A Theoretical Analysis of Bagging as a Linear Combination of Classifiers

Giorgio Fumera, *Member, IEEE,* Fabio Roli, *Member, IEEE,* Alessandra Serrau

Department of Electrical and Electronic Engineering, University of Cagliari

Piazza d'Armi, 09123 Cagliari, Italy

Giorgio Fumera: e-mail fumera@diee.unica.it, phone +39 070 657 5754, fax +39 070 675 5782

Fabio Roli (corresponding author): e-mail roli@diee.unica.it, phone +39 070 657 5779, fax +39 070 675 5782

Alessandra Serrau: e-mail serrau@diee.unica.it, phone +39 070 657 5776, fax +39 070 675 5782

Abstract

We apply an analytical framework for the analysis of linearly combined classifiers to ensembles generated by bagging. This provides an analytical model of bagging misclassification probability as a function of the ensemble size, which is a novel result in the literature. Experimental results on real data sets confirm the theoretical predictions. This allows us to derive a novel and theoretically grounded guideline for choosing bagging ensemble size. Furthermore, our results are consistent with explanations of bagging in terms of classifier instability and variance reduction, support the optimality of the simple average over the weighted average combining rule for ensembles generated by bagging, and apply to other randomization-based methods for constructing classifier ensembles. Although our results do not allow to compare bagging misclassification probability with the one of an individual classifier trained on the *original* training set, we discuss how the considered theoretical framework could be exploited to this aim.

Index Terms

Multiple Classifier Systems, Bagging, Linear Combiners, Classifier Fusion, Pattern Classification.

I. Introduction

Several methods for the construction of classifier ensembles, like bagging [3], the random subspace method [12], tree randomization [6] and random forests [4], are based on introducing some kind of randomness into the design process of individual classifiers. Bagging is perhaps the most popular method, and its effectiveness has been empirically shown in many real pattern recognition problems. However it still exhibits at least two main open issues. First, there is no clear understanding yet of the conditions under which bagging outperforms an individual classifier, as well as other ensemble construction methods. Second, only empirical and rough guidelines have been proposed so far to determine a suitable ensemble size for a task with given computational requirements in terms of memory size and CPU time. With regard to the former issue, currently the main explanation of bagging operation is given in terms of its capability to reduce the variance component of the misclassification probability, which was related by Breiman [3] to the degree of "instability" of the base classifier, informally defined as the tendency of undergoing large changes in its decision function as a result of small changes in the training set: the more unstable a classifier, the higher the variance component of its misclassification probability and thus the improvement attained by bagging. For classification problems, this

explanation is supported by empirical evidence [2], [6], [16], [20], according to several bias-variance decompositions proposed so far, although alternative explanations have been proposed as well (for instance [5], [7], [10]), and some works showed that bagging can also reduce bias [2], [20]. With regard to the latter issue above, it is well known that bagging miclassification rate tends to an asymptotic value as the ensemble size increases. Works in the literature focussed on determining the ensemble size sufficient to reach the asymptotic miclassification rate, empirically showing that suitable values are between 10 and 20 depending on the particular data set and base classifier [2], [3], [16]. Several researchers also proposed methods to select classifiers generated by bagging, with the aim of improving the ensemble accuracy and reducing its size [1], [14]–[16].

The main aim of this work was to apply a theoretical framework for the analysis of linearly combined classifiers, developed in [18], [19] and extended in [8], to ensembles generated by bagging, to investigate whether it could give any contribution to some of the open issues mentioned above. The framework is summarized in Sect. II-A. Preliminary results were presented in [9]. In Sect. II-B we show that, when applied to bagging, this framework provides a simple analytical model of bagging misclassification probability as a function of the ensemble size, which is a novel result in the literature. Experiments carried out on twenty-one benchmark data sets (Sect. III) show that it allows to accurately model the behaviour of bagging misclassification rate as a function of the ensemble size. This allows us to formulate a simple, quantitative and theoretically grounded guideline for choosing bagging ensemble size, which is an advance with respect to empirical guidelines reported in the literature, and can be useful in applications characterized by strict requirements on computational complexity at operation time. Our results give thus a contribution to the second open issue mentioned above. Although they can not be directly exploited to invesigate under which conditions bagging outperforms an individual classifier trained on the *original* training set, we discuss in Sect. II-C how the considered theoretical framework could be applied to this aim. As a byproduct, this framework gives a theoretical support to the optimality of the simple average combining rule over the weighted average for classifiers generated by bagging. We finally show that our results are not limited to bagging, but apply to other randomization-based methods for the construction of classifier ensembles.

II. A Theoretical Analysis of Bagging

The following notation will be used in this paper. T: a random variable consisting of n samples $\{(X_i, Y_i)\}_{i=1}^n$ drawn independently according to a given probability distribution $\mathbb{P}[X, Y]$, being X the feature vector and Y the class label. t: a realization of the random variable T, representing a given data set of n samples. t^{b}: a single bootstrap replicate of a given data set t, consisting of n samples drawn independently and with replacement from t, each with identical probability $1/n$. t^{B}: a random variable defined by the distribution over all possible bootstrap replicates of the single data set t (note that there are n^n different and equiprobable bootstrap replicates of a given data set of size n). Accordingly, t^{b} is a realization of t^{B}. $\mathbb{E}_{\cdot}[\cdot]$ and $\mathbb{V}_{\cdot}[\cdot]$: the expectation and variance operators (the subscript will be omitted when the random variable to which the operators apply is clear from the context).

A. An Analytical Framework for Linear Combiners

The analytical framework developed in [18], [19] and extended in [8] applies to classifiers which provide estimates of the a posteriori probabilities. It allows to approximate the component of the additional misclassification probability over Bayes error (named *added error*) around a given ideal class boundary, in the case when the effect of using the estimated posteriors leads to a boundary between the same classes shifted from the ideal one (see Fig. 1), and to compare the expected added error of an individual classifier with the one obtained by linearly combining the estimates provided by an ensemble of classifiers. For readers who are not familiar with this framework, a good introduction and discussion about its limitations can be found in [13]. In the following we summarize the derivation of the expression of the expected added error, which will be exploited in the next section, omitting only the intermediate steps which can be found in [8], [13], [18], [19].

