DOI 10.15622/ia.22.2.6

YU. ZELENKOV

OPTIMIZATION OF THE REGRESSION ENSEMBLE SIZE

Zelenkov Yu. **Optimization of the Regression Ensemble Size.**

Abstract. Ensemble learning algorithms such as bagging often generate unnecessarily large models, which consume extra computational resources and may degrade the generalization ability. Pruning can potentially reduce ensemble size as well as improve performance; however, researchers have previously focused more on pruning classifiers rather than regressors. This is because, in general, ensemble pruning is based on two metrics: diversity and accuracy. Many diversity metrics are known for problems dealing with a finite set of classes defined by discrete labels. Therefore, most of the work on ensemble pruning is focused on such problems: classification, clustering, and feature selection. For the regression problem, it is much more difficult to introduce a diversity metric. In fact, the only such metric known to date is a correlation matrix based on regressor predictions. This study seeks to address this gap. First, we introduce the mathematical condition that allows checking whether the regression ensemble includes redundant estimators, i.e., estimators, whose removal improves the ensemble performance. Developing this approach, we propose a new ambiguity-based pruning (AP) algorithm that bases on error-ambiguity decomposition formulated for a regression problem. To check the quality of AP, we compare it with the two methods that directly minimize the error by sequentially including and excluding regressors, as well as with the state-of-art Ordered Aggregation algorithm. Experimental studies confirm that the proposed approach allows reducing the size of the regression ensemble with simultaneous improvement in its performance and surpasses all compared methods.

Keywords: ensemble pruning, regression, ensemble learning, error-ambiguity decomposition, diversity of regressors.

1. Introduction. Ensemble learning is a method that combines several models, which are obtained by applying a learning process to a given problem. The main idea of this approach is that models view the problem from different points. Therefore, their combination improves robustness and accuracy either in classification or regression. However, the existing ensemble learning algorithms often generate unnecessarily large ensembles, which consume extra computational resources and may degrade the generalization ability [1]. There are theoretical and empirical publications that have shown that small ensembles can be better than large ensembles [1, 2].

The ensemble learning process can be described as the overproduce-and-choose approach [3]. The overproduction phase is aimed to produce a large set $\mathcal{F}_0 = [f_i, i = 1 \dots M_0]$ of candidate base models f_i. The choice phase is intended to select the subset of models $\mathcal{F} \subseteq \mathcal{F}_0$ that can be combined to achieve optimal performance.

In general, there are two ways to realize the choice phase. The first is a sequential selection when the algorithm starts from an empty set and

sequentially adds models according to some metric. Often the selection is combined with model generation. The second is pruning, in that case, the ensemble includes all candidate models, and the goal is to choose their optimal subset according to some metric.

Both the selection and pruning have the potential advantage of reducing ensemble size, and improving performance [4]. However, the selection and pruning of classifiers, rather than regressors, has previously received more attention from researchers [5, 6]. Some of these methods have been adapted to the regression task [7], but there is a lack of theoretical and empirical works dedicated exclusively to the regression problem.

There are theories considering the specifics of regression, in particular, these are the error-ambiguity decomposition [8, 9], which can be applied to develop a pruning algorithm. Here we present an ambiguity-based pruning algorithm that sequentially removes regressors with the worst generalization ability. We compare the performance of this algorithm with a state-of-the-art Ordered Aggregation [10] method also as with two algorithms based on direct optimization of the quality metric.

The rest of the paper organizes as follows. After the literature review, we introduce the mathematical condition that allows checking whether the regression ensemble includes redundant estimators, i.e., estimators, whose removal improves the ensemble performance. Next, on the basis of this approach, we propose the Ambiguity-based Pruning (AP) algorithm. In the last part of the paper, we present the results of experiments on real datasets that confirm that the proposed approach outperforms known methods in terms of accuracy and model complexity.

2. Literature Review. We consider the typical regression problem, and for a clear presentation, we establish the notation that will be used below. Take X to be the vector space of all possible inputs, and $Y \in \mathbb{R}$ to be the vector space of all possible outputs and there exists some unknown probability distribution over the product space $X \times Y$. The training set $D_{train} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_N, y_N)\}$ is made up of N samples from this probability distribution. Every $\mathbf{x}_i$ is an input vector from the training data, and y_i is the corresponding output. The goal is to induce on the basis of the training set a function $f: X \to Y$ that approximates an unknown true function such as $f(\mathbf{x}) \sim y$. The quality of the approximation is given by the generalization error, which usually is a mean squared error:

$$MSE(f) = \mathbb{E}[(f(\mathbf{x}) - y)^2].$$

Because it is not possible to determine this true error of a model f, the error is estimated on a different set of data D_{test}, containing K samples:

$$MSE(f) \approx \frac{1}{K}\sum_{i=1}^{K}[(f(\mathbf{x}_i) - y_i)]^2.$$

We will consider a regression ensemble, i.e., the combination of a few models since this should improve robustness and accuracy. In regression problems, ensemble integration most often is performed using a linear combination of the base models [3, 6].

$$f_E(\mathbf{x}) = \sum_{i=1}^{M}[w_i(\mathbf{x}) * f_i(\mathbf{x})],$$

where $w_i(\mathbf{x})$'s are the weighting functions, and M is a number of models $f_i(\mathbf{x}) \in \mathcal{F}$ in the ensemble (models f_i are often also referred to as estimators, predictors, regressors or learners). It follows from this that the problem of the models' selection is closely related to choosing the optimal weights. From now on, we will use notation f instead of $f(\mathbf{x})$ and w instead of $w(\mathbf{x})$ for simplicity.

Study [8] proposed the ambiguity decomposition of ensemble error that separates the weighted average error of the individual regressors and variability among their estimations at an arbitrary single data point:

$$(f_E - y)^2 = \sum_{i=1}^{M} w_i(f_i - y)^2 - \sum_{i=1}^{M} w_i(f_i - f_E)^2. \quad (1)$$

The first term $w_i(f_i - y)^2$ is the weighted error of the i-th ensemble member. The second, $w_i(f_i - f_E)^2$, is the ambiguity term, measuring the amount of variability among the ensemble member answers for this pattern. This equation explains why the quadratic error of the ensemble is less than or equal to the average quadratic error of the component estimators. Note that this decomposition is valid only for convex ensembles [9], i.e., when $f_E = \sum_{i=1}^{M} w_i f_i$ and $\sum_{i=1}^{M} w_i = 1, w_i \geq 0$.

An essential assumption of ensemble learning is that the base models should be sensitive to variations in the training set, so Decision Trees (DT) and Neural Networks (NN) usually are used.

The most popular ensemble learning algorithms for regression are Bagging, Random Forest, Negative Correlation and Gradient Boosting. The *Bagging* algorithm [11] employs bootstrap sampling to generate many training sets from the original training set and then trains a model for each of those training sets. The component predictions are combined via simple averaging for regression tasks. Bagging can be used both with DT and NN.

The *Random Forest* [12] algorithm is similar to Bagging in that they both resample the data. However, Random Forest is based exclusively on DT, when it performs splitting, a random sample of the features is also selected. In *Negative Correlation* learning [13], all the individual estimators in the ensemble are trained simultaneously and interactively through the correlation penalty terms in their error functions. This approach is used exclusively with NN since in that case there is a possibility to include a penalty in the formula for weights tuning in the backpropagation method. The *Gradient Boosting* algorithm [14] on each iteration computes pseudo-residuals and trains a new model using them as a target. Thus, each new estimator attempts to correct the error of its predecessors. The weight of each member is found in the process of a linear search.

The stochastic nature of Bagging and Random Forest leads to ensembles that can be significantly improved by pruning. Many authors used this fact in their research [7]. The family of Boosting methods (including AdaBoost) produces more balanced ensembles in general. However, some researchers report on successful applications of pruning especially in case of the classification problem solved with the AdaBoost algorithm [1, 2].

