1 2

An experimental bias–variance analysis of SVM ensembles based on resampling techniques

Giorgio Valentini

Abstract

Recently bias–variance decomposition of error has been used as a tool to study the behavior of learning algorithms and to develop new ensemble methods well-suited to the bias–variance characteristics of base learners. We propose methods and procedures, based on Domingos unified bias–variance theory, to evaluate and quantitatively measure the bias–variance decomposition of error in ensembles of learning machines. We apply these methods to study and compare the bias–variance characteristics of single SVMs and ensembles of SVMs based on resampling techniques, and their relationships with the cardinality of the training samples. In particular we present an experimental bias–variance analysis of bagged and random aggregated ensembles of Support Vector Machines, in order to verify their theoretical variance reduction properties. The experimental bias–variance analysis quantitatively characterizes the relationships between bagging and random aggregating, and explains the reasons why ensembles built on small subsamples of the data work with large databases. Our analysis also suggests new directions for research to improve on classical bagging.

Index Terms

Ensemble of learning machines, bias–variance analysis, Support Vector Machines, bagging.

I. Introduction

Ensemble methods represent one of the main current research lines in machine learning [1]–[3]. Several theories have been proposed to explain their behavior and characteristics. For instance, Allwein et al. interpreted the improved generalization capabilities of ensembles of learning machines in the framework of large margin classifiers [4], Kleinberg in the context of Stochastic Discrimination Theory [5], and Breiman and Friedman in the light of the bias–variance analysis adopted from classical statistics [6], [7].

Historically, the bias–variance insight was borrowed from the field of regression, using squared–loss as the loss function [8]. For classification problems, where the $0/1$ loss is the main criterion, several authors proposed bias–variance decompositions related to $0/1$ loss [9]–[12]. A few authors explicitly analyzed ensemble methods in the

G. Valentini is with the DSI - Dipartimento di Scienze dell’Informazione, Università degli Studi di Milano, Italy.
e-mail: valentini@dsi.unimi.it

framework of bias–variance tradeoff, considering the correlation between the base learners as one of the main factors affecting their effectiveness [13], [14]. In particular Bauer and Kohavi and Zhou, Wu and Tang analyzed bias–variance decomposition of error in bagging and other ensemble methods, using decision trees, Naive-Bayes, and neural networks as base learners [15], [16].

We recently applied bias–variance decomposition of error as a tool to study the behavior of learning algorithms and to develop new ensemble methods well-suited to the bias–variance characteristics of base learners [17], [18]. In particular we characterized bias and variance in Support Vector Machines (SVMs) [19] with respect to the kernel and its parameters in order to gain insight into the way SVMs learn from data and to study if and in which conditions we may develop SVM-based ensemble methods [20].

Indeed it is an open question if SVMs may be suitable base learners for ensemble methods: some authors claim that, because the SVMs directly implement the structural risk minimization principle [21], there is nothing to be gained by combining them; on the other hand, several results show the effectiveness of the ensembles of SVMs [22]–[24]. To help unravel this question, in this paper we extend the bias–variance analysis previously performed on single SVMs [20] to ensembles of SVMs based on resampling techniques.

In particular our aim consists of characterizing the bias–variance decomposition of error in terms of the kernel parameters of the base learners in bagged and random aggregated (RA) ensembles of SVMs. In this way we can *quantitatively* verify the theoretical results obtained by Breiman [25] about the variance reduction properties of random aggregating, and we may also understand the extent to which the results Breiman obtained for random aggregating can be extended to bagging, when SVMs are used as base learners. Indeed Breiman showed that random aggregating and bagging may be effective only if unstable predictors are used. Moreover he also showed that in regression problems, the aggregation of predictors cannot worsen the performance of single predictors, while in classification problems, if most of the predictions performed by the base classifier are non-optimal (in Bayes' sense), aggregation may also degrade the performance of the resulting ensemble [25].

To this end we present general methods and procedures, based on Domingos' unified bias–variance theory [11], to estimate the bias–variance decomposition of error in ensembles of SVMs. The proposed procedures are quite general and can be employed to analyze the bias–variance characteristics of other ensembles with different base learners.

Ensemble methods based on resampling techniques have been successfully applied to large data-mining problems [26]. Our bias–variance analysis of random aggregated ensembles may also yield insight into the reasons why voting by many classifiers built on small subsets of data, such as Breiman's "Pasting Small Votes" ensembles [27] and their distributed counterpart [26], [28] work with large databases.

Another issue raised by our experiments consists of understanding the reasons why Lobag [24], an ensemble method based on resampling and bias–variance analysis techniques, works at least when small samples are used.

The paper is structured as follows. In the following section the theoretical properties of random aggregating are summarized and bagging is introduced as an approximation of random aggregating. Sect. III provides an outline of the bias–variance decomposition of error for the $0/1$ loss function, according to Domingos'theory, and methods to measure bias and variance in ensembles of learning machines are presented. Sect. IV presents the results of an extended experimental analysis of bias–variance decomposition of error in bagged and random aggregated ensembles, using SVMs as base learners. Relationships between bias, variance and cardinality of the training samples are also studied, and some insights into the role of noise in bias–variance decomposition of error are provided. We then discuss the results, comparing the bias–variance decomposition in single, bagged and random-aggregated ensembles of SVMs and we consider some possible reasons why, on the one hand, ensemble methods built on small samples work with large databases, and why,on the other hand, Lobag ensembles work when small sample are available.

II. Random aggregating and bagging

In this section we summarize Breiman's main theoretical results for bagging and random aggregating [25], emphasising their variance reduction properties.

Let D be a set of m points drawn identically and independently from U according to P, where U is a population of labeled training data points $(\mathbf{x}_j, t_j)$, and $P(\mathbf{x}, t)$ is the joint distribution of the data points in U, with $\mathbf{x} \in \mathbb{R}^d$.

Let $\mathcal{L}$ be a learning algorithm, and define $f_D = \mathcal{L}(D)$ as the predictor produced by $\mathcal{L}$ applied to a training set D. The model produces a prediction $f_D(\mathbf{x}) = y$. Suppose that a sequence of learning sets $\{D_k\}$ is given, each i.i.d. from the same underlying distribution P. Breiman proposed aggregating the f_D trained with different samples drawn from U to get a better predictor $f_A(\mathbf{x}, P)$ [25]. For regression problems, $t_j \in \mathbb{R}$ and $f_A(\mathbf{x}, P) = E_D[f_D(\mathbf{x})]$ where $E_D[\cdot]$ indicates the expected value with respect to the distribution of D, while, in classification problems, $t_j \in S \subset \mathbb{N}$ and $f_A(\mathbf{x}, P) = \arg\max_j |\{k | f_{D_k}(\mathbf{x}) = j\}|$.

Because the training sets D are randomly drawn from U, we name the procedure to build f_A *random aggregating*. In order to simplify the notation, we denote $f_A(\mathbf{x}, P)$ as $f_A(\mathbf{x})$.

A. Random aggregating

If $\mathbf{X}$ and T are random variables having joint distribution P and representing values of the labeled data points $(\mathbf{x}, t)$, the expected squared loss EL for a single predictor $f_D(\mathbf{X})$ trained on a data set D is:

$$EL = E_D[E_{T,\mathbf{X}}[(T - f_D(\mathbf{X}))^2]] \tag{1}$$

where $E_{T,\mathbf{X}}[\cdot]$ indicates the expected value with respect to the distribution of T and $\mathbf{X}$.

The expected squared loss EL_A for the aggregated predictor is:

$$EL_A = E_{T,\mathbf{X}}[(T - f_A(\mathbf{X}))^2] \tag{2}$$

Developing the square in eq. 1 we have:

$$\begin{aligned} EL &= E_D[E_{T,\mathbf{X}}[T^2 + f_D^2(\mathbf{X}) - 2Tf_D(\mathbf{X})]] \\ &= E_T[T^2] + E_D[E_{\mathbf{X}}[f_D^2(\mathbf{X})]] - 2E_T[T]E_D[E_{\mathbf{X}}[f_D(\mathbf{X})]] \\ &= E_T[T^2] + E_{\mathbf{X}}[E_D[f_D^2(\mathbf{X})]] - 2E_T[T]E_{\mathbf{X}}[f_A(\mathbf{X})] \end{aligned} \tag{3}$$

In a similar way, developing the square in eq. 2 we have:

$$\begin{aligned} EL_A &= E_{T,\mathbf{X}}[T^2 + f_A^2(\mathbf{X}) - 2Tf_A(\mathbf{X})] \\ &= E_T[T^2] + E_{\mathbf{X}}[f_A^2(\mathbf{X})] - 2E_T[T]E_{\mathbf{X}}[f_A(\mathbf{X})] \\ &= E_T[T^2] + E_{\mathbf{X}}[E_D[f_D(\mathbf{X})]^2] - 2E_T[T]E_{\mathbf{X}}[f_A(\mathbf{X})] \qquad (4) \end{aligned}$$

Let be $Z = E_D[f_D(\mathbf{X})]$. Using $E[Z^2] \geq E[Z]^2$, considering eq. 3 and 4 we have that $E_D[f_D^2(\mathbf{X})] \geq E_D[f_D(\mathbf{X})]^2$ and hence $EL \geq EL_A$.

The reduction of error in randomly aggregated ensembles depends on how much different the two terms $E_{\mathbf{X}}[E_D[f_D^2(\mathbf{X})]]$ and $E_{\mathbf{X}}[E_D[f_D(\mathbf{X})]^2]$ of eq. 3 and 4. As outlined by Breiman, the effect of instability is clear: if $f_D(\mathbf{X})$ does not change too much with replicated data sets D, the two terms will be nearly equal and aggregation will not help. The more $f_D(\mathbf{X})$ varies, the greater the improvement that aggregation may produce.

In other words, the reduction of error depends on the instability of the prediction, i.e. on how unequal the two sides of eq. 5 are:

$$E_D[f_D(\mathbf{X})]^2 \leq E_D[f_D^2(\mathbf{X})] \qquad (5)$$

There is a strict relationship between the instability and the variance of the base predictor. Indeed the variance $V(\mathbf{X})$ of the base predictor is:

$$\begin{aligned} V(\mathbf{X}) &= E_D[(f_D(\mathbf{X}) - E_D[f_D(\mathbf{X})])^2] \\ &= E_D[f_D^2(\mathbf{X}) + E_D[f_D(\mathbf{X})]^2 - 2f_D(\mathbf{X})E_D[f_D(\mathbf{X})]] \\ &= E_D[f_D^2(\mathbf{X})] - E_D[f_D(\mathbf{X})]^2 \qquad (6) \end{aligned}$$

Comparing eq.5 and 6 we see that higher the instability of the base classifiers, the higher their variance is. The reduction of error in random aggregation is due to the reduction of the variance component (eq. 6) of error, because $V(\mathbf{X})$ will be strictly positive if and only if $E_D[f_D^2(\mathbf{X})] > E_D[f_D(\mathbf{X})]^2$, i.e. if and only if the base predictor is unstable (eq. 5).

In classification problems we may also obtain error reduction, but only if we use classifiers that provide the optimal prediction (in the Bayes sense) for the majority of the input patterns. On the other hand, ensembling Bayesian optimal predictors for all the input patterns is meaningless, because it is not possible to enhance the optimal Bayes classifier, and of course in this case we would have no diversity in the ensemble.

Breiman showed that, unlike regression, aggregating classifiers can lower performance, whereas in regression, aggregating predictors can lead to better performances, as long as the base predictor is unstable [25], [29].

More precisely let $f_D(\mathbf{X})$ be a base classifier that predicts a class label $t \in \mathcal{C}, \mathcal{C} = \{1, \ldots, C\}$, and let $\mathbf{X}$ be a random variable as in previous regression case and T a random variable with values in $\mathcal{C}$.

Thus the probability $p(D)$ of correct classification for a fixed data set D, considering a non deterministic assignment for the labels of the class, is:

$$p(D) = P(f_D(\mathbf{X}) = T) = \sum_{j=1}^{C} P(f_D(\mathbf{X}) = j|T = j)P(T = j) \tag{7}$$

In order to make the probability of correct classification p independent from the choice of a specific learning set, we average for D:

$$\begin{aligned} p &= \sum_{j=1}^{C} E_D[P(f_D(\mathbf{X}) = j|T = j)]P(T = j) \\ &= \sum_{j=1}^{C} \int P(f_D(\mathbf{X}) = j|\mathbf{X} = \mathbf{x}, T = j)P(T = j|\mathbf{X} = \mathbf{x})P_{\mathbf{X}}(d\mathbf{x}) \end{aligned} \tag{8}$$

Bearing in mind that $f_A(\mathbf{X}) = \arg\max_i P_D(f_D(\mathbf{x}) = i)$, the probability of correct classification p_A for random aggregation is:

$$\begin{aligned} p_A &= \sum_{j=1}^{C} \int P(f_A(\mathbf{X}) = j|T = j)P(T = j|\mathbf{X} = \mathbf{x})P_{\mathbf{X}}(d\mathbf{x}) \\ &= \sum_{j=1}^{C} \int I(\arg\max_i [P_D(f_D(\mathbf{X}) = i] = j)P(T = j|\mathbf{X} = \mathbf{x})P_{\mathbf{X}}(d\mathbf{x}) \end{aligned} \tag{9}$$

where I is the indicator function.