Let $\mathbb{P}[\omega_k|x]$ denote the posterior probability of class ω_k at point x of a one-dimensional feature space,[1] and $f_k(x) = \mathbb{P}[\omega_k|x] + \epsilon_k(x)$ denote the corresponding estimate provided by a given classifier, where the estimation error $\epsilon_k(x)$ is viewed as a realization of a random variable. Without loosing generality, for the purposes of this work we will assume that the randomness

[1]The more complex case of multidimensional feature spaces is discussed in [17], where it is shown that the same results of the one-dimensional case hold.

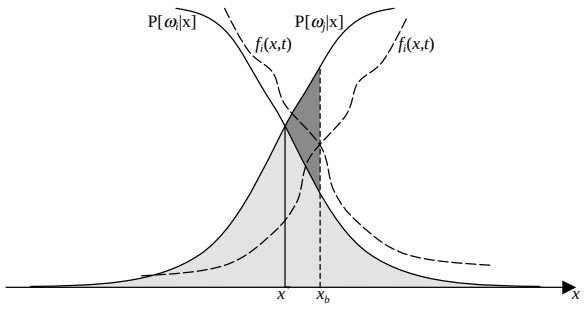

Fig. 1. True posterior probabilities around the boundary x^* between classes ω_i and ω_j (solid lines), and estimated posteriors (dashed lines), leading to the boundary $x_{\mathrm{b}}(t)$. Lightly and darkly shaded areas represent the contribution of this class boundary to Bayes error and to added error, respectively.

is due only to the choice of the training set t, and will explicitly denote this by writing $f_k(x;t)$ and $\epsilon_k(x;t)$. Fig. 1 shows the ideal boundary x^* between any two classes ω_i and ω_j, and the estimated boundary $x_{\mathrm{b}}(t)$, which is shifted from the ideal one by an amount $b = x_{\mathrm{b}}(t) - x^*$ (without loosing generality, in Fig. 1 it is assumed that $b > 0$). Using the estimated boundary causes an added error $e_{\mathrm{add}}(t) = \int_{x^*}^{x_{\mathrm{b}}(t)} \left(\mathbb{P}[\omega_j|x] - \mathbb{P}[\omega_i|x]\right) \mathbb{P}[x] \mathrm{d}x$, due to the fact that patterns in the interval $(x^*, x_{\mathrm{b}}(t))$ are assigned to class ω_i instead of ω_j. Using a first order approximation of the true posteriors of classes ω_i and ω_j and a zero-order approximation of the probability distribution $\mathbb{P}[x]$ around x^*, one obtains $e_{\mathrm{add}}(t) = \frac{\mathbb{P}[x^*]s}{2} b^2$, where $s = \mathbb{P}'[\omega_j|x^*] - \mathbb{P}'[\omega_i|x^*]$ [18], [19]. The expected value with respect to training sets T, denoted with e_{add}, is thus given by $\frac{\mathbb{P}(x^*)s}{2}(\beta_b^2 + \sigma_b^2)$, being β_b and σ_b^2 respectively the mean and variance of b. Accordingly, the corresponding summands in the expression of e_{add} are called in [18], [19] respectively the *bias* and *variance* components of the expected added error. Using the approximations above, one also obtains that $b = \frac{\epsilon_i(x_{\mathrm{b}}(t);t) - \epsilon_j(x_{\mathrm{b}}(t);t)}{s}$, which allows to rewrite the added error as a function of the estimation errors: $e_{\mathrm{add}}(t) = \frac{\mathbb{P}[x^*]}{2s}[\epsilon_i(x_{\mathrm{b}}(t);t) - \epsilon_j(x_{\mathrm{b}}(t);t)]^2$. This is the expression we will use in the rest of this paper. For the sake of simplicity, in the following we will omit the constant term $\frac{\mathbb{P}[x^*]}{2s}$ which is present in all the expressions of the added error, and will denote $\epsilon_i(x;t) - \epsilon_j(x;t)$ as $\epsilon(x;t)$. The added error can thus be rewritten as $e_{\mathrm{add}}(t) = \epsilon^2(x_{\mathrm{b}}(t);t)$, and its expected value with respect to training sets T is given by

$$e_{\mathrm{add}} = \mathbb{E} e_{\mathrm{add}}(T) = \mathbb{E}^2 \epsilon(x_{\mathrm{b}}(T);T) + \mathbb{V}\epsilon(x_{\mathrm{b}}(T);T) \ . \tag{1}$$

Note that the two summands in the rightmost-hand side of (1) correspond respectively to the bias and variance components of the added error defined above.

Consider now a linear combination of the estimates of the a posteriori probabilities provided by m classifiers, $f_k^{\text{ave}}(x;t_1,\ldots,t_m) = \frac{1}{m}\sum_{p=1}^m f_k^p(x;t_p) = \mathbb{P}[\omega_k|x] + \frac{1}{m}\sum_{p=1}^m \epsilon_k(x;t_p)$. Denoting with b^{ave} the corresponding estimated boundary, the added error is given by $e_{\text{add}}^{\text{ave}}(t_1,\ldots,t_m) = \left[\frac{1}{m}\sum_{p=1}^m \epsilon(x_{b^{\text{ave}}}(t_p);t_p)\right]^2$, and its expected value over $T_1,\ldots,T_m$ is:

$$e_{\text{add}}^{\text{ave}} = \mathbb{E}^2\left[\frac{1}{m}\sum_{p=1}^m \epsilon(x_{\text{b}}(T_p);T_p)\right] + \mathbb{V}\left[\frac{1}{m}\sum_{p=1}^m \epsilon(x_{\text{b}}(T_p);T_p)\right] \quad . \tag{2}$$

The main limits of this framework are due to its focus on the shift of ideal class boundaries as the only effect of estimation errors on the posteriors, disregarding other possible effects like failing to detect a boundary or creating one where there is none [13]. Despite this, this framework turned out to provide useful insights into the behaviour of linear combiners as well as guidelines for their design [8], [18], [19]. This motivated our interest in applying it to analyze bagging.

