An Ensemble Pruning Primer

Grigorios Tsoumakas, Ioannis Partalas, and Ioannis Vlahavas

Abstract Ensemble pruning deals with the reduction of an ensemble of predictive models in order to improve its efficiency and predictive performance. The last 12 years a large number of ensemble pruning methods have been proposed. This work proposes a taxonomy for their organization and reviews important representative methods of each category. It abstracts their key components and discusses their main advantages and disadvantages. We hope that this work will serve as a good starting point and reference for researchers working on the development of new ensemble pruning methods.

1 Introduction

Ensemble methods [5, 13] has been a very popular research topic during the last decade. It has attracted the interest of scientists from several fields including Statistics, Machine Learning, Pattern Recognition and Knowledge Discovery in Databases. The success of ensemble methods arises largely from the fact that they offer an appealing solution to several interesting learning problems of the past and the present, such as improving predictive performance, learning from multiple physically distributed data sources, scaling inductive algorithms to large databases and learning from concept-drifting data streams.

Typically, ensemble methods comprise two phases: the *production* of multiple predictive models and their *combination*. Recent work [2, 4, 7, 9, 15, 18, 16, 29, 21, 22, 35] considers an additional intermediate phase that deals with the reduction of the ensemble size prior to combination. This phase is com-

Grigorios Tsoumakas · Ioannis Partalas · Ioannis Vlahavas
Department of Informatics, Aristotle University of Thessaloniki,
54124 Thessaloniki, Greece
e-mail: {greg,partalas,vlahavas}@csd.auth.gr

monly called *ensemble pruning*, while other names include *selective ensemble*, *ensemble thinning* and *ensemble selection*.

Ensemble pruning is important for two reasons: *efficiency* and *predictive performance*. Having a very large number of models in an ensemble adds a lot of computational overhead. For example, decision tree models may have large memory requirements [16] and lazy learning methods have a considerable computational cost during execution. The minimization of run-time overhead is crucial in certain applications, such as stream mining. In addition, when models are distributed over a network, the reduction of models leads to the reduction of communication costs.

Equally important is the second reason, predictive performance. An ensemble may consist of both high and low predictive performance models. The latter may negatively affect the overall performance of the ensemble. In addition, an ensemble may contain many models that are very similar to each other. This reduces its diversity and capability for error correction. Pruning low performance models while maintaining a high diversity among the remaining members of the ensemble is typically considered a proper recipe for an effective ensemble.

Note that ensemble pruning is different from *ensemble weighting* [34], where the decisions of all models in the ensemble are considered, but with a different weight. Ensemble weighting is concerned solely with increasing the predictive performance, as it needs to maintain all models of the ensemble.

One of the first ensemble pruning approaches is discussed in [24]. The 12 years that followed have witnessed the development of several diverse methods for ensemble pruning. This work proposes a taxonomy for their organization and reviews important representative methods of each category. It steers clear of a mere enumeration of particular approaches in the related literature and instead attempts to abstract their key components, discuss their main advantages and disadvantages and analyze their complexity whenever possible. We hope that this work will serve as a good starting point and reference for researchers working on the development of new ensemble pruning methods.

The remainder of this chapter is structured as follows. Section 2 contains background material on ensemble production and combination. Section 3 presents the proposed taxonomy, introduces notation and discusses issues that are common for all methods. Sections 4 to 7 review important representative methods of each category in the taxonomy. Finally, the conclusions of this work are presented in Section 8.

2 Background

This section provides background material on ensemble methods. More specifically, information about the different ways of producing models are presented as well as different methods for combining the decisions of the models.

2.1 Producing the Models

An ensemble can be composed of either *homogeneous* or *heterogeneous models.* Homogeneous models derive from different executions of the same learning algorithm. Such models can be produced by using different values for the parameters of the learning algorithm, injecting randomness into the learning algorithm or through the manipulation of the training instances, the input attributes and the model outputs [6]. Popular methods for producing homogeneous models are *bagging* [3] and *boosting* [25].

