Contents

0-8493-0052-5/00/$0.00+$.50

Chapter 1

Ensemble methods: a review

Matteo Re
DSI, Dipartimento di Scienze dell'Informazione Università degli Studi di Milano, Italy

Giorgio Valentini
DSI, Dipartimento di Scienze dell'Informazione Università degli Studi di Milano, Italy

1.1 Introduction

Ensemble methods are statistical and computational learning procedures reminiscent of the human social learning behaviour of seeking several opinions before making any crucial decision. The idea of combining the opinions of different "experts" to obtain an overall "ensemble" decision is rooted in our culture at least from the classical age of ancient Greece, and it has been formalized during the Enlightenment with the *Condorcet Jury Theorem* [45]), that proved that the judgment of a committee is superior to those of individuals, provided the individuals have reasonable competence.

Ensembles are sets of learning machines that combine in some way their decisions, or their learning algorithms, or different views of data, or other specific characteristics to obtain more reliable and more accurate predictions in supervised and unsupervised learning problems [48, 116]. A simple example is

0-8493-0052-5/00/$0.00+$.50

represented by the *majority vote* ensemble, by which the decisions of different learning machines are combined, and the class that receives the majority of "votes" (that is, the class predicted by the majority of the learning machines) is the class predicted by the overall ensemble [158].

In the literature, a plethora of terms other than ensembles has been used, such as fusion, combination, aggregation, committee, to indicate sets of learning machines that work together to solve a machine learning problem [123, 108, 99, 40, 56, 19, 66], but in this paper we maintain the term *ensemble* in its widest meaning, in order to include the whole range of combination methods.

Nowadays ensemble methods represent one of the main current research lines in machine learning [48, 116], and the interest of the research community on ensemble methods is witnessed by conferences and workshops specifically devoted to ensembles, first of all the *Multiple Classifier Systems* conference organized by Roli, Kittler, Windeatt and other researchers of this area [173, 149, 85, 14, 62].

Several theories have been proposed to explain the characteristics and the successful application of ensembles to different application domains. For instance, Allwein, Schapire and Singer interpreted the improved generalization capabilities of ensembles of learning machines in the framework of large margin classifiers [177, 4], Kleinberg in the context of Stochastic Discrimination Theory [112], and Breiman and Friedman in the light of the bias–variance analysis borrowed from classical statistics [21, 70].

Empirical studies showed that both in classification and regression problems ensembles improves on single learning machines, and moreover large experimental studies compared the effectiveness of different ensemble methods on benchmark data sets [11, 49, 10, 188]. The interest in this research area is motivated also by the availability of very fast computers and networks of workstations at a relatively low cost that allow the implementation and the experimentation of complex ensemble methods using off-the-shelf computer platforms. However, as explained in Sect. 1.2 of this paper there are deeper reasons to use ensembles of learning machines, motivated by the intrinsic characteristics of the ensemble methods.

The main aim of this chapter is to introduce ensemble methods and to provide an overview and a bibliography of the main areas of research, without pretending to be exhaustive or to explain the detailed characteristics of each ensemble method. The paper is organized as follows. In the next section the main theoretical and practical reasons for combining multiple learners are introduced. Sect. 1.3 depicts the main taxonomies on ensemble methods proposed in the literature. In Sect. 1.4 and 1.5 we present an overview of the main supervised ensemble methods reported in the literature, adopting a simple taxonomy, originally proposed in [201]. Applications of ensemble methods are only marginally considered, but a specific section on some relevant applications of ensemble methods in astronomy and astrophysics has been added

(Sect. 1.6). The conclusions (Sect. 1.7) end this paper and lists some issues not covered in this work.

1.2 Theoretical and practical reasons for combining classifiers

There is no a unified theory underlying ensemble methods, and several authors outlined that consistent and theoretically sound explanations of the success of classifier ensembles are not available [116] or are incomplete or assumption-bound [90]. This is not surprising, considering the variety of the proposed approaches and the relative youngness of this research area.

Theories on ensemble methods. Despite these negative considerations, we would like to cite at least three theories able to explain the effectiveness of some of the most widely known and used supervised ensemble methods. The first one consider the ensembles in the framework of large margin classifiers [133], showing that ensembles enlarge the margins, enhancing the generalization capabilities of Output Coding [4] and boosting-based ensemble algorithms [177]. This interpretation is strictly related to the Vapnik's Statistical Learning Theory [203], that is the likely most accredited theory within the machine learning community.

The second is based on the classical bias–variance decomposition of the error [76], and it shows that ensembles can reduce variance [20, 124] or both bias and variance [113, 22, 176]. Recently Domingos proved that these two theories are two faces of the same coin. Indeed Schapire's notion of margins [177] can be expressed in terms of bias and variance and viceversa, and hence Schapire's bounds of ensemble's generalization error can be equivalently expressed in terms of the distribution of the margins or in terms of the bias–variance decomposition of the error, showing the equivalence of margin-based and bias–variance-based approaches [53, 52].

Another general theory about ensemble methods has been proposed by Kleinberg [111, 110]. His Stochastic Discrimination theory is founded on a set-theoretic abstraction to remove all the algorithmic details of classifiers and training procedures. By this abstraction the classifiers are considered as a combination of subsets of points of the feature space underlying a given problem, classifiers' decision regions are considered only in form of point sets, and the set of classifiers is just a sample into the power set of the feature space. A rigorous mathematical treatment starting from the "representativeness" of the examples used in machine learning problems leads to the design of ensemble of weak classifiers, whose accuracy is governed by the law of large numbers [109, 38].

Statistical, representational and computational reasons for com-

bining multiple learners. Without pretending to depict a general theory on ensemble methods, Thomas Dietterich suggested three main reasons why an ensemble of classifier might be better than a single classifier [48].

The first one is statistical. Indeed learning algorithms try to find an hypothesis in a given space $\mathcal{H}$ of hypotheses, and in many cases if we have sufficient data they can find the optimal one for a given problem, but in real cases we have only limited data sets and sometimes only few examples are available. The "subregion" $\mathcal{S} \subset \mathcal{H}$ of the "optimal" hyphotheses with respect to the training error, may correspond to classifiers having different generalization performances. We could in principle try to select among them the simplest or the one with the lowest capacity, but in practice this is difficult: we can avoid this problem by averaging or combining the base classifiers to get a good approximation of the unknown true hypothesis.

Continuing to follow Dietterich's analysis, the second reason for combining multiple learners arises from the limited representational capability of learning algorithms. In many cases the unknown function to be approximated is not present in $\mathcal{H}$, but a combination of hypotheses drawn from $\mathcal{H}$ can expand the space of representable functions, possibly embracing also the true one. It is well-known that many learning algorithms enjoy universal approximation properties [94, 151], but these asymptotic features do not hold with finite data sets, since the effective space of hypotheses explored by the learning algorithm with small-sized data can be significantly smaller than the virtual $\mathcal{H}$ considered in the asymptotic case. From this standpoint ensembles can enlarge the effective hypotheses coverage, expanding the space of representable functions.

The third reason is computational, in the sense that training algorithms may get stuck in local optima. For instance multi-layer perceptrons apply gradient descent techniques to minimize an error function over the training data, and inductive decision trees employ a greedy local optimization approach, both resulting in suboptimal solutions due to multiple local minima of the underlying error function to be minimized. As a consequence, even if the learning algorithm could in principle find the best hypothesis, we actually may not be able to find it. An ensemble merging different local suboptimal solutions may achieve a better approximation, at least avoiding the worst local minima solutions.

The accuracy-diversity trade-off. A simple example, due another time to Dietterich [48], is useful to introduce another open theoretical issue about ensemble methods: the so-called accuracy-diversity trade-off [121]. Having a set of L classifiers whose error is lower than random guessing for a two-class classification problem (that is an error lower than 0.5), it is easy to see that the overall error of the majority voting ensemble, given by the area under the binomial distribution where more than $L/2$ hypotheses are wrong, is significantly lower than the error of the base classifier. It is worth noting that this result is known since the end of the XVIII century in the context of social sciences: in fact the *Condorcet Jury Theorem* [45]) proved that the judgment

of a committee is superior to those of individuals, provided the individuals have reasonable competence (that is, on the assumption that their probability of being correct is higher than 0.5). Nevertheless, this result holds only if the base classifiers (individuals) are independent: if their decisions are dependent, that is similar for a given input data sets, we have not a series of independent Bernoulli experiments, and we have no guarantee of a reduced error when a majority voting ensemble is applied. This fact is a particular case of the more general problem of the relationships between accuracy and diversity of base learners within an ensemble: the performance of an ensembles depend on the accuracy of the component base learners, but also on their diversity, that is on their capability of responding differently to the same input. On one hand, if each base learner provides the same predictions, there is no utility in combining their outputs, but on the other hand, if the base learners are maximally accurate, they generate the same correct predictions, with no diversity between them. The resulting trade-off between accuracy and diversity has been actively studied, starting from the pioneering work of Tumer and Gosh, who showed how ensemble error decreases as base model error decreases and diversity increases [195]. Nevertheless, Kuncheva studies showed that the relationships between accuracy and diversity of base learners are more complex, showing e.g. that the way the ensemble method generates base learners and the behaviour of the base learning algorithms play a crucial role in determining the characteristics of the accuracy-diversity trade-off [120, 121].

Empirical reasons for combining multiple learners. There are also practical reasons for using ensemble methods, as witnessed by their successful applications in several domains [143, 144]. Indeed employing multiple learners can derive from the application context, such as when multiple sources data are available, inducing a natural decomposition of the problem. In more general cases we can dispose of different training sets, collected at different times, having eventually different features and we can use different specialized learning machine for each different item. Predictive performances of single models have been improved by the ensemble methodology in several application fields, ranging from information security [139], astronomy and astrophysics [13], geography and remote sensing [17], image retrieval [190], finance [128], to medicine [160], bioinformatics [166] and chemioinformatics [140].

1.3 Taxonomies of ensemble methods

Considering the variety of ensemble techniques and the large number of combination schemes proposed in literature, it is not surprising that a very large number of ensemble methods and algorithms are now available to the research community. To help the researchers and practitioners to get their bearings and to develop new methods and techniques, several taxonomies of

ensemble methods have been proposed. Indeed combination techniques can be grouped and analyzed in different ways, depending on the main classification criterion adopted. For instance, if we consider the representation of the input patterns as the main criterion, we can identify two distinct large groups, one that uses the same and one that uses different representations of the inputs [107, 108].

Another criterion distinguishes between strictly speaking ensemble systems and modular systems: the latter is characterized by component learners devoted to different tasks by which the original problem has been decomposed, the former by a combination of a set of classifiers, each of which solves the same original task [184, 122]. A taxonomy can also be based on the way diversity between base learners is achieved, i.e. implicitly between randomization methods like bagging and random subspace techniques, or by methods that explicitly improves diversity through a proper metric [28]. The differentiation between trainable and non-trainable ensembles represents another key to classify ensemble methods: non-trainable ensembles do not need training after the base learners have been induced (they apply "fixed" rules to combine base classifiers), while trainable ensembles imply the training of the combiner module, either during or after the base learners have been trained [58, 116].