B. Application to Bagging

Bagging consists of combining individual classifiers (either by majority voting or by linear combination of their soft outputs) trained on bootstrap replicates of the original training set [3]. In the following we will apply the analytical framework described in Sect. II-A to evaluate the expected added error of an ensemble of linearly combined classifiers generated by bagging.

According to [3], when the linear combining rule is used the *ideal* bagging is defined as a classifier whose output $f_k^{\text{bag}}(x;t)$ is the expectation of the output of a classifier trained on a (random) bootstrap replicate of t:

$$f_k^{\text{bag}}(x;t) = \mathbb{E}f_k(x;t^{\text{B}}) = \mathbb{P}[\omega_k|x] + \mathbb{E}\epsilon_k(x;t^{\text{B}}) \quad . \tag{3}$$

Following the same steps that lead to (1), and using the same notation, it is easy to see that the added error $e_{\text{add}}^{\text{bag}}(t)$ of this classifier is given by $\mathbb{E}^2\epsilon(x_{\text{b}}(t^{\text{B}}),t^{\text{B}})$. The expected added error over training sets T is therefore given by

$$\mathbb{E}_T\left[\mathbb{E}^2_{t^{\text{B}}|T=t}\epsilon(x_{\text{b}}(t^{\text{B}});t^{\text{B}})\right] \quad , \tag{4}$$

where the subscripts make clear that the outer expectation is taken over training sets T, while the inner expectations are taken over bootstrap replicates of a *given* realization of T.

The "real" bagging is a finite approximation of (3), obtained by averaging over a finite number of bootstrap replicates. Consider first a given training set t and a single classifier trained on a given boostrap replicate t^{b}, with the corresponding added error $e_{\mathrm{add}}(t^{\mathrm{b}})$. From (1), the expected value of the added error over bootstrap replicates of the same t is

$$\mathbb{E}e_{\mathrm{add}}(t^{\mathrm{B}}) = \mathbb{E}^2\epsilon(x_{\mathrm{b}}(t^{\mathrm{B}});t^{\mathrm{B}}) + \mathbb{V}\epsilon(x_{\mathrm{b}}(t^{\mathrm{B}});t^{\mathrm{B}}) \ .$$

Taking now the expectation over training sets T, the expected added error is given by

$$e_{\mathrm{add}} = \mathbb{E}_T\left[\mathbb{E}^2_{t^{\mathrm{B}}|T=t}\epsilon(x_{\mathrm{b}}(t^{\mathrm{B}});t^{\mathrm{B}}) + \mathbb{V}_{t^{\mathrm{B}}|T=t}\epsilon(x_{\mathrm{b}}(t^{\mathrm{B}});t^{\mathrm{B}})\right] \ . \tag{5}$$

Consider now the linear combination of m classifiers trained on bootstrap replicates of the same t, denoted as $t^{\mathrm{b}}_p, p = 1,\ldots,m$. From (2), the expected value of the added error over $t^{\mathrm{B}}_1,\ldots,t^{\mathrm{B}}_m$ is given by:

$$\mathbb{E}^2\frac{1}{m}\sum_{p=1}^{m}\epsilon(x_{\mathrm{b}}(t^{\mathrm{B}}_p);t^{\mathrm{B}}_p) + \mathbb{V}\frac{1}{m}\sum_{p=1}^{m}\epsilon(x_{\mathrm{b}}(t^{\mathrm{B}}_p);t^{\mathrm{B}}_p) \ .$$

The expected added error is obtained by taking the expectation of the above expression over T:

$$e^{\mathrm{ave}}_{\mathrm{add}} = \mathbb{E}_T\left[\mathbb{E}^2_{t^{\mathrm{B}}_1,\ldots,t^{\mathrm{B}}_m|T=t}\frac{1}{m}\sum_{p=1}^{m}\epsilon(x_{\mathrm{b}}(t^{\mathrm{B}}_p);t^{\mathrm{B}}_p) + \mathbb{V}_{t^{\mathrm{B}}_1,\ldots,t^{\mathrm{B}}_m|T=t}\frac{1}{m}\sum_{p=1}^{m}\epsilon(x_{\mathrm{b}}(t^{\mathrm{B}}_p);t^{\mathrm{B}}_p)\right] \ . \tag{6}$$

It turns out that it is possible to rewrite (6) as a function of the expected added error (5) of a single classifier trained on a bootstrap replicate of the original training set. To this aim, let us compute the expectations between square brackets in (6). Note first that the bootstrap replicates of a given set of samples t, $t^{\mathrm{B}}_p, p = 1,\ldots,m$, are by construction i.i.d. random variables: this implies that the estimation errors $\epsilon(x_{\mathrm{b}}(t^{\mathrm{B}}_p);t^{\mathrm{B}}_p), p = 1,\ldots,m$ are i.i.d. as well. The first term on the right-hand side of (6) is the bias of the expected added error, this now becomes:

$$\frac{1}{m^2}\sum_{p,q=1}^{m}\left[\mathbb{E}\epsilon(x_{\mathrm{b}}(t^{\mathrm{B}}_p);t^{\mathrm{B}}_p)\right]\left[\mathbb{E}\epsilon(x_{\mathrm{b}}(t^{\mathrm{B}}_q);t^{\mathrm{B}}_q)\right] = \mathbb{E}^2\epsilon(x_{\mathrm{b}}(t^{\mathrm{B}});t^{\mathrm{B}}) \ .$$

For the same reason above, it is easy to see that the second term (the variance of the expected added error) can be rewritten as

$$\frac{1}{m^2}\sum_{p=1}^{m}\mathbb{V}\epsilon(x_{\mathrm{b}}(t^{\mathrm{B}}_p);t^{\mathrm{B}}_p) = \frac{1}{m}\mathbb{V}\epsilon(x_{\mathrm{b}}(t^{\mathrm{B}});t^{\mathrm{B}}) \ . \tag{7}$$

It follows that (6) can be rewritten as:

$$e^{\mathrm{ave}}_{\mathrm{add}} = \mathbb{E}_T\left\{\mathbb{E}^2_{t^{\mathrm{B}}|T=t}\epsilon(x_{\mathrm{b}}(t^{\mathrm{B}});t^{\mathrm{B}}) + \frac{1}{m}\left[\mathbb{V}_{t^{\mathrm{B}}|T=t}\epsilon(x_{\mathrm{b}}(t^{\mathrm{B}});t^{\mathrm{B}})\right]\right\} \ . \tag{8}$$

What (8) says is that the expected added error of m bagged classifiers $e_{\text{add}}^{\text{ave}}$ is given by the same bias component of the expected added error e_{add} (5) of a single bootstrap replicate, plus $1/m$ times its variance component. In other words, as the ensemble size m increases, bagging expected added error decreases as $1/m$, and tends to an asymptotic value equal to the bias component of a single bootstrap replicate, $\mathbb{E}_T\left[\mathbb{E}^2_{t^{\text{B}}|T=t}\epsilon(x_{\text{b}}(t^{\text{B}});t^{\text{B}})\right]$ (by the way, the asymptotic value equals the expected added error of the ideal bagged classifier (4)). We point out that this result holds even for the case of bootstrap replicates drawn from a single training set t, i.e. without taking the expectation over T in (8).