We now split the patterns into a set O, corresponding to the optimal (Bayesian) predictions performed by the aggregated classifier, and into a set O', corresponding to non-optimal predictions. The set of optimally classified patterns O is:

$$O = \{\mathbf{x}| \arg\max_j P(T = j|\mathbf{X} = \mathbf{x}) = \arg\max_j P_D(f_D(\mathbf{x}) = j)\}$$

According to the proposed partition of the data we can split the probability p_A of correct classification for random aggregation into two terms:

$$p_A = \int_{\mathbf{x} \in O} \max_j P(T = j|\mathbf{X} = \mathbf{x})P_{\mathbf{X}}(d\mathbf{x}) + \int_{\mathbf{x} \in O'} \sum_{j=1}^{C} I(f_A(\mathbf{x}) = j)P(T = j|\mathbf{X} = \mathbf{x})P_{\mathbf{X}}(d\mathbf{x}) \tag{10}$$

If $\mathbf{x} \in O$ we have:

$$\arg\max_j P(T = j|\mathbf{X} = \mathbf{x}) = \arg\max_j P_D(f_D(\mathbf{x}) = j) \tag{11}$$

In this case, considering eq. 8 and 9:

$$\sum_{j=1}^{C} P(f_D(\mathbf{X}) = j|\mathbf{X} = \mathbf{x}, T = j)P(T = j|\mathbf{X} = \mathbf{x}) \leq \max_j P_D(f_D(\mathbf{x}) = j)$$

and hence $p_A \geq p$. On the contrary, if $\mathbf{x} \in O'$, eq. 11 does not hold, and it may occur that:

$$\sum_{j=1}^{C} I(f_A(\mathbf{x}) = j)P(T = j|\mathbf{X} = \mathbf{x}) < \sum_{j=1}^{C} P(f_D(\mathbf{X}) = j|T = j)P(T = j|\mathbf{X} = \mathbf{x})$$

As a consequence, if the set of optimally predicted patterns O is large, aggregation improves performances; on the contrary, if set O' is large, aggregation can worsen performances.

B. *Bagging as an approximation of random aggregating*

In most cases we deal only with data sets of limited size, and moreover we do not know the probability distribution underlying the data. In such cases, we could try to simulate random aggregation by bootstrap replicates of the data [30] and by aggregating the predictors trained on the bootstrapped data.

Bagging [25] shows the same limits as random aggregating: only if the base learners are unstable can we achieve error reduction for single base learners. Of course, if the base learner is near to the Bayes error we cannot expect improvements by bagging.

Bagging is an approximation of random aggregating, for at least two reasons.

First, bootstrap samples are not real data samples: they are drawn from a data set D, which in turn is sample of the population U. On the contrary f_A uses samples drawn directly from U.

Second, bootstrap samples are drawn from D through a uniform probability distribution that is only an approximation of the unknown true distribution P.

For these reasons we can only hope that this is a good enough approximation of f_A to result in substantial variance reduction (eq. 2) [31].

With bagging, each base learner, on the average, uses only 63.2% of the available data for training and so we can expect a larger bias for each base learner, since the effective size of the learning set is reduced. This may also affect the bias of the bagged ensemble, which critically depends on the bias of the component base learners: we may expect an increment of the bias of the bagged ensemble compared to the unaggregated predictor trained on the entire available training set.

Bagging is a variance-reduction method, but we cannot expect such large decreases of variance as in random aggregating. The intuitive reason consists of the fact that, in random aggregating, the base learners use more variable training sets drawn from U according to the distribution P. Random aggregating thus exploits more information from the population U, while bagging can exploit only the information from a single data set D drawn from U, through bootstrap replicates of the data from D.

Breiman showed that, in regression problems, random aggregating always reduces variance. But what about bagging in classification? Some authors experimentally showed that bagging reduces variance in classification problems, using decision trees or neural networks as base classifiers [15], [16]. Here we investigate whether this property also holds when SVMs are used as base learners. Moreover we study the relationships between bias–variance decomposition of error in bagged and random aggregated ensembles of SVM, in order to get a bias–variance interpretation of the reasons why voting many classifiers built on small subsets of data [26], [27] works successfully. To this purpose we need methods and procedures to quantitatively estimate the bias–variance decomposition of error in ensembles of learning machines.

III. Bias–variance decomposition of error in ensembles of learning machines

In this section we show how to measure bias and variance in ensemble methods, outlining also the main ideas behind Domingos' bias–variance decomposition of error with the 0/1 loss. For a more detailed introduction to

Domingos' bias-variance theory, see [11], [32].

A. Bias–Variance Decomposition in classification problems

For classification problems, where the $0/1$ loss is the main criterion, several authors proposed bias–variance decompositions related to $0/1$ loss [6], [9], [10], [33]–[35]. These decompositions have significant shortcomings: in particular they lose the relationship to the original squared loss decomposition [8], in most cases forcing bias and variance to be purely additive.

We consider classification problems and the $0/1$ loss function in the Domingos' unified framework of bias–variance decomposition of the error [11], [36]. According to this approach, bias and variance are defined for an arbitrary loss function, showing that the resulting decomposition specializes to the standard for squared loss, but it also holds for the $0/1$ loss.

Let $L(t,y)$ be the $0/1$ loss function, that is $L(t,y) = 0$ if $y = t$, and $L(t,y) = 1$ otherwise.

The expected loss EL of a learning algorithm $\mathcal{L}$ at point $\mathbf{x}$ can be written by considering both randomness due to the choice of the training set D and randomness in t due to the choice of a particular test point $(\mathbf{x}, t)$:

$$EL(\mathcal{L}, \mathbf{x}) = E_D[E_t[L(t, f_D(\mathbf{x}))]] \tag{12}$$

where $E_D[\cdot]$ and $E_t[\cdot]$ indicate the expected value with respect to the distribution of D, and to the distribution of t.

The purpose of bias-variance analysis consists of decomposing this expected loss into terms that separate the bias and the variance. To derive this decomposition, we need to define the *optimal prediction* and the *main prediction*: bias and variance can be defined in terms of these quantities.

The *optimal prediction* y_* for point $\mathbf{x}$ minimizes $E_t[L(t,y)]$:

$$y_*(\mathbf{x}) = \arg\min_y E_t[L(t,y)] \tag{13}$$

The noise $N(\mathbf{x})$, is defined in terms of the optimal prediction, and represents the remaining loss that cannot be eliminated, even by the optimal prediction:

$$N(\mathbf{x}) = E_t[L(t, y_*)]$$

The *main prediction* y_m at point $\mathbf{x}$ is defined as

$$y_m = \arg\min_{y'} E_D[L(f_D(\mathbf{x}), y')] \tag{14}$$

i.e., it is the label for $\mathbf{x}$ that the learning algorithm "wishes" were correct, or, in other words, it represents its systematic prediction. For 0/1 loss, the main prediction is the most predicted class.

The *bias* $B(\mathbf{x})$ is the loss of the main prediction compared to the optimal prediction:

$$B(\mathbf{x}) = L(y_*, y_m)$$

It represents the systematic error of the learning algorithm. For the 0/1 loss, the bias is always 0 or 1. We will say that $\mathcal{L}$ is *biased at point* $\mathbf{x}$, if $B(\mathbf{x}) = 1$.

The *variance* $V(\mathbf{x})$ is the average loss of the predictions relative to the main prediction:

$$V(\mathbf{x}) = E_D[L(y_m, f_D(\mathbf{x}))] \tag{15}$$

It captures the extent to which the various predictions $f_D(\mathbf{x})$ vary depending on D.

Domingos distinguishes between two opposite effects of variance on error: in the unbiased case variance increases the error, while in the biased case variance decreases error.

As a result we can define an *unbiased variance*, $V_u(\mathbf{x})$, which is the variance when $B(\mathbf{x}) = 0$ and a *biased variance*, $V_b(\mathbf{x})$, which is the variance when $B(\mathbf{x}) = 1$. Finally we can also define the *net variance* $V_n(\mathbf{x})$ to take into account the combined effect of unbiased and biased variance:

$$V_n(\mathbf{x}) = V_u(\mathbf{x}) - V_b(\mathbf{x})$$

If we can disregard noise, unbiased variance captures the extent to which the learner deviates from the correct prediction y_m (in the unbiased case $y_m = y_*$), while the biased variance captures the extents to which the learner deviates from the incorrect prediction y_m (in the biased case $y_m \neq y_*$).

From this standpoint, variance can be interpreted as a measure of diversity between classifiers trained with different data sets D. Moreover the effects of this kind of diversity on error depend on the type of the variance, as we need to distinguish when $\mathcal{L}$ is biased or unbiased at a specific point $\mathbf{x}$.

This decomposition for a single point $\mathbf{x}$ can be generalized to the entire population by defining $E_{\mathbf{x}}[\cdot]$ to be the expectation with respect to $P(\mathbf{x})$. Then we can define the *average bias* $E_{\mathbf{x}}[B(\mathbf{x})]$, the *average unbiased variance* $E_{\mathbf{x}}[V_u(\mathbf{x})]$, and the *average biased variance* $E_{\mathbf{x}}[V_b(\mathbf{x})]$. In the noise-free case, the expected loss over the entire population is

$$E_{\mathbf{x}}[EL(\mathcal{L}, \mathbf{x})] = E_{\mathbf{x}}[B(\mathbf{x})] + E_{\mathbf{x}}[V_u(\mathbf{x})] - E_{\mathbf{x}}[V_b(\mathbf{x})].$$

B. Measuring bias–variance decomposition of error in ensembles of learning machines

In this subsection we summarize how to estimate the decomposition of error in bias, net-variance, unbiased and biased variance with ensembles of learning machines.

To represent the samples used in our bias–variance decomposition procedures, we make use of the following notation:

- $\mathcal{D}$: the overall data set we used to train the ensembles
- D_i : a subsample drawn from the training set $\mathcal{D}$, $1 \leq i \leq n$.
- $\mathbf{S}_i$: a set of samples, $1 \leq i \leq n$
- D_{ij} : samples belonging to the set $\mathbf{S}_i$, $1 \leq j \leq m$
- $\mathcal{T}$: the test set to estimate the bias–variance decomposition of error

With bagging a single set of bootstrapped data is used to train the base learners. Here our aim is to estimate the bias–variance decomposition of error in bagged ensemble and this estimate has to be independent of the particular choice of the training set (see: eq. 12): hence we need multiple sets $\mathbf{S}_i$ of samples to train multiple ensembles with the same learning parameters. Each set is obtained by bootstrapping a training set D_i (as it is usual in bagging), but here we need multiple training sets to bootstrap multiple sets.

In our experimental set-up, n training sets D_i are obtained by subsampling with replacement according to the uniform probability distribution from a much larger available training set $\mathcal{D}$. In particular we subsampled n relatively small training sets D_i of size s to guarantee that there will not be too much overlap between different D_i. Finally from each D_i we bootstrapped a set $\mathbf{S}_i = \{D_{ij}\}_{j=1}^m$ of m samples that we used to train the base learners of the bagged ensemble.

For random aggregation the procedure is similar, but with the substantial difference that the set $\mathbf{S}_i = \{D_{ij}\}_{j=1}^m$ is directly drawn from $\mathcal{D}$, by subsampling with replacement according to the uniform probability distribution.

The estimate of the bias–variance decomposition of error is performed on a separate test set $\mathcal{T}$ not used to train the ensembles.

In the rest of this section we present the experimental procedures adopted in greater detail. We may distinguish between two main steps:

1) Generation of the data for ensemble training.
2) Bias-variance estimate on a separate test set.

The design and the implementation of the first step depends on the specific ensemble method to be evaluated. In the following section, we present procedures to generate training data for bagged and random aggregated ensembles. The second step uses the data sets previously generated to train the ensemble and to evaluate the bias-variance decomposition of error on a separate test set. This second step is not ensemble specific and can be applied unmodified to any ensemble of learning machines. Here we describe an approach (bias–variance estimate using a single and separate test set) feasible when a large test set is available. We can easily extend the bias–variance estimate to small test sets, using, for instance, bootstrap or cross-validation techniques.

1) Generating training data to estimate bias–variance decomposition in bagged and random aggregated ensembles: As a first step we need to generate the data to train the base learners. This step is different in random aggregating and bagging.

For *bagging* we draw with replacement from a learning set $\mathcal{D}$ n samples D_i of size s, according to the uniform probability distribution. Note that $|D_i| << |\mathcal{D}|$, as $\mathcal{D}$ represents the universe population from which training data D_i are drawn.

From each D_i, $1 \leq i \leq n$, we generate by bootstrap m replicates D_{ij}, collecting them in n different sets $\mathbf{S}_i = \{D_{ij}\}_{j=1}^m$. Such n sets will be used to train n ensembles composed by m base learners.