Sharkey [182] proposes a multi-dimensional taxonomy, founded on three dichotomies: 1) selection or combination of the multiple base learners; 2) methods based or not on the direct combination of base learner outputs; 3) methods based on ensembles or modular systems. Extending this approach, a more complex five-dimensional taxonomy has been proposed by Rokach, based on combiner usage, classifier dependency, diversity generation, ensemble size and the capability of ensemble methods to be applied with different base learning algorithms [171].

In her fundamental book on ensemble methods, Lucy Kuncheva proposes a four level taxonomy based on the way ensembles are constructed [116]. At a first level the author highlights the combination rules to assemble multiple classifiers, distinguishing between fusion methods that combine in some way the outputs of the base learners and selection methods, by which a single classifier is selected among the set of available base classifiers. At a second level we may consider different models, and we may design base learners for specific ensemble methods. At feature level different subsets of features can be used for the classifiers. Finally, different data subsets, so that each base classifier in the ensemble is trained on its own data, can be used to build up the committee of learning machines.

In this survey we adopt the classification scheme originally proposed in [201] (with some minor modifications borrowed from [116]), not because we consider this taxonomy better than others, but simply because it is quite simple and clean and facilitates us to introduce the main ensemble methods presented in the literature. Indeed ensemble methods are characterized by two basic features: 1) the algorithms by which different base learners are combined; 2) the techniques by which different and diverse base learners are generated.

Our proposed taxonomy basically distinguishes between *non-generative* ensemble methods that mainly rely on the former feature of ensemble methods, and *generative* ensembles that mainly focus on the latter. It is worth noting that the "combination" and the "generation" of base learners are somehow both present in all ensemble methods: the distinction between these two large classes depends on the predominance of the combination or of the generation component of the ensemble algorithm.

More precisely, *non-generative* ensemble methods confine themselves to combine a set of possibly well-designed base classifiers: they do not actively generate new base learners but try to combine in a suitable way a set of existing base classifiers. Examples are methods that combine the output of a set of base learners by majority voting [158], or methods that select the best subset of base learners on the basis of their accuracy [156], or methods that combine the probabilistic output of a set of classifiers according to the Bayes rule [57]. Note that in all these cases the emphasis is placed on the way the base learners are combined or selected, and not on the way different and diverse classifiers are generated.

On the contrary, *generative* ensemble methods generate sets of base learners acting on the base learning algorithm or on the structure of the data set to try to actively improve diversity and accuracy of the base learners. In this case the emphasis is placed on the way diverse base learners are constructed, while the combination technique does not represent the main issue of the ensemble algorithm. Examples are resampling methods, that train base learners on different bootstrap replicates of the data [20], or random subspace algorithms that generate diverse base learners by using different randomly selected subsets of features [87], or mixture of experts methods, where a gating network performs the division of the input space and an ensemble of neural networks perform the effective calculation at each assigned region separately [103].

Table 1.1 provides a high-level scheme of the taxonomy of ensemble methods proposed in this paper. *Non generative* methods (Sect. 1.4) are partitioned in *Ensemble fusion* (Sect 1.4.1) and *Ensemble selection* (Sect. 1.4.2) methods, while the other high level and more heterogeneous branch of the taxonomy, i.e. *Generative* ensembles (Sect. 1.5), is subdivided in *Resampling* (Sect. 1.5.1), *Feature selection* (Sect. 1.5.2), *Mixture of experts* (Sect. 1.5.3), *Output Coding* (Sect. 1.5.4), and *Randomized ensembles* (Sect. 1.5.5) methods.

It is worth noting that semi-supervised and unsupervised ensemble methods have been recently proposed. Unfortunately, for lack of space we do not discuss these topics, and we refer the reader to Section 8.3 of Kuncheva's book [116], or to the brief review provided in [77]. However, some examples of the application of unsupervised ensemble methods to astronomy and astrophysics problems are described in Sect. 1.6 of this chapter.

TABLE 1.1: A taxonomy for ensemble methods

Non generative ensembles	*Ensemble fusion methods*	- Majority voting - Naive Bayes rule - Behavior-Knowledge-Space - Algebraic operators fusion - Fuzzy fusion - Decision Template - Meta Learning - Multi-label hierarchical methods
	Ensemble selection methods	- Test and select - Cascading classifiers - Dynamic classifier selection - Clustering based selection - Pruning by statistical tests - Pruning by semidef. programming - Forward/Backward selection
Generative ensembles	*Resampling methods*	- Bagging - Boosting - Arcing - Cross-validated committees
	Feature selection and extraction methods	- Random Subspace - Similarity based selection - Input decimation - Feature subset search - Rotation forests
	Mixture of experts	- Gating network selection - Hierarchical mixture of experts - Hybrid experts
	Output Coding methods	- One Per Class - Pairwise and Correcting Classifiers - ECOC - Data driven ECOC
	Randomized methods	- Randomized decision trees - Random forests - Pasting small votes

1.4 Non-generative ensembles

Following the taxonomy proposed in [116], we can subdivide non-generative strategies in *ensemble fusion* and *ensemble selection* methods. Both these approaches share the very general common property of using a predetermined set of learning machines previously trained with suitable algorithms. The base learners are then put together by a combiner module that may vary depending on the requirement of the output of the individual learning machines: for instance the combiner may need the labels of the classes, or a ranking of the classes, or a support (e.g. the a posteriori probability estimation) for each class [210]. Moreover we may distinguish between combiners that are trainable and not trainable [58]. Very schematically ensemble fusion methods combine all the outputs of the base classifiers, while ensemble selection methods try to choose the "best classifiers" among the set of the available base learners.

1.4.1 Ensemble fusion methods

The most popular ensemble fusion method is represented by the *majority voting* ensemble, by which each base classifier "votes" for a specific class, and the class that collects the majority of votes is predicted by the ensemble [106, 158, 124]. By generalizing this approach, Xu et al. proposed a *thresholded plurality vote*: by imposing a threshold on the number of votes to select the class, we may move from an *unanimity vote* rule, by which we choose a class only if all the base classifiers agree on the corresponding label, to the simple *majority voting* rule, by which it sufficient to achieve the majority of votes to select the class; intermediate cases can be considered by moving the threshold of votes, at the expenses of some possible unclassified example [210]. This approach can be refined assigning different weights to each classifier to optimize the performance of the combined classifier, according to the base learner accuracy estimated on a validation set [123, 147]. By generalizing the analysis to linear combiners, Fumera and Roli showed the reasons why weighted average improves on simple average combining rule [73].

Assuming conditional independence between classifiers, a *Naive-Bayes decision rule* selects the class with the highest posterior probability computed through the estimated class conditional probabilities and the Bayes' formula [54, 57]. A Bayesian approach has also been used in *Consensus based classification* of multisource remote sensing data [16, 15, 27], outperforming conventional multivariate methods for classification. To overcome the problem of the independence assumption (that is unrealistic in most cases), the Behavior-Knowledge Space (BKS) method [96] considers each possible combination of class labels, filling a look-up table using the available data set, but this technique requires a huge volume of training data.

Other simple operators such as *Minimum*, *Maximum*, *Average*, *Product*

and *Ordered Weight Averaging* have been applied to combine multiple classifiers [175, 26, 114]. On the basis of a common Bayesian framework, Josef Kittler provided a theoretical underpinning of many existing classifier combination schemes based on the product and the sum rule, showing also that the sum rule is less sensitive to the errors of subsets of base classifiers [108].

Fuzzy set theory has also been applied to combine multiple learners through proper fuzzy aggregation connectives [40, 39, 105, 207, 205]. In particular the fuzzy integral has been reported to give excellent results as a classifier combiner. Its effectiveness comes from the fact that it measures the "strength" of every subset of classifiers, and not only the strength of each individual classifier: as a consequence, for each example to be classified, the decision of the ensemble is based on the competence of every subset of base learners [121]. If the classifier outputs are possibilistic, *Dempster-Schafer* combination rules can be applied [169].

A valuable approach that takes into account the prediction-scores (e.g. the support) of the base learners for each class to be predicted is represented by the *Decision templates* (DT) [117]. The main idea behind decision templates consists in comparing a "prototypical answer" of the ensemble for the examples of a given class (the template), to the current answer of the ensemble to a specific example whose class needs to be predicted (the decision profile). Different similarity measures can be used to evaluate the matching between the matrix of classifier outputs for a given input, that is the decision profiles, and the matrix templates (one for each class) found as the class means of the classifier outputs. This approach can be easily applied to combine multiple set of features or sources of data to improve predictions [47, 164]. In [165] the authors analyzed the tolerance of DT and other ensemble methods to noisy data, and showed that DT are particularly resistant to the addition of noisy data sets.

Ensemble fusion can be performed also by second-level trainable combiners, through *meta-learning* techniques [59]. For instance, in *stacking* methods the outputs of the base learners are interpreted as features in an intermediate space: the outputs are fed into a second-level machine to perform a trained combination of the base learners [208]. By extending this approach a new method based on multiresponse linear regression have been shown to outperform the original Wolpert's stacking approach [60]. Stacking requires a careful training of the base learners and of the combiner: if we use the same training data for both, it is likely to incur in overfitting. To avoid this problem, L_2 norm (ridge regression) and L_1 norm (Lasso regression) penalization have been introduced in linear models for stacked generalization [167]. Another type of meta-learning ensemble is represented by methods that use an arbiter or a combiner to decide recursively in a hierarchically structured input space on the basis of the predictions made by the base learners. The aim of this strategy is respectively to provide an alternate classification when the base learners disagree (*arbiter trees*) [34] or to combine the outputs of the base classifiers by learning their relationships with the correct labels (*combiner trees*) [35, 95].

Several ensemble methods have been devoted to the *classification of hierarchically structured classes*, as those related to the classification of texts in the web or to the classification of genes in functional genomics [204, 100, 3]. Different ensemble based algorithms have been proposed ranging from methods restricted to multilabels with single and no partial paths [46] to methods extended to multiple and also partial paths [31]. In this context hierarchical ensemble methods are in general characterized by a two-step strategy: a) flat learning of the classes as a set of independent classification problems; b) combination of the predictions by exploiting the relationships between classes that characterize the hierarchy. Some recently published works clearly demonstrated that this approach ensures an increment in precision with respect to "flat" methods, but this comes at expenses of the overall recall [83, 142, 33]. A recently proposed hierarchical ensemble approach, proposed in the context of functional genomics, partially overcomes this problem, and can be in principle extended to other application domains characterized by the unbalance of the classes and hierarchically structured relationships between classes [198]. Hierarchically structured ensembles, originally proposed in [33, 198] have been also successfully combined with majority voting ensembles and kernel fusion methods to integrate multiple sources of data in the context of gene function prediction problems [32]. Finally, a different method, based on ensembles of hierarchical multi-label decision trees, developed for the prediction of gene functions, is general enough to be adapted to other classification problems with classes structured according to a direct acyclic graph [179].

1.4.2 Ensemble selection methods

This general approach tries to identify the "best" base classifier among the set of base learners for a specified input, and the output of the ensemble is the output of the selected best classifier. From a more general standpoint also a subset of base classifiers can be chosen. In this case we need to decide to pick one of the selected outputs as the ensemble output, or to combine the output of the base learners, according, e.g., to one of the ensemble fusion methods described in the previous section. To design an ensemble selection method we need to decide how to build the individual classifiers, how to evaluate the competence of each classifier on a specific input, what selection strategy to use [116].