Different authors proposed different classification schemes of pruning algorithms. In study [6] the authors classify them as partitioning-based and as search-based. Partitioning-based methods divide the pool of models into subgroups. Then, for each subgroup, one or more models are selected using a given selection criterion. Search-based algorithms, in turn, are divided into (1) exponential that search the complete space of models, (2) randomized that use stochastic methods, such as evolutionary algorithms, and (3) sequential that search for a subset of the original pool by iteratively adding or removing models.

In study [1] the authors split pruning algorithms into two categories, (1) selection-based that do not weight each model by a weighting coefficient and either select or reject the learner, and (2) weight-based algorithms that improve the generalization performance of the ensemble by tuning the weight on each ensemble member.

In paper [6] the authors reviewed regression ensemble pruning approaches published before 2008, here we will consider some recent publications on the basis of the approach to the classification proposed in [1]. First, we review some selection-based algorithms.

Study [5] reviewed a family of pruning methods based on modifying the order of estimators in a Bagging ensemble. This order in the original Bagging algorithm is unspecified, and the error of the ensemble generally exhibits a monotonic decrease as a function of the number of estimators.

According to pruning strategies based on ordered aggregation, from the subensemble $\mathcal{F}_{L-1}$ of size $L-1$, the subensemble $\mathcal{F}_L$ of size L is constructed by incorporating a single estimator selected from the set $\mathcal{F}_0 \backslash \mathcal{F}_{L-1}$, which contains the estimators from the original ensemble not included in $\mathcal{F}_{L-1}$. This estimator is identified using a rule that attempts to optimize the performance of the augmented ensemble $\mathcal{F}_L$. The ordered ensemble that includes $L < M$ estimators generally exhibits the error that is below the error of the complete bagging ensemble.

Assuming that the generalization error of the regression ensemble can be expressed as:

$$E = \frac{1}{M^2}\sum_{i=1}^{M}\sum_{j=1}^{M} C_{ij}\,, C_{ij} = \frac{1}{N}\sum_{n=1}^{N}\left[\left(f_i(\mathbf{x}_n) - y_n\right)\left(f_j(\mathbf{x_n}) - y_n\right)\right], \qquad (2)$$

where the correlation matrix C is estimated over a training dataset. In paper [10] the authors proposed Ordered Aggregation (OA) algorithm. The algorithm starts with an empty ensemble and then selects at each iteration the regressor that, when incorporated, reduces the training error (2) of the new ensemble the most.

As for the disadvantages of this method, we can state the following. First, this approach based on the assumption that minimizing training error leads to the minimization of generalization error but in fact, this usually leads to overfitting. Second, time complexity grows exponentially. Third, the number of ensemble members is an external factor; there is no internal stopping criterion.

Later the same authors [7] proposed to use Semidefinite Programming (SDP) introduced in [15] for the classification task. In that case, it is necessary to find a sub-ensemble for which the sum of the elements in the corresponding sub-matrix of C is as low as possible. Note, that it is also NP-hard computational problem.

Authors reported that the minimum of test error obtained either with OA and SDP-pruning is generally below the asymptotic error of the complete bagging ensemble, and pruned ensembles obtained by retaining only 20% of the original bagging ensemble have the best overall performance. The main conclusion of [7] is that the key to improvement in generalization performance is the selection of subsets of regressors whose bias is low and whose correlations are small or negative.

Study [4] extended the OA approach using dynamic ensemble selection technique. Their algorithm consists of two steps. First, the base regressors are trained on bootstrap samples of the training dataset, and the regressor order is found for every instance in the training set. In the second stage, the regressor order that is associated with the training instance closest

to the test instance is retrieved. To find the closest training instance the k-Nearest Neighbors method is used. Empirical testing on several data sets showed that in most cases this approach outperforms original OA-pruning.

There are also examples of the use of evolutionary methods for selecting the optimal set of estimators. For example, a genetic algorithm that searches in the space of candidate base models $\mathcal{F}_0$. In that case, binary fixed-length strings $\{0,1\}^{M_0}$ where $M_0 = |\mathcal{F}_0|$ represent ensembles that form the evolving population. The pruned ensemble includes only estimators that have a value of 1 in the corresponding position of the coding string. In study [16] the authors used such an approach for pruning classification ensembles obtained by the AdaBoost algorithm.

In paper [17] the authors generalized this approach as a multi-objective optimization problem; they proposed simultaneously to minimize two variables – the generalization error of the ensemble and its size.

Some authors claim that weight-based pruning is a more general approach than selection-based [1]. According to [2], the optimal weights of the regression ensemble can be obtained as:

$$w_i = \frac{\sum_{j=1}^{M}(C^{-1})_{ij}}{\sum_{k=1}^{M}\sum_{j=1}^{M}(C^{-1})_{kj}}.$$

However, in real-world applications, some estimators can be quite similar, which makes the correlation matrix C (2) ill-conditioned [2]. The second problem of this formulation is that the optimal combination of weights is computed from the training set, which can lead to overfitting [1].

In paper [1] the authors presented the ensemble pruning algorithm by expectation propagation that approximates the posterior estimation of the weight vector. It produces a «sparse» combination of weights, most of which are zeros. For experiments with the regression, authors used Bagging and Random Forest algorithms with 100 Decision Trees, they reported that the size of the pruned ensemble was reduced, on average, approximately ten times.

In study [18] the authors explored two other weight-based pruning techniques: one based on a cocktail ensemble (CE) algorithm [19] and the second on stacking generalization [20].

CE that was designed for generating the ensemble of ensembles is the following. Since the combination of multiple regressors is an NP-hard problem, the authors of [19] proposed to use the pair-wise combinations of estimators. Through a linear combination of models f_1 and f_2, a new ensemble is formed:

$$f_E(\mathbf{x}) = w_1 f_1(\mathbf{x}) + (1 - w_1) f_2(\mathbf{x}) \text{ w.r.t. } w_1 \in [0,1].$$

Following the error-ambiguity decomposition [8], in study [19] the authors proved that given E_1 and E_2 as generalization errors of f_1 and f_2 respectively, the optimal weight of f_1 is $w_1 = (E_2 - E_1)/2\Delta + 0.5$, where $\Delta = \mathbb{E}\left[\left(f_1(\mathbf{x}) - f_2(\mathbf{x})\right)^2\right]$ is the squared output difference of the two ensembles. E_1, E_2 and Δ can be estimated from training data. CE algorithm starts from the selection base model with the lowest error; then each subsequently selected regressor is the one which reduces the combined estimated error the most.

Stacking is a popular ensemble learning strategy, where the weights of the base models are the regression coefficients of the meta-level regressor [20]. The authors of [18] used the linear regression as a meta-level; the final ensemble consists only of models with greater than zero weight. According to the experimental results, in some cases, the stacked-based approach provides results with less error than standard Bagging and CE and generates the shortest ensembles [18].

Other authors use various randomized algorithms to search for the optimal combination of weights in the ensemble. Study [2] utilized the genetic algorithm with a floating coding scheme to represent weights. Authors reported that their approach outperforms on regression task original Bagging and AdaBoost.R2 algorithms with NN as the base model, an average number of networks after pruning was less than 4.

All approaches discussed above are based on the set of models obtained by ensemble methods that produce estimators using a single learning algorithm. An alternative approach is the usage of heterogeneous models. In this case, a large number of models is generated using various learning algorithms, and the goal is to select those, which produce better generalization. This task also can be formulated as a pruning problem, there is an ensemble that includes all models, and it is necessary to remove non-effective ones. Most of the works in this scientific flow are dedicated to the selection problem [21].