Heterogeneous models derive from running different learning algorithms on the same data set. Such models have different views about the data, as they make different assumptions about it. For example, a neural network is robust to noise in contrast with a k-nearest neighbor classifier.

2.2 Combining the Models

Common methods for combining an ensemble of predictive models include *voting*, *stacked generalization* and *mixture of experts.*

In voting, each model outputs a class value (or ranking, or probability distribution) and the class with the most votes is the one proposed by the ensemble. When the class with the maximum number of votes is the winner, the rule is called *plurality voting* and when the class with more than half of the votes is the winner, the rule is called *majority voting.* A variant of voting is weighted voting where the models are not treated equally as each of them is associated with a coefficient (weight), usually proportional to its classification accuracy.

Let x be an instance and m_i, $i = 1..k$ a set of models that output a probability distribution $m_i(x, c_j)$ for each class c_j, $j = 1..n$. The output of the (weighted) voting method $y(x)$ for instance x is given by the following mathematical expression:

$$y(x) = \arg\max_{c_j} \sum_{i=1}^{k} w_i m_i(x, c_j),$$

where w_i is the weight of model i. In the simple case of voting (unweighted), the weights are all equal to one, that is, $w_i = 1, i = 1..k$.

Stacked generalization [32], also known as *stacking* is a method that combines models by learning a meta-level (or level-1) model that predicts the correct class based on the decisions of the base level (or level-0) models. This model is induced on a set of meta-level training data that are typically produced by applying a procedure similar to k-fold cross validation on the training data. The outputs of the base-learners for each instance along with the true class of that instance form a meta-instance. A meta-classifier is then trained on the meta-instances. When a new instance appears for classification, the output of the all base-learners is first calculated and then propagated to the meta-classifier, which outputs the final result.

The mixture of experts architecture [12] is similar to the weighted voting method except that the weights are not constant over the input space. Instead there is a gating network which takes as input an instance and outputs the weights that will be used in the weighted voting method for that specific instance. Each expert makes a decision and the output is averaged as in the method of voting.

3 A Taxonomy of Ensemble Pruning Methods

We propose the organization of the various ensemble pruning methods into the following categories:

- *Ranking* based. Methods of this category are conceptually the simplest. They order the models of the ensemble once according to an evaluation function and select models in this *fixed* order.
- *Clustering* based. Methods of this category comprise two stages. Initially, they employ a clustering algorithm in order to discover groups of models that make similar predictions. Subsequently, each cluster is separately pruned in order to increase the overall diversity of the ensemble.
- *Optimization* based. Ensemble pruning can be posed as an optimization problem as follows: find the subset of the original ensemble that optimizes a measure indicative of its generalization performance (e.g. accuracy on a separate validation set). Exhaustive search of the space of ensemble subsets is infeasible for a moderate ensemble size.
- *Other.* This category includes methods that don't fall into one of the previous categories.

Before proceeding to the description of the main characteristics of each category, some common notation is introduced. The original ensemble is denoted as $H = \{h_t, t = 1, 2, \ldots, T\}$. All methods employ a function that evaluates the suitability of single models, model pairs or ensembles of more than two models for inclusion in the final ensemble. Evaluation is typically based on the predictions of the models on a set of data, which will be called the *pruning set*. The role of the pruning set can be performed by the training set, a separate validation set, or even a set of - naturally existing or artificially produced - instances with unknown value for the target variable. The pruning set will be denoted as $D = \{(\boldsymbol{x_i}, y_i), i = 1, 2, \ldots, N\}$, where $\boldsymbol{x_i}$ is a vector with feature values and y_i is the value of the target variable, which may be unknown.

4 Ranking-based Methods

The main point of differentiation among the methods of this category is the evaluation measure used for model ranking. Using the predictive performance of individual models is too simplistic and does not achieve satisfying results [24, 33]. Information-theoretic measures were also used in [33] for the evaluation of Bayesian models, with equally disappointing results.

