Ensembles of Learning Machines

Giorgio Valentini[1,2] and Francesco Masulli[1,3]

[1] INFM, Istituto Nazionale per la Fisica della Materia, 16146 Genova, Italy
[2] DISI, Università di Genova, 16146 Genova, Italy
valenti@disi.unige.it
[3] Dipartimento di Informatica, Università di Pisa, 56125 Pisa, Italy
masulli@di.unipi.it

Abstract. Ensembles of learning machines constitute one of the main current directions in machine learning research, and have been applied to a wide range of real problems. Despite of the absence of an unified theory on ensembles, there are many theoretical reasons for combining multiple learners, and an empirical evidence of the effectiveness of this approach. In this paper we present a brief overview of ensemble methods, explaining the main reasons why they are able to outperform any single classifier within the ensemble, and proposing a taxonomy based on the main ways base classifiers can be generated or combined together.

1 Introduction

Ensembles are sets of learning machines whose decisions are combined to improve the performance of the overall system. In this last decade one of the main research areas in machine learning has been represented by methods for constructing ensembles of learning machines. Although in the literature [86, 129, 130, 69, 61, 23, 33, 12, 7, 37] a plethora of terms, such as committee, classifier fusion, combination, aggregation and others are used to indicate sets of learning machines that work together to solve a machine learning problem, in this paper we shall use the term *ensemble* in its widest meaning, in order to include the whole range of combining methods. This variety of terms and specifications reflects the absence of an unified theory on ensemble methods and the youngness of this research area. However, the great effort of the researchers, reflected by the amount of the literature [118, 70, 71] dedicated to this emerging discipline, achieved meaningful and encouraging results.

Empirical studies showed that both classification and regression problem ensembles are often much more accurate than the individual base learner that make them up [8, 29, 40], and recently different theoretical explanations have been proposed to justify the effectiveness of some commonly used ensemble methods [69, 112, 75, 3].

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. 2 of this paper there are deeper reasons to use ensembles of learning machines. motivated by the intrinsic characteristics of the ensemble methods.

This work presents a brief overview 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 reasons for combining multiple learners are depicted. Sect. 3 presents an overview of the main ensemble methods reported in the literature, distinguishing between *generative* and *non-generative* methods, while Sect. 4 outlines some open problems not covered in this paper.

2 Reasons for Combining Multiple Learners

Both empirical observations and specific machine learning applications confirm that a given learning algorithm outperforms all others for a specific problem or for a specific subset of the input data, but it is unusual to find a single expert achieving the best results on the overall problem domain. As a consequence multiple learner systems try to exploit the local different behavior of the base learners to enhance the accuracy and the reliability of the overall inductive learning system. There are also hopes that if some learner fails, the overall system can recover the error. Employing multiple learners can derive from the application context, such as when multiple sensor 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.

However, there are deeper reasons why ensembles can improve performances with respect to a single learning machine. As an example, consider the following one given by Tom Dietterich in [28]. If we have a dichotomic classification problem and L hypotheses whose error is lower than 0.5, then the resulting majority voting ensemble has an error lower than the single classifier, as long as the error of the base learners are uncorrelated. In fact, if we have 21 classifiers, and the error rates of each base learner are all equal to $p = 0.3$ and the errors are independent, the overall error of the majority voting ensemble will be given by the area under the binomial distribution where more than $L/2$ hypotheses are wrong:

$$P_{error} = \sum_{(i=\lceil L/2 \rceil)}^{L} \binom{L}{i} p^i (1-p)^{L-i} \quad \Rightarrow P_{error} = 0.026 \ll p = 0.3$$

This result has been studied by mathematicians since the end of the XVIII century in the context of social sciences: in fact the *Condorcet Jury Theorem* [26]) proved that the judgment of a committee is superior to those of individuals, provided the individuals have reasonable competence (that is, a probability of being correct higher than 0.5). As noted in [85], this theorem theoretically justifies recent research on multiple "weak" classifiers [63, 51, 74], representing an interesting research direction diametrically opposite to the development of highly accurate and specific classifiers.