The experimental procedure to generate the training data for bagging from an available data set $\mathcal{D}$ is summarized in Fig. 1. The procedure `Generate_bootstrap_samples` (Fig. 1) generates sets $\mathbf{S}_i$ of bootstrapped samples, drawing, at first, a subsample D_i (of size s) from the training set $\mathcal{D}$ according to the uniform probability distribution (procedure `Draw_with_replacement`) and then drawing from D_i m bootstrap replicates (procedure `Bootstrap_replicate`).

For *random aggregating* we draw with replacement from $\mathcal{D}$ n sets of samples $\mathbf{S}_i$, according to the uniform

```
Procedure Generate_bootstrap_samples
Input arguments:
    - $\mathcal{D}$: Overall training set
    - $n$: number of sets of samples $\mathbf{S}$
    - $m$: number of bootstrap replicates for each set $\mathbf{S}$
    - $s$: size of each bootstrap replicate
Output:
    - Sets $\mathbf{S}_i = \{D_{ij}\}_{j=1}^{m},\ 1 \leq i \leq n$ of bootstrapped samples
begin procedure
    for $i = 1$ to $n$
    begin
        $D_i$ = Draw_with_replacement($\mathcal{D}, s$)
        $\mathbf{S}_i = \emptyset$
        for $j = 1$ to $m$
        begin
            $D_{ij}$ = Bootstrap_replicate($D_i$)
            $\mathbf{S}_i = \mathbf{S}_i \cup D_{ij}$
        end
    end
end procedure.
```

Fig. 1. Procedure to generate samples for bias–variance analysis in bagging

probability distribution. Each set of samples $\mathbf{S}_i$ is composed by m samples D_{ij} drawn with replacement from $\mathcal{D}$, using the uniform probability distribution. Each sample D_{ij} is composed by s examples, and the D_{ij} samples are collected in n sets $\mathbf{S}_i = \{D_{ij}\}_{j=1}^{m}$. Note that in this case each sample D_{ij} is directly drawn from $\mathcal{D}$ and not from the samples $D_i \subset \mathcal{D}$. Fig. 2 summarizes the experimental procedure we adopted to generate the data for random aggregating. Sets $\mathbf{S}_i$ of samples are drawn from $\mathcal{D}$ by the procedure `Draw_with_replacement`. This process is repeated n times, giving rise to n sets of samples that will be used to train n random aggregated ensembles, each composed by m base learners (procedure `Generate_samples`, Fig. 2).

Note that this is only an approximation of random aggregating. Indeed with random aggregating we should draw the samples from the universe of the data according to their unknown distribution P. Of course this is in general not possible (except for synthetic data), but in our experiments we used synthetic data or comfortably large data sets $\mathcal{D}$, setting the size of the samples D_{ij} to relatively small values, and using a uniform probability distribution instead of the unknown distribution P. From this standpoint, we approximated random aggregation by randomly drawing data from the universe population U represented by a comfortably large training set $\mathcal{D}$.

2) Estimate of bias–variance decomposition of error: In this step we use the n sets of samples $\mathbf{S}_i$ to train n ensembles, each composed by m learners, each one trained with different resampled data, and we repeat this process for all the considered ensemble models. In order to properly compare the effect of different choices of the learning parameters on bias–variance decomposition of error, each ensemble model is represented by a different choice of the learning parameters and is trained with the same sets of samples $\mathbf{S}_i,\ 1 \leq i \leq n$.

For each model, bias–variance decomposition of error is evaluated on a separate test set $\mathcal{T}$, significantly larger

```
Procedure Generate_samples
Input arguments:
    - $\mathcal{D}$: Overall training set
    - $n$: number of sets of samples $\mathbf{S}$
    - $m$: number of samples collected in each set $\mathbf{S}$
    - $s$: size of each sample
Output:
    - $\mathbf{S}_i = \{D_{ij}\}_{j=1}^{m},\ 1 \leq i \leq n$: sets of samples
begin procedure
    for $i = 1$ to $n$
    begin
        $\mathbf{S}_i = \emptyset$
        for $j = 1$ to $m$
        begin
            $D_{ij}$ = Draw_with_replacement($\mathcal{D}, s$)
            $\mathbf{S}_i = \mathbf{S}_i \cup D_{ij}$
        end
    end
end procedure.
```

Fig. 2. Procedure to generate samples for bias–variance analysis in random aggregation

```
Procedure Bias–Variance_analysis
Input arguments:
    - $n$: number of ensembles
    - $\{\mathbf{S}_i\}_{i=1}^{n}$: sets of samples
    - $\mathcal{T}$: test set
    - $\mathcal{A}$: set of learning parameters
Output:
    - $BV = \{bv(\alpha)\}_{\alpha \in \mathcal{A}}$: error, bias, net-variance, unbiased and biased variance
    of the bagged ensembles having base learners with learning parameters $\alpha \in \mathcal{A}$.
begin procedure
    for each $\alpha \in \mathcal{A}$
    begin
        Ensemble_Set($\alpha$) = $\emptyset$
        for $i = 1$ to $n$
        begin
            ensemble($\alpha,\ \mathbf{S}_i$) = Ensemble_train ($\alpha,\ \mathbf{S}_i$)
            Ensemble_Set($\alpha$) = Ensemble_Set($\alpha$) $\cup$ ensemble($\alpha,\ \mathbf{S}_i$)
        end
        $bv(\alpha)$ = Perform_BV_analysis(Ensemble_Set ($\alpha$), $\mathcal{T}$)
        $BV = BV \cup bv(\alpha)$
    end
end procedure.
```

Fig. 3. Procedure to perform bias–variance analysis in ensembles of learning machines

than the training sets, using the n ensembles trained with the n sets $\mathbf{S}_i$.

The experimental procedure to estimate the bias–variance decomposition of error is summarized in Fig. 3. In the procedure `Bias-Variance_analysis` (Fig. 3) different ensembles are trained (procedure `Ensemble_train`)

```
Procedure Perform_BV_analysis
Input:
    - $F(\alpha) = \{e_i\}_{i=1}^{n}$: set of $n$ ensembles $e_i$ trained with parameters $\alpha$
    - $\mathcal{T}$: test set
Output:
    - $bv(\alpha)$: estimate of the bias–variance decomposition with learning parameters $\alpha$:
    - $bv(\alpha).loss$ estimate of the loss
    - $bv(\alpha).bias$ estimate of the bias
    - $bv(\alpha).netvar$ estimate of the net variance
    - $bv(\alpha).var_u$ estimate of the unbiased variance
    - $bv(\alpha).var_b$ estimate of the biased variance

begin procedure
    for each $\mathbf{x} \in \mathcal{T}$
    begin
        $p_1(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} ||e_i(\mathbf{x}) = 1||$
        $p_{-1}(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} ||e_i(\mathbf{x}) = -1||$
        $y_m = \arg\max(p_1, p_{-1})$
        $B(\mathbf{x}) = \frac{y_m - t}{2}$
        $V_u(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} ||(B(\mathbf{x}) = 0) \text{ and } (y_m \neq e_i(\mathbf{x}))||$
        $V_b(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} ||(B(\mathbf{x}) = 1) \text{ and } (y_m \neq e_i(\mathbf{x}))||$
        $V_n(\mathbf{x}) = V_u(\mathbf{x}) - V_b(\mathbf{x})$
        $Err(\mathbf{x}) = B(\mathbf{x}) + V_n(\mathbf{x})$
    end
    $p = card(\mathcal{T})$
    $bv(\alpha).loss = \frac{1}{p}\sum_{\mathbf{x}\in\mathcal{T}} Err(\mathbf{x})$
    $bv(\alpha).bias = \frac{1}{p}\sum_{\mathbf{x}\in\mathcal{T}} B(\mathbf{x})$
    $bv(\alpha).netvar = \frac{1}{p}\sum_{\mathbf{x}\in\mathcal{T}} V_n(\mathbf{x})$
    $bv(\alpha).var_u = \frac{1}{p}\sum_{\mathbf{x}\in\mathcal{T}} V_u(\mathbf{x})$
    $bv(\alpha).var_b = \frac{1}{p}\sum_{\mathbf{x}\in\mathcal{T}} V_b(\mathbf{x})$
end procedure.
```

Fig. 4. Procedure to perform bias variance decomposition of error for ensembles of learning machines. $B(\mathbf{x})$, $V_u(\mathbf{x})$, $V_b(\mathbf{x})$, $V_n(\mathbf{x})$, $Err(\mathbf{x})$ are respectively the bias, the unbiased, biased and net variance, and the overall error. Note that $F(\alpha)$ is a set of ensembles e_i, $1 \leq i \leq n$, trained with the same learning parameter α.

using the same sets of samples generated through the procedure `Generate_samples` (random aggregating) or `Generate_bootstrap_samples` (bagging). Note that Ensemble_Set(α) represents a set of ensembles characterized by the same learning parameter α; ensemble(α, $\mathbf{S}_i$) is an ensemble trained with a specific set $\mathbf{S}_i$ of training samples. The learning parameter α depends on the choice of the base learners: for instance, with gaussian kernels it represents the C regularization parameter and the width σ of the gaussian function.

Bias–variance decomposition of error is performed on the separate test set $\mathcal{T}$ using the previously trained ensembles (procedure `Perform_BV_analysis`, Fig. 4).

The procedure `Perform_BV_analysis` provides an estimate of bias–variance decomposition of error for a given model. Note that in Fig. 4 the function $||z||$ is equal to 1 if z is true, and 0 otherwise.

For instance, in order to perform a bias–variance analysis with a bagged ensemble having a training set $\mathcal{D}$, a separate test set $\mathcal{T}$, using n sets of m bootstrapped samples of cardinality s, it is sufficient to call the two procedures

`Generate_bootstrap_samples` and `Bias-Variance_analysis`:

```
$\{\mathbf{S}_i\}_{i=1}^n = $ Generate_bootstrap_samples($\mathcal{D}, n, m, s$);
$BV = $ Bias-Variance_analysis($n, \{\mathbf{S}_i\}_{i=1}^n, \mathcal{T}, \mathcal{A}$);
```

With random aggregated ensembles the overall procedure is quite similar: the only difference consists of the way the initial resampling procedure is performed:

```
$\{\mathbf{S}_i\}_{i=1}^n = $ Generate_samples($\mathcal{D}, n, m, s$);
$BV = $ Bias-Variance_analysis($n, \{\mathbf{S}_i\}_{i=1}^n, \mathcal{T}, \mathcal{A}$);
```

IV. Experimental results

This section summarizes the results of an extended bias–variance analysis of ensembles of SVMs, using a set of two-class classification problems, while the discussion is postponed until the next section. Full experimental results and graphics are available on the web (see the appendix for details).

We performed the following experimental tasks:

- Bias–variance analysis of bagged and random aggregated (RA) ensembles of SVMs with respect to the kernel parameters
- Comparison between bias–variance characteristics of single, bagged and RA SVMs
- Check and quantitative evaluation of the variance reduction properties of RA and bagged ensembles of SVMs.
- Comparison of the bias–variance characteristics of single, bagged and RA SVMs, while varying the cardinality of the training data
- Comparison of single SVMs trained on large data sets with RA SVM ensembles trained on small samples
- Evaluation of the effect of noisy data on the bias–variance decomposition of error in bagged and RA ensembles of SVMs

We used linear, polynomial and gaussian SVMs as base learners. The bias–variance decomposition of error has been evaluated with respect to different settings of the kernel parameters: for gaussian kernels we selected a set of values of σ such that $\sigma \in [0.01, 100]$, for polynomial kernels we considered degrees between 2 and 10, and we selected the regularization parameter $C \in [0.1, 1000]$.

We also studied the relationships between the cardinality of the training samples and the bias–variance characteristics of single, bagged and RA SVMs with respect to different choices of kernel and regularization parameters of SVMs. We then considered the comparison of RA SVM ensembles trained on small samples with single SVMs trained on large data sets, to evaluate the loss/gain of the ensemble approach with respect to the accuracy and the computation time. Finally we analyzed the effect of noisy data into the bias–variance characteristics of bagged and RA ensembles of SVMs.

Considering all the data sets, we trained and tested more than 160000 different SVM ensembles and a total of more than 10 millions of single SVMs. To perform the experimental analysis we used a cluster of Linux workstations, and we developed new classes and specific C++ applications, extending the *NEURObjects* software library [37] [1].

[1] The extended version of the *NEURObjects* library is available at: `http://homes.dsi.unimi.it/∼valenti/sw/NEURObjects`

A. Data sets

In the experiments we employed 7 different data sets, both synthetic and “real”.

P2 is a synthetic bidimensional two–class data set; each region is delimited by one or more of four simple polynomial and trigonometric functions [2].

The synthetic data set *Waveform* is generated from a combination of 2 of 3 “base” waves; we reduced the original three classes of *Waveform* to two, deleting all samples pertaining to class 0. The other data sets are all from the UCI repository [38].