Kappa pruning [16] employs a diversity measure for evaluation. It ranks all *pairs* of classifiers in H based on the κ statistic of agreement calculated on the training set. Its time complexity is $O(T^2N)$. Kappa pruning could be generalized by accepting a parameter to specify any pairwise diversity measure for either classification or regression models, in place of the κ statistic. However, it would still beg for one fundamental theoretical question: Do two diverse pairs of models, lead to one diverse ensemble of four models? The intuitive answer is no. In fact, kappa pruning has been shown to be non-competitive for pruning classifier ensembles produced via bagging [17].

An efficient and effective ranking-based pruning method for ensembles of classifiers is orientation ordering [19]. A key concept in orientation ordering is the *signature vector* of a classifier h_t; an N-dimensional vector with elements taking the value +1 if $h_t(x_i) = y_i$ and -1 if $h_t(x_i) \neq y_i$. The average signature vector of all classifiers in an ensemble is called the *ensemble signature vector*. It is indicative of the ability of the ensemble to correctly classify each example in the pruning set (the training set in this method) using majority voting for classifier combination. The *reference vector* is a vector perpendicular to the ensemble signature vector that corresponds to the projection of the first quadrant diagonal onto the hyper-plane defined by the ensemble signature vector.

Orientation ordering ranks the classifiers by increasing value of the angle between their signature vector and the reference vector. Essentially this ordering gives preference to models that correctly classify those examples that are incorrectly classified by the full ensemble. Orientation ordering is among the fastest methods for ensemble pruning, with time complexity of $O(TN)$. In addition, its predictive performance is not significantly worse than state-of-the-art methods for pruning classifier ensembles produced via bagging [17].

Another interesting issue in ranking-based ensemble pruning methods concerns the choice of the final number of models from the obtained ranking. One approach is to use a fixed user-specified amount or percentage of models. In kappa pruning for example, classifier pairs are selected in ascending order of agreement until a specified number of models has been reached. If the goal of pruning is to improve efficiency, then this approach can be used in order to obtain the desired amount of models, which may be dictated by constraints (memory and speed) in the application environment.

A second approach is to dynamically select the size based on the evaluation measure or the predictive performance of ensembles of different size. In orientation ordering for example, only the classifiers whose angle is less than $\pi/2$ are included in the final ensemble, while in [33], the models whose evaluation measure is lower than the average of all models are pruned. This approach is more preferable when the goal of pruning is to improve predictive performance, as it is more flexible and can sacrifice efficiency for effectiveness.

5 Clustering-based Methods

A first issue for the methods of this category is the choice of clustering algorithm. Past approaches have used hierarchical agglomerative clustering [9], k means [8, 15] and deterministic annealing [1].

Clustering algorithms are based on the notion of distance. Therefore, a second issue for clustering based methods is the choice of an appropriate distance measure. The probability that classifiers don't make coincident errors in a separate validation set was used as a distance measure in [9]. This measure is actually equal to one minus the *double fault* diversity measure [14]. The Euclidean distance in the training set is used in [8, 15]. Actually, any distance measure suitable for nominal (classifiers) or numeric (regressors) output could be used. Note that there is no need for a labeled pruning set in this case [1]. Artificially generated data could be used instead.

Another important issue concerns the process of pruning each cluster. An elegant approach was used in [1], where a new model is trained for each cluster, using the cluster centroids as values of the target variable. Another interesting approach is to select from each cluster the single classifier that is most distant to the rest of the clusters [9]. The approach followed in [15] was to iteratively remove models from the least to the most accurate, until the accuracy of the entire ensemble starts to decrease. This, however, does not guarantee the selection of a single model from each cluster. The most accurate model of each cluster was selected in [8].

A final issue worth mentioning is the choice of the number of clusters. This could be determined based on the performance of the method on a validation set [8]. In [15], the number of clusters was gradually increased until the disagreement between the cluster centroids started to deteriorate.

6 Optimization-based Methods

In the following subsections we look into ensemble pruning methods that are based on three different optimization approaches: *genetic algorithms*, *semi-definite programming* and *hill climbing*. The last approach is examined at a greater level of detail, as a large number of this kind of ensemble pruning methods have been recently proposed.

