **return** Ψ

the BOOS algorithm is used in combination with BAGG and RAPT methods to produce the i-th regressor. The goal of the combination is to create weak predictors, which are experts in different subspaces of the training dataset (instances and features). The keypoint of the combining-based idea proposed in this work is indicated by the lines $4 - 12$ of the Algorithm 1.

BAGG and RAPT methods are applied by using the `SAMPLE` function, which is described in Algorithm 2. The input parameters of the `SAMPLE` function are D_s, the sample source; $\mathbf{w}$, the vector of probability weights of the samples; and finally, S_b, F_{min}, and F_{max}, the same parameters adopted in Algorithm 1. The `SAMPLE` function selects instances χ and features Φ from D_s to build a new dataset Δ. As can be seen in Algorithm 2, the instances χ are taken randomly with replacement and using the vector of probability weight $\mathbf{w}$, while the features Φ are selected without replacement.

When the regressor R^i is trained without boosting (when i is not multiple of 3), the vector $\mathbf{w}$ is assumed to be $\mathbf{w}_u$, which represents the uniform distribution. In other words, all instances in D_s will have the same probability to be taken. Furthermore, the complete training set D is used as source of the samples (line 5 of the Algorithm 1). Differently, when boosting is used for training (i is multiple of 3), a dataset Γ is created with instances that were not used for training the two prior regressors (R^{i-1} and R^{i-2}), i.e., $\Gamma = D - (D^{i-1} \cup D^{i-2})$. The dataset Γ is used as input for the `SAMPLE` function, a first aspect that differs from the approach without boosting. A second aspect regards to the

Algorithm 2 Sample function using probability weights and combining BAGG and RAPT methods.

Inputs:
D_s, training set
Returns:
Δ, sampled data

```
function SAMPLE( D_s, S_b, F_min, F_max, w)
    χ ← Take S_b samples from instances of D_s, with
        replacement using probability weights in w;
    Φ ← Take between F_min and F_max samples
        from features of D_s without replacement;
    Δ ← New dataset with instances χ and features Φ;
    return Δ
```

probability weights, which are calculated from errors made by the regressors R^{i-1} and R^{i-2} when predicting instances of Γ.

Considering K as the number of instances in the dataset Γ, such that $\Gamma = [\gamma_1, \ldots, \gamma_j, \ldots, \gamma_K]$. We define a vector of probability weights $\mathbf{w} = [w_1, \ldots, w_j, \ldots, w_K]$, where the j-th probability weight w_j, associated with each instance γ_j, is given by

$$w_j = \varepsilon_j^{i-1} + \varepsilon_j^{i-2}\,, \tag{4}$$

where ε_j^{i-1} and ε_j^{i-2} are the prediction errors of the regressors R^{i-1} and R^{i-2}, respectively, for the instance γ_j. That is, the probability of selecting γ_j is proportional to the sum of its prediction errors related to the two previous regressors. Therefore, the new regressor R^i will focus on instances that have been poorly learned. Finally, the training dataset D^i is constructed from the set Γ, and after the training, the regressor R^i is included in the pool Ψ.

Results

The proposed method is experimentally evaluated by using R programming language, with emphasis on the package caret (Kuhn and Johnson 2013). For all experiments, we use the k-Nearest Neighbors (k-NN) algorithm as the base regressor and assume pools of $M = 36$ predictors. For all regressors in the pool, the 10-fold cross-validation re-sampling technique is used for model tuning, while the best k is searched in the interval $[1; 11]$. The k-NN technique was chosen because its computational cost has a direct relationship to the dataset properties (number of instances and features), which allow us to analyze the computational cost trade-offs. Indeed, the proposed method can be easily extended to other regressors, such as ANNs or support vector regressors (SVRs).

Table 1 shows ten public datasets used in the experiments. These datasets cover different types of problems, having continuous, discrete, and categorical features. Furthermore, they differ in number of training instances and features.

The measurements previously defined in Subsection are used to evaluate diversity. For each dataset listed in Table 1, 90% of the instances are used for pool generation (training set), whereas 10% are used to evaluate the diversity. Pools of regressors using BAGG, RAPT, BOOS methods and our approach are created for each dataset to compare diversity. BAGG, RAPT, and BOOS methods are referred as our benchmarks, while the proposed approach is denoted by the combining-based (CBAS) method, from now on. Throughout this Section, the best method for each dataset and each diversity metric is highlighted in bold when we present the results. To verify if the differences between the performances of the pool generation methods are statistically relevant, Friedman test with the Nemenyi post-hoc test are applied (Demšar 2006). When the difference between the methods is not statistically relevant, we have a draw, with more than one method indicated in bold.