In the next section we will discuss the scope and the implications of the above results on the open issues mentioned in Sect. I.

C. Discussion

The main theoretical result of Sect. II-B is an analytical expression of bagging misclassification probability as a function of the ensemble size, which to our knowledge is a novel result in the literature. Taking into account the limits of the theoretical framework from which this result has been derived (pointed out in Sect. II-A), it is interesting to experimentally investigate whether and to what extent the behaviour of the overall bagging misclassification rate agrees with the theoretical prediction given by (8) on real pattern recognition problems. This issue is of practical relevance as well, since (8) could be exploited to formulate guidelines for choosing bagging ensemble size: it is addressed in Sect. III.

As can be seen from (1) and (2), the considered framework leads to a particular bias-variance decomposition of the misclassification probability of a classifier (which turns out to be additive as in regression problems thanks to the assumptions and approximations on which the framework is based, see Sect. II-A). In particular, (8) implies that the expected added error of an ensemble of size m generated by bagging has the same bias component, and a variance component reduced by a factor $1/m$, with respect to a classifier trained on a single bootstrap replicate of the original training set. This *qualitatively* agrees with the fact that bagging is a variance reduction mechanism and is effective for unstable classifiers, as reported in previous works since [3]. Indeed, in the context of Tumer and Ghosh model the concept of classifier "instability" can be related to the variance of the estimated class boundary x_{b} (see Sect. II-A and Fig. 1): the higher the variance of x_{b}, the more unstable the classifier. Accordingly, the variance component of the expected

added error (1) of an unstable classifier will be high, as well as the one of a classifier trained on a bootstrap replicate of the original training set (5). Instead, the variance component of the error of $m > 1$ bagged classifiers will be smaller than that of a single bootstrap replicate, according to (8), and *can be expected* to be smaller than that of a classifier trained on the original training set (1) as well. We point out however that this last conclusion can not be formally derived from the results of Sect. II-B, since the bias-variance decomposition of bagging error given by (8) is related to the error of a single classifier trained not on the original training set (1), but on a bootstrap replicate of it (5). For the same reason, the results of Sect. II-B do not allow an analytical comparison between the bias component of bagging and that of a classifier trained on the original training set. An analytical comparison between the expected added error of bagging (8) and the one of a single classifier trained on the original training set (1) would be a very interesting contribution to the first open issue about bagging mentioned in Sect. I, but unfortunately it seems not possible without any assumption on the form of $\epsilon(x;t)$ and on the distribution $\mathbb{P}[X,Y]$, and even under assumptions of this kind it would be a quite complex task: for this reason, this line of research was out of the scope of this work and was left for future investigations. Morover, an experimental comparison between the bias and variance components of the corresponding error rates on real data sets is not possible either, since the decomposition resulting from Tumer and Ghosh framework (1), (2) refers to the expected value of estimation errors, which in real data sets is unknown, and to regions of the feature space around ideal class boundaries, which are unknown as well.

Consider now the fact that (8) implies that bagging never performs worse than a single classifier trained on a single bootstrap replicate of the original training set. We would like to stress that this result does not apply to a single classifier trained on the original training set, as explained above, and thus does not contradict previus works in which such an individual classifier was empirically found to outperform bagging. Morever, for the same reason (8) does not imply that the bias of bagging is identical to that of a single classifier trained on the original training set, and thus is not conflicting with previous empirical results which showed that bagging bias can be lower (for instance, [2], [20]). Instead, (8) is in agreement with empirical results showing that the bias of bagging does not depend on the ensemble size [20].

Let us now discuss the impact of data set size on bagging behaviour, in light of the fact that the theoretical model of Sect. II-A does not take into account the training set size explicitly. It

is known that bootstrap replicates drawn from a large data set are likely to be more similar to each other (and to the data set they are drawn from) than the ones drawn from a small data set. In other words, the "variance" of a classifier can be low, if a large training set is used. In this case bagging would be not effective: it could attain a small (or even negative) improvement over an individual classifier trained on the original training set. Our model does not take into account the training set size explicitly as pointed out above, and predicts that bagging error always decreases as the ensemble size increases. However this model is not conflicting with the possible behaviour of bagging on large data sets explained above, for two reasons. First, as explained above our model does not predict that bagging always outperforms an individual classifier trained on the original training set. Second, the reduction of bagging error predicted by our model as the ensemble size increases is equal to the variance component of the error corresponding to a single bootstrap replicate. If such variance component is small (as can happen on large data sets due to the similarity between bootstrap replicates), the predicted reduction of bagging error is small as well, but in principle this does not prevent such reduction to exhibit a $1/m$ rate according to (8).

We finally point out a byproduct of the results of Sect. II-B. According to [8], the expected added error of an ensemble of linearly combined classifiers is minimized by giving them identical weights, if their estimation errors are i.i.d. This gives a theoretical support to the common use of the simple average rule for combining classifiers generated by bagging, since their estimation errors satisfy by construction the i.i.d. condition (Sect. II-B). This is in agreement with the experimental results reported in [7], in which using the weighted average rule with bagging did not show any improvement over the simple average.