6.1 Genetic Algorithms

The Gasen-b method [36] performs stochastic search in the space of model subsets using a genetic algorithm. The ensemble is represented as a bit string, using one bit for each model. Models are included or excluded from the ensemble depending on the value of the corresponding bit. Gasen-b performs standard genetic operations such as mutations and crossovers and uses default values for the parameters of the genetic algorithm. The fitness function for an individual $S \subseteq H$ is the accuracy of S on a separate validation set using voting for model combination.

6.2 Semi-Definite Programming

Zhang et al. [35] formulate the ensemble pruning problem as a mathematical problem and apply semi-definite programming (SDP) techniques. In specific, the authors initially formulated the ensemble pruning problem as a quadratic integer programming problem that looks for a fixed-size subset of k classifiers with minimum misclassification and maximum divergence.

They subsequently found that this quadratic integer programming problem is similar to the *max cut with size k* problem, which can be approximately solved using an algorithm based on SDP. Their algorithm requires the number of classifiers to retain as a parameter and runs in polynomial time.

6.3 Hill Climbing

Hill climbing search greedily selects the next state to visit from the neighborhood of the current state. States, in our case, are the different subsets of models and the neighborhood of a subset $S \subseteq H$ consists of those subsets that can be constructed by adding or removing one model from S. We focus on the directed version of hill-climbing that traverses the search space from one end (empty set) to the other (complete ensemble). An example of the search space for an ensemble of four models is presented in Figure 1.

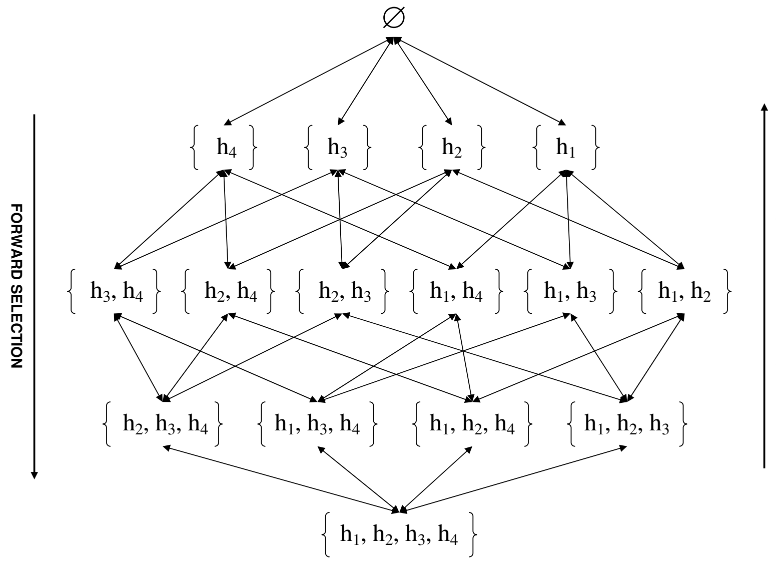

Fig. 1 An example of the search space of hill climbing ensemble pruning methods for an ensemble of 4 models.

Depending on the direction of search, we have forward selection [16, 7, 18, 4, 33] and backward elimination [2, 22, 33] methods. In both cases, the traversal requires the evaluation of $\frac{T(T+1)}{2}$ subsets, leading to a time complexity of $O(T^2 g(T,N))$. The term $g(T,N)$ concerns the complexity of the evaluation process, which is linear with respect to N and ranges from constant to quadratic with respect to T, as we shall see in the rest of this section.

Similarly to ranking-based methods, the main component that differentiates hill climbing ensemble pruning methods is the evaluation measure. Evaluation measures can be grouped into two major categories: *performance* based and *diversity* based.

The goal of performance based measures is to find the model h_t that maximizes the performance of the ensemble produced by adding (removing) h_t to (from) the current ensemble. Their calculation depends on the method used for ensemble combination, which usually is voting. Accuracy was used as an evaluation measure in [16, 7, 33], while [4] experimented with several metrics, including accuracy, root-mean-squared-error, mean cross-entropy, lift, precision/recall break-even point, precision/recall F-score, average precision and ROC area. Another measure is *benefit* which is based on a cost model and has been used in [7].