To conclude this review we can note that, in general, ensemble pruning is based on two metrics: diversity and accuracy [22, 23]. Formally the problem is to find a subset of candidate estimators $\{f_k\} \subset \mathcal{F}_0$ with both high accuracy (i.e., lower loss $L(\{f_k\})$) and maximum diversity $D(\{f_k\})$:

$$f_M = \operatorname{argmin}_k L(\{f_k\}) \text{ w.r.t. } D(\{f_k\}) \to \max.$$

Many diversity metrics are known for problems dealing with a finite set of classes defined by discrete labels [22, 24]. Therefore, much work on

ensemble pruning is focused on tasks of classification [25, 26], clustering [27, 28], and selection of an optimal set of features [29, 30].

For the regression problem, it is much more difficult to introduce a diversity metric. In fact, the only such metric known to date is the correlation matrix (2) proposed in [10]. This approach is used in recent papers: [31], but in general the number of studies on regression ensemble pruning published after 2010 is extremely small.

We propose a new approach to regressor selection based on error-ambiguity decomposition, which was introduced in [8], studied in detail in [9], and generalized to all supervised learning problems in [32]. Many authors have pointed out that this decomposition potentially opens ways of optimizing ensembles [33, 34]. Other applications of the error-ambiguity decomposition include, for example, adaptive sampling [35, 36].

3. Ensemble pruning on the basis of Error-Ambiguity decomposition. Study [37] proposed the Reduce-Error pruning algorithm. The first learner incorporated into the ensemble is the one with the lowest error, as estimated on the selection set. The remaining estimators are then sequentially incorporated in the ensemble, one at a time, in such a way that the error of the partial subensemble, estimated on the selection set, is as low as possible. This method was proposed for the classification problem, but we can adapt it to the regression task. So, the regressor f_M that should be incorporated into the subensemble f_E^{M-1} to construct f_E^M is selected by the rule:

$$f_M = \underset{k}{\operatorname{argmin}} \sum_{(x,y)\in D} MSE(f_E^{M-1}(x) \cup f_k(x), y), k \in \mathcal{F}_0 \backslash f_E^{M-1}. \quad (3)$$

Instead of the selection, we can also use a sequential reduction process, i.e., starting from an ensemble that includes all regressors, at each step, one of them whose removal reduces error maximally is deleted. This regressor is selected by the rule:

$$f_M = \operatorname{argmin}_k \sum_{(x,y)\in D} MSE(f_E^M(x) \backslash f_k(x), y), k \in f_E^M. \quad (4)$$

In what follows, we will refer to the algorithm presented by Equation (3) as a Direct Reduce Error (DR), and to the algorithm Equation (4) as a Reverse Reduce Error (RR).

Consider the conditions under which a regressor selected according to Equation (4) must meet to create the maximum effect. Averaging the error – ambiguity decomposition (1) for all N samples in the dataset, we obtain the following formula:

$$\mathbb{E}[(f_E^M - y)^2] = \sum_{i=1}^{M} w_i^M \mathbb{E}[(f_i - y)^2] - \sum_{i=1}^{M} w_i^M \mathbb{E}[(f_i - f_E^M)^2] = \sum_{i=1}^{M} w_i^M (\mathbb{E}[(f_i - y)^2] - \mathbb{E}[(f_i - f_E^M)^2]), \quad (5)$$

where w_i^M is the optimal weight of the i-th estimator in the ensemble of size M.

The first term $E_i = \mathbb{E}[(f_i - y)^2]$ is the mean squared error of regressors in the ensemble, and the second one $A_i^M = \mathbb{E}[(f_i - f_E^M)^2]$ is their ambiguity (note that its value depends on all ensemble members). Thus, to ensure $MSE(f_E^M) = 0$, it is enough to fulfill the conditions:

$$\forall i: E_i = A_i^M, i = 1, \dots, M. \quad (6)$$

This can be explained in terms of diversity. Formula (6) means that the i-th estimator, which satisfies this condition, differs from other members of the ensemble (i.e., its outputs are correlated with outputs of other members) in such a way that its error is compensated by this diversity.

Using Equation (5), we can find the condition when the removal of one regressor will not lead to a growth of the total ensemble error, i.e.:

$$\mathbb{E}[(f_E^M - y)^2] - \mathbb{E}[(f_E^{M-1} - y)^2] \geq 0.$$

Let the regressor that is to be removed have index M. Since the ensemble can be presented as:

$$f_E^M = \sum_{i=1}^{M} w_i^M f_i = \sum_{i=1}^{M-1} w_i^M f_i + w_M^M f_M,$$

after some algebra, we get the condition that the removed regressor must match in order not to reduce the overall performance:

$$w_M^M (E_M - A_M^M) - \left[\sum_{i=1}^{M-1} w_i^{M-1} (E_i - A_i^{M-1}) - \sum_{i=1}^{M-1} w_i^M (E_i - A_i^M) \right] \geq 0. \quad (7)$$

Accordingly, this regressor must have a maximal positive value defined by Equation (7). An essential consequence of this formula is that it allows us to evaluate the potential of the current ensemble for pruning. If

there are no regressors in the ensemble for which the value of Equation (7) is positive, then the pruning of such an ensemble is impossible.

Formula (7) is derived from Equation (4), but Equation (4) presents the algorithm that is more effective computationally since it requires just the computation of the MSE of each regressor. It can be done once before the pruning cycle since the errors of individual estimators do not depend on ensemble composition. The procedure presented by Equation (7) requires the computation of ambiguity value, which depends on the current ensemble composition and therefore changes on each step.

The term in square brackets in Equation (7) represents the difference between the errors of the pruned subensemble and the original ensemble, from which the prediction of deleted regressor is excluded. This value determines the threshold that the value of $w_M^M(E_M - A_M^M)$ of deleted regressor must exceed. But, since we determine these values based on the training set, we are not sure that this takes into account all the information about the actual distribution of data. Thus, the assumption is reasonable that we can omit the term in square brackets and present the rule for choosing a regressor to be deleted as:

$$\max_i [w_i^M (E_i - A_i^M) \geq 0], i = 1, \dots, M. \quad (8)$$

On the one hand, this simplification reduces computational complexity since we exclude the A_i^{M-1} term. On the other hand, it could facilitate the selection of subensemble with greater generalization since it introduces stochasticity in the selection process. In most cases, the regressor that is selected for removal from the ensemble by rule Equation (8) will coincide with the one chosen by Equation (4). Discrepancies can only be observed when the ensemble is already well optimized, and differences defined by Equation (7) tend to zero.

Since for the ensembles based on averaging like Bagging and Random Forest $w_i^M = 1/M$, this rule can be simplified as:

$$\max_i [(E_i - A_i^M) \geq 0], i = 1, \dots, M.$$

We can show that this is equivalent to choosing a regressor with a maximum distance to the line defined by Equation (4) on the half-plane $E_i \geq A_i^M$. The distance from the point $M(x_M, y_M)$ to the line determined by the equation $Ax + By + C = 0$ is defined as $d = |Ax_M + By_M + C|/\sqrt{A^2 + B^2}$. Substituting into this formula the

coordinate values of the ith regressor $(E_i,\ A_i^M)$ and the coefficients of the equation of the straight line $E - A^M = 0$, we obtain $d_i = |E_i - A_i^M|/\sqrt{2}$.

Figure 1 presents the example of the averaging ensemble of 4 regressors trained on the airfoil dataset (its parameters will be given below). The left chart shows the values of E_i and A_i^M of these regressors; a solid line presents Equation (6). The right graph shows the corresponding values of the two terms of Equation (7). As it follows from the data presented on the right graph best candidate for removing is estimator number 2 according to Equation (7) and estimator number 0 according to Equation (8). Corresponding points are marked by red and green circles on the left chart, respectively. The exclusion of estimator 2 will lead to better performance on the training set, but, as said above, it can also lead to a loss of the generalization since the pruned model will be overfitted to training data. The ambiguity of the remaining ensemble members changes when any estimator is removed; therefore, the use of equations (7) and (8) can lead to different final structures. In the next sections, we will present empirical data, which confirms that rule Equation (8) allows us to generate more effective ensembles.