The *test and select* methodology relies on a a greedy approach, by which a new learner is added to the ensemble only if the resulting squared error is reduced [158], but in principle any optimization technique can be used to select the "best" component of the ensemble, including genetic algorithms [146, 125].

Another possible approach is represented by *cascading classifiers*. The base learners are applied sequentially, and if the confidence of the first classifier is high, its prediction is taken, otherwise the prediction is recursively demanded to the next classifier and so on. This "cascade" model is useful especially with

real-time systems, since most of the inputs need to be processed by a few classifiers [6, 74].

The competence of each base classifier can be estimated dynamically (that is during the operational phase) by using only "a priori" information on the features of the input pattern (with no knowledge about its predicted labels) or by using "a posteriori" information on the labels predicted by all the classifiers, by applying e.g. k-NN (k Nearest Neighbours) estimates [209]. This approach is also known as *Dynamic Classifier Selection* [91, 209, 80] and it is based on the definition of a function selecting for each pattern the classifier which is likely the most accurate, estimating, for instance the accuracy of each classifier in a local region of the feature space surrounding an unknown test pattern [80].

Nevertheless, this dynamical approach might be too computationally intensive and a less demanding preestimation of the competence region can be applied. To this end, the region of competence for the base classifiers can be determined through clustering methods [130, 115]. Clustering algorithms have also been employed to discover groups of base learners that make similar predictions, then models are picked from each cluster to both select a subset of the available base learners and to improve the diversity of the ensemble [82, 127, 81].

Other greedy search methods are based on a ranking that gives preference to classifiers that are able to correct the incorrect predictions of the ensemble, as in *Orientation Ordering* ensembles, thus assuring the selection of base learners able to improve the prediction of the overall ensemble [132]. Ensembles of heterogeneous classifiers can also be pruned by statistical procedures that select only those classifiers with significantly better performances, combined in a second step through majority voting [192, 191].

Ensemble pruning can also be formulated as a *semi-definite programming* problem that minimizes misclassification and maximize diversity, with the constraints of selecting only a subset of the available classifiers: by setting the number of base classifiers to be selected as an input parameter Zhang et al. approximating solve the resulting quadratic integer programming problem in polynomial time [213].

Algorithms borrowed from the feature selection literature or for the solution of complex optimization tasks (tabu search) are proposed in [172]. Forward selection [30] and backward elimination [9] ensemble selection algorithms, adapted from the corresponding feature selection literature, respectively add or remove base classifiers selected according to the minimization of an objective function. Another valuable approach is represented by a greedy search that takes into account both ensemble decision strength and the diversity of the base learners to select the base learners [154]. Recently Partalas, Tsoumakas and Vlahavas proposed a new ensemble pruning method via directed hill climbing: they introduced a new measure to select models that takes into account the uncertainty of the decision of the ensemble. Through the proposed *Uncertainty Weighted Accuracy* measure, the authors select a

small subset of base classifiers, achieving state-of-art results on a large set of benchmark data sets [155].

1.5 Generative ensembles

In this section we introduce ensemble methods able to generate base learners by acting on the base learning algorithm or on the structure of the data set to try to actively boost diversity and accuracy of the base learners. These methods can perturb the structure and the characteristics of the available input data, as in *resampling* methods or in *feature selection/subsampling* methods, or can manipulate the aggregation and the coding of the classes (*Output Coding* methods), or can select base learners specialized for a specific input region (*mixture of experts* methods). They can also randomly modify the base learning algorithm, or apply randomized procedures to the learning processes to improve the diversity or to avoid local minima of the error (*randomized* methods).

1.5.1 Resampling methods

Resampling techniques can be used to generate different hypotheses. For instance, bootstrapping techniques [61] may be used to generate different training sets and a learning algorithm can be applied to the obtained subsets of data in order to produce multiple hypotheses. These techniques are effective especially with unstable learning algorithms, which are algorithms very sensitive to small changes in the training data, such as neural-networks and decision trees [64].

Bagging (an acronym for *b*ootstrap *agg*regat*ing*) builds up the ensemble by making bootstrap replicates of the training sets, and the multiple hypotheses, resulting from the application of a suitable learning algorithm to the perturbed data, are used to get an aggregated predictor [20]. The aggregation can be performed averaging the outputs in regression or by majority or weighted voting in classification problems [185, 186].

By applying procedures to estimate the bias and variance of each base learner, an enhanced version of bagging, tailored to the characteristics of Support Vector Machines is proposed in [199]: considering that bagging is a variance-reduction methods, by selecting the SVMs with a low bias, we may obtain ensembles that lower both the variance and the bias component of the error. This technique comes from a more general approach to design ensembles based on the bias-variance decomposition of the error, to exploit the specific learning characteristics of the base learners [200].

Another variant of bagging, based on a non-uniform probability to extract examples from the training set is *Wagging*: while in bagging each example

is drawn with equal probability from the available training data, in wagging each example is extracted according to a weight stochastically assigned [11].

Extending this approach, in *boosting* methods the learning algorithm at each iteration uses a different distribution or weighting over the training examples [175, 176, 174, 67, 56, 55], according to the errors of the base learners. The most known algorithm in this family is surely *AdaBoost* [69, 68]. This technique places the highest weight on the examples most often misclassified by the previous base learner: in this way the base learner focuses its attention on the hardest examples. Then the boosting algorithm combines the outputs of the base learners by weighted majority voting. Schapire and Singer showed that the training error exponentially drops down with the number of iterations [178] and Schapire et al. [177] proved that boosting enlarges the margins of the training examples, showing also that this fact translates into a superior upper bound on the generalization error. It is worth noting that this ensemble method is one of the most studied, with a solid theoretical background, and largely applied in several application domains.

Breiman proposed an algorithm similar to the AdaBoost algorithm, that he named *arcing* (adaptive resampling and combining) to investigate whether the success of AdaBoost is due to technical details or to the resampling scheme adopted [22, 24]. His conclusions showed that AdaBoost is a well-theoretically founded algorithm, while arcing is basically a heuristic with empirical results comparable to AdaBoost [22]. Different variants of boosting algorithms for multiclass problems, or real-valued classifier outputs have been proposed [178, 5].

Relationships of boosting with logistic regression have been analyzed in [71, 43], giving raise to a parametrized family of iterative algorithms [43] and to the *LogitBoost* algorithm, that casts AdaBoost in a statistical framework, by applying the cost functional of logistic regression and reinterpreting boosting as a generalized additive model [71]. Instead of re-weighting, a new boosting by resampling technique can be adopted: a local error for each training example is computed and then used to update the probability of drawing the example at the next iteration of the algorithm [212].

Experimental work showed that bagging is effective with noisy data, while boosting, concentrating its efforts on noisy data is quite sensitive to noise [163, 49]. Nevertheless, boosting algorithms designed for noisy data partially overcome this problem [148, 189]. Finally, another approach based on subsampling to achieve diversity consists in constructing training sets by leaving out disjoint subsets of the training data as in *cross-validated committees* [152, 153] or sampling without replacement [183].

1.5.2 Feature selection/extraction methods

Reducing the number of input features of the base learners, we can contrast the effects of the classical curse of dimensionality problem that characterize high-dimensional and sparse data [72]. For instance, by applying feature se-

lection algorithms or subsampling methods to draw subsets of features from the available data, we can construct sets of diverse base classifiers that can be combined through appropriate ensemble fusion techniques.

A random strategy can be applied to select sets of features. In *Random Subspace* methods [87, 120], a subset of features is randomly selected and assigned to an arbitrary learning algorithm: a random subspace of the original feature space is obtained, and classifiers are constructed inside this reduced subspace. The aggregation is usually performed using weighted voting on the basis of the base classifiers accuracy, but other techniques could be in principle applied. It has been shown that this method is effective for classifiers having a decreasing learning curve constructed on small and critical training sample sizes [187].

By using a dissimilarity representation of the objects, e.g. the distances between the pairs of examples in the training set, we can construct a "similarity-based" feature space that resembles an approach similar to kernel methods [180]. By adopting this approach a linear discriminant classifier applied on subsets of randomly selected dissimilarity features has been proposed [157].

An open problem in random subspace methods is represented by the choice of the dimension of the projected subspace. Ho suggested a dimension about equal to the half of the available features [87], and in [29] a method based on a random search in the feature subset spaces is proposed. However, both methods are heuristics, even if supported by empirical evidence of their effectiveness.

Following a different approach, sets of features can also be chosen by non-random selection methods. For instance, the *Input Decimation* approach [150, 104] reduces the correlation among the errors of the base classifiers, decoupling the base classifiers by training them with different subsets of the input features. It differs from the previous Random Subspace Method, since for each class the correlation between each feature and the output of the class is explicitly computed, and the base classifier is trained only on the most correlated subset of features.

Extending this approach, various criteria instead of the simple correlation between features and class labels have been introduced; moreover, base learner selection is accomplished by checking both the accuracy and the diversity of the base learners [194, 193]. Gunter and Bunke apply different feature subset search algorithms to find different subsets of features; to incrementally select the base learners, they take into account the diversity and the accuracy of the overall ensemble [84]. Genetic and evolutionary techniques have also been applied to construct ensembles based on feature subset selection [145, 118, 170].

Another effective method that relies on feature extraction techniques is represented by the *rotation forests* [168]. Features are randomly split into n subsets, and n axis rotation are performed to encourage simultaneously individual accuracy and diversity within the ensemble. In a comparative experi-

mental study rotation forests achieve better results than those obtained with bagging, boosting and random forests [119].

1.5.3 Mixture of experts

A general approach similar to ensemble selection is represented by the *mixture of experts* methods [98, 97]. It differs from selection methods by the fact that the recombination of the base learners is governed by a supervisor learning machine, that selects the most appropriate element of the ensemble on the basis of the available input data: a gating network performs the division of the input space and an ensemble of neural networks perform the effective calculation at each assigned region separately. The output of the gating network are interpreted as probabilities for selecting the expert responsible for the prediction on a given input. These probabilities can be used to stochastically select the expert, or to choose the expert according to a winner-takes-all paradigm, or as weights to combine the outputs of the multiple base learners (experts). Through this type of ensemble methods both the selector (the gating network) and the base classifiers can be trained with standard learning algorithms: the standard back-propagation algorithm or the expectation-maximization method [103]. An extension of this model is the *hierarchical mixture of experts* method, where the outputs of the different experts are non-linearly combined by different supervisor gating networks hierarchically organized [101, 102, 97].

Cohen and Intrator extended the idea of constructing local simple base learners for different regions of input space, searching for appropriate architectures that should be locally used and for a criterion to select a proper unit for each region of the input space [41, 42]. They proposed a hybrid MLP/RBF network by combining RBF and Perceptron units in the same hidden layer and using a forward selection procedure to add units until the error drops to a given threshold. Although the resulting *Hybrid Perceptron/Radial Network* is not in a strict sense an ensemble, the way by which the regions of the input space and the computational units are selected and tested could be in principle extended to ensembles of learning machines.

1.5.4 Output Coding methods

By manipulating the coding of classes in multi-class classification problems, we can construct ensembles able to partially correct errors committed by the base learners, exploiting the redundancy in the bit-string representation of the classes [138, 51, 48]. More precisely, *Output Coding* (OC) methods decompose a multiclass–classification problem in a set of two-class subproblems, and then recompose the original problem combining them to achieve the class label. An equivalent way of thinking about these methods consists in encoding each class as a bit string (named codeword), and in training a different two-class base learner (dichotomizer) in order to separately learn each codeword bit.