Table 1: Public datasets used in the experiments.

Dataset	Number of features	Number of instances
Airquality	5	153
CPUs	6	209
Autompg	7	392
Concrete	8	1030
Abalone	8	4177
Carseats	10	400
Wage	11	3000
Forestfires	12	513
Boston Housing	13	506
Hitters	19	322

Table 2: Correlation coefficients obtained for the 10 public datasets considering bagging (BAGG), random patches (RAPT), boosting (BOOS), and the combining-based (CBAS) methods.

Dataset	BAGG	RAPT	BOOS	CBAS
Airquality	0.91	0.76	0.84	**0.66**
CPUs	0.90	0.87	0.86	**0.80**
Autompg	0.91	0.80	0.84	**0.78**
Concrete	0.84	0.68	0.76	**0.59**
Abalone	0.74	0.53	0.63	**0.48**
Carseats	0.52	0.26	0.38	**0.20**
Wage	0.91	**0.50**	0.82	0.61
Forestfires	0.38	0.15	0.23	**0.08**
Boston Housing	0.89	***0.74**	0.85	***0.73**
Hitters	0.61	0.57	**0.49**	**0.46**
W-D-L	0-0-10	1-1-8	0-1-9	8-1-1
(*) Not statistically relevant				

Table 2 shows the correlation coefficients obtained for all datasets considering the benchmarks and the CBAS method. Also in Table 2, we have a win-draw-lose (W-D-L) comparison between the pool generation methods for each dataset. Considering that the lower correlation coefficient, the greater the diversity, the CBAS method wins in eight of ten datasets. We can also see that, for Boston Housing dataset, the difference between RAPT and CBAS methods is not statistically relevant, meaning that the two approaches are equivalent[1].

Table 3 illustrates the disagreement measures obtained for all datasets considering BAGG, RAPT, BOOS, and CBAS methods. In this case, the higher the disagreement, the greater the diversity. So, the results behave in an opposite way when compared to those obtained by correlation coefficient. Similarly to the previous case, the CBAS method is the winner for the same eight datasets. In addition, our proposal is equivalent to the RAPT method for the Boston Housing dataset,

Table 3: Disagreement measures obtained for the 10 public datasets considering bagging (BAGG), random patches (RAPT), boosting (BOOS), and the combining-based (CBAS) methods.

Dataset	BAGG	RAPT	BOOS	CBAS
Airquality	0.06	0.13	0.10	**0.18**
CPUs	0.04	0.06	0.07	**0.10**
Autompg	0.04	0.11	0.08	**0.12**
Concrete	0.09	0.21	0.14	**0.25**
Abalone	0.12	0.23	0.16	**0.24**
Carseats	0.26	0.40	0.34	**0.43**
Wage	0.04	**0.22**	0.08	0.18
Forestfires	0.34	0.41	0.40	**0.45**
Boston Housing	0.05	*0.11	0.06	*0.12
Hitters	0.23	0.22	0.25	**0.28**
W-D-L	0-0-10	1-1-8	0-0-10	8-1-1
(*) Not statistically relevant				

Table 4: Root mean square error (RMSE) obtained for the 10 public datasets considering the generation of three pools of regresssors (Kotsiantis and Kanellopoulos 2012) and our proposal (CBAS).

Dataset	3 pools of regressors	CBAS
Airquality	*12.17	*10.07
CPUs	*37.15	*26.53
Autompg	*2.96	*3.15
Concrete	*7.58	*7.86
Abalone	*2.58	*2.40
Carseats	*2.22	*1.94
Wage	*12.54	*8.30
Forestfires	*1.38	*1.34
Boston Housing	*3.22	*3.34
Hitters	*550.12	*541.44
W-D-L	0-10-0	0-10-0
(*) Not statistically relevant		

since the difference between them is not statistically relevant, as it was observed for the correlation coefficient.

Table 4 exhibits the root mean square error (RMSE) achieved for all datasets considering method proposed in (Kotsiantis and Kanellopoulos 2012). In this case, three pools of regressors are generated, one for each re-sampling technique (BAGG, RAPT and BOOS). For the final prediction, an averaging methodology is used. In the second column of Table 4, we have the RMSE for the same method, however using the CBAS approach to generate a single pool of regressors. Similar to the calculation of the diversity measures, the RMSE is obtained over 10% of the instances (test set). Friedman test with the Nemenyi post-hoc test shows that the difference between them is statistically irrelevant, meaning that the methods are equivalent[1]. On the other hand, our proposal reduces the number of pools without compromising the accuracy of the predictions.