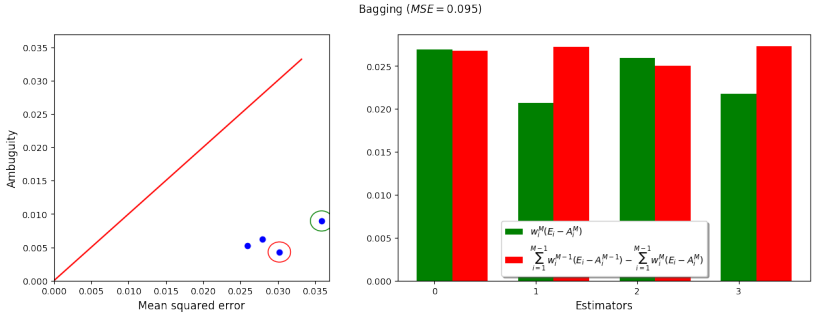

Fig. 1. Bagging ensemble of 4 regressors

The next issue that should be discussed is the data for the creation of an initial pool of regressors and their subsequent selection. Some authors use separated training and selection datasets. Researchers often use this approach to avoid overfitting, but it also can lead to degradation of the generalizing ability of the model since it reduces data available for the model generation. On the other hand, paper [10] reported that the size of the subensembles minimizing the error on the training data tends to be smaller than the optimal subensembles when the error is estimated on an independent selection set. Besides, the selection of the optimal subensemble is an NP-hard problem and not generally feasible in practice.

Our experiments in the preparation of this article confirm that the separation of data into training and selection sets does not improve overall performance. On the contrary, this often leads to a decrease in productivity since candidate models obtained on reduced training datasets do not have sufficient predictive ability.

We compared three different approaches. First, we divided all the data available for training D_{train} into independent sets: the first one for generating models and second for selecting D_{sel}. Table 1 shows the results for cases when the size of the D_{sel} includes 50% and 33% of the data available. In the third experiment, we performed the generation and selection on the same set D_{train}, which included all available data.

Table 1. The pruning model performance for various combinations of training and selection data

Dataset splitting	Bagging	AP	RR	OA	DR
$\|D_{sel}\| = 0.5\|D_{train}\|$	0.273	0.235	0.243	0.259	0.243
$\|D_{sel}\| = 0.33\|D_{train}\|$	0.250	0.228	0.226	0.234	0.219
Training and selection on D_{train}	0.223	0.189	0.189	0.202	0.197

Table 1 shows the best average MSE obtained with 10-fold cross-validation on the airfoil dataset. We used a single-layer artificial neural network (NN) as a base estimator. At each iteration, we first optimized the number of neurons in the NN, and then we generated the initial pool of 100 estimates using the Bagging algorithm and, finally, applied reduction and selection algorithms. We tested two pruning algorithms Reverse Reduce Error (RR) (4) and Ambiguity-based Pruning (AP) (8) and two selection methods Direct Reduced Error (DR) (3) and Ordered Aggregation (OA) [10].

As we can see, the critical factor that defines the pruned model performance is the performance of the initial ensemble that improves when the amount of data available for training grows. Therefore, here and below, we propose to use the single dataset for the generating of candidate models and the selection of optimal subsets.

Summing up the above, the proposed algorithm, which we call Ambiguity-Based Pruning (AP), is presented in Table 2. Since instead of using the exact expression (7) to select the regressor for reduction, we use an approximate value (8), we must stop the algorithm when the current MSE value begins to rise to control convergence and prevent the loss of generalization (lines 9-13).

Table 2. Ambiguity Based Pruning Algorithm

INPUT	Training dataset $D_{train} = \{(x_i, y_i)\}_{i=1}^{N}$, pool f_E of regressors fitted on D_{train}.
1	**Initialize** $e_E = MSE(f_E)$
2	**Repeat**
3	*ensPruned* **is** *False*
4	**For each** f_i **in** f_E **do**
5	$e_i = w_i^M \left(E_i - A_i^M\right)$
6	**End for**
7	$j = \text{argmax}(\boldsymbol{e})$
8	$f_{new} = f_E \backslash f_j$
9	**If** $MSE(f_{new}) \leq e_E$
10	$e_E = MSE(f_{new})$
11	$f_E = f_{new}$
12	*ensPruned* **is** *True*
13	**End if**
14	**Until** *ensPruned* **is** *True*
RETURN	f_E

4. Empirical Analysis and Evaluation. In this section, we assess the generalization performance of Ambiguity-Based Pruning (AP) and compare this technique with other algorithms. As a state-of-art method that sets the baseline for the performance of pruning algorithms, we use Ordered Aggregation (OA). In study [7] the authors conducted an experimental comparison of OA with conventional ensemble methods such as Bagging, AdaBoost.R2, and Negative Correlation, and showed that it surpasses them in performance. Also, according to [7], OA outperforms other known pruning methods such as hybrid ensembles [38], GASEN [2], regularized stacked generalization [39, 40], and performs slightly better than SDP-pruning [7], although the difference between average ranks of OA and SDP is not statistically significant.

In addition to OA, we also included in the set of methods for comparing two algorithms discussed in Section 3, Direct Reduce Error (DR), and Reverse Reduce Error (RR).

It should be noted that the authors of [7] defined the number of estimators in the OA-pruned ensemble as an external parameter. In our experiments, we used a slightly modified OA algorithm: the selection of regressors continues while the error in the training data does not increase (lines 9-13 in Table 2). Such an approach helps to identify the optimal size of the final subensemble. A similar condition was applied to other algorithms tested (RR and DR).

The experiments are carried out on 20 regression problems from the UCI-Repository and other sources. They include real-world problems from different fields of application: industry, biology, urban management, etc. All datasets were scaled to center around zero and have unit variance. Table 3 displays the number of instances (N), the number of attributes (K) and the source of the different datasets considered.

Table 3. Datasets used in the experiments

Dataset	Description	N / K	Source
abalone	Predicting the age of abalone	4177/8	KEEL
airfoil	Aerodynamic and acoustic tests of airfoil	1503/5	UCI
bank8FM	Simulation of how customers choose bank	8192/8	OML
bike	Count of rental bikes	17379/12	UCI
california	Median house value of the block groups	20640/8	KEEL
CASP	Properties of protein tertiary structure	45730/9	UCI
CCPP	Combined Cycle Power Plant	9568/4	UCI
compactiv	Computer systems activity measures	8192/21	KEEL
concrete	Predicting the concrete compressive	1030/8	KEEL
egrid	Electrical grid stability simulated data	10000/12	UCI
elevators	Action taken on the elevators of the aircraft	16599/18	KEEL
facebook	Facebook comment volume	40949/53	UCI
forestFires	Burned area of forest fires in Portugal	517/12	KEEL
house	Median house price in the regions of USA	22784/16	KEEL
kin8nm	Forward kinematics of an 8-link robot arm	8192/8	OML
laser	Far-Infrared-Laser in a chaotic state	993/4	KEEL
stock	Daily stock prices for aerospace companies	950/9	KEEL
supercond	Superconductors and their relevant features	21263/81	UCI
treasury	Economic data of USA	1049/15	KEEL
wankara	The weather information of Ankara	1609/9	KEEL

Data sources:
– KEEL – KEEL dataset: https://sci2s.ugr.es/keel/datasets.php
– OML – OpenML: https://www.openml.org/search?type=data
– UCI – UC Irvine Machine Learning Repository: http///archive.ics.uci.edu/ml/

The experimental protocol is the following. We used 10-fold cross-validation to estimate the mean squared error. The experiments involve building a bagging ensemble of $M = 100$ predictors. Following the work [7],

we use the feed-forward neural network (NN) with a single hidden layer of sigmoidal neurons and a linear unit in the output layer as a base learner. The networks are trained during 1000 epochs using the quasi-Newton optimization method BFGS. Before the generation of the bagging ensemble, the optimal number of hidden units in NN was explored using Bayesian optimization and a separate 5-fold cross-validation procedure. Once the best architecture of NN is found, a bagging ensemble is generated using neural networks with these hyper-parameters. Next, the bagging ensemble is pruned using various pruned algorithms. Finally, we computed the mean value of MSE on all testing folds for all compared algorithms, also as bagging ensemble, to use the last as a benchmark for performance comparison. All computations were performed using the Python programming language and scikit-learn and scikit-optimize libraries.