Table I summarizes the main features of the data sets used in the experiments. The first column refers to the cardinality of the overall training set (the data set $\mathcal{D}$ of Sect. III-B), from which training samples are subsampled. The second column refers to the cardinality of the separate test set, used to estimate the bias–variance decomposition of error. The last column represents the dimension of the input space.

For “real” data, we randomly drew the samples from the overall training set, while for synthetic data sets the data were generated through suitable computer programs. For the UCI data sets we randomly split all the available data in a training and a test set of about equal size, except for the *Grey-Landsat* data set for which we maintained the original size for both the training and test set.

To measure the bias–variance decomposition of error, for each data set we used 100 sets (parameter $n = 100$ in Sect. III-B), each set is composed by 100 samples (parameter $m = 100$ in Sect. III-B), and each sample is composed by 100 examples (parameter $s = 100$).

The choice of small-sized samples allows us to better evaluate the variance component of error and to obtain a quite large data diversity between the sets of samples: in such a way we may better simulate the variability of the data observed in real problems.

The relationships between cardinality of the training data and the bias–variance characteristics of the ensembles have been studied by considering different samples ranging from 25 to 3200 examples for each training set (see Sect. IV-D).

B. Bias–variance analysis in bagged SVM ensembles

We compared bias–variance decomposition in single SVMs and bagged ensembles of SVMs. In the figures of this section, the results referred to single SVMs are labeled with crosses, while bagged SVMs are labeled with triangles. The analyzed quantities (e.g. bias, net-variance, unbiased and biased variance) are represented with the same type of line both in single and bagged SVMs. Full experimental results are downloadable from `http://homes.dsi.unimi.it/∼valenti/papers/BV/bv-svm-bagging.pdf`.

1) Gaussian bagged SVM ensembles: Fig. 5 represents bias–variance decomposition of error in bagged and single RBF-SVM with respect to different values of σ (the “spread” of the kernel) and for a fixed value of the regularization parameter C. error follows a “sigmoid” trend, visible also in other data sets.

[2] The application `gensimple`, that we developed to generate the data, is available on line at `ftp://ftp.disi.unige.it/person/ValentiniG/BV/gensimple`.

TABLE I

DATA SETS USED IN THE EXPERIMENTS.

Data set .	card. overall train sets	card. test sets	number of attributes
P2	synthetic	10000	2
Waveform	synthetic	10000	21
Grey-Landsat	4425	2000	36
Letter-Two	614	613	16
Letter-Two w. noise	614	613	16
Spam	2301	2300	57
Musk	3299	3299	166

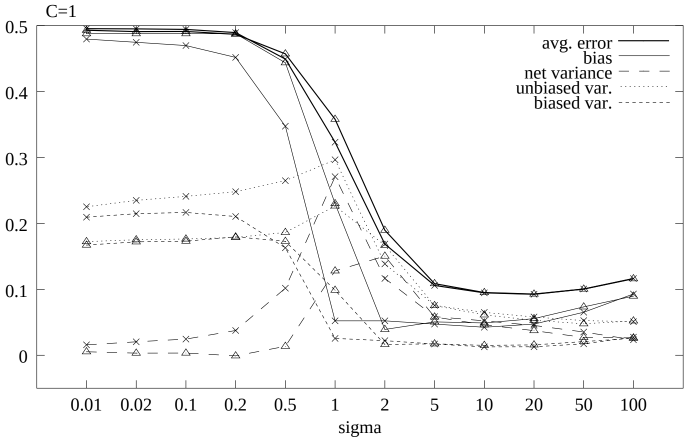


Fig. 5. Comparison of bias-variance decomposition between single gaussian SVMs (lines labeled with crosses) and bagged gaussian SVM ensembles (lines labeled with triangles), while varying σ and for $C = 1$ (Letter-Two data set).

The value of the σ parameter of the gaussian kernel determines three different regions (Fig. 5), previously just observed in single gaussian SVMs [17], [20].

High bias region (small σ values). In this region error of single and bagged SVMs is about equal, and it is characterized by a very high bias. Net-variance is close to 0, because biased variance is about equal to unbiased variance. In some cases they are both close to 0. In other cases they are equal but greater than 0 with slightly larger values in single than in bagged SVMs. In this region the error is ruled by the high bias due to the too low values of σ.

Transition region (intermediate σ values). In this region the bagged SVMs start to learn from data. The bias decreases very quickly both in single and bagged SVMs. Net-variance maintains the wave-shape just observed in single SVMs, but it is slightly lower. error diminishes at about the same rate in single and bagged SVMs (Fig. 5).

Stabilized region. For relatively large values of σ net-variance tends to stabilize. In this region net-variance of bagged SVMs is equal or less than the that of single SVMs, while bias remains substantially unchanged in both.

As a result, bagged SVMs show equal or lower average error with respect to single SVMs (Fig. 5).

Such behavior can be explained through the specific characteristics of gaussian kernels. Indeed for very small values of σ the training error is very small (about 0), while the number of support vectors is very high, and high also are error and bias. In particular the real-valued function computed by the SVM (that is the function computed without considering the sign function) is very spiky with small values of σ. The response of the SVM is high only in small areas around the support vectors, while in all the other areas "not covered" by the support vectors the response is very low (about 0): in other words the SVM is not able to get a decision, with a consequently very high bias [20]. These facts support the hypothesis of overfitting problems with small values of σ. Enlarging σ we obtain a wider response on the input domain: the real-valued function computed by the SVM becomes smoother, as the "bumps" around the support vectors become wider, and the SVM can decide also on unknown examples, while, at the same time, the number of support vectors decreases. As noted in [39], using very large values of sigma, we have a very smooth discriminant function (in practice a hyperplane), and increasing it even further nothing is changed. Moreover, enlarging too much σ we may obtain worse results, especially if the data are not linearly separable. See [20] for more details about this topic.

The main effect of bagging consists of a reduction of the unbiased variance component of error.

2) Polynomial and dot-product bagged SVM ensembles: In bagged polynomial SVMs, the trend of error with respect to the degree shows an "U" shape similar to that of single polynomial SVMs (Fig. 6). Bias and biased variance are unchanged with respect to single SVMs, while net-variance is slightly reduced (for the reduction of the unbiased variance). As a result, we have a slight reduction of the overall error. The "U" shape w.r.t. to the degree depends both on bias and net-variance. The classical trade-off between bias and variance is sometimes noticeable, but in other cases both bias and net-variance increase with the degree. As a general rule, for low degree polynomial kernel bias is relatively large and net variance is low, while the opposite occurs with high degree polynomials. The

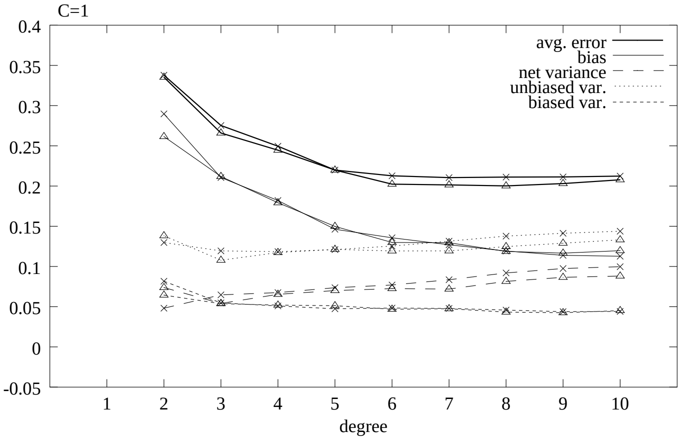


Fig. 6. Comparison of bias-variance decomposition between single polynomial SVMs (lines labeled with crosses) and bagged polynomial SVM ensembles (lines labeled with triangles), while varying the degree and for $C = 1$ (P2 data set).

regularization parameter C plays also an important role: large C values tend to decrease bias for low polynomial degrees too.

Fig. 7 shows the comparison of bias-variance decomposition between single and bagged dot-product SVMs. The reduction of error in bagged ensembles is due to the reduction of unbiased variance, while bias is unchanged or slightly increased. Biased variance also remains substantially unchanged. The shape of error curve is quite independent of the C values, at least for $C \geq 1$. Unbiased variance and bias show opposite trends both in single and bagged dot-product SVMs.

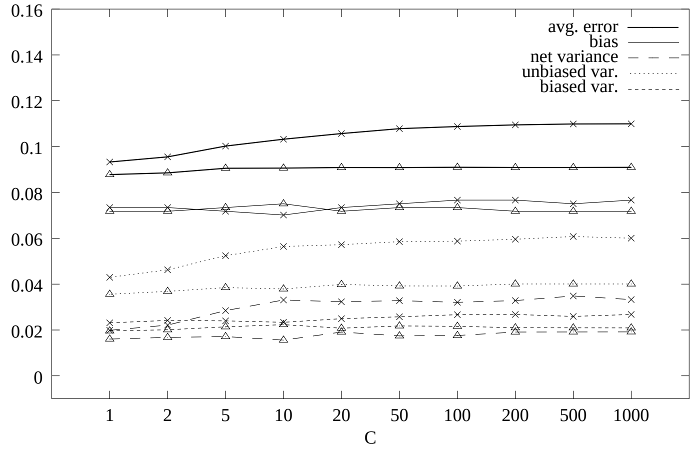


Fig. 7. Comparison of bias-variance decomposition between single dot-product SVMs (lines labeled with crosses) and bagged dot-product SVM ensembles (lines labeled with triangles), while varying the values of C (Letter-Two data set).

Considering bias–variance decomposition of error with respect to the number of base learners, we obtain most of error and unbiased variance reduction with only 10-20 base learners. Bias and biased variance remain substantially unchanged independently of the number of the base learners (Fig. 8).

C. Bias–variance analysis in random aggregated SVM ensembles

Similarly to the previous section, the results referred to single SVMs are labeled with crosses, while RA ensembles are labeled with triangles. Full experimental results are available in the supplementary material listed in the appendix.

In RA ensembles of SVMs net-variance is very close to 0. As a consequence, error is in practice reduced to bias. This property holds independently of the kernel used. For instance, in Fig. 9 that represents the compared bias–variance decomposition of error in single and random aggregated gaussian SVMs, net–variance is very close to 0, and it is quite difficult to distinguish the bias and overall error curves (labeled with triangles).

1) Gaussian random aggregated SVM ensembles: As in single and bagged SVMs, we can distinguish three main regions with respect to σ:

High bias region. In this region the error of single and random aggregated SVMs is about equal, and it is characterized by a very high bias. Net-variance is close to 0, because biased variance is about equal to unbiased

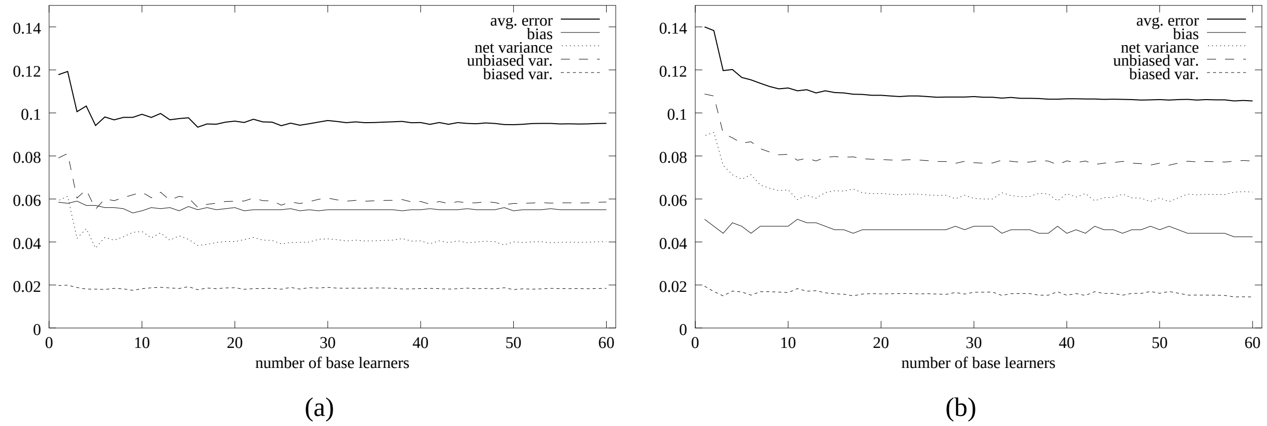


Fig. 8. Bias-variance decomposition of error in bias, net variance, unbiased and biased variance in bagged SVMs, with respect to the number of base learners. (a) Grey-Landsat data set, gaussian kernel ($\sigma = 2$, $C = 100$). (b) Letter-Two data set, dot-product kernel ($C = 100$).

variance. In most cases they are both close to 0 (Fig. 9). In some cases they are equal but greater than 0 with significantly larger values in single than in random aggregated SVMs (see supplementary material listed in the appendix).

Transition region. Bias decreases in the transition region at about the same rate in single and random aggregated SVM ensembles. Net-variance maintains the wave-shape also in random aggregated SVMs, but it is lower. In some data sets (Fig. 9), net-variance remains low with no significant variations also for small values of σ. For these reasons error decreases more quickly in random aggregated SVMs, and error of the ensemble is about equal to the bias.