When the dichotomizers are applied to classify new points, a suitable measure of dissimilarity between the codeword computed by the ensemble and the codeword classes is used to predict the class (e.g. Hamming distance) [50].

Different *decomposition schemes* have been proposed in literature: In the One-Per-Class (OPC) decomposition [8], each dichotomizer separates a single class from all others; in the *PairWise Coupling* (PWC) decomposition [86], the task of each dichotomizer consists in separating a pair of classes, ignoring all other classes; the *Correcting Classifiers* (CC) and the *PairWise Coupling Correcting Classifiers* (PWC-CC) are variants of the PWC decomposition scheme, that reduce the noise originated in the PWC scheme due to the processing of non pertinent information performed by the PWC dichotomizers [141].

Error Correcting Output Coding (ECOC) [50, 51] is the most studied OC method, and has been successfully applied to several classification problems [2, 18, 78, 196, 211]. This decomposition method tries to improve the error correcting capabilities of the codes generated by the decomposition through the maximization of the minimum distance between each pair of codewords [113]. This goal is achieved by means of the redundancy of the coding scheme [202].

The trade-off between error recovering capabilities and complexity/learnability of the dichotomies induced by the decomposition scheme has been studied in [4]. The effectiveness of ECOC decomposition methods depends mainly on the design of the learning machines implementing the decision units, on the similarity of the ECOC codewords, on the accuracy of the dichotomizers, on the complexity of the multiclass learning problem and on the correlation of the codeword bits [134, 135, 136, 137].

The design of ECOC codes tuned to the characteristics of the data, in order to obtain codes and dichotomies that are both "simple" and able to recover errors of the base learners, is another interesting issue considered in several works [138, 7]. In [44] it is shown that given a set of dichotomizers the problem of finding an optimal decomposition matrix is NP-complete: by introducing continuous codes and casting the design problem of continuous codes as a constrained optimization problem, we can achieve an optimal continuous decomposition using standard optimization methods.

Data-driven ECOC (DECOC) analyzes the distribution of data classes to optimize both the composition and the number of the base learners [214]. Compact ECOC codes, able to code classes using very compact but effective codes, well-suited for problems characterized by very large number of classes have been proposed in [161], and adapted to the characteristics and the distribution of the data [12]. Recently ternary ECOC codes (adding a "don't care" bit) have been extensively studied and a taxonomy of both binary and ternary ECOC codes is proposed in [63].

1.5.5 Randomized ensemble methods

Randomness plays a central role to generate set of diverse base classifiers: for instance we can randomly draw examples (bagging, Sect. 1.5.1) or features (random subspace, Sect. 1.5.2) from the training data to construct ensembles of diverse classifiers. Moreover several experimental results showed that randomized learning algorithms used to generate base elements of ensembles improve the performances of single non-randomized classifiers. For instance in [49] randomized decision tree ensembles outperform single C4.5 decision trees [162], and adding gaussian noise to the data inputs, together with bootstrap and weight regularization can result in large improvements in classification accuracy [163].

By extending this approach, Leo Breiman proposed a general class of ensembles, the *random forests* [25], using decision trees as base classifiers. A random forest can be constructed by randomly sampling from the data set, or by sampling from the feature set, or from both. For instance, along with selecting examples bootstrapped from the available training data, a random subset of features is drawn at each node of the tree and the best one is selected among this set to split the nodes. Random forests are also implicitly able to select the most relevant features associated to the classification problem they are applied to [25].

Randomness plays a role also when ensembles are built to deal with very large or distributed data sets. Indeed in these situations ordinary learning algorithms cannot directly process the data set as a whole. To this end *pasting small votes* techniques, by which individual classifiers are trained on relatively small subsets of the available data have been proposed [23]. By this method, training sets are sampled from a large data set either randomly (*Rvotes*, similar to bagging), or taking into account their importance for the classification task (*Ivotes*, similar to boosting). Breiman [23] and Chawla, et al. [36] showed that importance small sampling-based ensembles such as *Ivotes* and their distribute counterpart *DI-votes* may obtain also better results with respect to single learners trained on the entire available learning set. In particular Chawla et al. showed that this ensemble approach may improve accuracy by enhancing diversity between base learners, even if stable classifiers, such as Naive-Bayes, are used [37]. Through a comparative experimental analysis, the reasons why voting many classifiers built on small subsets of data work successfully are interpreted in the context of bias-variance analysis of the generalization error: the success of this approach is due to the very significant variance reduction, while bias remains substantially unchanged [197].

1.6 Ensemble methods in astronomy and astrophysics

The exponential raise in the available amount of astronomical and astrophysical data observed in the last years resulted into an ever increasing gap between the availability of information about the natural objects (and/or phenomena) under investigation and our ability to extract useful knowledge from them. Astronomy has been among the first scientific disciplines to experience this flood of data. The sheer volume of data routinely analyzed in astronomical and astrophysical research projects needs the application of a multidisciplinary approach often involving the concurrent use of data mining, statistical and machine learning techniques in order to effectively tackle the high levels of noise usually present in the data produced by large scale astronomical projects.

The primary goal of this section dedicated to the application of ensemble methods in astronomy and astrophysics is not to present an exhaustive list of all the scientific papers recently published, but rather to put in light the potential benefits introduced by the application of the ensemble learning approach to common problems investigated in these rapidly growing research areas.

In a very broad sense the aim of ensemble learning, as in the case of all the other branches of *machine learning*, is to leverage a computational machine able to extract patterns from the data and then to translate these patterns into novel (and hopefully useful) knowledge. Indeed a common problem faced by scientific investigators is not only to classify objects according to a pre-existing classification scheme but also to highlight the eventual existence of relationships between uncategorized (or unlabeled) and more characterized objects. From this point of view the choice of the ensemble method to be applied to the problem at hand is of paramount importance. We finally would like to stress that the described ensemble methods are not out-of-the-box solutions but rather *tools* that, if applied correctly, have strong potential for the production of interesting scientific results and are able to provide inspiration for new ideas and applications.

Supervised ensemble methods have been applied to several astronomical problems. They have proven to be effective in the automated annotation of the content of large publicly available catalogs. A good example of this class of problems is the automated morphological classification of galaxies.

In [13] the authors predicted the morphological class of 800 examples with both a single classifier and an ensemble of classifiers trained on bootstrap replicates of the training set (bootstrap aggregating or bagging, see Sect. 1.5.1). Performance were collected in the form of averaged classification errors obtained by means of a canonical 10-fold classification scheme in order to provide a good estimate of the generalization capabilities of the evaluated approaches. As component classifiers the authors evaluated Artificial Neural

Networks (commonly applied in astronomical classification problems) trained with backpropagation, a pruned decision tree and the naive Bayes classifier.

The collected experimental results clearly demonstrated the ability of the ensemble systems to reduce the overall classification error but with different patterns. The authors observed that the decrement in the classification error due to the application of the ensemble method was different according to the type of the component classifier and also that the error reduction effect was inversely related to the number of the output classes. Apart of the expected classification error reduction, a really crucial point emerging from the results presented in this work is that the same ensemble scheme may produce different results according to the nature of its component classifiers. Unfortunately, there is no way to know a priori which is the best type of classifier to be used to solve a particular problem being this strictly dependent not only on the nature of the problem at hand but also on the dataset to be evaluated. The only way to face this problem is to perform experiments involving ensemble methods based on different types of base learners. With respect to this problem an interesting feature of the ensemble methods is that their flexible nature allows the formation of the classifiers committee not only using instances of the same algorithm, but also using committees composed by instances of different algorithms trained on the same datasets resulting into a sort of mixture of experts able to exploit the strength of different classifiers.

The combination of bagging and random selection of a small subset of features for splitting at each node is known as a Random Forest (see Sect. 1.5.5). Random Forests have proven to be effective in the identification of quasars from the FIRST survey [65]. The Random Forest ensemble method has also been applied in multi wavelength classification problems of data collected from different databases including the Sloan Digital Sky Survey (SDSS), USNO, FIRST and ROSAT [75]. In this experiment the authors not only demonstrated that the investigated ensemble method is effective in astronomical objects classification, but also investigated the feature selection and feature weighting capabilities of this method when applied to data constituted by samples from optical and radio bands. The authors also noted the robustness of Random Forest in outlier detection.

In [79] the authors applied machine-learning methods to the automatic geomorphic mapping of planetary surfaces. In this experiment the authors casted remotely sensed topographic data into semantically meaningful maps of landforms producing maps that are valuable research tools for planetary science. The proposed framework is composed by two distinct steps: mapping of the available topographic data (achieved by means of scene segmentation) followed by supervised classification of segments. The method was applied to six test sites on Mars. The collected experimental results showed that a combination of K -means-based agglomerative segmentation and both SVM with a quadratic kernel and Bagging with C4.5 produce the best maps. This work demonstrates that the bagged ensemble of decision trees perform comparably with the Support Vector Machine in the classification step, and this raise a

crucial question: why should an investigator choose a classification scheme instead of another? As we discussed previously, there is not an easy way to answer this question (being it strictly related to the intrinsic nature of the investigated problem), but while the application of an ensemble method does not ensure an increment in the classification performance, it is granted to improve the generalization level, meaning that an ensemble system is expected to be more robust w.r.t. previously unseen data. and this can make the difference in research fields in which the amount of the publicly available data and the acquisition rate of novel data is constantly growing (as in the case of astronomy and astrophysics).

Supervised methods can be applied only in presence of a priori knowledge available in the form of a set of labels, but relevant problems is astronomy and astrophysics need an unsupervised approach.

A common problem faced by astronomers is the classification of celestial objects on the basis of their spectral emissions. A large volume of noisy, multidimensional data has been recently produced by using CCD imaging spectrometers and made available to the scientific community by large scale astronomical projects (data collected by the Chandra X-Ray Observatory and the X-Ray Multimirror Mission (XMM-Newton)). The sheer volume of these datasets raised the need to develop methods to classify and characterize the vast library of X-ray spectra in an unsupervised fashion. This is of paramount importance in order to create a counterpart to the current parametric model fits, usually employed to classify X-ray spectral data, and to provide the investigators with a family of tools able to produce an opinion originated by a different classification paradigm (the unsupervised one) w.r.t. the one upon which are based classical spectral classification models.

In a recent work [92] the authors applied an ensemble classifier consisting of agglomerative hierarchical clustering and K-means clustering applied to X-ray spectral classification. This method does not need spectral source models and can operate without information about the sources. It is also able to deal with massive amount of data, a feature making it attractive for the analysis of the data produced by large scale astrophysical investigations. This approach employs Principal Component Analysis for dimensional reduction of the spectral bands followed by clustering. While PCA provides a means to automatically define optimal spectral band definitions from the data set itself, the ensemble clustering method groups similar sources of X-ray emission by placing them in a three-dimensional spectral sequence and then grouping the ordered sources into clusters based on their spectra. The statistical issues behind this method are discussed in [93].

