Expert Pruning Based on Genetic Algorithm in Regression Problems

S.A. Jafari[1,5], S.Mashohor[2], Abd. R. Ramli[3], and M. Hamiruce Marhaban[4]

[1,2,3] Department of Computer and Communication Systems Engineering, Faculty of Engineering, University Putra Malaysia
[4] Department of Electrical and Electronics Engineering, Faculty of Engineering, University Putra Malaysia
[5] Petroleum Training Center of Mahmoudabad, Mazandaran, Iran
sajkenari@gmail.com,
{syamsiah,arr,hamiruce}@eng.upm.edu.my

Abstract. Committee machines are a set of experts that their outputs are combined to improve the performance of the whole system which tend to grow into unnecessarily large size in most of the time. This can lead to extra memory usage, computational costs, and occasional decreases in effectiveness. Expert pruning is an intermediate technique to search for a good subset of all members before combining them. In this paper we studied an expert pruning method based on genetic algorithm to prune regression members. The proposed algorithm searches to find a best subset of experts by creating a logical weight for each member and chooses which member that the related weight is equal to one. The final weights for selected experts are calculated by genetic algorithm method. The results showed that MSE and R-square for the pruned CM are 0.148 and 0.9032 respectively that are reasonable rather than all experts separately.

Keywords: expert pruning, committee machine, learning algorithms, genetic algorithm.

1 Introduction

The ensembles made by existing methods are sometimes needlessly large. The disadvantages of ensembles are; using the extra memory, the computational overhead, and the occasional decreases in effectiveness. There are also some individuals with low predictive performances that create negative effect on overall performance of an ensemble. Pruning committee members while preserving a high diversity among the remaining individuals is an efficient technique for increasing the predictive performance. In the other word, the most advantage of expert pruning is efficiency and predictive performance. In fact the expert pruning problem is similar as an optimization problem, which the objective is finding the best subset of individuals from the original Committee Machine (CM). As we know, in a set with N member there

J.-S. Pan, S.-M. Chen, N.T. Nguyen (Eds.): ACIIDS 2012, Part III, LNAI 7198, pp. 79–88, 2012.

is $(2^N - 1)$ subset, so in a CM with a moderate size the exhaustive search becomes intractable. In [1], authors proposed a clustering method for ensemble pruning. In their method, experts are divided to some sets according to their outputs similarity and then one single network from each cluster is selected. This method does not guarantee that the selected experts improve the generality prediction of the ensemble. In [2], the authors presented a pruning approach based on a Genetic Algorithm (GA) named GASEN (Genetic Algorithm based Selective Ensemble). At the first step, GASEN trains a number of neural networks and then assigned random or equal weights to those networks. At the second step, the framework employed GA to change the weights so that the optimum weights of the neural networks in constituting an ensemble with lowest error can be tackled. Finally the networks whose weight is bigger than a preset threshold λ could be selected to join the ensemble and the models of the ensemble that did not exhibit the predefined threshold are dropped. After selecting the ensemble members the new weights for all candidates can be obtained by normalizing the oldest weight or reapplying the GA to sub ensemble. In [3], the authors extended the technique of Stacked Regression to prune the members of the ensemble using an equation including both accuracy and diversity. The diversity is based on measuring the positive correlation between the errors of the ensemble members and the accuracy of each individual is calculated relative to the accuracy of the most accurate ensemble member. A drawback of this method is predefinition of a weighting GA parameter to balance accuracy and diversity which has to define by user. More details of GA can be found in [4] and [5].

In this paper we studied an expert pruning or ensemble pruning method that can be used to prune regression committee members. The proposed algorithm aims to search for a best subset of experts by making a logical weight for each expert and finally chooses the best experts which the related weight value is one. The rest of this paper is structured as follows. Section 2 will presents related work on ensemble pruning in regression problems. In Section 3 we will introduce our methodology which is based on GA. Data set preparation, experts design with different training algorithms, expert pruning, combine the obtained subset, results and discussion are presented in Section 4. In Section 5, conclusion will be summarized.

2 Related Works

In this section we introduce some important proposed methods in ensemble pruning that are applied in regression problems. The explanation will explore three type of methods which are; Directed Hill Climbing (DHC), Semi Definite Programming (SDP) and ordered aggregation.