The calculation of performance-based metrics requires the decision of the current ensemble S on all examples of the pruning set. Therefore, the complexity of these measures is $O(|S|N)$. However, this complexity can be optimized to $O(N)$ if the predictions of the current ensemble are updated incrementally each time a model is added to/removed from it.

It is generally accepted that an ensemble should contain diverse models in order to achieve high predictive performance. However, there is no clear definition of diversity, neither a single measure to calculate it. In their interesting study, Kuncheva and Whitaker [14], could not reach into a solid conclusion on how to utilize diversity for the production of effective classifier ensembles. In a more recent theoretical and experimental study on diversity measures [27], the authors reached to the conclusion that diversity cannot be explicitly used for guiding the process of hill climbing methods. Yet, certain approaches have reported promising results [18, 2, 22].

One issue worth mentioning here is how to calculate the diversity during the search in the space of ensemble subsets. For simplicity we consider the case of forward selection only. Let S be the current ensemble and $h_t \in H \setminus S$ a candidate classifier to add to the ensemble.

One could compare the diversities of ensembles $S' = S \cup h_t$ for all candidate $h_t \in H \setminus S$ and select the one with the highest diversity. Any pairwise and non-pairwise diversity measure can be used for this purpose. The time complexity of most non-pairwise diversity measures is $O(|S'|N)$, while that of pairwise diversity measures is $O(|S'|^2N)$. However, a straightforward optimization can be performed in the case of pairwise diversity measures. Instead of calculating the sum of the pairwise diversity for every pair of classifiers in each candidate ensemble S', one can simply calculate the sum of the pairwise diversities only for the pairs that include the candidate classifier h_t. The sum of the rest of the pairs is equal for all candidate ensembles. The same optimization can be achieved in backward elimination too. This reduces their time complexity to $O(|S|N)$.

Several methods [18, 2, 27, 22] use a different approach to calculate diversity during the search. They use pairwise measures to compare the candidate classifier h_t with the current ensemble S, which is viewed as a single classifier that combines the decisions of its members with voting. This way they calculate the diversity between the current ensemble as a whole and the candidate classifier. Such an approach has time complexity $O(|S|N)$, which can be optimized to $O(N)$ if the predictions of the current ensemble are updated incrementally each time a model is added to/removed from it.

In the past, the widely known diversity measures *disagreement*, *double fault*, *Kohavi-Wolpert variance*, *inter-rater agreement*, *generalized diversity* and *difficulty* were used for hill climbing ensemble pruning in [27]. Four diversity measures designed specifically for hill climbing ensemble pruning are introduced in [18, 2, 22]. We next present these measures using a common notation.

We can distinguish four events concerning the decision of the current ensemble and the candidate classifier:

$$\begin{aligned} e_{tf}(\boldsymbol{x_i}) &: y = h_t(\boldsymbol{x_i}) \land y \neq S(\boldsymbol{x_i}) \\ e_{ft}(\boldsymbol{x_i}) &: y \neq h_t(\boldsymbol{x_i}) \land y = S(\boldsymbol{x_i}) \\ e_{tt}(\boldsymbol{x_i}) &: y = h_t(\boldsymbol{x_i}) \land y = S(\boldsymbol{x_i}) \\ e_{ff}(\boldsymbol{x_i}) &: y \neq h_t(\boldsymbol{x_i}) \land y \neq S(\boldsymbol{x_i}) \end{aligned}$$

The *complementariness* [18] of a model h_k with respect to an ensemble S and a pruning set D is calculated as follows:

$$COM_D(h_k, S) = \sum_{i=1}^{N} I(e_{tf}(\boldsymbol{x_i})),$$

where $I(true) = 1$, $I(false) = 0$ and $S(x_i)$ is the classification of instance x_i from the ensemble S. This classification is derived from the application of an ensemble combination method to S, which usually is voting. The complementariness of a model with respect to an ensemble is actually the number of examples of D that are classified correctly by the model and incorrectly by the ensemble. A selection algorithm that uses the above measure, tries to add (remove) at each step the model that helps the ensemble classify correctly the examples it gets wrong.