This simple example shows also an important issue in the design of ensembles of learning machines: the effectiveness of ensemble methods relies on the independence of the error committed by the component base learner. In this example, if the independence assumption does not hold, we have no assurance that the ensemble will lower the error, and we know that in many cases the errors are correlated. From a general standpoint we know that the effectiveness of ensemble methods depends on the *accuracy* and the *diversity* of the base learners, that is if they exhibit low error rates and if they produce different errors [49, 123, 92]. The correlated concept of independence between the base learners has been commonly regarded as a requirement for effective classifier combinations, but recent works have shown that not always independent classifiers outperform dependent ones [84]. In fact there is a trade-off between accuracy and independence: more accurate are the base learners, less independent they are.

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. In these cases the learning algorithm can find different hypotheses that appear equally accurate with respect to the available training data, and although we can sometimes select among them the simplest or the one with the lowest capacity, we can avoid the problem averaging or combining them to get a good approximation of the unknown true hypothesis.

Another 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, embracing also the true one. Although many learning algorithms present universal approximation properties [55, 100], with finite data sets these asymptotic features do not hold: the effective space of hypotheses explored by the learning algorithm is a function of the available data and it 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.

Many learning algorithms apply local optimization techniques that may get stuck in local optima. For instance inductive decision trees employ a greedy local optimization approach, and neural networks apply gradient descent techniques to minimize an error function over the training data. Moreover optimal training with finite data both for neural networks and decision trees is NP-hard [13, 57]. As a consequence even if the learning algorithm can in principle find the best hypothesis, we actually may not be able to find it. Building an ensemble using, for instance, different starting points may achieve a better approximation, even if no assurance of this is given.

Another way to look at the need for ensembles is represented by the classical bias–variance analysis of the error [45, 78]: different works have shown that several ensemble methods reduce variance [15, 87] or both bias and variance [15, 39, 77]. Recently the improved generalization capabilities of different ensemble methods have also been interpretated in the framework of the theory of large margin classifiers [89, 113, 3], showing that methods such as boosting and ECOC enlarge the margins of the examples.

3 Ensemble Methods Overview

A large number of combination schemes and ensemble methods have been proposed in literature. Combination techniques can be grouped and analysed in different ways, depending on the main classification criterion adopted. 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 [68, 69].

Assuming the architecture of the ensemble as the main criterion, we can distinguish between serial, parallel and hierarchical schemes [85], and if the base learners are selected or not by the ensemble algorithm we can separate selection-oriented and combiner-oriented ensemble methods [61, 81]. In this brief overview we adopt an approach similar to the one cited above, in order to distinguish between *non-generative* and *generative* ensemble methods. Non-generative ensemble methods confine theirselves to combine a set of given possibly well-designed base learners: they do not actively generate new base learners but try to combine in a suitable way a set of existing base classifiers. Generative ensemble methods generate sets of base learners acting on the base learning algorithm or on the structure of the data set and try to actively improve diversity and accuracy of the base learners.

3.1 Non-generative Ensembles

This large group of ensemble methods embraces a large set of different approaches to combine learning machines. They 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 its adaptivity to the input patterns and on the requirement of the output of the individual learning machines.

The type of combination may depend on the type of output. If only labels are available or if continuous outputs are hardened, then *majority voting*, that is the class most represented among the base classifiers, is used [67, 104, 87].

This approach can be refined assigning different weights to each classifier to optimize the performance of the combined classifier on the training set [86], or, assuming mutual independence between classifiers, a *Bayesian decision rule* selects the class with the highest posterior probability computed through the estimated class conditional probabilities and the Bayes' formula [130, 122]. A

Bayesian approach has also been used in *Consensus based classification* of multi-source remote sensing data [10, 9, 19], 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 [56] 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.

Where we interpret the classifier outputs as the support for the classes, fuzzy aggregation methods can be applied, such as simple connectives between fuzzy sets or the fuzzy integral [23, 22, 66, 128]; if the classifier outputs are possibilistic, *Dempster-Schafer* combination rules can be applied [108]. Statistical methods and similarity measures to estimate classifier correlation have also been used to evaluate expert system combination for a proper design of multi-expert systems [58].

The base learners can also be aggregated using simple operators as *Minimum*, *Maximum*, *Average* and *Product* and *Ordered Weight Averaging* [111, 18, 80]. In particular, 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 [69].