The ensemble learning paradigm has also been applied in a popular astronomical research area: galaxy spectra modeling. In [131] a statistical approach, ensemble learning for independent component analysis (EL-ICA), was applied to the analysis of a synthetic galaxy spectral library. The authors found that the proposed method was able to reduce the data of the spectral library to six nonnegative independent components (ICs). They also found that the iden-

tified ICs were good templates for modeling normal galaxy spectra, as the ones contained in the Sloan Digital Sky Survey (SDSS). In this case ensemble learning was applied, according to what originally suggested in [126], in order to provide a sort of heuristic able to efficiently estimate the nonlinear parameters of an ICA model. The approach proposed in [131] is not only able to solve the problem of data compression characterizing the first step of a spectra modeling task, but is also expected to manage effectively the risk of overfitting the training data.

As a last example of application of ensemble methods in astronomy/astrophysics, we would like to highlight the potential benefits introduced by the usage of ensemble systems in classification tasks involving missing data. Any classification algorithm independently by its supervised or unsupervised nature, assumes that each instance is associated to a complete set of values but real observational data often contain missing features. A common approach to tackle the problems due to the presence of incomplete instances is to simply remove them from the dataset under investigation before to start its analysis. This approach is suboptimal when a large fraction of the data points have missing features. A common technique to avoid this "filtering" approach is to impute the missing value, but often imputation techniques are based on the assumption that missing values occur by chance which is not always the case. In particular, in astronomy the absence of a value could have a physical meaning. In order to avoid both the prefiltering of incompletely described instances and the imputation of missing values, a solution based on a adapted clustering approach was proposed in [206]. The proposed method (KSC) is based on soft constraints induced by the fully described instances to assist in the grouping of the incomplete ones. This approach is suitable only for exploratory (unsupervised) investigations but cannot be applied to supervised classification problems. In [159] the author proposed a solution based on the creation of an ensemble of classifiers, each trained with a random subset of the features, so that an instance with missing features can still be classified using learners whose training data did not include those attributes. The main parameters affecting the performance of the classifier are the number of random features used in the training of the component classifiers, and the total number of component learners to be generated in order to create the ensemble committee.

1.7 Conclusions

In this chapter we provided an overview and a bibliography of ensemble methods. Despite our efforts, we are sure that we missed important research works and maybe important research areas, since this field of machine learning is continuously growing, and new methods and innovative applications able to

stimulate the development of new ensemble methods are object of intensive research. To expand on this subject, we would like to mention an excellent book devoted to ensemble methods, the fundamental *Combining Pattern Classifiers* by Lucy Kuncheva [116], as well as the cited proceedings of the Multiple Classifier Systems (MCS) conference, especially for the discussion of recent advanced topics on ensemble systems [14, 62].

Our overview focused on supervised ensemble methods, since historically these were the first to be studied and applied to several application domains. More precisely, in this chapter a general taxonomy, distinguishing between *generative* and *non–generative* ensemble methods, has been proposed, considering the different ways supervised base learners can be generated or combined together. Nevertheless ensemble methods have been also developed in the context of semi-supervised and unsupervised ensemble methods, as witnessed by recent research works on the unsupervised exploratory analysis of data [77, 143, 144].

Several important issues have not been discussed in this paper. In particular the theoretical problems behind ensemble methods need to be reviewed and discussed more in detail, and the discussion of the application of ensemble methods to real-world problems has been limited to some relevant problems in astronomy and astrophysics, since a discussion extended to all the application domains of ensemble systems if far beyond the scope of this chapter. To gain a general overview of the applications of ensemble methods, a good starting point could be the quite recent Special Issue on Applications of Ensemble Methods of the Information Fusion Journal [1].

Other open problems not covered in this chapter are the relationships between ensemble methods and data complexity [88, 89, 129], a systematic research of hidden commonalities among all the combination approaches despite their superficial differences, and a general analysis of the applications (and of the applicability) of these methods to supervised tasks such as active learning [181], or to semi-supervised learning [215].

Acknowledgments

We would like to thank the anonymous reviewers for their comments and suggestions. The authors gratefully acknowledge partial support by the PASCAL2 Network of Excellence under EC grant no. 216886. This publication only reflects the authors' views.

Bibliography

[1] AA.VV. *Special Issue on Applications of Ensemble Methods*, volume 9 (1) of *Information Fusion Journal*. Elsevier, 2008.

[2] D. Aha and R. Bankert. Cloud classification using error-correcting output codes. In *Artificial Intelligence Applications: Natural Science, Agriculture and Environmental Science*, volume 11, pages 13–28. 1997.

[3] N. Alaydie, C.K. Reddy, and F. Fotouhi. Hierarchical multi-label boosting for gene function prediction. In *Proceedings of the International Conference on Computational Systems Bioinformatics (CSB)*, Stanford, CA, 2010.

[4] E.L. Allwein, R.E. Schapire, and Y. Singer. Reducing multiclass to binary: a unifying approach for margin classifiers. *Journal of Machine Learning Research*, 1:113–141, 2000.

[5] E.L. Allwein, R.E. Schapire, and Y. Singer. Reducing multiclass to binary: a unifying approach for margin classifiers. In *Proc. ICML'2000, The Seventeenth International Conference on Machine Learning*, 2000.

[6] E. Alpaydin and C. Kaynak. Cascading classifiers. *Kybernetika*, 34(4):369–374, 1998.

[7] E. Alpaydin and E. Mayoraz. Learning error-correcting output codes from data. In *ICANN'99*, pages 743–748, Edinburgh, UK, 1999.

[8] R. Anand, G. Mehrotra, C.K. Mohan, and S. Ranka. Efficient classification for multiclass problems using modular neural networks. *IEEE Transactions on Neural Networks*, 6:117–124, 1995.

[9] R.E. Banfield, L.O. Hall, K.W. Bowyer, and P. Kegelmeyer. Ensemble diversity measure and their application to thinning. *Information Fusion*, 6(1):49–62, 2005.

[10] R.E. Banfield, L.O. Hall, O. Lawrence, K.W. Bowyer, W. Kevin, and P. Kegelmeyer. A comparison of decision tree ensemble creation techniques. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 29(1):173–180, 2007.

[11] E. Bauer and R. Kohavi. An empirical comparison of voting classification algorithms: Bagging, Boosting and variants. *Machine Learning*, 36(1/2):105–139, 1999.

[12] M.A. Bautista, X. Baro, O. Pujol, P. Radeva, J. Vitria, and S Escalera. Compact evolutive design of error-correcting output codes. In O. Okun, M. Re, and G. Valentini, editors, *ECML-SUEMA 2010 Proceedings*, Barcelona, Spain, 2010.

0-8493-0052-5/00/$0.00+$.50

[13] D. Bazell and WD. Aha. Ensemble of classifiers for morphological galaxy classification. *The Astrophysical Journal*, 548:219–223, 2001.

[14] J. Benediktsson, F. Roli, and Kittler. *Multiple Classifier Systems, 8th International Workshop, MCS2009*, volume 5519 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, 2009.

[15] J. Benediktsson, J. Sveinsson, O. Ersoy, and P. Swain. Parallel consensual neural networks. *IEEE Transactions on Neural Networks*, 8:54–65, 1997.

[16] J. Benediktsson and P. Swain. Consensus theoretic classification methods. *IEEE Transactions on Systems, Man and Cybernetics*, 22:688–704, 1992.

[17] J.A. Benediktsson, J. Chanussot, and M. Fauvel. Multiple classifier systems in remote sensing: From basics to recent developments. In M. Haindl, J. Kittler, and F. Roli, editors, *Multiple Classifier Systems. Seventh International Workshop, MCS 2007, Prague, Czech Republic*, volume 4472 of *Lecture Notes in Computer Science*, pages 511–512. Springer, 2007.

[18] A. Berger. Error correcting output coding for text classification. In *IJCAI'99: Workshop on machine learning for information filtering*, 1999.

[19] C. M. Bishop. *Neural Networks for Pattern Recognition*. Clarendon Press, Oxford, 1995.

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

[21] L. Breiman. Bias, variance and arcing classifiers. Technical Report TR 460, Statistics Department, University of California, Berkeley, CA, 1996.

[22] L. Breiman. Arcing classifiers. *The Annals of Statistics*, 26(3):801–849, 1998.

[23] L. Breiman. Pasting Small Votes for Classification in Large Databases and On-Line. *Machine Learning*, 36:85–103, 1999.

[24] L. Breiman. Prediction games and arcing classifiers. *Neural Computation*, 11(7):1493–1517, 1999.

[25] L. Breiman. Random Forests. *Machine Learning*, 45(1):5–32, 2001.

[26] M. Breukelen van, R.P.W. Duin, D. Tax, and J.E. Hartog den. Combining classifiers for the recognition of handwritten digits. In *Ist IAPR TC1 Workshop on Statistical Techniques in Pattern Recognition*, pages 13–18, Prague, Czech republic, 1997.

[27] G.J. Briem, J.A. Benediktsson, and J.R. Sveinsson. Boosting. Bagging and Consensus Based Classification of Multisource Remote Sensing Data. In J. Kittler and F. Roli, editors, *Multiple Classifier Systems. Second International Workshop, MCS 2001, Cambridge, UK*, volume 2096 of *Lecture Notes in Computer Science*, pages 279–288. Springer-Verlag, 2001.

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

[29] R. Bryll, R. Gutierrez-Osuna, and F. Quek. Attribute bagging: Improving accuracy od classifier ensembles by uing random feature subsets. *Pattern Recognition*, 36:1291–1302, 2003.

[30] R. Caruana, A. Niculescu-Mizil, G. Crew, and A. Ksikes. Ensemble selction from libraries of models. In *21th Int. Conf. on Machine Learning, ICML 2004*, page 18. ACM Press, 2004.

[31] N. Cesa-Bianchi, C. Gentile, and L. Zaniboni. Hierarchical classification: Combining Bayes with SVM. In *Proc. of the 23rd Int. Conf. on Machine Learning*, pages 177–184. ACM Press, 2006.

[32] N. Cesa-Bianchi, M. Re, and G. Valentini. Functional inference in FunCat through the combination of hierarchical ensembles with data fusion methods. In *ICML-MLD 2nd International Workshop on learning from Multi-Label Data*, pages 13–20, Haifa, Israel, 2010.

[33] N. Cesa-Bianchi and G. Valentini. Hierarchical cost-sensitive algorithms for genome-wide gene function prediction. *Journal of Machine Learning Research, W&C Proceedings, Machine Learning in Systems Biology*, 8:14–29, 2010.

[34] P. Chan and S. Stolfo. A comparative evaluation of voting and meta-learning on partitioned data. In *Proc.* 12^{th} *ICML*, pages 90–98, 1995.

[35] P. Chan and S. Stolfo. On the accuracy of meta-learning for scalable data mining. *Journal of Intelligent Information Systems*, 8:5–28, 1997.

[36] N.V. Chawla, L.O. Hall, K.W. Bowyer, and W.P. Kegelmeyer. Learning Ensembles from Bites: A Scalable and Accurate Approach. *Journal of Machine Learning Research*, 5:421–451, 2004.

[37] N.V. Chawla, L.O. Hall, K.W. Bowyer, T.E. Moore, and W.P. Kegelmeyer. Distributed pasting of small votes. In *Multiple Classifier Systems. Third International Workshop, MCS2002, Cagliari, Italy*, volume 2364 of *Lecture Notes in Computer Science*, pages 52–61. Springer-Verlag, 2002.

[38] D.. Chen. *Statistical estimates for Kleinberg's method of Stochastic Discrimination*. PhD thesis, The State University of New York, Buffalo, USA, 1998.