The *concurrency* [2] of a model h_k with respect to an ensemble S and a pruning set D is calculated as follows:

$$CON_D(h_k, S) = \sum_{i=1}^{N} \Big(2 * I(e_{tf}(\boldsymbol{x_i})) + I(e_{tt}(\boldsymbol{x_i})) - 2 * I(e_{ff}(\boldsymbol{x_i}))\Big)$$

This measure is very similar to complementariness with the difference that it takes into account two extra cases.

The *focused ensemble selection* method [22] proposes a measure that uses all the events and also takes into account the strength of the current ensemble's decision:

$$\begin{aligned} FES(h_k, S) = \sum_{i=1}^{N} \Big(& NT_i * I(e_{tf}(\boldsymbol{x_i})) - NF_i * I(e_{ft}(\boldsymbol{x_i})) + \\ & + NF_i * I(e_{tt}(\boldsymbol{x_i})) - NT_i * I(e_{ff}(\boldsymbol{x_i}))\Big), \end{aligned}$$

where NT_i denotes the proportion of models in the current ensemble S that classify example $(\boldsymbol{x_i}, y_i)$ correctly, and $NF_i = 1 - NT_i$ denotes the number of models in S that classify it incorrectly.

The *margin distance minimization* method [18] is based on the same concepts as the orientation ordering ranking-based method (see Section 4). It searches for the ensemble S with the minimum distance between its signature vector c_S and a predefined vector o placed in the first quadrant of the N-dimensional hyperplane. Vector o corresponds to an ideal vector that correctly classifies all examples.

The method is based on a measure called *margin*. The margin, $MAR_D(h_k, S)$, of classifier h_k with respect to an ensemble S and a pruning set D is calculated as follows:

$$MAR_D(h_k, S) = d\left(o, \frac{1}{|S|+1}\left(c_{S\cup\{h_k\}}\right)\right)$$

where d is the Euclidean distance.

7 Other Methods

This category includes three approaches that don't belong to any of the previous categories. The first one is based on statistical procedures for directly selecting a subset of classifiers, the second is based on reinforcement learning and the third on boosting.

7.1 Statistical Procedures

Tsoumakas et al. [29, 28] prune an ensemble of heterogeneous classifiers using statistical tests that determine whether the differences in predictive performance among the classifiers of the ensemble are significant. Only the classifiers with significantly better performance than the rest are retained and subsequently combined with the method of (weighted) voting.

Such statistical tests are called *multiple comparisons procedures* [10]. Three of those that were used in [28] are Tukey's test [30], Hsu's test [11] and Scott & Knott's procedure [26], with the last one offering the largest benefit.

The disadvantage of these methods is that they don't take the diversity of the ensemble into consideration. However, they could potentially play the role of a first fast filtering of low performing models in a large ensemble, followed by a more advanced diversity-aware method.

7.2 Reinforcement Learning

Partalas et al. [21, 23] take a reinforcement learning approach to ensemble pruning. In specific, the problem of pruning an ensemble of T classifiers is modeled as an episodic task, where an agent takes T sequential actions, each one corresponding to either the inclusion or exclusion of classifier $h_t, t = 1 \dots T$ from the final ensemble. The Q-learning [31] algorithm is then used to approximate the optimal policy for this task.

7.3 Boosting

An approach similar to boosting was used for pruning an ensemble of classifiers produced via bagging in [20]. The algorithm iteratively selects the classifier with the lowest weighted error on the training set. Instance weights are initialized and updated according to the AdaBoost algorithm. The only difference is that instead of terminating the process when the weighted error is larger than 0.5, the algorithm resets all instance weights and continues selecting models. The complexity of this approach is $O(T^2N)$.