Recently Ljudmila Kuncheva has developed a global combination scheme that takes into account the decision profiles of all the ensemble classifiers with respect to all the classes, designing *Decision templates* that summarize in matrix format the average decision profiles of the training set examples. Different similarity measures can be used to evaluate the matching between the matrix of classifier outputs for an input x, that is the decision profiles referred to x, and the matrix templates (one for each class) found as the class means of the classifier outputs [81]. This general fuzzy approach produce soft class labels that can be seen as a generalization of the conventional crisp and probabilistic combination schemes.

Another general approach consists in explicitly *training combining rules*, using second-level learning machines on top of the set of the base learners [34]. This stacked structure makes use of the outputs of the base learners as features in the intermediate space: the outputs are fed into a second-level machine to perform a trained combination of the base learners.

3.2 Generative Ensembles

Generative ensemble methods try to improve the overall accuracy of the ensemble by directly boosting the accuracy and the diversity of the base learner. They can modify the structure and the characteristics of the available input data, as in *resampling* methods or in *feature selection* methods, they can manipulate the aggregation of the classes (*Output Coding* methods), can select base learners specialized for a specific input region (*mixture of experts* methods), can select a proper set of base learners evaluating the performance and the characteristics of the component base learners (*test-and-select* methods) or can randomly modify the base learning algorithm (*randomized* methods).

Resampling methods Resampling techniques can be used to generate different hypotheses. For instance, bootstrapping techniques [35] 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.

In *bagging* [15] the ensemble is formed by making bootstrap replicates of the training sets, and then multiple generated hypotheses are used to get an aggregated predictor. The aggregation can be performed averaging the outputs in regression or by majority or weighted voting in classification problems [120, 121].

While in bagging the samples are drawn with replacement using a uniform probability distribution, in *boosting* methods the learning algorithm is called at each iteration using a different distribution or weighting over the training examples [111, 40, 112, 39, 115, 110, 32, 38, 33, 32, 16, 17, 42, 41]. 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 base rules taking a weighted majority vote of the base rules. Schapire and Singer showed that the training error exponentially drops down with the number of iterations [114] and Schapire et al. [113] 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. Experimental work showed that bagging is effective with noisy data, while boosting, concentrating its efforts on noisy data seems to be very sensitive to noise [107, 29].

Another training set sampling method consists in constructing training sets by leaving out disjoint subsets of the training data as in *cross-validated committees* [101, 102] or sampling without replacement [116].

Another general approach, named *Stochastic Discrimination* [73, 74, 75, 72], is based on randomly sampling from a space of subsets of the feature space underlying a given problem, then combining these subsets to form a final classifier, using a set-theoretic abstraction to remove all the algorithmic details of classifiers and training procedures. By this approach the 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 [20].

Feature selection methods This approach consists in reducing the number of input features of the base learners, a simple method to fight the effects of the classical curse of dimensionality problem [43]. For instance, in the *Random*

Subspace Method [51, 82], a subset of features is randomly selected and assigned to an arbitrary learning algorithm. This way, one obtains a random subspace of the original feature space, and constructs classifiers inside this reduced subspace. The aggregation is usually performed using weighted voting on the basis of the base classifiers accuracy. It has been shown that this method is effective for classifiers having a decreasing learning curve constructed on small and critical training sample sizes [119]

The *Input Decimation* approach [124, 98] 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 as 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.

Feature subspace methods performed by partitioning the set of features, where each subset is used by one classifier in the team, are proposed in [130, 99, 18]. Other methods for combining different feature sets using genetic algorithms are proposed in [81, 79]. Different approaches consider feature sets obtained by using different operators on the original feature space, such as Principal Component Analysis, Fourier coefficients, Karhunen-Loewe coefficients, or other [21, 34]. An experiment with a systematic partition of the feature space, using nine different combination schemes is performed in [83], showing that there are no "best" combinations for all situations and that there is no assurance that in all cases a classifier team will outperform the single best individual.

Mixtures of experts methods The recombination of the base learners can be governed by a supervisor learning machine, that selects the most appropriate element of the ensemble on the basis of the available input data. This idea led to the *mixture of experts* methods [60, 59], where a gating network performs the division of the input space and small neural networks perform the effective calculation at each assigned region separately. An extension of this approach 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 [64, 65, 59].