III. Experimental results

As explained in Sect. II-C, it is interesting to investigate whether the expression of bagging expected added error (8) as a function of the ensemble size m can be practically exploited to formulate guidelines on the choice of m in real classification problems. Expression (8) suggests to model the average error rate of m linearly combined classifiers generated by bagging, denoted in the following as $E(m)$, as the the asymptotic error E_∞ plus $1/m$ times the difference between E_1 and E_∞, being E_1 the average error rate corresponding to a single bootstrap replicate:

$$E_\infty + \frac{1}{m}[E_1 - E_\infty] \ , \tag{9}$$

where the average is ideally taken over all possible ensembles of m bootstrap replicates of a given training set, and then over all possible training sets, according to (8). The aim of the experiments reported below is to check whether and to what extent (9) approximates $E(m)$, for $m > 1$.

The experiments were carried out on twenty small data sets taken from the UCI Machine Learning repository (http://www.ics.uci. edu/∼mlearn/MLRepository.html): Breast cancer, Diabetes, Glass, Heart, Ionosphere, Soybean, Waveform, Balance scale, DNA, E-Coli, Haberman, Image, Iris, Liver disorders, Optdigits, Page blocks, Wine and Yeast. Decision trees and multilayer perceptron (one hidden layer) neural networks were used as base classifiers, since these are well known unstable classifiers [3]. Note that decision trees and the first seven data sets mentioned above were used in the early paper by Breiman [3]: this allows a comparison with results reported in that work. We also used the Forest CoverType data set (taken from the UCI Knowledge Discovery in Databases Archive, http://kdd.ics.uci.edu), which is made up of about $500,000$ patterns, to assess how well our model fits bagging behaviour on large data sets.

The experiments were carried out as follows.

- Each data set D was subdivided into a training set L and a testing set T. The DNA, Optdigits and Image data sets were originally subdivided into a training and a testing set: we used the original test set as T, while L was constructed by randomly drawing the 80% of patterns from the training set. For all the other data sets L was obtained by randomly drawing the 80% of the patterns from D, while the remaining patterns were used as the testing set T (execept for the large data set Forest CoverType, for which the size of L and T was set respectively to 20% and 80%). The number of hidden units for neural network classifiers was chosen on L.
- We considered values of bagging ensemble size up to 50, since $E(50)$ turned out to be a good approximation of the asymptotic error E_∞ for all the considered data sets. We drew 50 bootstrap replicates from L, trained a base classifier on each replicate, and computed the bagging error rate obtained by linearly combining the first m replicates, $m = 1, \ldots, 50$. This process was repeated for ten times (three for Forest CoverType), and the average bagging misclassification rate was computed for each m value: this approximates the inner expectation in (8) over different sequences of bootstrap replicates drawn from a given training set. For completeness, we also computed on T the misclassification rate of an

individual classifier trained on the original training set L.

- All the above process was repeated ten times (three for Forest CoverType), using thus different training sets L, and the average bagging error rate resulting from each run was further averaged over these runs, obtaining the values $E(1), E(2), \ldots, E(50)$. This approximates the outer expectation in (8) over different training sets. The average error rate of an individual classifier trained on the original training set L of each run was also computed.

The values of $E(m),\ m = 2, \ldots, 49$, were compared to the values predicted by (9), where E_1 and E_∞ were set equal respectively to $E(1)$ and $E(50)$. The results are reported in Figs. 2-4.

Figs. 2 and 3 show that in most small data sets bagging error follows quite well the theoretical behaviour predicted by (9), especially when neural networks were used as base classifier. For instance, using decision trees (Fig. 2), for $m > 5$ the absolute difference between the observed values and the predicted ones was below 0.01 in fourteen out of twenty data sets (the exceptions are E-Coli, German, Liver disorders, Page blocks, Waveform and Yeast), while for $m > 10$ it was below 0.01 in all data sets. We point out that our results are in agreement with the ones reported by Breiman in [3]: quoting from Breiman's paper, "most of the improvement [is attained] using only 10 bootstrap replicates". In view of a practical guideline on the choice of the ensemble size, let us focus on the relative reduction of bagging error with respect to the difference $E_1 - E_\infty$, being $1/m$ the one predicted by (9). Despite bagging error exhibits deviations from (9) on most data sets, for practical purposes the observed reduction is quite close to the predicted one. For instance, referring again to decision trees (Fig. 2), if we consider $m = 10$, for which the predicted error reduction is 90%, the observed reduction attained in our experiments was between 75% and 108%. Similar results were obtained with neural networks as base classifier.

There are however two data sets, Haberman and Iris, in which the observed reduction significantly differ from the predicted one for several m values, when decision trees were used as base classifier (Fig. 2). To a lesser extent, this is also the case of Haberman with neural networks as base classifier. In particular, in these data sets the deviations of bagging error $E(m)$ from (9) lead to a minimum of $E(m)$ for a small m (respectively 6 and 7), clearly violating the predicted decreasing behaviour of $E(m)$ for increasing m. This deserves a further explanation. These failures of our theoretical model of bagging error, as well as the deviations from (9) observed in other data sets, are clearly due to the violation of some of the assumptions on which our model is based, described in Sect. II-A. Unfortunately it is nearly impossible to check if these