[39] S. Cho and J. Kim. Combining multiple neural networks by fuzzy integral and robust classification. *IEEE Transactions on Systems, Man and Cybernetics*, 25:380–384, 1995.

[40] S. Cho and J. Kim. Multiple network fusion using fuzzy logic. *IEEE Transactions on Neural Networks*, 6:497–501, 1995.

[41] S. Cohen and N. Intrator. A Hybrid Projection Based and Radial Basis Function Architecture. In J. Kittler and F. Roli, editors, *Multiple Classifier Systems. First International Workshop, MCS 2000, Cagliari, Italy*, volume 1857 of *Lecture Notes in Computer Science*, pages 147–156. Springer-Verlag, 2000.

[42] S. Cohen and N. Intrator. Automatic Model Selection in a Hybrid Perceptron/Radial Network. In *Multiple Classifier Systems. Second International Workshop, MCS 2001, Cambridge, UK*, volume 2096 of *Lecture Notes in Computer Science*, pages 349–358. Springer-Verlag, 2001.

[43] M. Collins, R.E. Schapire, and Y. Singer. Logistic regression, AdaBoost and Bregman distances. *Machine Learning*, 48:31–44, 2002.

[44] K. Crammer and Y. Singer. On the learnability and design of output codes for multiclass problems. In *Proceedings of the Thirteenth Annual Conference on Computational Learning Theory*, pages 35–46, 2000.

[45] N.C. de Condorcet. *Essai sur l' application de l' analyse à la probabilité des decisions rendues à la pluralité des voix*. Imprimerie Royale, Paris, 1785.

[46] O. Dekel, J. Keshet, and Y. Singer. Large margin hierarchical classification. In *Proc. of the 21st Int. Conf. on Machine Learning*, pages 209–216. Omnipress, 2004.

[47] C. Dietrich, G. Palm, and F. Schwenker. Decision templates for the classification of bioacoustic time series. *Information Fusion*, 4(2):101–109, 2003.

[48] T.G. Dietterich. Ensemble methods in machine learning. In J. Kittler and F. Roli, editors, *Multiple Classifier Systems. First International Workshop, MCS 2000, Cagliari, Italy*, volume 1857 of *Lecture Notes in Computer Science*, pages 1–15. Springer-Verlag, 2000.

[49] T.G. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting and randomization. *Machine Learning*, 40(2):139–158, 2000.

[50] T.G. Dietterich and G. Bakiri. Error - correcting output codes: A general method for improving multiclass inductive learning programs. In *Proceedings of AAAI-91*, pages 572–577. AAAI Press / MIT Press, 1991.

[51] T.G. Dietterich and G. Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, (2):263–286, 1995.

[52] P. Domingos. A Unified Bias-Variance Decomposition and its Applications. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 231–238, Stanford, CA, 2000. Morgan Kaufmann.

[53] P. Domingos. A Unified Bias-Variance Decomposition for Zero-One and Squared Loss. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence*, pages 564–569, Austin, TX, 2000. AAAI Press.

[54] P. Domingos and M. Pazzani. On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning*, 29:103–130, 1997.

[55] H. Drucker and C. Cortes. Boosting decision trees. In *Advances in Neural Information Processing Systems*, volume 8. 1996.

[56] H. Drucker, C. Cortes, L. Jackel, Y. LeCun, and V. Vapnik. Boosting and other ensemble methods. *Neural Computation*, 6(6):1289–1301, 1994.

[57] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern classification, 2nd ed.* Wiley & Sons, New York, 2001.

[58] R. Duin. The combining classifier: to train or not to train? In *Proc. of the 16th Int. Conf. on Pattern Recognition, ICPR'02*, pages 765–770, Canada, 2002.

[59] R.P.W. Duin and D.M.J. Tax. Experiments with Classifier Combination Rules. In J. Kittler and F. Roli, editors, *Multiple Classifier Systems. First International Workshop, MCS 2000, Cagliari, Italy*, volume 1857 of *Lecture Notes in Computer Science*, pages 16–29. Springer-Verlag, 2000.

[60] S. Dzeroski and B. Zenko. Is combining classifiers with stacking better than selcting the best one? *Machine Learning*, 54(3):255–273, 2004.

[61] B. Efron and R. Tibshirani. *An introduction to the Bootstrap*. Chapman and Hall, New York, 1993.

[62] N. El Gayar, F. Roli, and Kittler. *Multiple Classifier Systems, 9th International Workshop, MCS2010*, volume 5997 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, 2010.

[63] S. Escalera, O. Pujol, and P. Radeva. On the decoding process in ternary error-correcting output codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(1):120–134, 2010.

[64] T. Evgeniou, M. Pontil, and A. Elisseeff. Leave one out error, stability, and generalization of voting combinations of classifiers. *Machine Learning*, 55(1):71–97, 2004.

[65] Eric D. Feigelson, G. Jogesh Babu, Leo Breiman, Michael Last, and John Rice. Random forests: Finding quasars. In *Statistical Challenges in Astronomy*, pages 243–254. Springer New York, 2003.

[66] E. Filippi, M. Costa, and E. Pasero. Multi-layer perceptron ensembles for increased performance and fault-tolerance in pattern recognition tasks. In *IEEE International Conference on Neural Networks*, pages 2901–2906, Orlando, Florida, 1994.

[67] Y. Freund. Boosting a weak learning algorithm by majority. *Information and Computation*, 121(2):256–285, 1995.

[68] Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and Systems Sciences*, 55(1):119–139, 1997.

[69] Y. Freund and R.E. Schapire. Experiments with a new boosting algorithm. In *Proceedings of the 13th International Conference on Machine Learning*, pages 148–156. Morgan Kauffman, 1996.

[70] J. Friedman and P. Hall. On Bagging and Nonlinear Estimation. Technical Report Tech. Report, Statistics Department, University of Stanford, CA, 2000.

[71] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: A statistical view of boosting. *The Annals of Statistics*, 38(2):337–374, 2000.

[72] J.H. Friedman. On bias, variance, 0/1 loss and the curse of dimensionality. *Data Mining and Knowledge Discovery*, 1:55–77, 1997.

[73] G. Fumera and F. Roli. A Theoretical and Experimental Analysis of Linear Combiners for Multiple Classifer Systems. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(6):942–956, 2005.

[74] J. Gamma and P. Brazdil. Cascade generalization. *Machine Learning*, 41(3):315–343, 2000.

[75] Zhang Y. Gao D. and Zhao Y. Random forest algorithm for classification of multiwavelength data. *Research in Astron. Astrophysics*, 9(2):220–226, 2009.

[76] S. Geman, E. Bienenstock, and R. Doursat. Neural networks and the bias-variance dilemma. *Neural Computation*, 4(1):1–58, 1992.

[77] R. Ghaemi, M. Sulaiman, H. Ibrahim, and N. Mustapha. A survey: clustering ensembles techniques. In *World Academy of Science, Engineering and Technology 50*, pages 636–645, 2009.

[78] R. Ghani. Using error correcting output codes for text classification. In *ICML 2000: Proceedings of the 17th International Conference on Machine Learning*, pages 303–310, San Francisco, US, 2000. Morgan Kaufmann Publishers.

[79] S. Ghosh, T.F. Stepinski, and R. Vilalta. Automatic annotation of planetary surfaces with geomorphic labels. *Geoscience and Remote Sensing, IEEE Transactions on*, 48(1):175–185, 2010.

[80] G. Giacinto and F. Roli. Dynamic Classifier Fusion. In J. Kittler and F. Roli, editors, *Multiple Classifier Systems. First International Workshop, MCS 2000, Cagliari, Italy*, volume 1857 of *Lecture Notes in Computer Science*, pages 177–189. Springer-Verlag, 2000.

[81] G. Giacinto and F. Roli. An approach to the automatic design of multiple classifier systems. *Pattern Recognition Letters*, 22(1):25–33, 2001.

[82] G. Giacinto, F. Roli, and G. Fumera. Design of effective multiple classifier systems by clustering of classifiers. In *15th International Conference on Pattern Recognition ICPR 2000*, pages 160–163, 2000.

[83] Y Guan, C.L. Myers, D.C. Hess, Z. Barutcuoglu, A. Caudy, and O.G. Troyanskaya. Predicting gene function in a hierarchical context with an ensemble of classifiers. *Genome Biology*, 9(S2), 2008.

[84] S. Gunter and H. Bunke. Feature selection algorithms for the generation of multiple classifier systems and their application to handwritten word recognition. *Pattern Recognition Letters*, 25:1323–1336, 2004.

[85] M. Haindl, F. Roli, and Kittler. *Multiple Classifier Systems, 7th International Workshop, MCS2007*, volume 4472 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, 2007.

[86] T. Hastie and R. Tibshirani. Classification by pairwise coupling. *The Annals of Statistics*, 26(1):451–471, 1998.

[87] T.K. Ho. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):832–844, 1998.

[88] T.K. Ho. Complexity of Classification Problems ans Comparative Advantages of Combined Classifiers. In J. Kittler and F. Roli, editors, *Multiple Classifier Systems. First International Workshop, MCS 2000, Cagliari, Italy*, volume 1857 of *Lecture Notes in Computer Science*, pages 97–106. Springer-Verlag, 2000.

[89] T.K. Ho. Data Complexity Analysis for Classifiers Combination. In J. Kittler and F. Roli, editors, *Multiple Classifier Systems. Second International Workshop, MCS 2001, Cambridge, UK*, volume 2096 of *Lecture Notes in Computer Science*, pages 53–67, Berlin, 2001. Springer-Verlag.

[90] T.K. Ho. Multiple classifier combinations: Lessons and the next steps. In A. Kandel and K. Bunke, editors, *Hybrid Methods in Pattern Recognition*, pages 171–198. World Scietific, 2002.

[91] T.K. Ho, J.J. Hull, and S.N. Srihari. Decision combination in multiple classifiers. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(4):405–410, 1997.

[92] S.M. Hojnacki et al. An x-ray spectral classification algorithm with application to young stellar clusters. *The Astrophysical Journal*, 659, 2007.

[93] S.M. Hojnacki et al. An unsupervised, ensemble clustering algorithm: A new approach for classification of x-ray sources. *Statistical Methodology*, 5:350–360, 2008.

[94] K. Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4:251–257, 1991.

[95] T. Hothorn and B. Lausen. Bundling classifers by bagging trees. *Computational Statistics and Data Analysis*, 49:1068–1078, 2005.

[96] Y.S. Huang and Suen. C.Y. Combination of multiple experts for the recognition of unconstrained handwritten numerals. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17:90–94, 1995.

[97] R.A. Jacobs. Methods for combining experts probability assessment. *Neural Computation*, 7:867–888, 1995.

[98] R.A. Jacobs, M.I. Jordan, S.J. Nowlan, and G.E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):125–130, 1991.

[99] A. Jain, R. Duin, and J. Mao. Statistical pattern recognition: a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:4–37, 2000.

[100] X. Jiang, N. Nariai, M. Steffen, S. Kasif, and E. Kolaczyk. Integration of relational and hierarchical network information for protein function prediction. *BMC Bioinformatics*, 9(350), 2008.

[101] M. Jordan and R. Jacobs. Hierarchies of adaptive experts. In *Advances in Neural Information Processing Systems*, volume 4, pages 985–992. Morgan Kauffman, San Mateo, CA, 1992.