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 input space [24, 25]. They proposed a hybrid MLP/RBF network by combining RBF and Perceptron units in the same hidden layer and using a forward selection [36] to add units until an error goal is reached. 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.

Output Coding decomposition methods *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 [94, 90, 28]. 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 misure of similarity between the codeword computed by the ensemble and the codeword classes is used to predict the class.

Different *decomposition schemes* have been proposed in literature: In the One-Per-Class (OPC) decomposition [5], each dichotomizer f_i has to separate a single class from all others; in the *PairWise Coupling* (PWC) decomposition [50], the task of each dichotomizer f_i consists in separating a class C_i form class C_j, 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 [96].

Error Correcting Output Coding [30, 31] is the most studied OC method, and has been successfully applied to several classification problems [1, 11, 46, 6, 126, 131]. Thisdecomposition method tries to improve the error correcting capabilities of the codes generated by the decomposition through the maximization of the minimum distance between each couple of codewords [77, 90]. This goal is achieved by means of the redundancy of the coding scheme [127].

ECOC methods present several open problems. The tradeoff between error recovering capabilities and complexity/learnability of the dichotomies induced by the decomposition scheme has been tackled in several works [3, 125], but an extensive experimental evaluation of the tradeoff has to be performed in order to achieve a better understanding of this phenomenon. A related problem is the analysis of the relationship between codeword length and performances: some preliminary results seem to show that long codewords improve performance [46]. Another open problem, not sufficiently investigated in literature [46, 91, 11], is the selection of optimal dichotomic learning machines for the decomposition unit. Several methods for generating ECOC codes have been proposed: exhaustive codes, randomized hill climbing [31], random codes [62], and Hadamard and BCH codes [14, 105]. An open problems is still the joint maximization of distances between rows and columns in the decomposition matrix. Another open problem consists in designing codes for a given multiclass problem. An interesting greedy approach is proposed in [94], and a method based on soft weight sharing to learn error correcting codes from data is presented in [4]. In [27] 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.

The work in [91] highlights that 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. In particular, Peterson and Weldon [105] showed that if errors on different code bits are dependent, the effectiveness of error correcting code is reduced. Consequently, if a decomposition matrix contains very similar rows (dichotomies), each error of an assigned dichotomizer will be likely to appear in the most correlated dichotomizers, thus reducing the effectiveness of ECOC. These hypotheses have been experimentally supported by a quantitative evaluation of the dependency among output errors of the decomposition unit of ECOC learning machines using mutual information based measures [92, 93].

Test and select methods The *test and select* methodology relies on the idea of selection in ensemble creation [117]. The simplest approach is a greedy one [104], where a new learner is added to the ensemble only if the resulting squared error is reduced, but in principle any optimization technique can be used to select the "best" component of the ensemble, including genetic algorithms [97].

It should be noted that the time complexity of the selection of optimal subsets of classifiers is exponential with respect to the number of base learners used. From this point of view heuristic rules, as the "choose the best" or the "choose the best in the class", using classifiers of different types strongly reduce the computational complexity of the selected phase, as the evaluation of different classifier subsets is not required [103]. Moreover test and select methods implicitly include a "production stage", by which a set of classifiers must be generated.

Different selection methods based on different search algorithm mututated from feature selection methods (forward and backward search) or for the solution of complex optimization tasks (tabu search) are proposed in [109]. Another interesting approach uses clustering methods and a misure of diversity to generate sets of diverse classifiers combined by majority voting, selecting the ensemble with the highest performance [48]. Finally, *Dynamic Classifier Selection* methods [54, 129, 47] are based on the definition of a function selecting for each pattern the classifier which is probably the most accurate, estimating, for instance the accuracy of each classifier in a local region of the feature space surrounding an unknown test pattern [47].

Randomized ensemble methods Injecting randomness into the learning algorithm is another general method to generate ensembles of learning machines. For instance, if we initialize with random values the initial weights in the backpropagation algorithm, we can obtain different learning machines that can be combined into an ensemble [76, 101].

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 [29] randomized decision tree ensembles outperform single C4.5 decision trees [106], and adding gaussian noise to the data inputs, together with bootstrap and weight regularization can achieve large improvements in classification accuracy [107].