2.1 Directed Hill Climbing Method (DHC)

Hill climbing search greedily moves from current state to the next state, which is in its neighborhood. Let $H = \{h_t; t = 1,2,...,T\}$ be an original ensemble with T members and $S \subseteq T$ is a sub ensemble. At first, the method selects an empty (or full) initial sub

ensemble S and continues with searching in the space of different ensembles by iteratively expanding (or contracting) the set S by a single member h_t. The search is guided by an evaluation measure that is the main component of the hill climbing algorithm which proposed in different methods by some authors. The experimental results obtained by different hill climbing methods report good predictive performance results [6-8]. Figure 1; illustrate the search space for an ensemble with four members. One of the important parameters in designing DHC method is the direction of search that is in two types which are; forward selection and backward elimination.

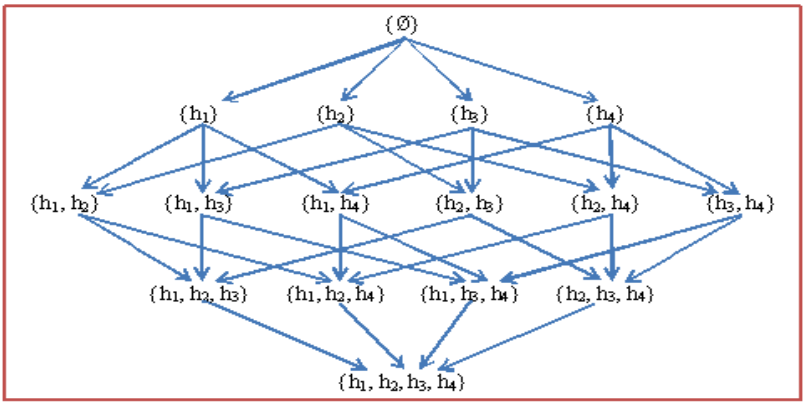


Fig. 1. An example of forward search in DHC

In forward selection, firstly, the sub ensemble S is initialized to the empty set and then the algorithm continues by iteratively adding the individual $h_t \in H \setminus S$ to set S so that optimizes the predefined evaluation function. In backward elimination, the sub ensemble S is equal with *H* at the start and then the algorithm continues by iteratively remove the individual $h_t \in S$ to optimize the evaluation function [6]. The second significant parameter in designing DHC method is the type of evaluation measure that are based on performance and diversity. In performance basis, the objective function is defined based on increasing the performance of the produced ensemble which is created by adding or removing a model to or from the current ensemble [7, 9]. In [10, 11], the authors have used diversity as an evaluation measure. Finding a suitable calculating method for diversity during the search of sub ensemble plays very significant role and needs much more attention. The third significant parameter is the evaluation of data sets. The evaluation function scores the candidate sub ensemble according to its diversity or accuracy. These procedures need a set of data for performance which will be called the as pruning set but it is clear that training set or separate validation can also be used as pruning set. In [12], the authors used a k-fold cross validation such that the remaining fold of the training set are used to create an ensemble. The same fold is used as the pruning set for models and sub ensembles of the ensemble. Finally, the evaluations are averaged across all folds. Because the evaluation of models is based on unseen data that were not used for their training, therefore this method

is less prone to over fitting. The last important parameter is amount of pruning that is the size of the final sub ensemble which can be determine in two ways. One way is using the fixed number or fixed percentage for sub ensemble size which can be defined by user before starting the test. Another way is dynamic population size which is based on predictive performance of the sub ensemble with different size. In this approach, the performance of the whole sub ensemble scores during the search from initial population to final in forward or backward selection will be analyzed. Finally sub ensemble with best performance will be selected as final ensemble which successfully improves the efficiency.

2.2 Semi Definite Programming (SDP)

In [13], the authors proposed a pruning method based on Semi Definite Programming (SDP) for classification tasks, and in [14], the author used this method for regression problem. They formulated an ensemble pruning problem as a quadratic programming to obtain sub ensemble classifiers with optimal accuracy and diversity trade off. At first, they defined an error matrix P, which records the misclassification of each classifier on the training set as follows:

$$P_{ij} = \begin{cases} 0, & \text{if } jth \text{ classifier on data point } i \text{ is correct} \\ 1, & \text{otherwise} \end{cases} \tag{1}$$

Let $G = P^t P$, which the diagonal arrays G_{ii} are the whole errors made by classifier i and the term G_{ij} is the whole errors made by classifier i and j. The goal is to find a sub matrix of G, such that the sums of the elements in row or column are minimum. After normalization, the matrix $\tilde{G}$ will be:

$$\tilde{G} = \begin{cases} G_{ii} / N & \text{if } i = j \\ \frac{1}{2}\left(\frac{G_{ij}}{G_{ii}} + \frac{G_{ij}}{G_{jj}} \right) & \text{if } i \neq j \end{cases} \tag{2}$$