Stabilized region. Net-variance stabilizes, but at lower values (very close to 0) compared with net-variance of single SVMs. Hence we have a reduction of error for random aggregated SVM ensembles in this region. Note that the reduction of error depends largely on the level of the unbiased variance.

2) Polynomial and dot-product random aggregated SVM ensembles: The error is almost entirely due to the bias also in random aggregated polynomial SVMs. The bias component is about equal in random aggregated and single SVMs.

In single SVMs sometimes opposite trends between bias and unbiased variance are observed: bias decreases, while unbiased variance increases with the degree (Fig. 10). On the contrary in random aggregated ensembles net-variance is very close to 0 and the error is almost entirely due to the bias (Fig. 10).

Hence in random aggregated SVMs, the shape of error with respect to the degree depends on the shape of bias, and consequently the error curve shape is bias-dependent, while in single and bagged SVMs is variance or bias-variance dependent.

The general shape of error with respect to the degree resembles an “U” curve, or can be flatted in dependence of the bias trend, especially with relatively large C values. Also with random aggregated dot-product SVMs the

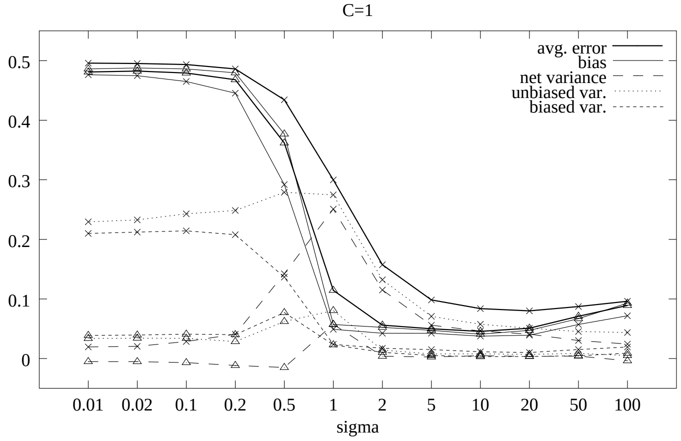


Fig. 9. Comparison of bias-variance decomposition between single gaussian SVMs (lines labeled with crosses) and random aggregated ensembles of gaussian SVMs (lines labeled with triangles), while varying σ and for $C = 1$ (Letter-Two data set).

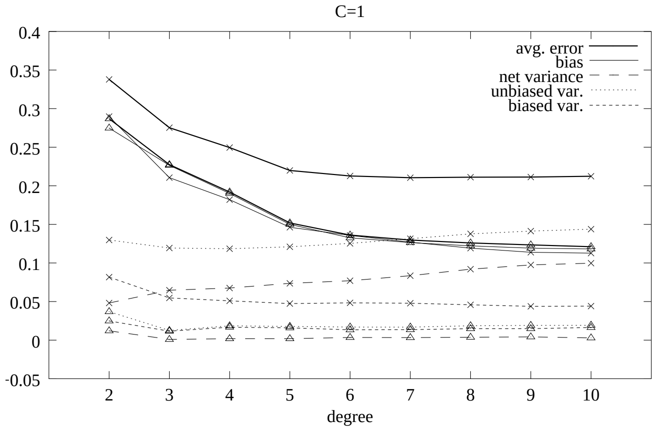


Fig. 10. Comparison between bias-variance decomposition between single polynomial SVMs (lines labeled with crosses) and random aggregated polynomial SVM ensembles (lines labeled with triangles), while varying the degree and for $C = 1$ (P2 data set).

error is about equal to the bias, that remains substantially unchanged with respect to single SVMs. Hence the error shape is equal to the bias shape. As a result we a have a significant error reduction due to decrement of unbiased variance (Fig. 11).

Fig. 12 shows that bias remains constant with respect to the number of the base learners. Most of error decrement is achieved with only 20 base learners, and it is almost entirely due to the decrement of unbiased variance. Error is reduced to bias, when the number of base learners is sufficiently large. Biased variance is low and slowly decreases, while unbiased variance continues to decrease, but most of its decrement occurs within the first 20 base learners

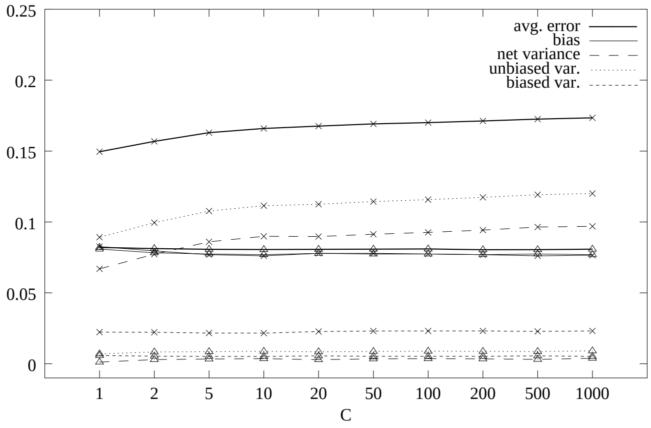


Fig. 11. Comparison of bias-variance decomposition between single dot-product SVMs (lines labeled with crosses) and random aggregated dot-product SVM ensembles (lines labeled with triangles), while varying the values of C. (Spam data set).

(Fig. 12).

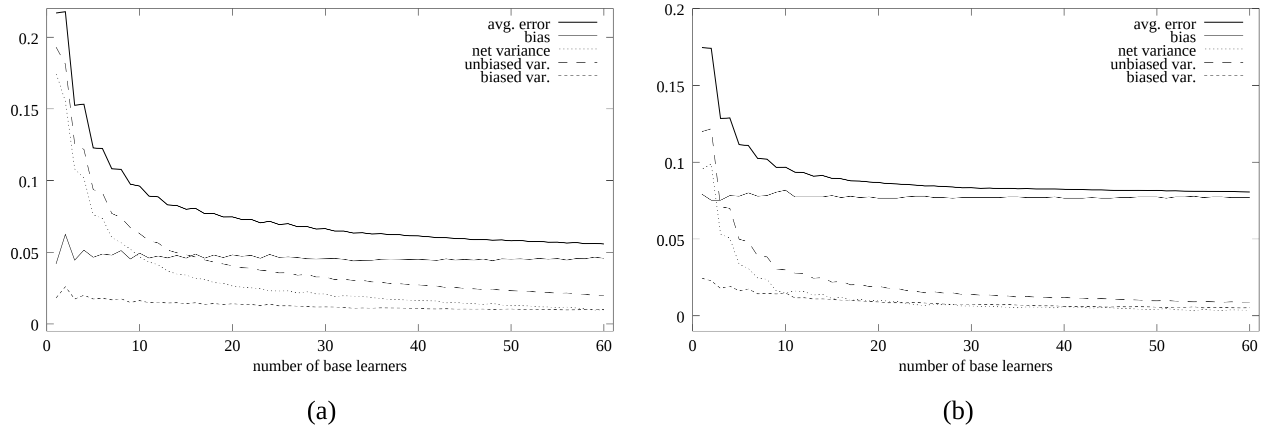
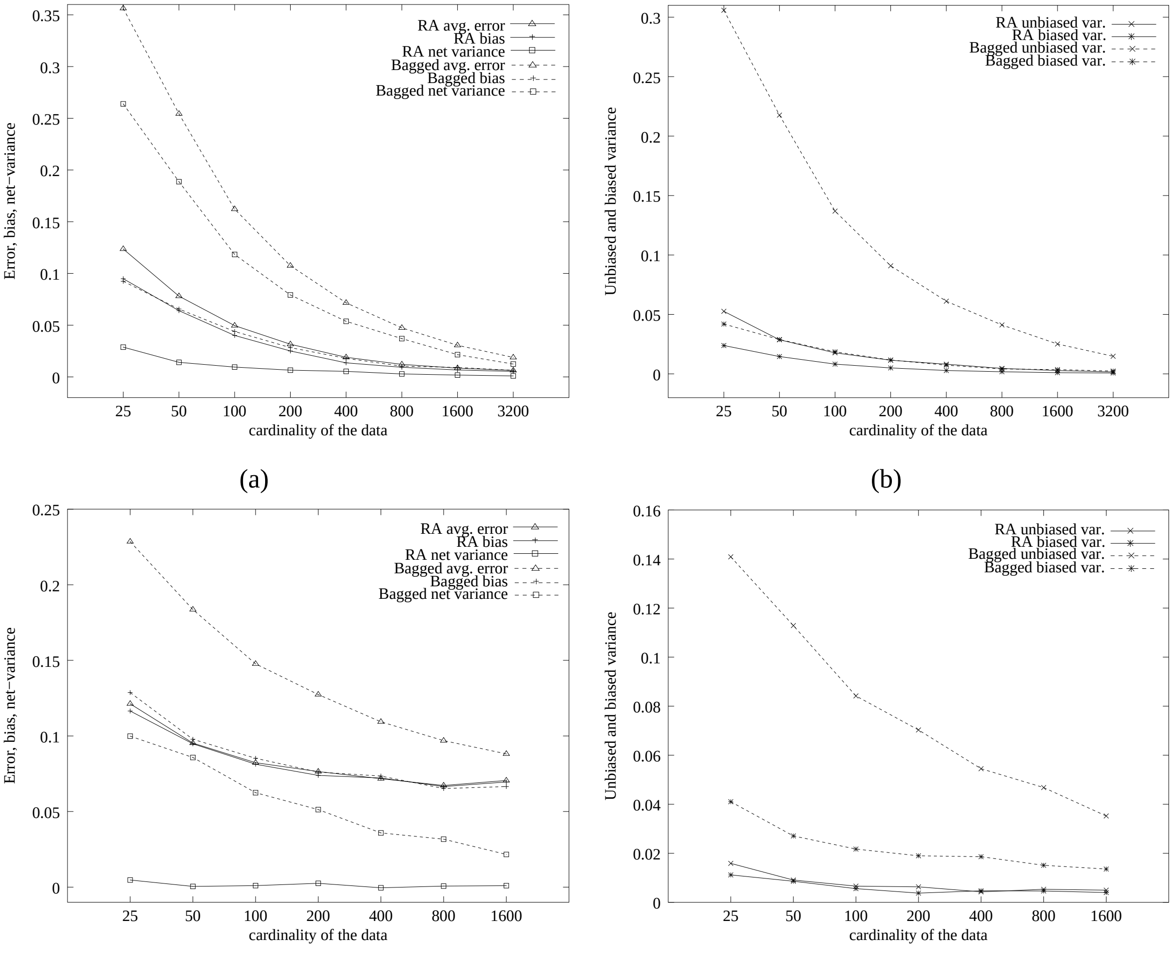


Fig. 12. Bias-variance decomposition of error in bias, net variance, unbiased and biased variance in random aggregated SVMs, with respect to the number of base learners. (a) Gaussian kernel, $C = 1$ $\sigma = 0.2$, (P2 data set) (b) Dot-product kernel, $C = 100$ (Spam data set).

D. Relationships between bias–variance decomposition of error and cardinality of the data in bagged and RA-SVM ensembles.

To understand the relationships between bias–variance decomposition of error and the cardinality of the data, we performed a bias–variance analysis considering training samples from 25 to 3200 examples with the P2 data set and from 25 to 1600 examples with the Spam data set. For full experimental results and details, see the supplementary material listed in the appendix.



(a) (b)

(c) (d)

Fig. 13. Comparison of the bias-variance decomposition of error in bagged and random aggregated SVMs, while varying the cardinality of the data. Continuous lines: RA SVMs; dashed lines: bagged SVMs. (a) P2 data set: average error, bias and net variance (gaussian kernel with $\sigma = 0.5$ and $C = 100$) (b) P2 data set: unbiased and biased variance (gaussian kernel with $\sigma = 0.5$ and $C = 100$) (c) Spam data set: average error, bias and net variance (polynomial kernel of 3^{rd} degree and $C = 100$) (d) Spam data set: unbiased and biased variance (polynomial kernel of 3^{rd} degree and $C = 100$)

While for small samples the difference of net and unbiased variance between bagged and random aggregated ensembles is very large, error and variance tend to converge to the same values when the cardinality is increased (Fig. 13). Bias, as expected, is in general quite similar both in bagged and random aggregated ensembles (Fig. 13 a and c), while unbiased and biased variance is significantly smaller in RA ensembles, especially with small samples (Fig. 13 b and d).

On the other hand, if we consider the relative reduction of error, bias and variance reduction for RA and bagged ensembles w.r.t. single SVMs, the scenario is quite different (Fig. 14 a and c). The relative error reduction for

TABLE II

COMPARISON OF THE RESULTS BETWEEN SINGLE AND RA-SVMS, WHILE VARYING THE CARDINALITY OF THE SAMPLES.