Table 4 shows the average mean squared error and its standard deviation estimated by 10-fold cross-validation for each dataset and each prediction method.

Table 4. Average mean squared error normalized by the corresponding scaling factor

Dataset	Scaling factor	Bagging	AP	DR	RR	OA
abalone	10^{-1}	4.309 (0.272)	4.318 (0.272)	4.315 (0.272)	4.324 (0.273)	4.319 (0.272)
airfoil	10^{-1}	2.225 (0.112)	1.890 (0.099)	1.973(0.104)	1.887 (0.094)	2.023 (0.109)
bank8FM	10^{-2}	3.470 (0.002)	3.443 (0.002)	**3.440 (0.002)**	3.442 (0.002)	3.467 (0.002)
bike	10^{-1}	1.448 (0.091)	**1.154 (0.065)**	**1.163 (0.066)**	**1.155 (0.065)**	**1.209 (0.071)**
cadata	10^{-1}	2.775 (0.096)	2.685 (0.088)	2.684 (0.089)	2.683 (0.088)	2.756 (0.095)
CASP	10^{-1}	5.609 (0.012)	**5.522 (0.013)**	**5.521 (0.013)**	**5.521 (0.012)**	**5.601 (0.012)**
CCPP	10^{-2}	5.889 (0.005)	**5.833 (0.005)**	**5.833 (0.005)**	**5.833 (0.005)**	5.889 (0.005)
compactiv	10^{-2}	1.738 (0.001)	**1.674 (0.001)**	**1.672 (0.001)**	**1.674 (0.001)**	**1.707 (0.001)**
concrete	10^{-2}	8.629 (0.020)	**8.206 (0.019)**	**8.139 (0.019)**	**8.140 (0.019)**	**8.333 (0.021)**
egrid	10^{-2}	4.773 (0.003)	**4.523 (0.003)**	**4.528 (0.003)**	**4.529 (0.003)**	**4.623 (0.003)**
elevators	10^{-2}	8.829 (0.014)	**8.575 (0.013)**	**8.575 (0.013)**	**8.575 (0.013)**	8.754 (0.014)
facebook	10^{-1}	4.599 (0.188)	4.621 (0.181)	4.672 (0.177)	4.659 (0.180)	4.498 (0.181)
forestFires	10^{0}	1.218 (1.664)	1.281 (1.585)	2.722 (2.429)	1.408 (1.553)	1.533 (1.614)
house	10^{-1}	4.207 (0.057)	**4.138 (0.055)**	**4.140 (0.056)**	**4.140 (0.056)**	**4.193 (0.057)**
kin8nm	10^{-2}	9.233 (0.008)	**8.648 (0.008)**	**8.652 (0.008)**	**8.646 (0.008)**	**8.816 (0.008)**
laser	10^{-2}	1.362 (0.015)	1.360 (0.015)	1.411 (0.016)	1.430 (0.017)	1.369 (0.015)
stock	10^{-1}	1.486 (0.112)	1.498 (0.126)	1.541 (0.128)	1.514 (0.126)	1.535 (0.121)
supercond	10^{-1}	1.759 (0.128)	1.737 (0.123)	1.741 (0.123)	1.735 (0.123)	1.754 (0.127)
treasury	10^{-3}	4.027 (0.002)	3.818 (0.002)	3.822 (0.002)	3.804 (0.002)	4.003 (0.002)
wankara	10^{-3}	6.099 (0.001)	6.022 (0.001)	6.043 (0.001)	6.081 (0.001)	6.152 (0.001)

The figures displayed are scaled by a factor shown in the first column of the table. To check whether the observed improvements in error are statistically significant, a paired Wilcoxon test was performed on cross-validation data for each dataset. Error values that are significantly better than bagging at an α - value of 0.05 are highlighted in boldface. Error values that are significantly better than OA at an α-value of 0.05 are underlined. As it follows from Table 4, there were no algorithms that perform analysis statistically worse than bagging and OA on the datasets selected for the experiment.

Following the framework proposed by J. Demšar [41], we also conducted the Friedman test to compare the overall performance of different methods in the collection. The results obtained (F_F = 59.14, the corresponding p-value is 2E-11, and the critical value of χ^2 distribution is 11.07) confirm that the hypothesis of the equivalent performance of all algorithms should be rejected at $\alpha = 0.05$. If the null hypothesis is rejected, we can proceed with a post-hoc Nemenyi test. Figure 2 displays the results of this test for $\alpha = 0.05$. The differences in performance between algorithms whose average ranks are further than a critical distance (CD) are statistically significant. The obtained value of the CD is 1.686. In Figure 2, algorithms whose differences in performance are not statistically significant are connected with a solid line.

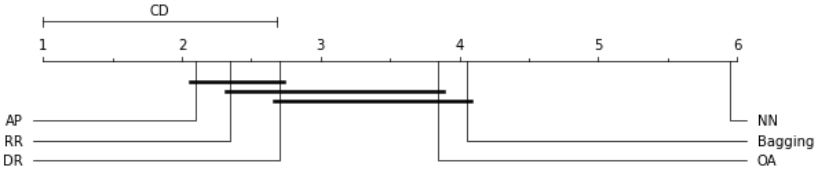

Fig. 2. Comparison of algorithms' performance against each other with the Nemenyi test. Groups of methods that are not significantly different (at $\alpha = 0.05$) are connected with a solid black line

As follows from the data presented in Figure 2, four groups of models are distinguished, and models inside one group have statistically comparable results. The first group includes algorithms of pruning AP, RR, and DR, while AP has a slight advantage within this group. The second group consists of the algorithms RR, DR, and OA, the OA; this group showed the worst results. The third group comprises the bagging algorithm, as well as methods for pruning the DR and OA. The fourth group is represented only by a single-layer feed-forward neural network (NN).

The following conclusions can be drawn from the presented results. First, pruning methods can improve the performance of the initial set of regressors generated using the bagging algorithm. Data n Table 4 confirms this conclusion generally, but we should note that there are a few exceptions, namely results for datasets: abalone, forestFires, and stock. Second, methods based on the sequential exclusion of estimators from the initial ensemble (AP and RR) are superior to sequential inclusion algorithms (DR and OA). One possible explanation is that in the case of the exclusion, the algorithm has more information about the interaction of all estimators in the ensemble. In the case of the AP algorithm, this information is presented explicitly in the ambiguity term (1). When the algorithm implements an inclusion scenario, only information about the pair interaction of estimators is available explicitly given by correlation matrix Equation (2) in the OA algorithm.

Another critical point associated with the quality of pruning algorithms is the size of the ensemble obtained after the algorithm application. Table 5 presents these data.