Where N is number of training data and whole array of matrix in $\tilde{G}$ are belong to (0, 1). The diagonal array (G_{ii}) are the error rate of classifier i and the off-diagonal array (G_{ij}) are the overlap of errors between classifier i and j. Note that G_{ij} / G_{ii} is the conditional probability that classifier j misclassifies a point, given that classifier i does. Taking the average of G_{ij} / G_{ii} and G_{ij} / G_{jj} as the off-diagonal elements of $\tilde{G}$ makes the matrix symmetric. Naturally, sum of diagonal array in matrix $\tilde{G}$ measures the overall strength and sum of off-diagonal array measures the diversity of the ensemble. Then a combination of diagonal and off-diagonal array in $\tilde{G}$ should be a good approximation of the ensemble error and the ensemble is good if the whole array is small values. The equation is a quadratic integer programming formula which can be

used for the subset selection problem with a fixed-size subset of classifiers (k), and the objective function is the sum of the corresponding elements in the $\tilde{G}$ matrix that should be minimized.

$$\begin{aligned} &\min \ w^t \tilde{G} w \\ &\text{s.t.} \begin{cases} \sum_i w_i = k \\ w_i \in \{0,1\} \end{cases} \end{aligned} \tag{3}$$

The weight w_i represents whether *i-th* classifier is included in the sub ensemble (when $w_i = 1$) or no ($w_i = 0$). If $w_i = 1$, it means that the corresponding diagonal and off-diagonal array will be counted in the objective function and vice versa. The equation is a standard 0-1 optimization problem, which is NP-hard in general and therefore they formulate it as a "max cut problem" with size k. In graph theory the max cut problem is a method that partitions the vertices of an edge weighted graph into two sets with same size k, so that the total weight of edges crossing the partition is maximized.

2.3 Ordered Aggregation

In practice, committee members are trained sequentially and then collected as they are created from the different methods. In this method of creating an ensemble, by increasing the ensemble member, the individual error shows a monotonic decreasing. Ordered aggregation is a method in ensemble pruning which reorders the members of an original ensemble and then selects a sub ensemble. In the past two decades a number of researchers have sought to determine sub ensemble by ordered aggregation methods [7, 11, 14]. In [15], authors used ordered aggregation technique to prune a regression bagging ensemble. From the initial pool of M predictors generated by bagging, ordered aggregation builds a sequence of nested ensembles, in which the sub ensemble of size u contains the sub ensemble of size $(u - 1)$. Such as forward selection in hill climbing method, the algorithm starts with an empty set of experts and grows by adding a new expert in each iteration such that the new set reduces the training error of the extended sub ensemble. In this method, the algorithm creates a variance-covariance matrix C that has to be minimized. For example in iteration u, the new *k-th* expert is selected to add to the sub ensemble. The matrix $C(u)$ will be obtained by adding all covariance between last $(u-1)$ experts and the new expert k with the variance of this expert. In other words, the matrix $C(u)$ is created by the matrix $C(u-1)$ and all variance-covariance between new expert and the oldest experts. The objective function is the sum all array in matrix $C(u)$ which has to be less than the sum of array in matrix $C(u-1)$. In this situation, the expert k is suitable to be added to the set of $(u-1)$ experts to create new sub ensemble. This algorithm repeats until the whole of experts are tested. In [16], the authors proposed an AdaBoost algorithm using reweighting to compute the weighted training error and ordering them with considering the aggregation of the members generated by the Bagging method. Their

objective was modifying the aggregation order in a bagging ensemble. Finally they used the AdaBoost weighting method to compute the training error and the expert with the lowest weighted error is selected from a pool of experts generated by bagging. This process can be applied to any other parallel ensemble building methods.