	$N.examples$	$Error$	$Time\ (sec.)$	$Speed - up$
S-SVM	100000	0.0023	5474.68	——
RA-SVM	10000	0.0032	4089.82	1.3
RA-SVM	5000	0.0043	855.85	6.4
RA-SVM	2000	0.0076	124.93	43.8
RA-SVM	1000	0.0127	38.49	142.2
RA-SVM	200	0.0358	2.92	1874.9
RA-SVM	100	0.0539	1.69	3239.4

bagging is computed in the following way:

$$\text{Relative error reduction} = \frac{\text{Single SVM error} - \text{Bagged SVM error}}{\max(|\text{Single SVM error}|, |\text{Bagged SVM error}|)} \tag{16}$$

Fig. 14 b and d show the relative reduction of unbiased and biased variance. The most significant fact is the very large relative reduction of unbiased and net-variance in RA ensembles. Such reduction remains consistent and constant independently of the size of the samples (Fig. 14). In bagged SVMs, we have not such a large net and unbiased variance reduction: it is independent of the sample size for the P2 data set and greater with samples larger than 100 examples for the Spam data set (Fig. 14 c and d).

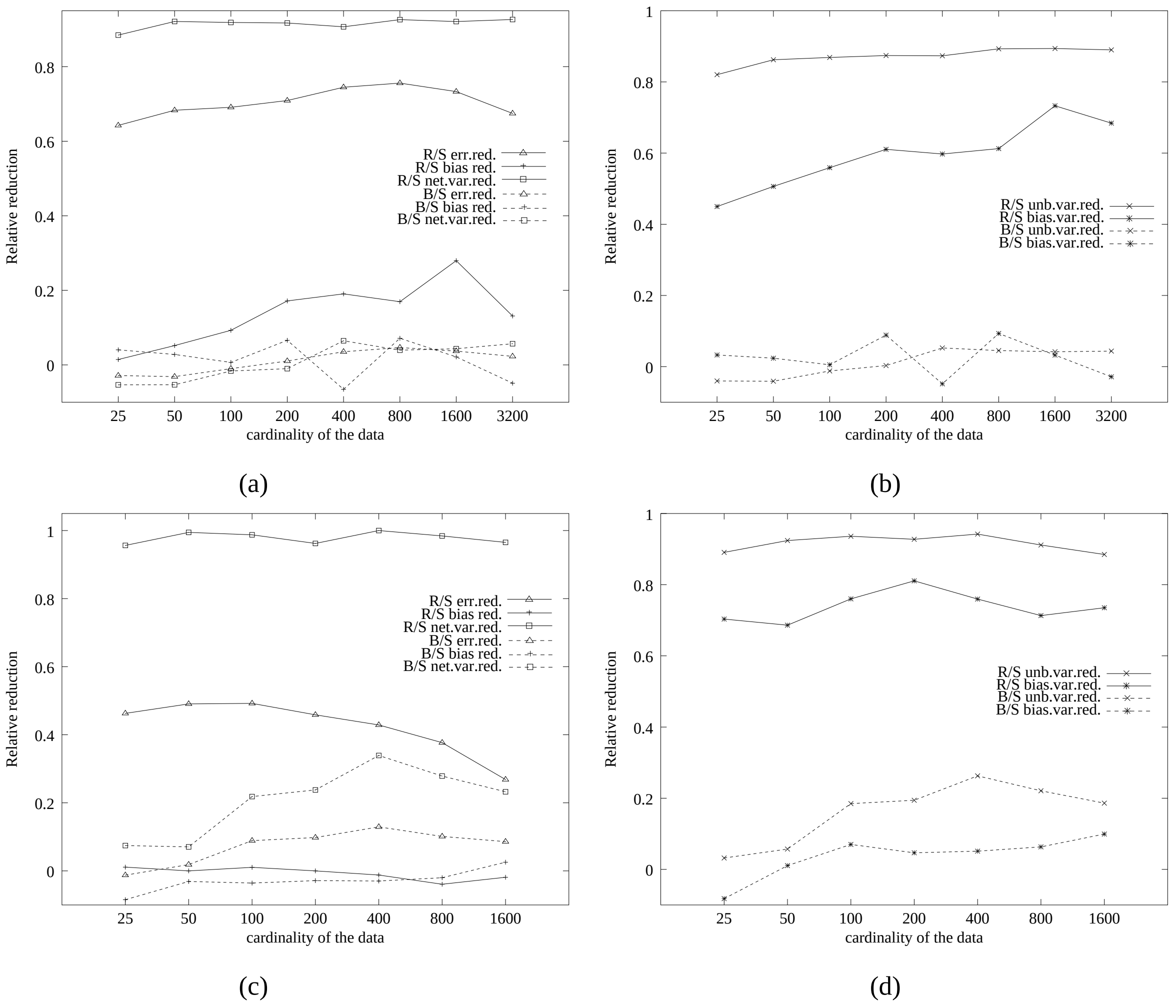
In our experiments bias remains substantially unchanged in both RA and bagged ensembles w.r.t. single SVMs, and sometimes with bagging the bias, as expected, is also increased (Fig. 14 a and c). Anyway in some situations bias relative reduction is not negligible both in bagged and RA ensembles. For instance with the P2 data set we observe a consistent bias relative reduction in RA ensembles, especially if large samples are used (Fig. 14 a). Moreover, depending of the choice of the σ parameter of gaussian kernels, with bagging we may also have a consistent reduction of bias if small samples are used, and with RA ensembles if quite large samples are used (e.g. for $\sigma = 0.1$, see figures in the supplementary material listed in the appendix).

E. Comparing single SVMs trained on large data sets with RA-SVM ensembles trained on small samples

The bias–variance analysis of random aggregated ensembles showed that the variance component of error is strongly reduced, while bias remains unchanged or is lowered (Sect. IV-C).

These facts suggest to apply RA-SVMs to large scale classification problems, considering also that the SVM algorithm, as well as other learning algorithms, does not scale too well when very large samples are available [40]. If variance reduction due to random aggregation is comparable with bias increment due to the reduction of sample size, using small samples to train the ensemble, we may obtain an accuracy similar to that of a single SVM trained on an entire large training set.

To yield insight into this hypothesis we performed a preliminary experiment with the synthetic data set *P2*, using



(a)

(b)

(c)

(d)

Fig. 14. Comparison of the relative reduction of error and bias-variance in bagged and random aggregated SVMs with respect to single SVMs, while varying the cardinality of the data. Continuous lines refer to random aggregated SVMs, dashed lines to bagged SVMs. R/S stands for Random aggregated vs. single SVMs and B/S bagged vs. single. Negative values indicate better results of single SVMs. (a) P2 data set: Comparing relative reduction of error, bias and net variance (gaussian kernel with $\sigma = 0.5$ and $C = 100$) (b) P2 data set: Comparing relative unbiased and biased variance (gaussian kernel with $\sigma = 0.5$ and $C = 100$) (c) Spam data set: Comparing relative reduction of error, bias and net variance (polynomial kernel of 3^{rd} degree and $C = 100$) (d) Spam data set: Comparing relative unbiased and biased variance (polynomial kernel of 3^{rd} degree and $C = 100$).

a quite large learning set of 10^5 examples, and comparing the results of single and RA-SVMs on a separate large testing set.

Table II summarizes the results of the experiments with gaussian kernels: S-SVM stands for single SVMs trained on the entire available learning set; RA-SVM stands for Random aggregated SVMs trained on subsamples of the available training set, whose cardinality are shown in the column $N.examples$; the column $Error$ shows error on a separate test set (composed by 100000 examples); the column $Time$ shows the the training time in seconds, using an AMD Athlon 2000+ processor with 512 Mb RAM, and the last column the speed-up achieved.

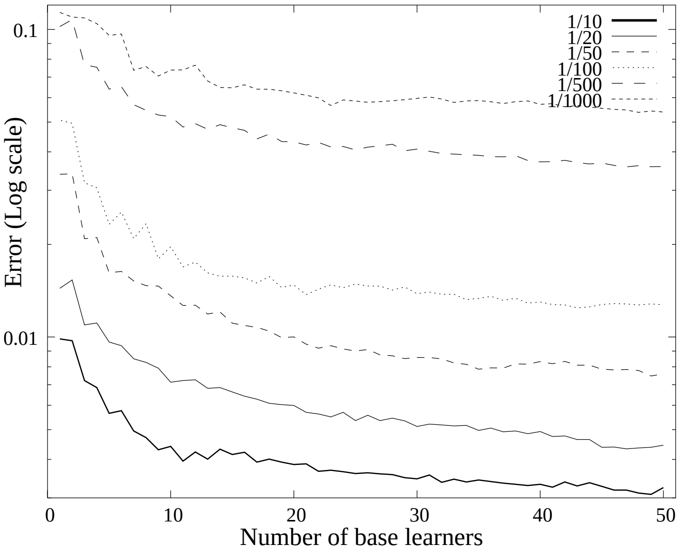


Fig. 15. Error (Log scale) as a function of the number of base learners (gaussian SVM) employed. The different curves refer to ensembles with base learners trained with different fractions of the learning set.

The results show that with RA-SVMs we may obtain a consistent speed-up, at the expense of a certain decrement of the accuracy: for instance, if accuracy is not the main concern, using RA-SVM ensembles trained with $1/50$ of the available data we can compute the same task in about 2 minutes against one hour and a half needed for a single SVM trained on the entire data set (Table II). Note that we achieve most of error reduction with about 30 base learners (Fig. IV-E), and using a parallel implementation we may also expect a further speed-up linear in the number of the base learners.

F. Effect of noisy data on bias–variance decomposition.

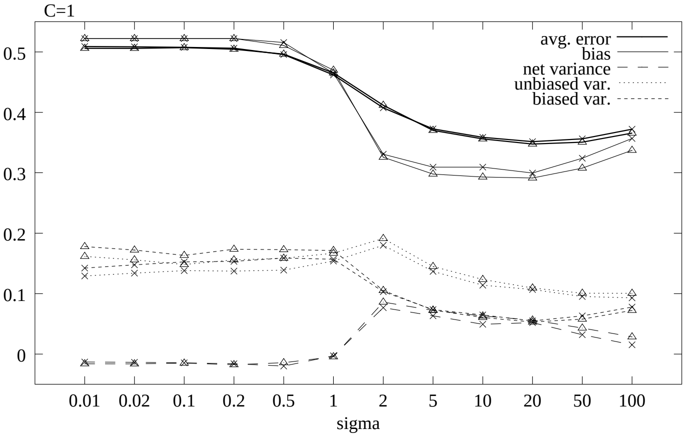
To simplify the computation and the overall analysis, in our experiments we did not explicitly consider noise, because its estimation with “real” data is a difficult task [12]. Anyway, noise may play a significant role in bias–variance analysis.

More precisely, Domingos [11] showed that for a quite general loss function the expected loss is:

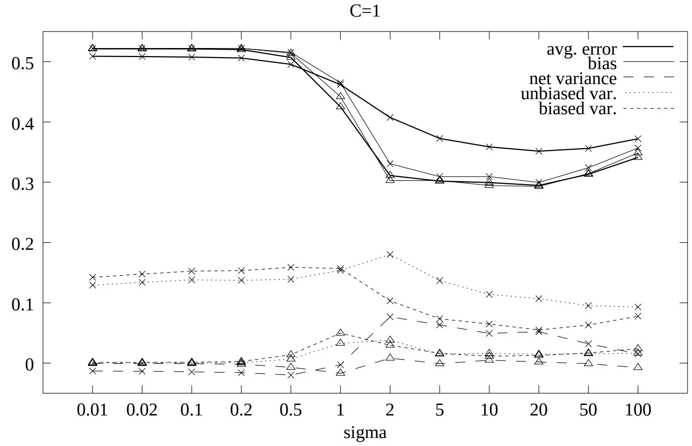
$$EL(\mathcal{L}, \mathbf{x}) = c_1 N(\mathbf{x}) + B(\mathbf{x}) + c_2 V(\mathbf{x}) \tag{17}$$

where N, B and V represent respectively the noise, bias and variance. For the $0/1$ loss function c_1 is $2P_D(f_D(\mathbf{x}) = y_*) - 1$; c_2 is +1 if $B(\mathbf{x}) = 0$ and -1 if $B(\mathbf{x}) = 1$. Hence, according to Domingos, the noise is linearly added to error with a coefficient equal to $2P_D(f_D(\mathbf{x}) = y_*) - 1$ (eq. 17). If the classifier is accurate, i.e. if $P_D(f_D(\mathbf{x}) = y_*) \gg 0.5$, then the noise $N(\mathbf{x})$, if present, influences the expected loss. In the opposite situation also, with very bad classifiers, that is when $P_D(f_D(\mathbf{x}) = y_*) \ll 0.5$, the noise influences the overall error in the opposite sense: it reduces the expected loss. If $P_D(f_D(\mathbf{x}) = y_*) \approx 0.5$, i.e. if the classifier performs a sort of random guessing, then $2P_D(f_D(\mathbf{x}) = y_*) - 1 \approx 0$ and the noise has no substantial impact on error.