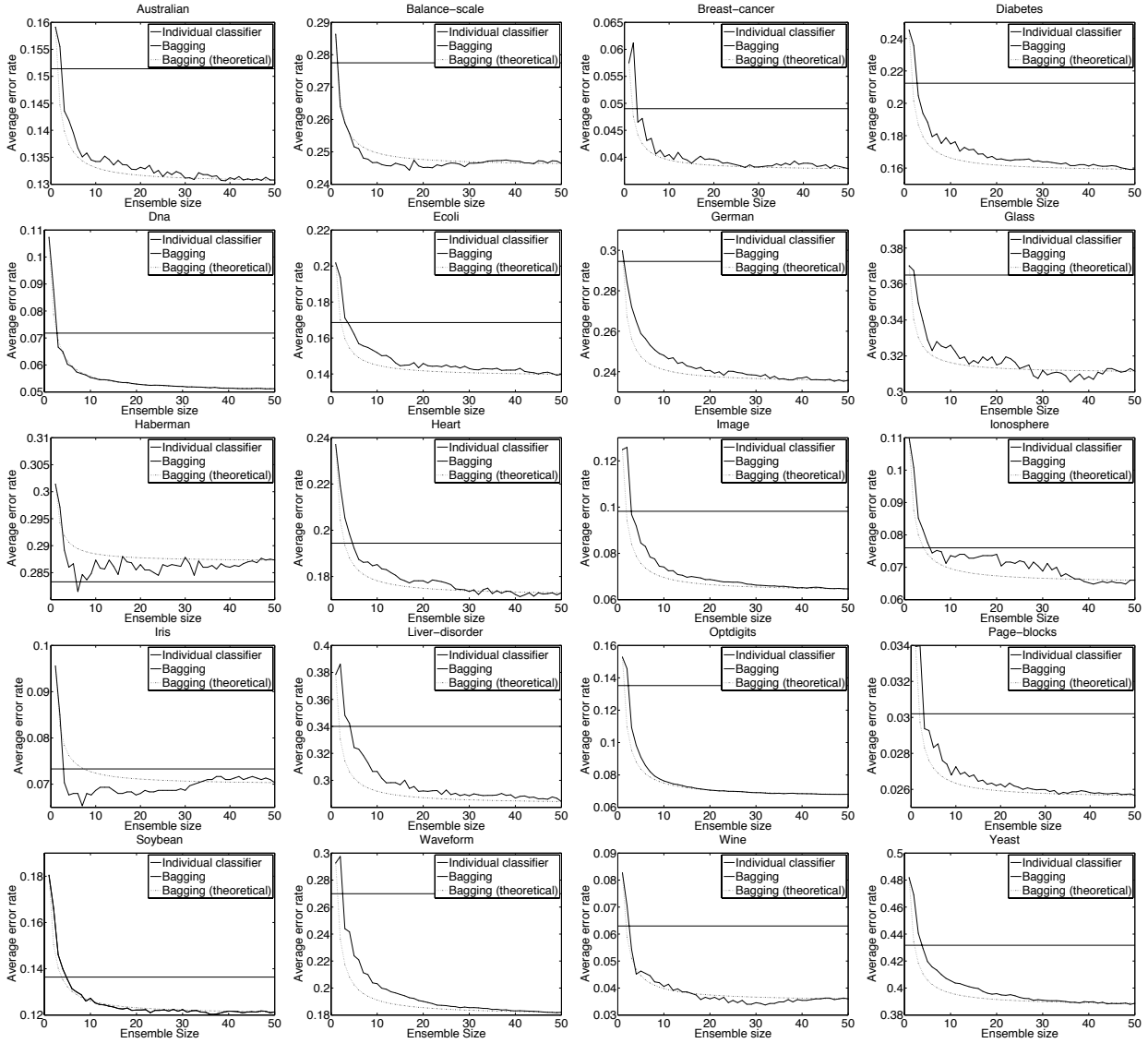

Fig. 2. Small data sets: test set misclassification rate of bagging as a function of the ensemble size, compared to the one predicted by (9) and the one of an individual classifier trained on the original training set. Decision trees were used as base classifier. The ensemble size value of 1 refers to an individual classifier trained on a bootstrap replicate of the original training set.

assumptions hold in real data sets, as well as to analytically assess to what extent bagging error will differ from the predicted one, when these assumptions do not hold. Nevertheless, from our experimental results it is at least possible to argue that the accuracy of our model is related to the difference $E_1 - E_\infty$ between the average error of an individual classifier trained on a bootstrap replicate of the original training set and the asymptotic bagging error. First, we point