3 Methodology

In this paper we proposed a pruning method based on GA that has been used for regression problems. The objective in any regression problem is to learn a predictor (expert) of the dependent variable $y \in R$ (in one dimensional output) as a function of the independent variables $X = (x_1, x_2, ..., x_n) \in R^n$ (attributes) using the training data $D = \{(x_i, y_i), i = 1,2,...,n\}$ that is drawn from a probability distribution $P(D)$. Assume $\hat{f}_i(x)$ is the prediction given by the ith expert on sample data *D*. The final predicted output of the CM with N members is a combination of the individual output with the weights w_i that are showed as below:

$$F_{CM}(x) = \sum_{i=1}^{N} w_i \hat{f}_i(x); \qquad 0 \le w_i \le 1 \quad \& \quad \sum_{i=1}^{N} w_i = 1 \tag{4}$$

The error of the CM is

$$E = \int (F(x) - f(x))^2 P(x) dx \tag{5}$$

Where $f(x)$ is target function to approximate and $P(x)$ is the probability density distribution in attribute space. To prune the regression CM we try to select a sub-CM with M members that minimize the error function which has been introduced in Equation (5). For selection of the sub-CM we use mean square error based on the training error and expect this estimation to be similar to calculation that is done over the true distribution of the data. Then we assume that minimizing the error given by training data leads to the minimization of the generalization error. Actually in real regression problem, the training error minimization usually leads to over fitting. Indeed, the experiments carried out show that the size of the sub-CM based on training error tend to be smaller than the optimal sub-CMs based on test data [15]. With assumption of a subset of original CM with lower generalization error, the process of finding this subset is complex and needs generating the whole $(2^N - 1)$ non empty sub-CM. In literature, finding an optimal subset of members that minimizes the error estimated on data set is defined as the NP-hard problem and is not generally feasible in practice. Our proposed method is a practical approach to expert pruning based on GA to find out whose experts that should be excluded from the CM. The main idea behind this proposed method is heuristics, i.e. assuming each expert can be assigned a weight that would characterize the fitness function, and then the experts whose weight is equal to one could be selected to join the sub-CM. Suppose that the weight of the *i-th* member of committee is w_i, which satisfies in below equation where *K* is the cardinality of sub-CM which is user defined.

$$w_i \in \{0,1\}, \qquad \sum_{i=1}^{N} w_i = K \tag{6}$$

Then we get a weight vector $W = (w_1, w_2, ..., w_N)$ that w_i is equal to zero or one for each $i = 1, 2, ..., N$. This weight vector should minimize the generalization error of the CM which is Mean Square Error (MSE) as a fitness function in GA. If $w_i = 0$, that means *i-th* expert is not belong to our final subset and if $w_i = 1$, which means the *i-th* expert is belong to our final pruned CM. Then after minimizing the objective function we have K weight, which are equal to one and $(N - K)$ weight, which are equal to zero. Each weight that is equal to one means the related expert is belong to our final pruned CM. Then we have K members for final sub-CM and they can be combined with different method and get a new weight for them that satisfies the Equation (4).