4 Conclusions

Ensemble methods have shown to be effective in many applicative domains and can be considered as one of the main current directions in machine learning research. We presented an overview of the ensemble methods, showing the main areas of research in this discipline, and the fundamental reasons why ensemble methods are able to outperform any single classifier within the ensemble. A general taxonomy, distinguishing between *generative* and *non–generative* ensemble methods, has been proposed, considering the different ways base learners can be generated or combined together.

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, even if a general theoretical framework for ensemble methods has not been developed. Other open problems not covered in this work are the relationships between ensemble methods and data complexity [52, 53, 88], a systematic research of hidden commonalities among all the combination approaches despite their superficial differences, and a general analysis of the relationships between ensemble methods and the characteristics of the base learners used in the ensemble itself.

Acknowledgments

This work has been partially funded by INFM.

References

[1] 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.

[2] K.M. Ali and M.J. Pazzani. Error reduction through learning multiple descriptions. *Machine Learning*, 24(3):173–202, 1996.

[3] 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.

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

[5] 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.

[6] G. Bakiri and T.G. Dietterich. Achieving high accuracy text-to-speech with machine learning. In *Data mining in speech synthesis*. 1999.

[7] R. Battiti and A.M. Colla. Democracy in neural nets: Voting schemes for classification. *Neural Networks*, 7:691–707, 1994.

[8] E. Bauer and R.. Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting and variants. *Machine Learning*, 36(1/2):525–536, 1999.

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

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

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

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

[13] A. Blum and R.L. Rivest. Training a 3-node neural network is NP-complete. In *Proc. of the 1988 Workshop ob Computational Learning Learning Theory*, pages 9–18, San Francisco, CA, 1988. Morgan Kaufmann.

[14] R.C. Bose and D.K. Ray-Chauduri. On a class of error correcting binary group codes. *Information and Control*, (3):68–79, 1960.

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

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

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

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

[19] 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.

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

[21] K.J. Cherkauker. Human expert-level performance on a scientific image analysis task by a system using combined artificial neural networks. In Chan P., editor, *Working notes of the AAAI Workshop on Integrating Multiple Learned Models*, pages 15–21. 1996.

[22] 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.

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

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

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

[30] 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.

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

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

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

[34] 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.

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

[36] S.E. Fahlman and C. Lebiere. The cascade-correlation learning architecture. In D.S. Touretzky, editor, *Advances in Neural Information Processing Systems*, volume 2, pages 524–532. Morgan Kauffman, San Mateo, CA, 1990.

[37] 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.

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

[39] 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.

[40] 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.

[41] J. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 39(5), 2001.

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

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

[44] C. Furlanello and S. Merler. Boosting of Tree-based Classifiers for Predictive Risk Modeling in GIS. 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 220–229. Springer-Verlag, 2000.

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

[46] 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.

[47] 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.

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

[49] T. Hastie and R. Tibshirani. *Generalized Additive Models*. Chapman and Hall, London, 1990.

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

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

[52] 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.

[53] 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.

[54] 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.

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

[56] 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.

[57] L. Hyafil and R.L. Rivest. Constructing optimal binary decision tree is np-complete. *Information Processing Letters*, 5(1):15–17, 1976.

[58] S. Impedovo and A. Salzo. A New Evaluation Method for Expert Combination in Multi-expert System Designing. 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.

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

[60] 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.

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

[62] G. James. *Majority vote classifiers: theory and applications*. PhD thesis, Department of Statistics - Stanford University, Stanford, CA, 1998.

[63] C. Ji and S. Ma. Combinination of weak classifiers. *IEEE Trans. Neural Networks*, 8(1):32–42, 1997.

[64] 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.

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

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

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

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

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

[70] J. Kittler and F. (editors) Roli. *Multiple Classifier Systems, Proc. of 1st International Workshop, MCS 2000, Cagliari, Italy*, volume 1857 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, 2000.

[71] J. Kittler and F. (editors) Roli. *Multiple Classifier Systems, Proc. of 2nd International Workshop, MCS2001, Cambridge, UK.* Springer-Verlag, Berlin, 2001.