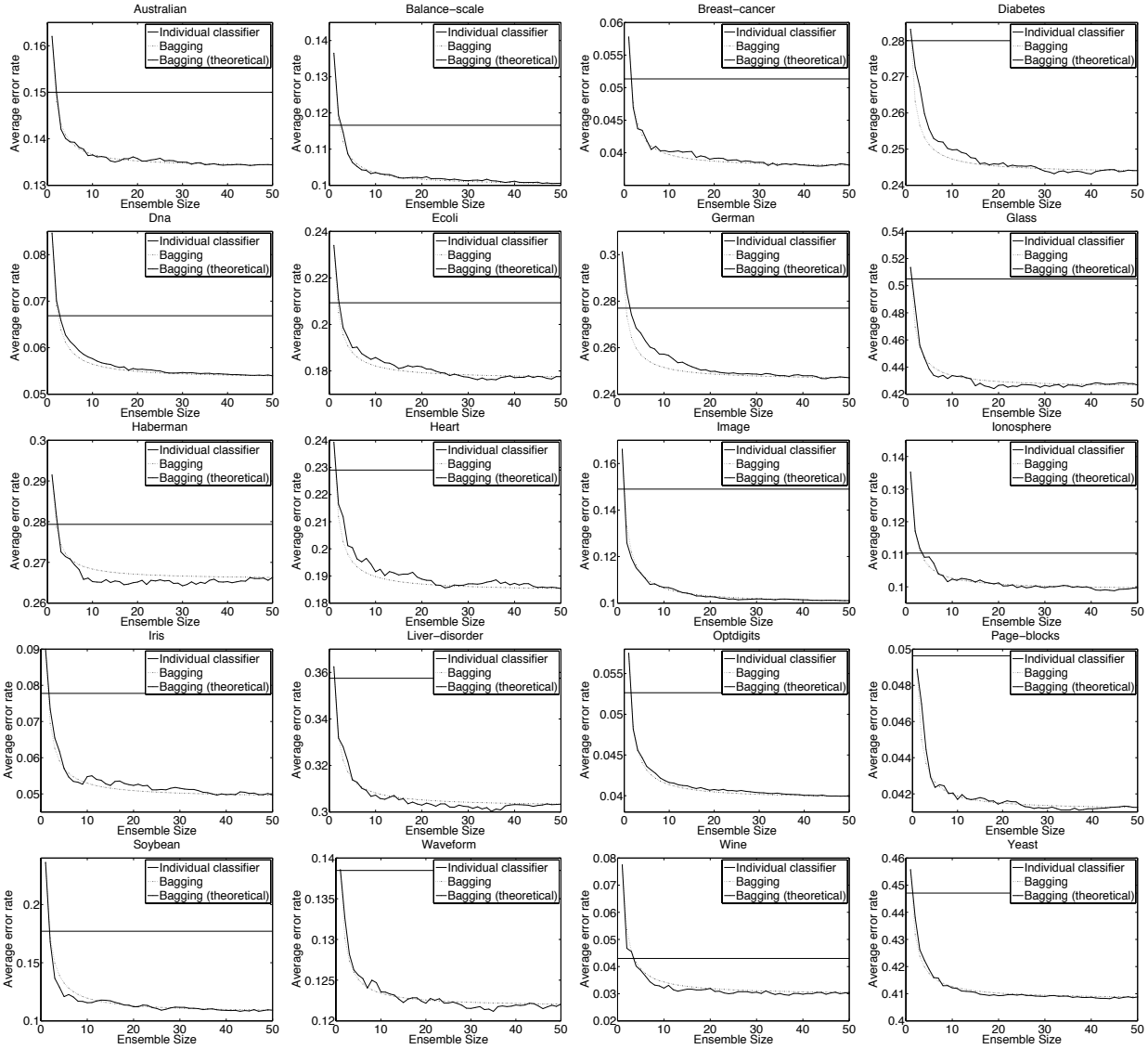

Fig. 3. Small data sets: results obtained with neural networks as base classifier (see caption of Fig. 2 for the details).

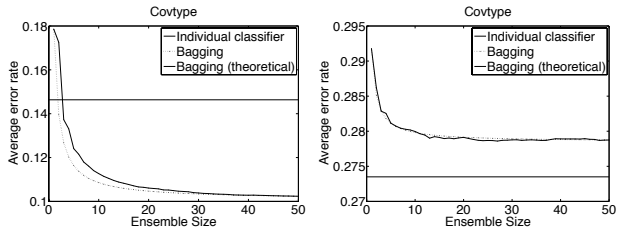

Fig. 4. Large data set. Base classifier: decision trees (left), neural networks (right). See caption of Fig. 2 for the details.

out that despite in some data sets like Glass and Ionosphere the *absolute* deviations between the observed bagging error and the predicted one are similar or even higher than in Haberman and Iris (see Fig. 2), the accuracy of our model in predicting the relative error reduction with respect to $E_1 - E_\infty$ is better. Consider now that, among all the considered small data sets, Haberman and Iris are characterized by the smallest values of $E_1 - E_\infty$, respectively about 0.015 and 0.025, with the only exception of Breast cancer, for which $E_1 - E_\infty$ is about 0.020 (we refer again to decision trees as base classifier). For instance, for Glass and Ionosphere $E_1 - E_\infty$ is respectively about 0.060 and 0.045. Accordingly, we argue that the accuracy of our model in predicting the relative error reduction attained by bagging as a function of the ensemble size can be lower when the difference $E_1 - E_\infty$ is lower. This is however not a general trend, since our model can work well even when $E_1 - E_\infty$ is small, as shown by the results on Breast cancer (decision trees) and on other data sets as Optdigits and Waveform (neural networks).

Consider finally the results obtained on the Forest CoverType data set (Fig. 4). When neural networks were used, bagging was outperformed (although slightly) by an individual classifier trained on the original training set. Moreover, the reduction of the error rate attained by bagging over a single bootstrap replicate was relatively small with respect to other data sets and to the overall error rate. This is a likely behaviour on a large data set (see the discussion in Sect. II-C). When decision trees were used, bagging behaviour was instead similar to the one exhibited on small data sets: the improvements over an individual classifier and over a single bootstrap replicate were higher than the ones attained using decision trees. Anyway, in both cases bagging error exhibited a decreasing behaviour as a function of the ensemble size, which turned out to be well approximated by (9).

To sum up, the above results provide evidence that the theoretical model of bagging error as a function of the ensemble size, given by (9), allows to accurately approximate, for practical purposes, the relative error reduction attained by bagging with respect to $E_1 - E_\infty$, for a given ensemble size (with the possible limits discussed above). This allows to derive the following novel guideline for the choice of bagging ensemble size: by combining m bagged classifiers, one can expect to reach on average a fraction of $(m-1)/m$ of the overall error reduction which can be attained by bagging from E_1 to E_∞. This is an improvement with respect to previous empirical guidelines, which simlpy suggested that choosing m between 10 and 20 is sufficient to approach the asymptotic bagging error (see for instance [3], [11], [16]). In particular, our guideline appears

to be very useful for applications characterized by strict requirements on memory size and CPU time, which require to find a trade-off between the attainable classification accuracy and the ensemble size.

Finally, it is worth noting that we repeated the above experiments using the majority voting rule, which is commonly used with bagging as well as the linear combination rule [2], [3], [6]. Interestingly, we found that also the behaviour of majority voting follows quite well the one predicted by (9) (we do not report the corresponding graphs due to limited space). This suggests that the above guideline on the choice of bagging ensemble size can be extended to ensembles combined by majority voting: this is relevant for applications in which the chosen base classifier provides only class labels, but not estimates of the a posteriori probabilities.

IV. Discussion and Conclusions

We applied an analytical framework for linear combiners developed in [18], [19], and extended in [8], to the particular case of linearly combined classifiers generated by bagging. This provided two main contributions. First, an analytical model of bagging expected added error as a function of the ensemble size. Second, based on such model, a practical guideline on the choice of bagging ensemble size which is an advance with respect to emprical guidelines proposed in the literature. We also showed that our theoretical results support the optimality of the simple average combining rule for classifier ensembles generated by bagging.

The analytical model derived in this work, and the corresponding guideline for choosing bagging ensemble size, hold for the case in which the classifier ensemble is obtained from a random sequence of bootstrap replicates of the original training set, which is the standard procedure followed in bagging, and refer to the average bagging error (over random sequences of bootstrap replicates). These results do not exclude the possibility that the misclassification rate can be further improved, being equal the ensemble size, by appropriately *selecting* a sequence of bootstrap replicates. Selection methods for bagging have indeed already been proposed, for instance in [1], [14]–[16], and their effectiveness was empirically shown. We point out however that selection methods proposed so far are not based on theoretical results, and in particular do not allow to predict the attainable performance improvement over standard bagging.