[102] M.I. Jordan and R.A. Jacobs. Hierarchical mixture of experts and the em algorithm. *Neural Computation*, 6:181–214, 1994.

[103] M.I. Jordan and L. Xu. Convergence results for the EM approach to mixture of experts architectures. *Neural Networks*, 8:1409–1431, 1995.

[104] Tumer. K. and C.N. Oza. Input decimated ensembles. *Pattern Analysis and Applications*, 6:65–77, 2003.

[105] J.M. Keller, P. Gader, H. Tahani, J. Chiang, and M. Mohamed. Advances in fuzzy integration for pattern recognition. *Fuzzy Sets and Systems*, 65:273–283, 1994.

[106] F. Kimura and M. Shridar. Handwritten Numerical Recognition Based on Multiple Algorithms. *Pattern Recognition*, 24(10):969–983, 1991.

[107] J. Kittler. Combining classifiers: a theoretical framework. *Pattern Analysis and Applications*, (1):18–27, 1998.

[108] J. Kittler, M. Hatef, R.P.W. Duin, and J. Matas. On combining classifiers. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(3):226–239, 1998.

[109] E.M. Kleinberg. Stochastic Discrimination. *Annals of Mathematics and Artificial Intelligence*, pages 207–239, 1990.

[110] E.M. Kleinberg. An overtraining-resistant stochastic modeling method for pattern recognition. *Annals of Statistics*, 4(6):2319–2349, 1996.

[111] E.M. Kleinberg. A Mathematically Rigorous Foundation for Supervised Learning. In J. Kittler and F. Roli, editors, *Multiple Classifier Systems. First International Workshop, MCS 2000, Cagliari, Italy*, volume 1857 of *Lecture Notes in Computer Science*, pages 67–76. Springer-Verlag, 2000.

[112] E.M. Kleinberg. On the Algorithmic Implementation of Stochastic Discrimination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(5):473–490, 2000.

[113] E. Kong and T.G. Dietterich. Error - correcting output coding correct bias and variance. In *The XII International Conference on Machine Learning*, pages 313–321, San Francisco, CA, 1995. Morgan Kauffman.

[114] L.I. Kuncheva. An application of OWA operators to the aggregation of multiple classification decisions. In *The Ordered Weighted Averaging operators. Theory and Applications*, pages 330–343. Kluwer Academic Publisher, USA, 1997.

[115] L.I. Kuncheva. Switching between selection and fusion in combining classifiers: An experiment. *IEEE Trans. on Systems, Man and Cybernetics*, 32(2):146–156, 2002.

[116] L.I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience, New York, 2004.

[117] L.I. Kuncheva, J.C. Bezdek, and R.P.W. Duin. Decision templates for multiple classifier fusion: an experimental comparison. *Pattern Recognition*, 34(2):299–314, 2001.

[118] L.I. Kuncheva and L.C. Jain. Designing classifier fusion systems by genetic algorithms. *IEEE Trnsactions on Evolutionary Computation*, 4(4):327–336, 2000.

[119] L.I. Kuncheva and J. Rodriguez. An experimental study on rotation forest ensembles. In M. Haindl, J. Kittler, and F. Roli, editors, *Multiple Classifier Systems. Seventh International Workshop, MCS 2007, Prague, Czech Republic*, volume 4472 of *Lecture Notes in Computer Science*, pages 459–468. Springer, 2007.

[120] L.I. Kuncheva, F. Roli, G.L. Marcialis, and C.A. Shipp. Complexity of Data Subsets Generated by the Random Subspace Method: An Experimental Investigation. In J. Kittler and F. Roli, editors, *Multiple Classifier Systems. Second International Workshop, MCS 2001, Cambridge, UK*, volume 2096 of *Lecture Notes in Computer Science*, pages 349–358. Springer-Verlag, 2001.

[121] L.I. Kuncheva and C.J. Whitaker. Measures of diversity in classifier ensembles. *Machine Learning*, 51:181–207, 2003.

[122] L. Lam. Classifier combinations: Implementations and theoretical issues. In *Multiple Classifier Systems. First International Workshop, MCS 2000, Cagliari, Italy*, volume 1857 of *Lecture Notes in Computer Science*, pages 77–86. Springer-Verlag, 2000.

[123] L. Lam and C. Sue. Optimal combination of pattern classifiers. *Pattern Recognition Letters*, 16:945–954, 1995.

[124] L. Lam and C. Sue. Application of majority voting to pattern recognition: an analysis of its behavior and performance. *IEEE Transactions on Systems, Man and Cybernetics*, 27(5):553–568, 1997.

[125] W.B. Langdon and B.F. Buxton. Genetic programming for improved receiver operating characteristics. In J. Kittler and F. Roli, editors, *Second International Conference on Multiple Classifier System*, volume 2096 of *LNCS*, pages 68–77, Cambridge, 2001. Springer Verlag.

[126] H. Lapplainen. Nonlinear independent component analysis using ensemble learning: Theory. In *Proc. 1st Int. Workshop on Independent Component Analysis and Blind Signal Separation*, page 7, 1998.

[127] A. Lazarevic and Z. Obradovic. Effective pruning of neural network classifiers. In *Proceedings of the IEEE International Joint Conference on Neural Networks IJCNN'01*, pages 796–801. IEEE, 2001.

[128] W. Leigh, R. Purvis, and J.M. Ragusa. Forecasting the NYSE composite index with technical analysis, pattern recognizer, neural networks and genetic algorithm: A case study. *Decision Support Systems*, 32(4):361–377, 2002.

[129] M. Li and P Vitanyi. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer-Verlag, Berlin, 1993.

[130] R. Liu and B. Yuan. Multiple classifier combination by clustering and selection. *Information Fusion*, 2:163–168, 2001.

[131] H. et al. Lu. Ensemble learning for independent component analysis of normal galaxy spectra. *The Astronomical Journal*, 131:790–805, 2006.

[132] G. Martinez-Muniz and A. Suarez. Pruning in ordered bagging ensembles. In *23th Int. Conf. on Machine Learning, ICML 2006*, pages 609–616. ACM Press, 2006.

[133] L. Mason, P. Bartlett, and J. Baxter. Improved generalization through explicit optimization of margins. *Machine Learning*, 2000.

[134] F. Masulli and G. Valentini. Effectiveness of error correcting output codes in multiclass learning problems. In *Lecture Notes in Computer Science*, volume 1857, pages 107–116. Springer-Verlag, Berlin, Heidelberg, 2000.

[135] F. Masulli and G. Valentini. Quantitative Evaluation of Dependence among Outputs in ECOC Classifiers Using Mutual Information Based Measures. In K. Marko and P. Webos, editors, *Proceedings of the International Joint Conference on Neural Networks IJCNN'01*, volume 2, pages 784–789, Piscataway, NJ, USA, 2001. IEEE.

[136] F. Masulli and G. Valentini. Effectiveness of Error Correcting Output Coding decomposition schemes in ensemble and monolithic learning machines. *Pattern Analysis and Application*, 6:285–300, 2003.

[137] F. Masulli and G. Valentini. An experimental analysis of the dependence among codeword bit errors in ecoc learning machines. *Neurocomputing*, 57:189–214, 2004.

[138] E. Mayoraz and M. Moreira. On the decomposition of polychotomies into dichotomies. In *The XIV International Conference on Machine Learning*, pages 219–226, Nashville, TN, July 1997.

[139] E. Menahem, A Shabtai, L. Rokach, Y. Elovici, and A Troiha. Improving malware detection by applying multi-iducer ensemble. *Computational Statistics and Data Analysis*, 53(4):1483–1494, 2009.

[140] C. Merkwirth et al. Ensemble methods for classification in cheminformatics. *Journal of Chemical Information and Modeling*, 44(6):1971–1978, 2009.

[141] M. Moreira and E. Mayoraz. Improved pairwise coupling classifiers with correcting classifiers. In C. Nedellec and C. Rouveirol, editors, *Lecture Notes in Artificial Intelligence, Vol. 1398*, pages 160–171, Berlin, Heidelberg, New York, 1998.

[142] G. Obozinski, G. Lanckriet, C. Grant, Jordan. M., and W.S. Noble. Consistent probabilistic output for protein function prediction. *Genome Biology*, 9(S6), 2008.

[143] O. Okun and G. (eds.) Valentini. *Supervised and Unsupervised Ensemble Methods and their Applications*, volume 126 of *Studies in Computational Intelligence*. Springer-Verlag, Berlin, 2008.

[144] O. Okun and G. (eds.) Valentini. *Applications of Supervised and Unsupervised Ensemble Methods*, volume 245 of *Studies in Computational Intelligence*. Springer-Verlag, Berlin, 2009.

[145] D.W Opitz. Feature selection for ensembles. In *Proc. of the 16th National Conference on Artificla Intelligence, AAAI*, pages 379–384, 1999.

[146] D.W. Opitz and J.W. Shavlik. Actively searching for an effective neural network ensemble. *Connection Science*, 8(3/4):337–353, 1996.

[147] D.W. Opitz and J.W. Shavlik. Generating accurate and diverse members of a neural-network ensemble. In D. Touretzky, M. Mozer, and M. Hasselmo, editors, *Advances in Neural Information Processing Systems*, volume 8, pages 535–541, Cambridge, MA, 1996. MIT Press.

[148] N.C. Oza. Aveboost2: Boosting for noisy data. In *Multiple Classifier Systems. Fifth International Workshop, MCS 2004, Cagliari, Italy*, volume 3077 of *Lecture Notes in Computer Science*, pages 31–40. Springer-Verlag, 2004.

[149] N.C. Oza, R. Polikar, F. Roli, and Kittler. *Multiple Classifier Systems, 6th International Workshop, MCS2005*, volume 3541 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, 2005.

[150] N.C. Oza and K. Tumer. Input Decimation Ensembles: Decorrelation through Dimensionality Reduction. In J. Kittler and F. Roli, editors, *Multiple Classifier Systems. Second International Workshop, MCS 2001, Cambridge, UK*, volume 2096 of *Lecture Notes in Computer Science*, pages 238–247. Springer-Verlag, 2001.

[151] J. Park and I.W. Sandberg. Approximation and radial basis function networks. *Neural Computation*, 5(2):305–316, 1993.

[152] B. Parmanto, P. Munro, and H. Doyle. Improving committe diagnosis with resampling techniques. In D.S. Touretzky, M. Mozer, and M. Hesselmo, editors, *Advances in Neural Information Processing Systems*, volume 8, pages 882–888. MIT Press, Cambridge, MA, 1996.

[153] B. Parmanto, P. Munro, and H. Doyle. Reducing variance of committee prediction with resampling techniques. *Connection Science*, 8(3/4):405–416, 1996.

[154] I. Partalas, G. Tsoumakas, and I. Vlahavas. Focused ensemble selection: A diversity-based method for greedy ensemble selection. In *Proceeding of the 2008 conference on ECAI 2008: 18th European Conference on Artificial Intelligence*, pages 117–121. IOS-Press, 2008.

[155] I. Partalas, G. Tsoumakas, and I. Vlahavas. An ensemble uncertainty aware measure for directed hill climbing ensemble pruning. *Machine Learning*, 81(3):257–282, 2010.

[156] D. Partridge, and W.B Yates. Engineering multiversion neural-net systems. *Neural Computation*, 8:869–893, 1996.