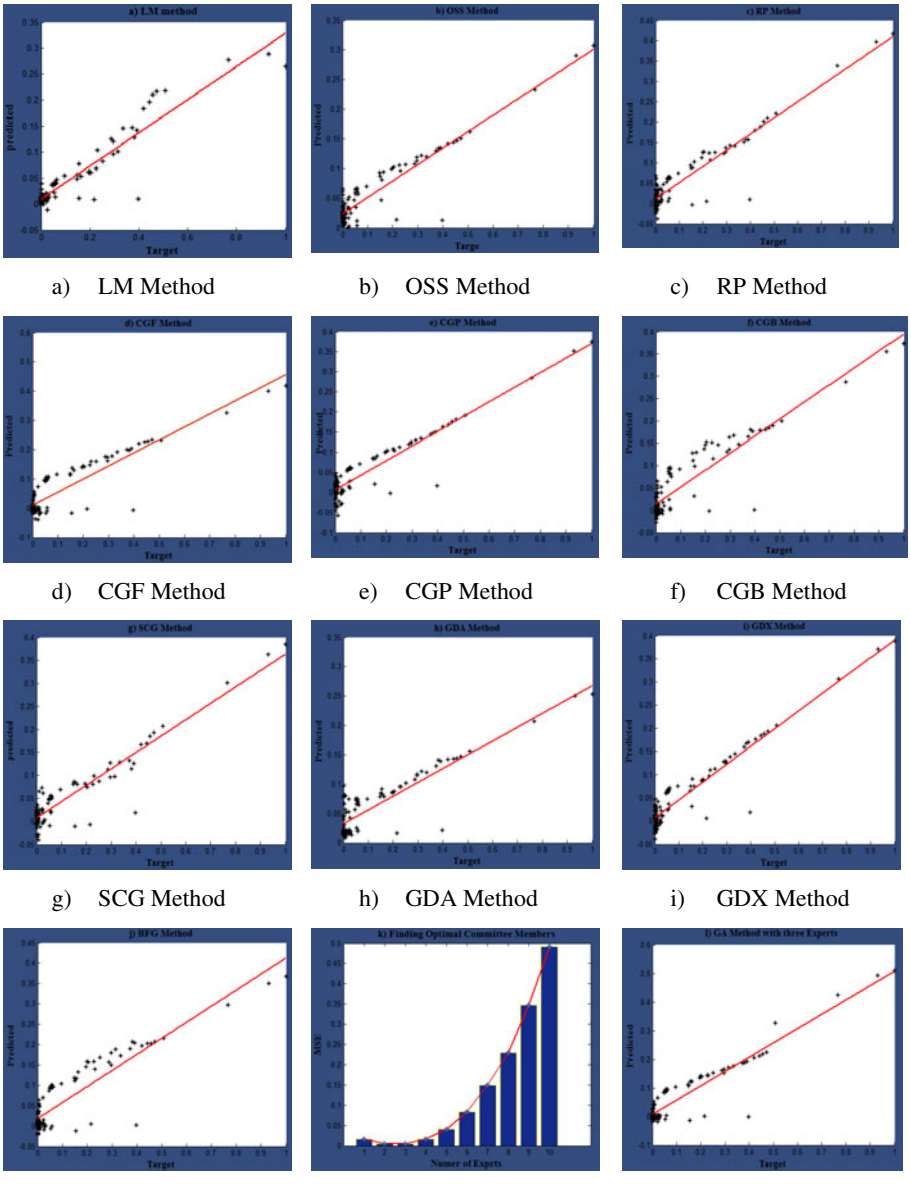
4 Simulation Results and Discussion

The simulation is performed under Matlab on a personal computer with Intel core 2 duo 2.7 GHZ processor and 4 GB of memory. We carried out experiments over a set of oil data from National Iranian Oil Company (NIOC) that were well log data and core permeability data. Sonic transit time (DT), Neutron log (NPHI), Density log (RHOB), Gamma Ray (GA), and True Formation Resistivity (Rt) are the input data and rock permeability (K) is a target data. Ten neural networks with different training algorithms have been used as committee members that were: Variable learning rate back propagation (GDA, GDX), Resilient back propagation (RP), Conjugate gradient (CGF, CGP, CGB, SCG), Quasi-Newton (BFG, OSS), Levenberg-Marquardt (LM). Figure 2 (a-j) illustrated the cross plot between target and predicted value by different training algorithms for testing data. After that the proposed method for expert pruning are implemented on the initial CM. The performance of the GA method on all individuals showed that three experts can decrease the error compared with other subset of the initial CM. Figure 2(k) illustrates the MSE versus the number of experts. Therefore by applying our proposed method, experts with no. 3, 4 and 10 are selected as the best subset. On average, it takes about 30 minutes in each iteration. After finding the optimal subset, the final weights for these three experts are calculated by GA method which are 0.25, 0.6 and 0.15 respectively. Table 1, illustrates MSE and correlation coefficient for all ten experts and the final weight for three selected experts based on GA method for making the final sub CM.

Table 1. MSE and R^2 for ten experts and the final GA calculated weight for selected experts

Method no.	(1) LM	(2) OSS	(3) RP	(4) CGF	(5) CGP	(6) CGB	(7) SCG	(8) GDA	(9) GDX	(10) BFG
MSE	0.0023	0.024	0.018	0.016	0.02	0.019	0.021	0.026	0.019	0.018
R^2	0.887	0.868	0.882	0.82	0.867	0.82	0.876	0.779	0.897	0.804
Weight	---	---	0.25	0.6	---	---	---	---	---	0.15

According to the results, the pruned CM has produced the minimum error and reasonable correlation coefficient rather than all experts that are 0.148 for MSE and 0.9032 for R-square. Figure (2-l) shows the scatter plot of target and predicted final output with GA method as a combination method for sub CM.



a) LM Method b) OSS Method c) RP Method

d) CGF Method e) CGP Method f) CGB Method

g) SCG Method h) GDA Method i) GDX Method

j) BFG Method k) Different subset l) Output of sub CM

Fig. 2. The performance of different sub CM (k); The crossplot between target value and predicted value for ten learning algorithms (a)-(j) and for combined sub CM with GA (l)