We finally point out that the theoretical results of Sect. II-B can be extended to all randomization-based methods for the construction of classifier ensembles, provided that such methods can be

modelled as follows (we use the notation in [4]): for the p-th individual classifier, a random variable Θ_p is generated, independent of the previous variables $\Theta_1, \ldots, \Theta_{p-1}$ but with the same distribution; then an individual classifier $h(x; \Theta_p)$ is constructed. Besides bagging, where Θ_p corresponds to the training set of the p-th individual classifier (obtained as a bootstrap replicate of a given training set), examples of this methods are the random subspace method [12], in which feature subsets are randomly selected out of the original feature set for each individual classifier, tree randomization [6], in which the attribute corresponding to each node of a decision tree is randomly selected among the best k ones, and random forests [4], in which a combination of the above strategies can be used (for instance, training sets can be generated as in bagging and attribute selection at each node of a decision tree can be randomized). With regard to the analytical model of Sect. II-A, the key point is that the estimation errors $\epsilon_k(x; \Theta_p)$ on the a posteriori probabilities of the different classifiers obtained using such methods are functions of the i.i.d. variables Θ_p, and thus they are i.i.d. themselves. This implies that all the theoretical results of Sect. II-B, and in particular the relationship (8) between the expected added error of the ensemble and that of an individual classifier obtained with the same procedure (where the inner expectations are to be intended in the general case with respect to random variables Θ_p) hold also for these ensemble construction methods. Experiments carried out with the random subspace method, analogous to the ones in Sect. III (not reported here due to limited space) actually showed that also for this method the average misclassification rate follows the behaviour predicted by (9). Our results give therefore a unifying view of such randomization-based ensemble construction methods, suggesting that the main difference among them is the amount of the bias component in the expected misclassification probability of *individual* classifiers generated by these methods: the lower the bias component, the lower the asymptotic expected misclassification probability of a classifier ensemble, independently on the variance component which can be arbitrarily reduced towards zero by increasing the ensemble size. An immediate practical implication is that the guideline for choosing the ensemble size derived in Sect. III can be extended to all such ensemble construction methods.

As explained in Sect. II-C, although our theoretical results are consistent with previous explanations of bagging in terms of classifier instability and of variance reduction, they can not be applied in a straightforward way to compare the asymptotic expected added error of bagging (4) with the one of a single classifier trained on the original training set (1). This is

however a very interesting direction for future developments of this work, since it could allow to improve the understanding of the conditions under which bagging outperforms an individual classifier, which is one of the main open issues mentioned in Sect. I. This could even give analogous insights on the other ensemble construction methods to which our theoretical results apply.

ACKNOWLEDGMENT

The authors are indebted to Gavin Brown for his valuable comments and suggestions on an earlier version of this paper. We would like to thank Battista Biggio for carrying out some of the experiments. We also thank the reviewers for their constructive remarks.

REFERENCES

[1] R. E. Banfield, L.O. Hall and K.W. Bowyer, "A comparison of Decision Tree Ensemble Creation techniques," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 29, pp. 173-180, 2007.

[2] E. Bauer and R. Kohavi, "An empirical comparison of voting classification algorithms: bagging, voting, and variants," *Machine Learning,* vol. 36, pp. 105-139, 1999.

[3] L.Breiman, "Bagging Predictors," *Machine Learning,* vol. 24, pp. 123-140, 1996.

[4] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, pp. 5-32, 2001.

[5] N. Chawla, T.E. Moore Jr. and K.W. Bowyer, "Bagging is a small-data-set phenomenon," *Proc. Int'l Conf. Computer Vision and Pattern Recognition,* pp. 684-689, 2001.

[6] T.G. Dietterich, "An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting and randomization," *Machine Learning,* vol. 40, pp. 1-22, 1999.

[7] P. Domingos, "Why does bagging work? A Bayesian account and its implications," *Proc. 3rd Int. Conf. Knowledge Discovery and Data Mining,* pp. 155-158, 1997.

[8] G. Fumera and F. Roli, "A Theoretical and Experimental Analysis of Linear Combiners for Multiple Classifier Systems," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 27, pp. 942-956, 2005.

[9] G. Fumera, F. Roli and A. Serrau, "Dynamics of Variance Reduction in Bagging and Other Techniques Based on Randomisation," *Proc. Int'l Workshop Multiple Classifier Systems,* Springer LNCS vol. 3541, pp. 316-325, 2005.

[10] Y. Grandvalet, "Bagging Equalizes Influence," *Machine Learning,* vol. 55, pp. 251-270, 2004.

[11] S. Guenter and H. Bunke, "Multiple classifier systems in offline handwritten word recognition - on the influence of training set and vocabulary size," *Int'l J. Patt. Rec. Artif. Int.*, vol. 18, pp. 1303-1320, 2004.

[12] T.K. Ho, "The random subspace method for constructing decision forests," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, pp. 832-844, 1998.

[13] L.I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms,* Hoboken, N.J.: Wiley, 2004.

[14] P. Latinne, O. Debeir and C. Decaestecker, "Limiting the number of trees in random forests," *Proc. Int'l Workshop Multiple Classifier Systems,* Springer LNCS vol. 2096, pp. 178-187, 2001.

[15] G. Martínez-Muñoz and Alberto Suárez, "Using boosting to prune bagging ensembles," *Pattern Recognition Letters*, vol. 28, no. 1, pp. 156-165, Jan. 2007.

[16] M. Skurichina, R.P.W. Duin, "Bagging for linear classifiers," *Pattern Recognition*, vol. 31, pp. 909-930, 1998.

[17] K. Tumer, "Linear and order statistics combiners for reliable pattern classification," PhD dissertation, The University of Texas, Austin, 1996.

[18] K. Tumer and J. Ghosh, "Analysis of Decision Boundaries in Linearly Combined Neural Classifiers," *Pattern Recognition*, vol. 29, pp. 341-348, 1996.

[19] K. Tumer and J. Ghosh, "Linear and order statistics combiners for pattern classification," *Combining Artificial Neural Nets*, A.J.C. Sharkey, ed., pp. 127-155. London: Springer, 1999.

[20] G. Valentini, "An experimental bias-variance analysis of SVM ensembles based on resampling techniques," *IEEE Trans. Systems, Man and Cybernetics, Part B*, vol. 35, pp. 1252-1271, 2005