[72] E.M. Kleinberg. On the Algorithmic Implementation of Stochastic Discrimination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

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

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

[75] 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.

[76] J. Kolen and Pollack J. Back propagation is sensitive to initial conditions. In *Advances in Neural Information Processing Systems*, volume 3, pages 860–867. Morgan Kauffman, San Francisco, CA, 1991.

[77] 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.

[78] A. Krogh and J. Vedelsby. Neural networks ensembles, cross validation and active learning. In D.S. Touretzky, G. Tesauro, and T.K. Leen, editors, *Advances in Neural Information Processing Systems*, volume 7, pages 107–115. MIT Press, Cambridge, MA, 1995.

[79] L.I. Kuncheva. Genetic algorithm for feature selection for parallel classifiers. *Information Processing Letters*, 46:163–168, 1993.

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

[81] 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.

[82] 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.

[83] L.I. Kuncheva and C.J. Whitaker. Feature Subsets for Classifier Combination: An Enumerative Experiment. 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 228–237. Springer-Verlag, 2001.

[84] L.I. Kuncheva et al. Is independence good for combining classifiers? In *Proc. of 15th Int. Conf. on Pattern Recognition*, Barcelona, Spain, 2000.

[85] 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.

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

[87] 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.

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

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

[90] F. Masulli and G. Valentini. Comparing decomposition methods for classification. In R.J. Howlett and L.C. Jain, editors, *KES'2000, Fourth International Conference on Knowledge-Based Intelligent Engineering Systems & Allied Technologies*, pages 788–791, Piscataway, NJ, 2000. IEEE.

[91] 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.

[92] F. Masulli and G. Valentini. Dependence among Codeword Bits Errors in ECOC Learning Machines: an Experimental Analysis. In *Lecture Notes in Computer Science*, volume 2096, pages 158–167. Springer-Verlag, Berlin, 2001.

[93] 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.

[94] 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.

[95] S. Merler, C. Furlanello, B. Larcher, and A. Sboner. Tuning Cost-Sensitive Boosting and its Application to Melanoma Diagnosis. 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 32–42. Springer-Verlag, 2001.

[96] 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.

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

[98] 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.

[99] H.S. Park and S.W. Lee. Off-line recognition of large sets handwritten characters with multiple Hidden-Markov models. *Pattern Recognition*, 29(2):231–244, 1996.

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

[101] 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.

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

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

[104] 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.

[105] W.W. Peterson and E.J.Jr. Weldon. *Error correcting codes.* MIT Press, Cambridge, MA, 1972.

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

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

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

[109] 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.

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

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

[112] 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.

[113] 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.

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

[115] H. Schwenk and Y. Bengio. Training methods for adaptive boosting of neural networks. In *Advances in Neural Information Processing Systems*, volume 10, pages 647–653. 1998.

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

[117] A Sharkey, N. Sharkey, U. Gerecke, and G. Chandroth. The test and select approach to ensemble combination. 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 30–44. Springer-Verlag, 2000.

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

[119] M. Skurichina and R.P.W. Duin. Bagging, boosting and the randon subspace method for linear classifiers. *Pattern Analysis and Applications*. (in press).

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

[121] 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.

[122] C. Suen and L. Lam. Multiple classifier combination methodologies for different output levels. In *Multiple Classifier Systems. First International Workshop, MCS 2000, Cagliari, Italy*, volume 1857 of *Lecture Notes in Computer Science*, pages 52–66. Springer-Verlag, 2000.

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

[124] K. Tumer and N.C. Oza. Decimated input ensembles for improved generalization. In *IJCNN-99, The IEEE-INNS-ENNS International Joint Conference on Neural Networks*, 1999.

[125] G. Valentini. Upper bounds on the training error of ECOC-SVM ensembles. Technical Report TR-00-17, DISI - Dipartimento di Informatica e Scienze dell' Informazione - Università di Genova, 2000. ftp://ftp.disi.unige.it/person/ValentiniG/papers/TR-00-17.ps.gz.

[126] G. Valentini. Gene expression data analysis of human lymphoma using Support Vector Machines and Output Coding ensembles. *Artificial Intelligence in Medicine* (to appear).

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

[128] 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.

[129] 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.

[130] 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.

[131] C. Yeang et al. 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.