5 Conclusion

In this paper we presented an expert pruning method based on GA. Ten neural networks with different training algorithms are created as experts for initial CM. After that the proposed pruning method is applied to find the optimal sub CM. As mentioned in section 2, the size of the final subset can be defined by user before starting the test. In this paper we tried to find optimal subset size by adding one by one member to initial empty subset. In this approach, the subset with best performance is selected as final sub CM which successfully improved the MSE as evaluation measure. As illustrated in figure 2(k) the subset with three experts is the optimal subset for our investigation that are RP, CGF and BFG learning algorithms respectively. After finding the optimal subset, the final weights for these three experts are calculated by GA method which are 0.25, 0.6 and 0.15 respectively (refer table 1). Mean square error and correlation coefficient obtained by applying GA in final sub CM are 0.148 and 0.9032 respectively. Therefore in comparing with MSE and R-square obtained for all members, these sub CM results appears more reasonable. Figure (2-l) shows the scatter plot of measured and predicted final output with GA method as a combination method for sub CM. The advantage of the proposed method related to DHC, SDP and OA that mentioned in section 2, is the similarity computation in our method that is very significant for optimal subset selection.

Acknowledgements. The authors wish to express their appreciation to the National Iranian Oil Company (NIOC) for sponsorship, data preparation and their collaboration. Also, we would like to thank engineer Mr. .Hamid Mosalmannejad from Iranian Offshore Oil Company (IOOC) for his invaluable experiences shared during this research.

References

1. Bakker, B., Heskes, T.: Clustering ensembles of neural network models. Neural Netw. 16(2), 261–269 (2003)
2. Zhou, Z.-H., Wu, J., Tang, W.: Ensembling neural networks: Many could be better than all. Artificial Intelligence 137(1-2), 239–263 (2002)
3. Rooney, N., Patterson, D., Nugent, C.: Reduced Ensemble Size Stacking. In: 6th IEEE International Conference on Tools with Artificial Intelligence, pp. 266–271 (2004)
4. Patel, R., Shrawankar, U.N., Raghuwanshi, M.M.: Genetic Algorithm with Histogram Construction Technique. In: Second International Conference on Emerging Trends in Engineering & Technology, pp. 615–618 (2009)
5. Huang, H.-C., Chen, Y.-H.: Genetic fingerprinting for copyright protection of multicast media. Soft Computing 13(4), 383–391 (2008)
6. Banfield, R.E., et al.: Ensemble diversity measures and their application to thinning. Information Fusion 6(1), 49–62 (2005)
7. Caruana, R., et al.: Ensemble selection from libraries of models. In: Proceedings of the Twenty-First International Conference on Machine Learning (2004)
8. Partalas, I., Tsoumakas, G., Vlahavas, I.: An ensemble uncertainty aware measure for directed hill climbing ensemble pruning. Machine Learning 81(3), 257–282 (2010)

9. Fan, W., et al.: Pruning and dynamic scheduling of cost-sensitive ensembles. In: Eighteenth National Conference on Artificial Intelligence, pp. 146–151 (2002)
10. Brown, G., Wyatt, J.L., Peter: Managing Diversity in Regression Ensembles. Machine Learning Research 6, 1621–1650 (2005)
11. Martinez-Munoz, G., Suarez, A.: Pruning in ordered bagging ensembles. In: Proceedings of the 23rd International Conference on Machine Learning, pp. 609–616 (2006)
12. Caruana, R., Munson, A., Niculescu-Mizil, A.: Getting the most out of ensemble selection. In: International Conference on Data Mining, pp. 828–833 (2006)
13. Zhang, Y., Burer, S., Street, W.N.: Ensemble Pruning Via Semi-definite Programming. Jornal of Machin. Learnning Research 7, 1315–1338 (2006)
14. Martinez-Muoz, G., Hernandez-Lobato, D., Suarez, A.: An Analysis of Ensemble Pruning Techniques Based on Ordered Aggregation. IEEE Transactions on Pattern Analysis and Machine Intelligence 31(2), 245–259 (2009)
15. Hernandez-Lobato, D., Martinez-Munoz, G., Suarez, A.: Pruning in Ordered Regression Bagging Ensembles. In: Proceedings of the IEEE World Congress on Computational Intelligence, pp. 1266–1273 (2006b)
16. Martínez-Muñoz, G., Suárez, A.: Using boosting to prune bagging ensembles. Pattern Recognition Letters 28(1), 156–165 (2007)