In a previous work we showed that with single SVMs, if noise is present, but not explicitly considered, its main effect consists of incrementing bias and consequently the average error [20]. The same effect can be observed also with bagged and RA ensembles. Indeed, with gaussian kernels, adding 20 % noise to the Letter-Two data set (Fig. 16 (a) and (b)), bias is raised to about 0.3 (that is 30 %) both in bagged (a) and RA (b) ensembles, with an increment of about 0.25 with respect to the data set without noise (Fig. 5 a and 9 a), while the net–variance is only sligthly incremented. A similar behavior is also observed with polynomial and dot-product kernels (for full details, see the supplementary material listed in the appendix).



(a)



(b)

Fig. 16. Letter-Two with noise data set: comparison of bias–variance decomposition of error between: (a) Bagged and single SVMs (b) RA and single SVMs. Lines labeled with crosses refer to single SVMs, while lines labeled with triangles refer to bagged (a) and RA (b) ensembles. In abscissa are reported values of the σ parameter of the gaussian kernel.

TABLE III

COMPARISON OF THE RESULTS BETWEEN SINGLE AND BAGGED SVMS.

	$\mathbf{E_{SVM}}$	$\mathbf{E_{bag}}$	**% Error reduction**	**% Bias reduction**	**% NetVar reduction**	**% UnbVar reduction**
Data set *P2*						
RBF-SVM	0.1517	0.1500	1.14	-2.64	3.18	2.19
Poly-SVM	0.2088	0.1985	4.95	4.85	5.08	5.91
D-prod SVM	0.4715	0.4590	2.65	1.11	34.09	15.28
Data set *Waveform*						
RBF-SVM	0.0707	0.0662	6.30	-1.41	26.03	17.82
Poly-SVM	0.0761	0.0699	8.11	0.36	23.78	17.94
D-prod SVM	0.0886	0.0750	15.37	-0.22	37.00	28.20
Data set *Grey-Landsat*						
RBF-SVM	0.0384	0.0378	1.74	2.94	-7.46	3.94
Poly-SVM	0.0392	0.0388	1.05	-4.76	24.80	12.06
D-prod SVM	0.0450	0.0439	2.58	16.87	-165.72	-62.21
Data set *Letter-Two*						
RBF-SVM	0.0745	0.0736	1.20	-25.00	21.63	12.29
Poly-SVM	0.0745	0.0733	1.55	-15.79	13.92	10.41
D-prod SVM	0.0955	0.0878	8.09	2.22	27.55	23.06
Data set *Letter-Two with added noise*						
RBF-SVM	0.3362	0.3345	0.49	1.75	-5.78	0.40
Poly-SVM	0.3432	0.3429	0.09	-0.58	3.06	0.91
D-prod SVM	0.3486	0.3444	1.21	-0.56	10.23	6.09
Data set *Spam*						
RBF-SVM	0.1292	0.1290	0.14	-0.48	1.57	2.22
Poly-SVM	0.1323	0.1318	0.35	2.11	-5.83	-1.19
D-prod SVM	0.1495	0.1389	7.15	-3.16	19.87	16.38
Data set *Musk*						
RBF-SVM	0.0898	0.0920	-2.36	-6.72	22.91	13.67
Poly-SVM	0.1225	0.1128	7.92	-10.49	38.17	37.26
D-prod SVM	0.1501	0.1261	15.97	-2.41	34.56	29.38

V. DISCUSSION

A. Bias–Variance characteristics of bagged and RA SVM ensembles

In Table III the compared results of bias–variance decomposition between single SVMs and bagged SVM ensembles are summarized. E_{SVM} stands for the estimated error of single SVMs, E_{bag} for the estimated error of bagged ensembles of SVMs, % $Error\ reduction$ stands for the percent error reduction of error between single and bagged ensembles, and it is computed as in eq.16.

% $Bias\ reduction$, % $NetVar\ reduction$ and % $UnbVar\ reduction$ corresponds respectively to the percent reduction of bias, net–variance and unbiased variance between single and bagged ensembles of SVMs. The negative signs means that a larger error in the bagged ensemble is obtained. Note that sometimes the decrement of the net–variance can be larger than 100 %: indeed net–variance can be negative, when biased variance is larger than unbiased variance.

TABLE IV

COMPARISON OF THE RESULTS BETWEEN SINGLE AND RANDOM AGGREGATED SVMS.

	$\mathbf{E_{SVM}}$	$\mathbf{E_{agg}}$	**% Error reduction**	**% Bias reduction**	**% NetVar reduction**	**% UnbVar reduction**
Data set *P2*						
RBF-SVM	0.1517	0.0495	67.37	24.52	99.04	85.26
Poly-SVM	0.2088	0.1030	50.65	19.56	92.26	83.93
D-prod SVM	0.4715	0.4611	2.21	0.89	142.65	91.08
Data set *Waveform*						
RBF-SVM	0.0707	0.0501	29.08	1.14	100.58	89.63
Poly-SVM	0.0761	0.0497	34.59	3.68	97.12	89.44
D-prod SVM	0.0886	0.0498	43.74	3.84	99.12	90.69
Data set *Grey-Landsat*						
RBF-SVM	0.0384	0.0300	21.87	3.22	99.95	85.42
Poly-SVM	0.0392	0.0317	19.13	3.17	83.79	80.95
D-prod SVM	0.0450	0.0345	23.33	19.27	69.88	72.57
Data set *Letter-Two*						
RBF-SVM	0.0745	0.0345	53.69	0.00	95.32	92.48
Poly-SVM	0.0745	0.0346	53.54	-5.26	95.46	92.71
D-prod SVM	0.0955	0.0696	27.11	2.22	109.73	92.31
Data set *Letter-Two with added noise*						
RBF-SVM	0.3362	0.2770	17.55	2.92	90.26	87.04
Poly-SVM	0.3432	0.2775	19.13	1.75	95.96	89.42
D-prod SVM	0.3486	0.2925	16.07	-1.68	106.4	89.97
Data set *Spam*						
RBF-SVM	0.1292	0.0844	34.67	6.75	99.74	90.05
Poly-SVM	0.1323	0.0814	38.47	22.33	95.22	86.03
D-prod SVM	0.1495	0.0804	46.22	6.90	94.91	90.24
Data set *Musk*						
RBF-SVM	0.0898	0.0754	16.02	0.39	106.70	93.85
Poly-SVM	0.1225	0.0758	38.12	1.53	97.52	94.02
D-prod SVM	0.1501	0.0761	49.28	0.80	98.30	93.03

As expected, bagging usually does not reduce bias (on the contrary, sometimes bias slightly increases). Net-variance is only partially reduced, and its decrement ranges from 0 to about 35 % with respect to single SVMs. Its reduction is due to the unbiased variance reduction, while biased variance is unchanged. As a result, error slightly decreases, ranging from 0 to about 15 % with respect to single SVMs, depending on the kernel and the data set. The overall shape of the curves of error, bias and variance are very close to that of single SVMs (Fig. 5, 6, 7).

In Table IV are summarized the compared results of bias–variance decomposition between single SVMs and random aggregated SVM ensembles. E_{SVM} stands for the estimated error of single SVMs, E_{agg} for the estimated error of random aggregated ensembles of SVMs, % $Error\ reduction$ stands for the percent error reduction of error between single and random aggregated ensembles.

Random aggregated ensembles of SVMs strongly reduce net-variance. Indeed in all the data sets net-variance is near to 0, with a reduction close to 100 % with respect to single SVMs, confirming the ideal behavior of random aggregating. Unbiased variance reduction is responsible for this fact, as in all data sets its decrement amounts to about 90 % with respect to single SVMs (Table IV). As expected, bias remains substantially unchanged, but with the *P2* data set we register a not negligible decrement of the bias, at least with polynomial and gaussian kernels. As a result, error decreases from 15 to about 70 % with respect to single SVMs, depending on the kernel and on the characteristics of the data set. Note that RA ensembles obtain less scattered error estimates with respect to bagged ensembles: this is due to the fact that for bagging the estimate of the error strongly depends on the choice of the sample from which bootstrapped data are drawn, while for RA ensembles the samples are drawn directly from the complete data set.

The overall shape of the curves of error resembles that of bias of single SVMs, with a characteristic sigmoid shape for gaussian kernels (Fig. 9), an "U" shape for polynomial kernels (Fig. 10), while it is relatively independent of the C values (at least for sufficiently large values of C) for random aggregated linear SVMs (Fig. 11).

Friedman showed that bagging an estimator leaves the linear part unchanged, but reduces the variability of the non linear component by replacing it with an estimate of its expected values (thus reducing variance) [7]. Thus bagging should be effective with highly non-linear methods such as decision trees or neural networks, but not so effective with linear methods (and viceversa also with linear problems). Our results with bagged and RA ensembles confirm the theoretical analysis of Friedman (Table III and IV). However we obtained sometimes significant reduction of the error also with linear SVMs. These results can be interpreted in the Friedman's theoretical framework considering that we used in our experiments soft-margin linear SVMs: in this setting the regularization introduced in the quadratic optimization problem associated with the SVM algorithm may reduce error in non linear problems, even if linear classifiers are used [21], [41].

Summarizing, in our experiments, as expected, we obtained a smaller reduction of the average error with bagged SVMs (from 0 to 15 %), due to a lower decrement of the net-variance (about 35% against a reduction of about 90 % with random aggregated ensembles), while bias remains unchanged or slightly increases (Fig. 17).

B. Related experimental work on bias–variance analysis of bagging

Bauer and Kohavi performed an experimental analysis of bias-variance decomposition of error in bagged Naive-Bayes and decision tree ensembles [15], and Zhou, Wu and Tang studied bagged neural networks [16]. In both cases the authors adopted the bias–variance decomposition scheme proposed by Kohavi and Wolpert [34]. More precisely, let T be the random variable representing the label $t \in \mathcal{C}$ of an example $\mathbf{x} \in \mathbb{R}^d$ and Y the random variable representing the prediction of a classifier with respect to an example $\mathbf{x}$, where $\mathcal{C}$ is a set of discrete values corresponding to different classes. Then, according to Kohavi and Wolpert, the bias $B_{KW}(\mathbf{x})$ and variance $V_{KW}(\mathbf{x})$ are:

$$B_{KW}(\mathbf{x}) = \frac{1}{2} \sum_{t \in \mathcal{C}} \left(P(T = t|\mathbf{x}) - P(Y = t|\mathbf{x})\right)^2 \tag{18}$$

$$V_{KW}(\mathbf{x}) = \frac{1}{2} \left(1 - \sum_{t \in \mathcal{C}} P(Y = t|\mathbf{x})^2\right) \tag{19}$$

Even if this scheme captures the main concepts behind bias and variance, the resulting estimates of bias and variance are slightly biased [11]. Indeed Kohavi and Wolpert defined bias and variance in terms of quadratic functions of $P(T = t|\mathbf{x})$ and $P(Y = t|\mathbf{x})$, while the loss function used in classification is the $0/1$ loss. Thus the resulting decomposition is purely additive, while we know that in some cases the variance should be subtracted to error (see Sect. III-A, and [20] for more details). The bias, as defined in eq. 18, is not restricted to taking on the values 0 or 1 (as it should be natural with the $0/1$ loss function). Moreover it is easy to see that with the Kohavi Wolpert decomposition scheme the optimal Bayes classifier may have non zero bias. Anyway, if biased variance is not too high and the estimated bias in not too far from 0 or 1, Kohavi and Wolpert decomposition of error for classification problems in not too seriously biased.

Comparing our experimental setup with that of Bauer and Kohavi and Zhou et al., we used smaller ratios between samples and overall training set sizes, in order to take into account the variability of the data for a given size. We employed also a larger number of samples to obtain better approximation of the bias–variance estimates. On the other hand Bauer and Kohavi explicitly selected the size of the learning set in order to permit improvements with ensembles (in the sense that too large training sets may generate Bayes-optimal classifiers), while we used only relatively small sample sizes, without testing if there was room for improvements. Moreover we explicitly considered the effects of learning parameters (Sect. IV-B) on bias–variance decomposition of error, while this is only partially considered in Bauer and Kohavi work. On the contrary Zhou et al. used only a single architecture and MATLAB default learning parameters for all the neural networks used in their work.

Anyway the overall results obtained by Bauer and Kohavi are quite similar to the ones we obtained in our experiments. Indeed they achieved a comparable average relative variance reduction, without significant average bias reduction, confirming the theoretical property that bagging is mainly a variance reduction ensemble method, at least when unstable base learners are used. The overall reduction of error obtained with bagged decision trees is quite larger with respect to the average error reduction we registered with bagged SVMs. Note that we used different base learners, different data sets and also different measures to estimate the bias–variance decomposition of error. Anyway Bauer and Kohavi and our results basically fpund the same overall bias–variance trade–off in bagged ensembles of unstable base learners.