This approach ranks individual classifiers, but it does so based on their weighted error on the training set. Since at each step of the algorithm the instance weights depend on the classifiers selected up to that step, we refrained from categorizing this approach to ranking-based methods, where each model can be independently evaluated and ranked independently of the currently selected models.

8 Conclusions

This work presented a taxonomy of ensemble pruning methods. We believe that such a taxonomy is necessary for researchers working on new methods. It will help them identify the main categories of methods and their key points, and avoid duplication of work. Due to the large amount of existing methods and the different parameters of an ensemble selection framework (heterogeneous/homogeneous ensemble, algorithms used, size of ensemble, etc), it is possible to devise a new method, which may only differ in small, perhaps unimportant, details from existing methods. A generalized view of the methods, as offered from this work, will help avoid work towards such small differences, and perhaps may lead to more novel methods.

We do not argue that the proposed taxonomy is perfect. On the contrary, it is just a first step in abstracting and categorizing the different methods. We made an effort to include most of the important ensemble pruning methods, but no doubt, some high quality methods may have been left outside this study. For example, we haven't considered instance-base ensemble pruning methods that dynamically prune the ensemble for each test instance.

This work refrained from performing experimental comparisons between the methods. However, we would like to stress the importance of the following guidelines for empirical ensemble pruning studies. Firstly the ensemble should consist of a moderate size of models (e.g. 100 or more). For small ensemble sizes (e.g. 10 models), an exhaustive search for the best subset of models is computationally feasible, and perhaps even faster than some more complex methods of the literature. Secondly the study should include a large number of datasets, and include appropriate statistical tests for the comparison of different methods, in order to derive safe and useful conclusions.

References

1. Bakker, B., Heskes, T.: Clustering ensembles of neural network models. Neural Networks **16**(2), 261–269 (2003)
2. Banfield, R.E., Hall, L.O., Bowyer, K.W., Kegelmeyer, W.P.: Ensemble diversity measures and their application to thinning. Information Fusion **6**(1), 49–62 (2005)
3. Breiman, L.: Bagging predictors. Machine Learning **24**(2), 123–140 (1996)
4. Caruana, R., Niculescu-Mizil, A., Crew, G., Ksikes, A.: Ensemble selection from libraries of models. In: Proceedings of the 21st International Conference on Machine Learning (2004)
5. Dietterich, T.G.: Machine-learning research: Four current directions. AI Magazine **18**(4), 97–136 (1997)
6. Dietterich, T.G.: Ensemble Methods in Machine Learning. In: Proceedings of the 1st International Workshop in Multiple Classifier Systems, pp. 1–15 (2000)
7. Fan, W., Chu, F., Wang, H., Yu, P.S.: Pruning and dynamic scheduling of cost-sensitive ensembles. In: Eighteenth national conference on Artificial intelligence, pp. 146–151. American Association for Artificial Intelligence (2002)
8. Fu, Q., Hu, S.X., Zhao, S.Y.: Clusterin-based selective neural network ensemble. Journal of Zhejiang University SCIENCE **6A**(5), 387–392 (2005)
9. Giacinto, G., Roli, F., Fumera, G.: Design of effective multiple classifier systems by clustering of classifiers. In: 15th International Conference on Pattern Recognition, ICPR 2000, pp. 160–163 (2000)
10. Hochberg, Y., Tamhane, A.C.: Multiple comparison procedures. Wiley (1987)
11. Hsu, J.C.: Constrained simultaneous confidence intervals for multiple comparisons with the best. Annals of Statistics **12**, 1136–1144 (1984)
12. Jacobs, R.A., Jordan, M., Nowlan, S., Hinton, G.: Adaptive mixtures of local experts. Neural Computation **3**, 79–87 (1991)
13. Kuncheva, L.I.: Combining Pattern Classifiers: Methods and Algorithms. Wiley-Interscience (2004)
14. Kuncheva, L.I., Whitaker, C.J.: Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. Machine Learning **51**(2), 181–207 (2003)
15. Lazarevic, A., Obradovic, Z.: The effective pruning of neural network classifiers. In: 2001 IEEE/INNS International Conference on Neural Networks, IJCNN 2001, pp. 796–801 (2001)
16. Margineantu, D., Dietterich, T.: Pruning adaptive boosting. In: Proceedings of the 14th International Conference on Machine Learning, pp. 211–218 (1997)
17. Martinez-Munoz, 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)
18. Martinez-Munoz, G., Suarez, A.: Aggregation ordering in bagging. In: International Conference on Artificial Intelligence and Applications (IASTED), pp. 258–263. Acta Press (2004)