Table 5. Average number of regressors in the pruned sub ensemble

Data	NN	AP	DR	RR	OA
abalone	10.5 (3.4)	12.6 (2.1)	**11.2 (2.5)**	11.9 (1.9)	98.2 (5.7)
airfoil	10.5 (3.5)	9.8 (3.0)	**7.6 (1.8)**	8.8 (2.3)	23.9 (5.2)
bank8FM	17.1 (1.8)	9.8 (2.0)	9.7 (1.9)	**9.6 (2.1)**	94.8 (4.0)
bike	14.9 (3.1)	11.0 (6.6)	**7.2 (1.9)**	7.7 (1.6)	22.3 (3.5)
cadata	13.2 (5.3)	8.8 (2.0)	**6.5 (1.8)**	8.0 (2.4)	65.2 (10.4)
CASP	19.1 (0.9)	10.8 (2.1)	10.4 (1.9)	**10.3 (1.9)**	89.7 (4.9)
CCPP	13.5 (2.0)	5.1 (1.5)	**4.2 (1.5)**	5.1 (1.5)	99.5 (1.0)
compactiv	11.2 (4.9)	9.9 (2.9)	8.5 (2.6)	**8.5 (2.1)**	53.4 (5.2)
concrete	16.1 (1.7)	14.7 (3.2)	**13.5 (2.5)**	13.9 (2.8)	27.2 (5.8)
egrid	18.9 (1.0)	14.4 (2.8)	**11.1 (1.3)**	12.4 (1.6)	34.2 (5.8)
elevators	7.9 (2.6)	11.8 (3.3)	**6.7 (2.6)**	9.7 (1.7)	70.5 (8.2)
facebook	11.9 (2.4)	8.7 (2.1)	8.3 (1.6)	**8.1 (1.4)**	31.9 (4.8)
forestFires	9.1 (3.1)	43.1 (10.1)	**13.1 (9.8)**	23.7 (3.8)	7.9 (3.3)
house	16.2 (2.3)	14.6 (3.0)	**12.3 (1.9)**	13.4 (2.9)	60.2 (7.4)
kin8nm	18.0 (0.9)	12.4 (2.4)	**9.3 (1.3)**	11.2 (1.3)	39.3 (4.2)
laser	11.1 (3.9)	13.8 (5.5)	**7.8 (2.9)**	8.1 (1.5)	24.0 (5.8)
stock	13.1 (5.0)	12.0 (3.4)	**9.7 (3.4)**	10.9 (3.1)	28.4 (3.8)
supercond	9.2 (4.4)	9.5 (2.0)	**8.9 (1.7)**	9.2 (1.8)	55.4 (8.6)
treasury	11.8 (4.4)	7.3 (2.2)	**5.5 (1.6)**	6.7 (1.6)	45.9 (7.5)
wankara	9.4 (3.7)	13.6 (3.3)	**9.9 (2.2)**	11.0 (1.8)	66.9 (10.5)

The first column of Table 5 (NN) contains the mean and the standard deviation of the number of neurons obtained during the optimization of NN at each iteration in the process of 10-fold cross-validation. Other columns show the mean and standard deviation of the number of base estimators (NN) obtained from the 10-fold cross-validation for each algorithm. Better value in each line is highlighted with bold. As we can see, the best value is achieved by methods that directly minimize the prediction error (DR and RR), AP creates somewhat larger ensembles, while significantly ahead of OA.

We also conducted the Friedman and Nemenyi tests for these data. Results confirm that the sizes of ensembles obtained with pruning algorithms are statistically different at α=0.05. Figure 3 presents the results of the Nemenyi test.

The ideal pruning algorithm should reduce both the error and size of the ensemble relative to the original bagging ensemble. This condition is satisfied in most cases; however, Table 4 shows that pruning algorithms increase the error for some datasets. However, this degradation is not statistically significant (Table 4), and the pruned model can be ten or more times smaller than the original bagging ensemble, which can be crucial for devices with limited memory (e.g., smart sensors). Distributions of AP, DR, and RR results are very similar (with a slight advantage of AP in terms of error); OA in our experiments demonstrates the worst performance in terms of error and model size.

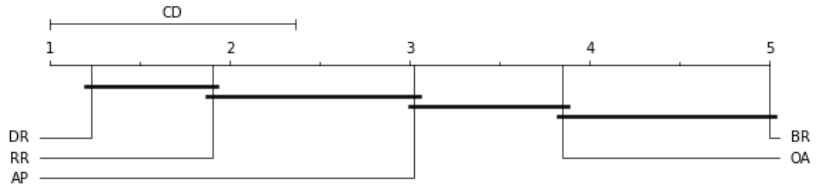

Fig. 3. Comparison of the size of ensembles with the Nemenyi test. CD = 1.364

5. Conclusions. In this work, we investigated regression ensembles pruning methods since previous research did not pay enough attention to this area.

The first of the main contributions of our work is the formal condition (7), which allows estimating the potential to reduce the size of the convex regression ensemble. If there are no regressors in the ensemble for which the value of Equation (7) is positive, then the reduction of such an ensemble is impossible.

Next, we proposed an Ambiguity-based pruning algorithm based on well-known error-ambiguity decomposition and compared its performance with other pruning techniques such as minimization of error using direct and reverse approaches. The results of experiments with real datasets show that Ambiguity-based pruning outperforms in most cases other algorithms also the state-of-art Ordered Aggregation algorithm.

References

1. Chen H., Tiňo P., Yao X. Predictive ensemble pruning by expectation propagation. IEEE Transactions on Knowledge & Data Engineering. 2009. vol. 21. no. 7. pp. 999–1013.
2. Zhou Z., Wu J., Tang W. Ensembling neural networks: many could be better than all. Artificial Intelligence. 2002. vol. 137. no. 1–2. pp. 239–263.
3. Sagi O., Rokach L. Ensemble learning: A survey. WIREs Data Mining and Knowledge Discovery. 2018. vol. 8. no. 4. e1249.
4. Dias K., Windeatt T. Dynamic ensemble selection and instantaneous pruning for regression. Proc. of the ESANN. Bruges, 2014. pp. 643–648.
5. Martínez-Muñoz G., Hernández-Lobato D., Suárez A. An analysis of ensemble pruning techniques based on ordered aggregation. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2009. vol. 31. no. 2. pp. 245–259.
6. Mendes-Moreira J., Soares C., Jorge A.M., de Sousa J.F. Ensemble approaches for regression: A survey. ACM Computing Surveys. 2012. vol. 45. no. 1. Article 10.
7. Hernández-Lobato D., Martínez-Muñoz G., Suárez A. Empirical Analysis and Evaluation of Approximate Techniques for Pruning Regression Bagging Ensembles. Neurocomputing. 2011. vol. 74. no. 12–13. pp. 2250–2264.
8. Krogh A., Vedelsby J. Neural network ensembles, cross validation, and active learning. Advances in neural information processing systems. 1995. pp. 231–238.
9. Brown G., Wyatt J.L., Tino P. Managing diversity in regression ensembles. Journal of Machine Learning Research. 2005. vol. 6. pp. 1621–1650.
10. Hernández-Lobato D., Martínez-Muñoz G., Suárez A. Pruning in ordered regression bagging ensembles. Proceedings of the International Joint Conference on Neural Networks, Vancouver, 2006. pp. 1266–1273.
11. Breiman L. Bagging predictors. Machine Learning. 1996. vol. 24. no. 2. pp. 123–140.
12. Breiman L. Random forests. Machine learning. 2001. vol. 45. no. 1. pp. 5–32.
13. Liu Y., Yao X. Ensemble learning via negative correlation. Neural networks. 1999. vol. 12. no. 10. pp. 1399–404.
14. Friedman J.H. Greedy function approximation: A gradient boosting machine. Annals of statistics. 2001. vol. 29. no. 5. pp. 1189–1232.
15. Zhang Y., Burer S., Street W.N. Ensemble pruning via semidefinite programming. Journal of Machine Learning Research. 2006. vol. 7. pp. 1315–1338.
16. Hernández-Lobato D., Hernández-Lobato J.M., Ruiz-Torrubiano R., Valle Á. Pruning adaptive boosting ensembles by means of a genetic algorithm. International Conference on Intelligent Data Engineering and Automated Learning. Springer, 2006. pp. 322–329.
17. Qian C., Yu Y., Zhou Z. Pareto Ensemble Pruning. Proceedings of the 29th AAAI Conference on Artificial Intelligence. Austin, 2015. pp. 2935–2941.
18. Sun Q., Pfahringer B. Bagging ensemble selection for regression. Australasian Joint Conference on Artificial Intelligence. Sydney, 2012. pp. 695–706.
19. Yu Y., Zhou Z.H., Ting K.M. Cocktail ensemble for regression. Proceedings of ICDM'07, 2007. pp. 721–726.
20. Wolpert D.H. Stacked generalization. Neural Networks. 1992. vol. 5. pp. 241–259.
21. Caruana R., Niculescu-Mozil A., Crew G., Ksikes A. Ensemble selection from libraries of models. Proceedings of the ICML'04. Banf, 2004. pp. 18–25.