Another important aspect to compare the pool generation

[1]Diversity measures and RMSE values used for the statistical tests are available on https://github.com/timotrob/CBASDiversity.

Table 5: Normalized average computational cost for BAGG, RAPT, BOOS, and CBAS methods for all datasets considered in this work.

Dataset	BAGG	RAPT	BOOS	CBAS
Airquality	**1.00**	1.05	1.05	1.05
CPUs	1.07	**1.00**	1.06	1.07
Autompg	1.00	**1.00**	1.06	1.06
Concrete	1.09	1.08	**1.00**	1.08
Abalone	**1.00**	1.16	1.22	1.16
Carseats	1.08	**1.00**	1.08	1.08
Wage	**1.00**	1.17	1.13	1.20
Forestfires	**1.00**	1.08	1.08	1.09
Boston Housing	1.10	1.11	**1.00**	1.10
Hitters	**1.00**	1.10	1.11	1.10
W-D-L	5-0-5	3-0-7	2-0-8	0-0-10

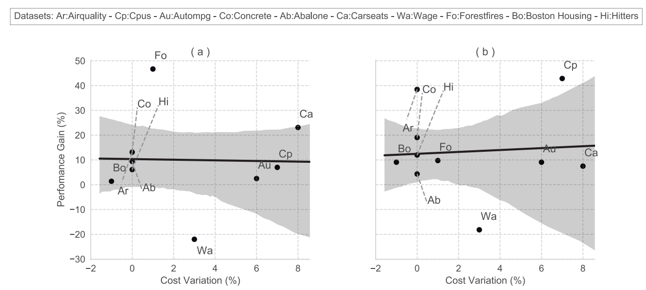

Figure 2: Complexity-performance trade-off considering the cost variation and the performance gain between the CBAS method and the best benchmark (BAGG, RAPT or BOOS): (a) Correlation coefficient, (b) Disagreement measure.

methods is the computational cost. Table 5 exhibits the normalized average computational cost for each pool generation method and each dataset. For all pool generation methods, it is assumed the use of the same hardware. For each method and dataset, the lowest time consumed is made equal to one (reference value highlighted in bold) and the other times are compared to it. For example, considering the Airquality database, the computational cost of the RAPT, BOOS, and CBAS methods are 5% larger than the cost of the BAGG one. Concerning the W-D-L comparison, we note that the BAGG method has the lowest computational cost among the strategies, winning in 50% of the datasets. For the remaining datasets, the RAPT method wins in three of them, while the BOOS approach is the least complex for two datasets (Concrete and Boston Housing). Although the CBAS method does not have the lowest computational cost for any of the datasets, it does not always have the highest cost.

The complexity-performance trade-off considering the cost variation and the performance gain between the CBAS method and the best benchmark is exhibited in Figs. 2(a) and 2(b) for correlation coefficient and disagreement measure, respectively. For each dataset (represented by a labeled point), the computational cost variation and the performance gain, both in percentage, of the CBAS method are compared to the same parameters of the best benchmark (BAGG, RAPT, or

BOOS).

In general, we observe that, depending on the dataset and the diversity measure, the complexity-performance trade-off presents a different behavior. For example, for correlation coefficient as diversity measure and assuming the Forestfires dataset, the best benchmark is the RAPT method (0.15 in Table 2). In this case, we can see in Fig. 2(a) that the CBAS method offers approximately 46.7% of performance gain associated to 1% of cost variation, i.e., practically without increasing complexity. Considering again the correlation coefficient, but now the Carseats dataset (RAPT is the best benchmark according to Table 2), we see that the CBAS method provides roughly 23.1% of performance gain at the price of 8% of cost increasing. Finally, for both diversity measures and assuming the Wage dataset, the best benchmark is the RAPT method (0.50 in Table 2 and 0.22 in Table 3), even better than the CBAS approach. In this case, we have a decreasing of the performance gain around 20% with a increasing of 3% in terms of computational cost.

With this in mind, the complexity-performance trade-off of the CBAS pool generation method depends on both the dataset and the diversity measurement. In most of cases, the CBAS method gives a performance gain with increasing or similar cost.