19. Martinez-Munoz, G., Suarez, A.: Pruning in ordered bagging ensembles. In: 23rd International Conference in Machine Learning (ICML-2006), pp. 609–616. ACM Press (2006)
20. Martinez-Munoz, G., Suarez, A.: Using boosting to prune bagging ensembles. Pattern Recognition Letters **28**(1), 156–165 (2007)
21. Partalas, I., Tsoumakas, G., Katakis, I., Vlahavas, I.: Ensemble pruning using reinforcement learning. In: 4th Hellenic Conference on Artificial Intelligence (SETN 2006), pp. 301–310 (2006)
22. Partalas, I., Tsoumakas, G., Vlahavas, I.: Focused ensemble selection: A diversity-based method for greedy ensemble selection. In: M. Ghallab, C.D. Spyropoulos, N. Fakotakis, N.M. Avouris (eds.) ECAI 2008 - 18th European Conference on Artificial Intelligence, Patras, Greece, July 21-25, 2008, Proceedings, *Frontiers in Artificial Intelligence and Applications*, vol. 178, pp. 117–121. IOS Press (2008)
23. Partalas, I., Tsoumakas, G., Vlahavas, I.: Pruning an ensemble of classifiers via reinforcement learning. Neurocomputing **(in press)** (2009)
24. Partridge, D., Yates, W.B.: Engineering multiversion neural-net systems. Neural Comput. **8**(4), 869–893 (1996)
25. Schapire, R.E.: The strength of weak learnability. Machine Learning **5**(2), 197–227 (1990)
26. Scott, A.J., Knott, M.: A cluster analysis method for grouping means in the analysis of variance. Biometrics **30**, 507–512 (1974)
27. Tang, E.K., Suganthan, P.N., Yao, X.: An analysis of diversity measures. Machine Learning **65**(1), 247–271 (2006)
28. Tsoumakas, G., Angelis, L., Vlahavas, I.: Selective fusion of heterogeneous classifiers. Intelligent Data Analysis **9**(6), 511–525 (2005)
29. Tsoumakas, G., Katakis, I., Vlahavas, I.: Effective Voting of Heterogeneous Classifiers. In: Proceedings of the 15th European Conference on Machine Learning, ECML2004, pp. 465–476 (2004)
30. Tukey, J.W.: The problem of multiple comparisons. Tech. rep., Princeton University (1953)
31. Watkins, C., Dayan, P.: Q-learning. Machine Learning **8**, 279–292 (1992)
32. Wolpert, D.H.: Stacked generalization. Neural Networks **5**, 241–259 (1992)
33. Yang, Y., Korb, K., Ting, K., Webb, G.: Ensemble selection for superparent-one-dependence estimators. In: AI 2005: Advances in Artificial Intelligence, pp. 102–112 (2005)
34. Yang, Y., Webb, G.I., Cerquides, J., Korb, K.B., Boughton, J., Ting, K.M.: To select or to weigh: A comparative study of linear combination schemes for superparent-one-dependence estimators. IEEE Transactions on Knowledge and Data Engineering **19**(12), 1652–1665 (2007)
35. Zhang, Y., Burer, S., Street, W.N.: Ensemble pruning via semi-definite programming. Journal of Machine Learning Research **7**, 1315–1338 (2006)
36. Zhou, Z.H., Tang, W.: Selective ensemble of decision trees. In: 9th International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing, RSFDGrC 2003, pp. 476–483. Chongqing, China (2003)