[157] M. Pekalska, E. an Skurichina and R. Duin. Combining Fisher linear discriminant for dissimilarity representations. In J. Kittler and F. Roli, editors, *Multiple Classifier Systems. First International Workshop, MCS 2000, Cagliari, Italy*, volume 1857 of *Lecture Notes in Computer Science*, pages 230–239. Springer-Verlag, 2000.

[158] M.P. Perrone and L.N. Cooper. When networks disagree: ensemble methods for hybrid neural networks. In Mammone R.J., editor, *Artificial Neural Networks for Speech and Vision*, pages 126–142. Chapman & Hall, London, 1993.

[159] R. Polikar et al. Learn++.mf: A random subspace approach for the missing feature problem. *Pattern Recognition*, 27, 2010.

[160] R. Polikar, A. Topalis, Parikh D., D. Green, J. Frymiare, J. Kounios, and C. Clark. An ensemble based data fusion approach for early diagnosis of alzheimer disease. *Information Fusion*, 9:83–95, 2008.

[161] O. Pujol, P. Radeva, and J. Vitria. Discriminant ECOC: A heuristic method for application dependent design of error correcting output codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28:1001–1007, 2006.

[162] J.R. Quinlan. *C4.5 Programs for Machine Learning*. Morgan Kauffman, 1993.

[163] Y. Raviv and N. Intrator. Bootstrapping with noise: An effective regularization technique. *Connection Science*, 8(3/4):355–372, 1996.

[164] M. Re and G. Valentini. Integration of heterogeneous data sources for gene function prediction using Decision Templates and ensembles of learning machines. *Neurocomputing*, 73(7-9):1533–37, 2010.

[165] M. Re and G. Valentini. Noise tolerance of Multiple Classifier Systems in data integration-based gene function prediction. *Journal of Integrative Bioinformatics*, 7(3:139), 2010.

[166] M. Re and G. Valentini. Simple ensemble methods are competitive with state-of-the-art data integration methods for gene function prediction. *Journal of Machine Learning Research, W&C Proceedings, Machine Learning in Systems Biology*, 8:98–111, 2010.

[167] S. Reid and G. Grudic. Regularized linear models in stacked generalization. In J. Kittler, J. Benediktsson, and F. Roli, editors, *Multiple Classifier Systems. Eighth International Workshop, MCS 2009, Reykjavik, Iceland*, volume 5519 of *Lecture Notes in Computer Science*, pages 112–121. Springer, 2009.

[168] J. Rodriguez and L.I. Kuncheva. Rotation forest: A new classifier ensemble method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1619–1630, 2006.

[169] G. Rogova. Combining the results of several neural neetworks classifiers. *Neural Networks*, 7:777–781, 1994.

[170] L. Rokach. Genetic algorithm-based feature set partitioning for classifiaction problems. *Pattern Recognition*, 41(5):1676–1700, 2008.

[171] L. Rokach. Taxonomy for characterizing ensemble methods in classification asks: A reveiw and annotated bibliography. *Computational Statistics and Data Analysis*, 53:4046–4072, 2009.

[172] F. Roli, G. Giacinto, and G. Vernazza. Methods for Designing Multiple Classifier Systems. In J. Kittler and F. Roli, editors, *Multiple Classifier Systems. Second International Workshop, MCS 2001, Cambridge, UK*, volume 2096 of *Lecture Notes in Computer Science*, pages 78–87. Springer-Verlag, 2001.

[173] F. Roli, J. Kittler, and T. Windeatt. *Multiple Classifier Systems, Fifth International Workshop, MCS2004*, volume 3077 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, 2004.

[174] R. Schapire and Y. Singer. Boostexter: A boosting-based system for text categorization. *Machine Learning*, 39(2/3):135–168, 2000.

[175] R.E. Schapire. The strenght of weak learnability. *Machine Learning*, 5(2):197–227, 1990.

[176] R.E. Schapire. A brief introduction to boosting. In Thomas Dean, editor, *16th International Joint Conference on Artificial Intelligence*, pages 1401–1406. Morgan Kauffman, 1999.

[177] R.E. Schapire, Y. Freund, P. Bartlett, and W. Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *The Annals of Statistics*, 26(5):1651–1686, 1998.

[178] R.E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37(3):297–336, 1999.

[179] L. Schietgat, C. Vens, J. Struyf, H. Blockeel, and S. Dzeroski. Predicting gene function using hierarchical multi-label decision tree ensembles. *BMC Bioinformatics*, 11(2), 2010.

[180] B. Scholkopf and A. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.

[181] B. Settles. Active learning literature survey. Technical Report Computer Sciences Technical Report 1648, University of Wisconsin, Madison, 2010.

[182] A. Sharkey. Types of multi-ney systems. In F. Roli and J. Kittler, editors, *Multiple Classifier Systems, Third International Workshop, MCS2002*, volume 2364 of *Lecture Notes in Computer Science*, pages 108–117. Springer-Verlag, 2002.

[183] A. Sharkey, N. Sharkey, and G. Chandroth. Diverse neural net solutions to a fault diagnosis problem. *Neural Computing and Applications*, 4:218–227, 1996.

[184] A. (editor) Sharkey. *Combining Artificial Neural Nets: Ensemble and Modular Multi-Net Systems*. Springer-Verlag, London, 1999.

[185] M. Skurichina and R.P.W. Duin. Bagging for linear classifiers. *Pattern Recognition*, 31(7):909–930, 1998.

[186] M. Skurichina and R.P.W. Duin. Bagging and the Random Subspace Method for Redundant Feature Spaces. In *Multiple Classifier Systems. Second International Workshop, MCS 2001, Cambridge, UK*, volume 2096 of *Lecture Notes in Computer Science*, pages 1–10. Springer-Verlag, 2001.

[187] M. Skurichina and R.P.W. Duin. Bagging, boosting and the random subspace method for linear classifiers. *Pattern Analysis and Applications*, 5(2):121–135, 2002.

[188] S.Y. Sohna and H.W. Shinb. Experimental study for the comparison of of classifier combination methods. *Pattern Recognition*, 40:33–40, 2007.

[189] Y. Sun, S. Todorovic, and L. Li. Reducing the overfitting of adaboost by controlling its data distribution skewness. *Int. J. of Pattern Recognition and Artificial Intelligence*, 20(7):1093–1116, 2006.

[190] D. Tao, X. Tang, X. Li, and X. Wu. Asymmetric bagging and random subspace for support vector machine-based relevance feedback in image retrieval. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(7):1088–1099, 2006.

[191] G. Tsoumakas, L. Angelis, and I. Vlahavas. Selective fusion of heterogeneous classifiers. *Intelligent Data Analysis*, 9(6):511–525, 2005.

[192] G. Tsoumakas, I. Katakis, and I. Vlahavas. Effective voting of heterogeneous classifiers. In *Proc. of the 15th European Conference on Machine Learning, ECML 2004*, pages 465–476, 2004.

[193] A. Tsymbal, M. Pechenizkiy, and P. Cunningham. Diversity in search strategies for ensemble feature selection. *Information Fusion*, 6:83–98, 2006.

[194] A. Tsymbal, S. Puuronen, and D.W. Patterson. Ensemble feature selection with the simple bayesian classifiaction. *Information Fusion*, 4:87–100, 2003.

[195] K. Tumer and J. Ghosh. Error correlation and error reduction in ensemble classifiers. *Connection Science*, 8(3/4):385–404, 1996.

[196] G. Valentini. Gene expression data analysis of human lymphoma using support vector machines and output coding ensembles. *Artificial Intelligence in Medicine*, 26(3):283–306, 2002.

[197] G. Valentini. An experimental bias-variance analysis of SVM ensembles based on resampling techniques. *IEEE Transactions on Systems, Man and Cybernetics-Part B: Cybernetics*, 35(6):1252–1271, 2005.

[198] G. Valentini. True path rule hierarchical ensembles for genome-wide gene function prediction. *IEEE ACM Transactions on Computational Biology and Bioinformatics*, 8(3) May/June 2011.

[199] G. Valentini and T.G. Dietterich. Low Bias Bagged Support Vector Machines. In T. Fawcett and N. Mishra, editors, *Machine Learning, Proceedings of the Twentieth International Conference (ICML 2003)*, pages 752–759, Washington D.C., USA, 2003. AAAI Press.

[200] G. Valentini and T.G. Dietterich. Bias–variance analysis of Support Vector Machines for the development of SVM-based ensemble methods. *Journal of Machine Learning Research*, 5:725–775, 2004.

[201] G. Valentini and F. Masulli. Ensembles of learning machines. In *Neural Nets WIRN-02*, volume 2486 of *Lecture Notes in Computer Science*, pages 3–19. Springer-Verlag, 2002.

[202] J. Van Lint. *Coding theory*. Spriger Verlag, Berlin, 1971.

[203] V. N. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.

[204] C. Vens, J. Struyf, L. Schietgat, S. Dzeroski, and H. Blockeel. Decision trees for hierarchical multi-label classification. *Machine Learning*, 73(2):185–214, 2008.

[205] A. Verikas, A. Lipnickas, K. Malmqvist, M. Bacauskiene, and A. Gelzinis. Soft combination of neural classifiers: A comparative study. *Pattern Recognition Letters*, 20:429–444, 1999.

[206] K. L. Wagstaff and V. G. Laidler. Making the most of missing values: Object clustering with partial data in astronomy. In *Astronomical Data Analysis Software and Systems XIV ASP Conference Series, Vol. 347, Proceedings of the Conference held 24-27 October, 2004 in Pasadena, California, USA.*, page 172, 2005.

[207] D. Wang, J.M. Keller, C.A. Carson, K.K. McAdoo-Edwards, and C.W. Bailey. Use of fuzzy logic inspired features to improve bacterial recognition through classifier fusion. *IEEE Transactions on Systems, Man and Cybernetics*, 28B(4):583–591, 1998.

[208] D.H. Wolpert. Stacked Generalization. *Neural Networks*, 5:241–259, 1992.

[209] K. Woods, W.P. Kegelmeyer, and K. Bowyer. Combination of multiple classifiers using local accuracy estimates. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(4):405–410, 1997.

[210] L Xu, C Krzyzak, and C. Suen. Methods of combining multiple classifiers and their applications to handwritting recognition. *IEEE Transactions on Systems, Man and Cybernetics*, 22(3):418–435, 1992.

[211] C. Yeang, S. Ramaswamy, P. Tamayo, S. Mukherjee, R.M. Rifkin, M. Angelo, M. Reich, E. Lander, J. Mesirov, and T. Golub. Molecular classification of multiple tumor types. In *ISMB 2001, Proceedings of the 9th International Conference on Intelligent Systems for Molecular Biology*, pages 316–322, Copenaghen, Denmark, 2001. Oxford University Press.

[212] C.X. Zhang and J.S. Zhang. A local boosting algorithm for solving classification problems. *Computational Statistics and Data Analysis*, 52(4):1928–1941, 2008.

[213] Y. Zhang, S. Burer, and W. Nick Street. Ensemble pruning via semi-definite programming. *Journal of Machine Learning Research*, 7:1315–1338, 2006.

[214] J. Zhou, H. Peng, and C. Suen. Data-driven decompositon for multi-class classification. *Pattern Recognition*, 41(1):67–76, 2008.

[215] X. Zhu and A. Goldberg. *Introduction to Semi-Supervised Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool, 2009.