Conclusion

In this study, a new combining-based method of pool generation was proposed to increase the diversity of ensemble learning systems in regression problems. Traditional ensemble-based algorithms such as bagging, random patches, and boosting, were used as reference for comparison. Also, our proposal is compared to a combining-based technique in terms of root mean square error. For all experiments, k-Nearest Neighbors (k-NN) algorithm was assumed as the base regressor and ten public datasets covering different types of problems were considered. In addition, correlation coeficient and disagreement measure were used to evaluate the diversity of the proposed method as well as the adopted benchmarks. Numerical results showed that the proposed combining-based method won the single ensemble generator approaches in 80% of the datasets considered in this work. In these cases, our proposal gave a performance gain with increasing or similar cost. In terms of root mean square error, our proposal of a single pool generation reached an equivalent performance, without compromising the accuracy of the predictions, when compared to another combined approach using three pools of regressors.

References

A. A. Mohammed, R. Minhas, Q. M. J. W., and Sid-Ahmed, M. A. 2012. Human face recognition based on multidimensional PCA and extreme learning machine. *Journal ACM Computing Survey* 44(10-11):2588–2597.

Avnimelech, R., and Intrator, N. 1999. ‘Boosting regression estimators. *Neural Computation* 11(2):499–520.

Bakker, B., and Heskes, T. 2003. Clustering ensembles of neural network models. *Neural networks* 16(2):261–269.

Breiman, L. 1996. Bagging predictors. *Machine Learning* 24(2):123–140.

Breiman, L. 2001. Random forests. *Machine learning* 45(1):5–32.

Brown, G.; Wyatt, J.; Harris, R.; and Yao, X. 2005. Diversity creation methods: a survey and categorisation. *Information Fusion* 6(1):5–20.

Brown, G. 2004. *Diversity in neural network ensembles*. Ph.D. Dissertation, University of Birmingham, UK.

C. K. S. Leung, R. K. M., and Wang, Y. 2014. A machine learning approach for stock price prediction. In *Proc. of the* 18th Int. Database Engineering & Applications Symp., 274–277.

Demšar, J. 2006. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research* 7(Jan):1–30.

Dietterich, T. G. 2000. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, 1–15. Springer.

Dutta, H. 2009. Measuring diversity in regression ensembles. In *Proc. of the* 4th Indian Int. Conf. on Artif. Intelligence (IICAI 2009), volume 9, 1–17.

Hadavandi, E.; Shahrabi, J.; and Shamshirband, S. 2015. A novel boosted-neural network ensemble for modeling multi-target regression problems. *Engineering Applications of Artificial Intelligence* 45:204–219.

Hernández-Lobato, D.; Martínez-Muñoz, G.; and Suárez, A. 2006. Pruning in ordered regression bagging ensembles. In *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, 1266–1273.

Ho, T. K. 1998. The Random Subspace Method for Constructing Decision Forest. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20(8):832–844.

J. Bollen, H. M., and Zeng, X. 2011. Twitter mood predicts the stock market. *Journal of Comput. Science* 2(1):1–8.

J. Mendes-Moreira, C. Soares, A. M. J., and de Sousa, J. F. 2012. Ensemble approaches for regression: a survey. *ACM Computing Surveys* 45(1):1–40.

Kotsiantis, S., and Kanellopoulos, D. 2012. Combining bagging, boosting and random subspace ensembles for regression problems. *International Journal of Innovative Computing, Information and Control* 8(6):3953–3961.

Kuhn, M., and Johnson, K. 2013. *Applied Predictive Modeling*. New York: Springer.

Kuncheva, L. I. 2004. *Combining pattern classifiers: methods and algorithms*. John Wiley & Sons.

Louppe, G., and Geurts, P. 2012. Ensembles on random patches. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 346–361.

Panov, P., and Džeroski, S. 2007. Combining bagging and random subspaces to create better ensembles. In *International Symposium on Intelligent Data Analysis*, 118–129.

Rooney, N.; Patterson, D.; Anand, S.; and Tsymbal, A. 2004. Dynamic integration of regression models. *Proceedings of the International Workshop on Multiple Classifier Systems. Lecture Notes in Computer Science, vol. 3181* 164–173.

Skalak, D. B., et al. 1996. The sources of increased accuracy for two proposed boosting algorithms. In *Proc. American Association for Artificial Intelligence, AAAI-96, Integrating Multiple Learned Models Workshop*, volume 1129, 1133–1273.

Zhang, C., and Ma, Y. 2012. *Ensemble machine learning: methods and applications*. Springer.