22. Bian Y., Wang Y., Yao Y., Chen H. Ensemble pruning based on objection maximization with a general distributed framework. IEEE Transactions on Neural Networks and Learning Systems. 2020. vol. 31. no. 9. pp. 3766–3774.
23. Mao S., Chen J., Jiao L., Gou S., Wang R. Maximizing diversity by transformed ensemble learning. Applied Soft Computing. 2019. vol. 82. p. 105580.
24. Zhou Z. Machine learning. Springer, 2021. 472 p.
25. Guo H., Liu H., Li R., Wu C., Guo Y., Xu M. Margin & diversity based ordering ensemble pruning. Neurocomputing. 2018. vol. 275. pp. 237–246.
26. Lustosa Filho J.A.S., Canuto A.M., Santiago R.H.N. Investigating the impact of selection criteria in dynamic ensemble selection methods. Expert Systems with Applications. 2018. vol. 106. pp. 141–153.
27. Fan Y., Tao L., Zhou Q., Han X. Cluster ensemble selection with constraints. Neurocomputing. 2017. vol. 235. pp. 59–70.
28. Golalipour K., Akbari E., Hamidi S.S., Lee M., Enayatifar R. From clustering to clustering ensemble selection: A review. Engineering Applications of Artificial Intelligence. 2021. vol. 104. p. 104388.
29. Zhang C., Wu Y., Zhu M. Pruning variable selection ensembles. Statistical Analysis and Data Mining: The ASA Data Science Journal. 2019. vol. 12. no. 3. pp. 168–184.
30. Baron G. Greedy selection of attributes to be discretized. (Ed.: Hassanien A.) Machine Learning Paradigms: Theory and Application. Studies in Computational Intelligence. Springer, Cham, 2019. vol. 801. pp. 45–67.
31. Khairalla M.A.E. Metaheuristic ensemble pruning via greedy-based optimization selection. International Journal of Applied Metaheuristic Computing. 2022. vol. 13. no. 1. pp. 1–22.
32. Jiang Z., Liu H., Fu B., Wu Z. Generalized ambiguity decompositions for classification with applications in active learning and unsupervised ensemble pruning. Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17), 2017. pp. 2073–2079.
33. Dong X., Yu Z., Cao W., Shi Y., Ma Q. A survey on ensemble learning. Frontiers of Computer Science. 2020. vol. 14. no. 2. pp. 241–258.
34. Shahhosseini M., Hu G., Pham H. Optimizing ensemble weights and hyperparameters of machine learning models for regression problems. Machine Learning with Applications. 2022. vol. 7. p. 100251.
35. Fuhg J., Fau A., Nackenhorst U. State-of-the-Art and Comparative Review of Adaptive Sampling Methods for Kriging. Archives of Computational Methods in Engineering. 2021. vol. 28. pp. 2689–2747.
36. Liu H., Ong Y.-S., Cai J. A survey of adaptive sampling for global metamodeling in support of simulation-based complex engineering design. Structural and Multidisciplinary Optimization. 2018. vol. 57. no. 1. pp. 393–416.
37. Margineantu D.D., Dietterich T.G. Pruning adaptive boosting. Proc. of 14th International Conference on Machine Learning. ICML, 1997. pp. 211–218.
38. Hsu K.W. A theoretical analysis of why hybrid ensembles work. Computational Intelligence and Neuroscience. 2017. vol. 2017. p. 1930702.
39. Yao Y., Pirš G., Vehtari A., Gelman A. Bayesian hierarchical stacking: Some models are (somewhere) useful. Bayesian Analysis. 2022. vol. 17. no. 4. pp. 1043–1071.
40. Nuzhny A.S. Bayes regularization in the selection of weight coefficients in the predictor ensembles. Proc. ISP RAS, 2019. vol. 31. no 4. pp. 113–120. (in Russ.).
41. Demšar J. Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research. 2006. vol. 7. pp. 1–30.

Zelenkov Yuri — Ph.D., Dr.Sci., Professor, Graduate school of business, HSE University. Research interests: modeling of soft systems, machine learning, knowledge management. The number of publications — 80. yzelenkov@hse.ru; 28/11, Shabolovka St., 119049, Moscow, Russia; office phone: +7(495)771-3232.

DOI 10.15622/ia.22.2.6

Ю.А. Зеленков

ОПТИМИЗАЦИЯ РАЗМЕРА АНСАМБЛЯ РЕГРЕССОРОВ

Зеленков Ю.А. **Оптимизация размера ансамбля регрессоров.**

Аннотация. Алгоритмы обучения ансамблей, такие как bagging, часто генерируют неоправданно большие композиции, которые, помимо потребления вычислительных ресурсов, могут ухудшить обобщающую способность. Обрезка (pruning) потенциально может уменьшить размер ансамбля и повысить точность; однако большинство исследований сегодня сосредоточены на использовании этого подхода при решении задачи классификации, а не регрессии. Это связано с тем, что в общем случае обрезка ансамблей основывается на двух метриках: разнообразии и точности. Многие метрики разнообразия разработаны для задач, связанных с конечным набором классов, определяемых дискретными метками. Поэтому большинство работ по обрезке ансамблей сосредоточено на таких проблемах: классификация, кластеризация и выбор оптимального подмножества признаков. Для проблемы регрессии гораздо сложнее ввести метрику разнообразия. Фактически, единственной известной на сегодняшний день такой метрикой является корреляционная матрица, построенная на предсказаниях регрессоров. Данное исследование направлено на устранение этого пробела. Предложено условие, позволяющее проверить, включает ли регрессионный ансамбль избыточные модели, т. е. модели, удаление которых улучшает производительность. На базе этого условия предложен новый алгоритм обрезки, который основан на декомпозиции ошибки ансамбля регрессоров на сумму индивидуальных ошибок регрессоров и их рассогласованность. Предложенный метод сравнивается с двумя подходами, которые напрямую минимизируют ошибку путем последовательного включения и исключения регрессоров, а также с алгоритмом упорядоченного агрегирования (Ordered Aggregation). Эксперименты подтверждают, что предложенный метод позволяет уменьшить размер ансамбля регрессоров с одновременным улучшением его производительности и превосходит все сравниваемые методы.

Ключевые слова: обрезка ансамбля, ансамбль регрессоров, обучение ансамбля, декомпозиция ошибка-разнообразие, разнообразие регрессоров.

Литература

1. Chen H., Tiňo P., Yao X. Predictive ensemble pruning by expectation propagation. IEEE Transactions on Knowledge & Data Engineering. 2009. vol. 21. no. 7. pp. 999–1013.
2. Zhou Z., Wu J., Tang W. Ensembling neural networks: many could be better than all. Artificial Intelligence. 2002. vol. 137. no. 1–2. pp. 239–263.
3. Sagi O., Rokach L. Ensemble learning: A survey. WIREs Data Mining and Knowledge Discovery. 2018. vol. 8. no. 4. e1249.
4. Dias K., Windeatt T. Dynamic ensemble selection and instantaneous pruning for regression. Proc. of the ESANN. Bruges, 2014. pp. 643–648.
5. Martínez-Muñoz G., Hernández-Lobato D., Suárez A. An analysis of ensemble pruning techniques based on ordered aggregation. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2009. vol. 31. no. 2. pp. 245–259.
6. Mendes-Moreira J., Soares C., Jorge A.M., de Sousa J.F. Ensemble approaches for regression: A survey. ACM Computing Surveys. 2012. vol. 45, no. 1. Article 10.

7. Hernández-Lobato D., Martínez-Muñoz G., Suárez A. Empirical Analysis and Evaluation of Approximate Techniques for Pruning Regression Bagging Ensembles. Neurocomputing. 2011. vol. 74. no. 12–13. pp. 2250–2264.
8. Krogh A., Vedelsby J. Neural network ensembles, cross validation, and active learning. Advances in neural information processing systems. 1995. pp. 231–238.
9. Brown G., Wyatt J.L., Tino P. Managing diversity in regression ensembles. Journal of Machine Learning Research. 2005. vol. 6. pp. 1621–1650.
10. Hernández-Lobato D., Martínez-Muñoz G., Suárez A. Pruning in ordered regression bagging ensembles. Proceedings of the International Joint Conference on Neural Networks, Vancouver, 2006. pp. 1266–1273.
11. Breiman L. Bagging predictors. Machine Learning. 1996. vol. 24. no. 2. pp. 123–140.
12. Breiman L. Random forests. Machine learning. 2001. vol. 45. no. 1. pp. 5–32.
13. Liu Y., Yao X. Ensemble learning via negative correlation. Neural networks. 1999. vol. 12. no. 10. pp. 1399–404.
14. Friedman J.H. Greedy function approximation: A gradient boosting machine. Annals of statistics. 2001. vol. 29. no. 5. pp. 1189–1232.
15. Zhang Y., Burer S., Street W.N. Ensemble pruning via semidefinite programming. Journal of Machine Learning Research. 2006. vol. 7. pp. 1315–1338.
16. Hernández-Lobato D., Hernández-Lobato J.M., Ruiz-Torrubiano R., Valle Á. Pruning adaptive boosting ensembles by means of a genetic algorithm. International Conference on Intelligent Data Engineering and Automated Learning. Springer, 2006. pp. 322–329.
17. Qian C., Yu Y., Zhou Z. Pareto Ensemble Pruning. Proceedings of the 29th AAAI Conference on Artificial Intelligence. Austin, 2015. pp. 2935–2941.
18. Sun Q., Pfahringer B. Bagging ensemble selection for regression. Australasian Joint Conference on Artificial Intelligence. Sydney, 2012. pp. 695–706.
19. Yu Y., Zhou Z.H., Ting K.M. Cocktail ensemble for regression. Proceedings of ICDM'07, 2007. pp. 721–726.
20. Wolpert D.H. Stacked generalization. Neural Networks. 1992. vol. 5. pp. 241–259.
21. Caruana R., Niculescu-Mozil A., Crew G., Ksikes A. Ensemble selection from libraries of models. Proceedings of the ICML'04. Banf, 2004. pp. 18–25.
22. Bian Y., Wang Y., Yao Y., Chen H. Ensemble pruning based on objection maximization with a general distributed framework. IEEE Transactions on Neural Networks and Learning Systems. 2020. vol. 31. no. 9. pp. 3766–3774.
23. Mao S., Chen J., Jiao L., Gou S., Wang R. Maximizing diversity by transformed ensemble learning. Applied Soft Computing. 2019. vol. 82. p. 105580.
24. Zhou Z. Machine learning. Springer, 2021. 472 p.
25. Guo H., Liu H., Li R., Wu C., Guo Y., Xu M. Margin & diversity based ordering ensemble pruning. Neurocomputing. 2018. vol. 275. pp. 237–246.
26. Lustosa Filho J.A.S., Canuto A.M., Santiago R.H.N. Investigating the impact of selection criteria in dynamic ensemble selection methods. Expert Systems with Applications. 2018. vol. 106. pp. 141–153.
27. Fan Y., Tao L., Zhou Q., Han X. Cluster ensemble selection with constraints. Neurocomputing. 2017. vol. 235. pp. 59–70.
28. Golalipour K., Akbari E., Hamidi S.S., Lee M., Enayatifar R. From clustering to clustering ensemble selection: A review. Engineering Applications of Artificial Intelligence. 2021. vol. 104. p. 104388.
29. Zhang C., Wu Y., Zhu M. Pruning variable selection ensembles. Statistical Analysis and Data Mining: The ASA Data Science Journal. 2019. vol. 12. no. 3. pp. 168–184.
30. Baron G. Greedy selection of attributes to be discretized. (Ed.: Hassanien A.) Machine Learning Paradigms: Theory and Application. Studies in Computational Intelligence. Springer, Cham, 2019. vol. 801. pp. 45–67.

31. Khairalla M.A.E. Metaheuristic ensemble pruning via greedy-based optimization selection. International Journal of Applied Metaheuristic Computing. 2022. vol. 13. no. 1. pp. 1–22.
32. Jiang Z., Liu H., Fu B., Wu Z. Generalized ambiguity decompositions for classification with applications in active learning and unsupervised ensemble pruning. Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17), 2017. pp. 2073–2079.
33. Dong X., Yu Z., Cao W., Shi Y., Ma Q. A survey on ensemble learning. Frontiers of Computer Science. 2020. vol. 14. no. 2. pp. 241–258.
34. Shahhosseini M., Hu G., Pham H. Optimizing ensemble weights and hyperparameters of machine learning models for regression problems. Machine Learning with Applications. 2022. vol. 7. p. 100251.
35. Fuhg J., Fau A., Nackenhorst U. State-of-the-Art and Comparative Review of Adaptive Sampling Methods for Kriging. Archives of Computational Methods in Engineering. 2021. vol. 28. pp. 2689–2747.
36. Liu H., Ong Y.-S., Cai J. A survey of adaptive sampling for global metamodeling in support of simulation-based complex engineering design. Structural and Multidisciplinary Optimization. 2018. vol. 57. no. 1. pp. 393–416.
37. Margineantu D.D., Dietterich T.G. Pruning adaptive boosting. Proc. of 14th International Conference on Machine Learning. ICML, 1997. pp. 211–218.
38. Hsu K.W. A theoretical analysis of why hybrid ensembles work. Computational Intelligence and Neuroscience. 2017. vol. 2017. p. 1930702.
39. Yao Y., Pirš G., Vehtari A., Gelman A. Bayesian hierarchical stacking: Some models are (somewhere) useful. Bayesian Analysis. 2022. vol. 17. no. 4. pp. 1043–1071.
40. Нужный А.С. Регуляризация Байеса при подборе весовых коэффициентов в ансамблях предикторов. Труды ИСП РАН. Т. 31. № 4. 2019.
41. Demšar J. Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research. 2006. vol. 7. pp. 1–30.

Зеленков Юрий Александрович — д-р техн. наук, профессор, высшая школа бизнеса, НИУ «Высшая школа экономики». Область научных интересов: моделирование социально-экономических систем, машинное обучение, управление знаниями. Число научных публикаций — 80. yzelenkov@hse.ru; улица Шаболовка, 28/11, 119049, Москва, Россия; р.т.: +7(495)771-3232.