Zhou et al. found that variance is reduced in bagged neural networks. In particular they found an overall variance and error reduction significantly larger compared with our results. Even if it is difficult to compare the results obtained from different data sets and different bias–variance measures, we guess that this could be not a specific characteristic of bagged neural networks. Indeed Zhou et al. considered only a particular architecture (one hidden layer with 5 hidden units), and the MATLAB default learning rate for the back-propagation algorithm in all the experiments they performed to evaluate the bias–variance decomposition of error. Anyway, it has been shown that learning parameters strongly affect the bias–variance decomposition of error [20]. Even if to my knowledge no extensive bias–variance decomposition of error with respect to the learning parameters and architectures of neural networks have been performed, we guess that these factors should strongly influence the bias–variance decomposition of the error both in single and ensemble-aggregated neural networks. To help unravel this question, we need extensive experiments with bagged neural networks, explicitly considering the effect of different architectures and learning parameters on the bias–variance decomposition of error.

C. Sample size, bagging and random aggregating.

The bias–variance curves in single, bagged and random aggregated SVMs depend also on the size of the samples. Enlarging sample size, absolute values of error, bias and variance of bagged and RA SVMs seem to converge to similar values (Fig. 13).

On the other hand, if we increase the sample size, net and unbiased variance relative reduction of bagged SVMs w.r.t. single SVMs do not converge to the corresponding net and unbiased variance relative reduction of RA SVMs (Fig. 14). As a consequence of the large relative difference of the variance, the relative error reduction also does not converge to the same value in bagged and RA ensembles (Fig. 14 a and c).

These results show that bagging is only an approximation of random aggregation, at least when unstable base learners, such as SVMs, are used with small samples. Random aggregation fully exploits the variability of the data drawn from the universe population, while bagging tries only to simulate the variability of the data through bootstrapping techniques. Our results show also that when the size of the samples increases, the differences between the two approaches tend to be smaller, at least if we consider the absolute values of error, bias and variance.

As conjectured in our previous work [20], the optimal choice of σ with gaussian kernels strongly depends on the size of the available training sets. For instance with 100 samples $\sigma = 0.1$ is largely sub-optimal with single and bagged SVMs: indeed with $\sigma = 5$ error is about halved. On the contrary, with 3200 samples, choosing $\sigma = 0.1$ we obtain a significantly smaller error w.r.t. $\sigma = 5$ (P2 data set, see figures in the supplementary material listed in the appendix). These results depend on the coverage of the input space. With small samples we need larger σ values to cover the input space, because with small σ we may have no response of the classifier on some regions of the input space. With larger samples small σ may be not so critical as we may have a larger integration of the localized response around each support vector (bear in mind that the output of a gaussian SVM is a weighted sum of gaussian kernels centered around the support vectors with spread equal to σ).

D. Bias reduction in bagging and random aggregating

Our results show also that sometimes we may have bias reduction both in bagged and RA ensembles w.r.t. single SVMs (Fig. 14 a and c). A bias reduction in bagged ensembles has been observed in a regression setting also by Friedman and Hall [7], by Bauer and Kohavi [15] with decision trees as base learners, and by Zhou et al. [16] using neural networks as base learners.

With decision trees, bias reduction may be due to a no pruning approach: for instance, in [15] using unpruned decision trees the resulting bagged ensemble showed a lower bias, with a variance decrease due to the aggregation by majority voting. This approach is quite similar to the the selection of low biased base learners in Lobag ensembles [20]. Anyway in some cases in bagged and random aggregated ensembles we may observe a bias reduction without a clear relationship with the complexity of the base learners (Table III and IV). Similar results have been also obtained with very simple single-node-split decision trees (decision stumps) used as base learners [15].

In our experiments with gaussian SVMs as base learners, we can observe that the possible bias reduction may depend on the choice of the σ parameter. Anyway, with linear SVMs we may also have sometimes consistent relative bias reduction in bagged and RA ensembles (e.g. with the the Grey-Landsat data set, Table III and IV). These results are not explained by the classical Breiman's theory about bagging and ensemble methods based on resampling techniques. We have no a clear explanation of this phenomenon.

E. Effectiveness of overproduce and select *ensemble methods*

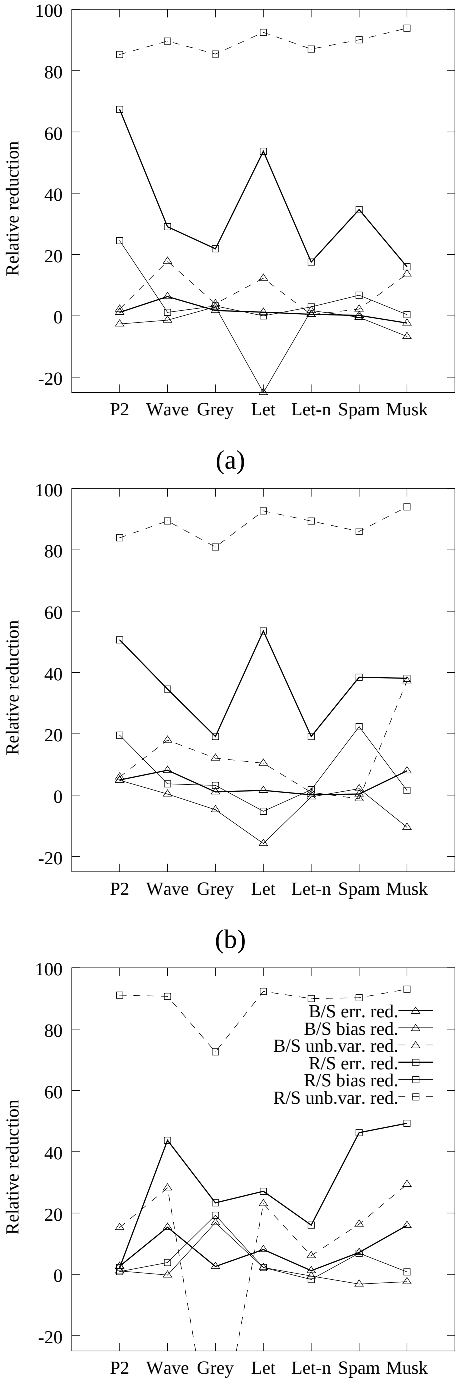
Considering the bias-variance decomposition with respect to the number of base learners, we may observe that most of the decrement of error occurs within the first iterations (from 10 to 30, depending on the data set), mainly for the decrement of net and unbiased variance, while bias and biased variance remains substantially unchanged (Fig. 12 and 8). These results suggest that we may employ relatively small SVM ensembles to achieve the same results as with larger ones.

Our results also support the "many could be better than all" theory [16], considering that if with 20 or 30 base learners we achieve about the same results that we may obtain with 100 base learners, we could try to select the best ones to enhance the overall performance of the ensemble. Indeed Zhou et al. showed that using genetic optimization techniques we may improve the accuracy of the ensemble with respect to standard bagging and Adaboost: the selection of the better base learners according to a fitness function related to the generalization error may significantly improve the accuracy of the ensemble [16].

From this standpoint our results support also other "overproduce and select" ensemble methods, based on the production of a pool of classifiers followed by a selection procedure to pick the classifiers that are most diverse and accurate [3]. Ensemble methods of this type are for instance the "Pruning adaptive boosting" approach [42], that uses the "kappa-error convex hull pruning" to select a subset of base learners out of the set of classifiers produced by Adaboost, the ensembles selected by double fault and Q statistic diversity measures [43], [44], and the "thinning the ensemble" approach by which the most incorrect classifiers on "uncertain examples" are removed from the ensemble [45].

F. Bias–variance analysis of bagged SVMs suggests how to improve Lobag

A variant of bagging, named *Lobag*, has been proposed to enhance the performance of standard bagging [18]. This approach is based on the selection of low-biased base learners through bias–variance analysis techniques; the selected base learners are successively aggregated in order to reduce the variance. Our experimental analysis shows that this approach is effective when the unbiased variance component of error is significant, as bagging reduces the unbiased variance, while bias remains substantially unchanged. Hence we may expect that Lobag works when small sized data sets are used. Effectively, it has been shown that Lobag significantly outperforms bagging when small samples are used [18].



(a)

(b)

(c)

Fig. 17. Comparison of the relative error, bias and unbiased variance reduction between bagged and single SVMs (lines labeled with triangles), and between random aggregated and single SVMs (lines labeled with squares). B/S stands for Bagged versus Single SVMs, and R/S for random aggregated versus Single SVMs. Results refers to 7 different data sets. (a) Gaussian kernels (b) Polynomial kernels (c) Dot-product kernels.

It has been observed that bagging unpruned trees reduces the bias of the bagged ensemble, while variance is reduced averaging between the base learners [15]. These unpruned trees may be interpreted as raw low bias base learners, and unpruned bagged trees as raw Lobag ensembles. Indeed in the Lobag algorithm the low bias learners are explicitly selected through out-of-bag estimates of the bias–variance decomposition of error, while in unpruned bagged trees no explicit bias measurements are performed.

Anyway Lobag selects as base learner the one with the lowest estimated bias, without taking into account the variance. Our experimental results show that net-variance is lowered of about 20 % in bagged SVMs (with a certain variability that depends on the kernel, kernel parameters and on the characteristics of the data sets). Hence an improvement of Lobag may consists of selecting the base learners according to the lowest sum of the estimated bias plus 20 % of its estimated variance. More refined approaches may try to estimate the net-variance reduction adaptively from the data [3].

G. *Voting many unstable classifiers built with small subsets of data strongly reduces variance*

Our experiments with RA ensembles can also explain the reasons why voting many unstable classifiers built on small subsets of data, such as Breiman's "Pasting Small Votes" ensembles [27] and their distributed counterpart [26], [28] work with large databases. Indeed random aggregated ensembles (using a bootstrap approximation of P) randomly draw small subsets of data from the universe population U. These approaches are effective when we have very large or distributed data sets. In these situations ordinary learning algorithms cannot directly process the data set as a whole. For instance several implementations of the SVM learning algorithm have a $\mathcal{O}(n^2)$ space complexity, where n is the number of examples. If n is relatively large (e.g. $n = 10^6$) we need room for 10^{12} elements, a too costly memory requirement for most current computers.

Comparing RA ensembles trained with small samples with single SVMs trained on the entire available large learning set (Sect. IV-E), we may obtain a significant speed-up at the expense of a certain decrement of the accuracy. Moreover we may further increment the speed-up with a distributed implementation as in the DR-vote ensembles proposed by Chawla et al. [26]. Our experiments (Sect. IV-C) show that the success of this approach is due to the unbiased variance reduction, while bias remains substantially unchanged. Anyway, with respect to single SVMs trained on the entire available learning data, RA ensembles trained on small samples achieved a lower accuracy (Sect. IV-E). Similar results have been obtained also by Evgeniou et al. [29], where SVM ensembles trained on small samples uniformly drawn from a large data set achieved similar or worse results of that of single SVMs trained on the entire data set. We suppose that this could be the effect of the bias increment due to the reduction of the size of the samples used in RA ensembles. with respect to the low bias of the SVM trained on the entire available learning set.

Breiman [27] and Chawla, Hall, Bowyer and Kegelmeyer [26] showed that importance sampling-based ensembles such as I-vote and DI-vote may obtain also better results with respect to single learners trained on the entire available

[3] These refinements of Lobag have been originally suggested by Tom Dietterich (personal communication), before that our experimental results confirmed the feasibility of this approach.

learning set. In particular Chawla et al. showed that this ensemble approach may improve accuracy by enhancing diversity between base learners, even if stable classifiers, such as Naive-Bayes, are used. We suppose that this approach may also reduce the bias component of error: indeed at each iteration the base learner focuses on the examples currently misclassified by the ensemble, in a way similar to arcing and boosting algorithms. In order to quantitatively evaluate the above hypothesis, an interesting experimental work could consists of explicitly analyzing the bias-variance decomposition of error in I-vote and DI-vote ensembles.

It is worth noting that other approaches may be more appropriate than random subsampling when very large data sets are available: as shown by Chawla et al., simply partitioning the original data into a set of disjoint partitions, we may obtain significantly better results with large databases [46]. We suppose that this approach may reduce the bias component more than random aggregating, as the effective size of the samples is larger and all the available information is used by the ensemble. Of course, to verify this hypothesis, we need to explicitly analyze the bias–variance characteristics of the “partition and aggregate” ensemble method.

VI. Conclusions

We conducted an extensive experimental analysis of bias–variance decomposition of error in random aggregated and bagged ensembles of SVMs, involving training an testing of more than 10 millions of SVMs.

Considering random aggregated ensembles, the most important fact we can observe consists of a very large reduction of net-variance. It is always strongly reduced, independently of the type of kernel used. This behavior is primarily due to the unbiased variance reduction, while bias remains unchanged with respect to single SVMs (Fig. 9, 10, 11).

Random aggregating shows a behavior very close to that predicted by theory (Sect. II-A), at least if well-tuned base learners are used: very low variance and bias unchanged with respect to single base learners.

On the other hand, experimental results confirm that bagging can be interpreted as an approximation of random aggregating, because net-variance is reduced, but not canceled by bootstrap aggregating techniques, while bias remains unchanged or slightly increases. Indeed our experiments showed that with random aggregating we can expect an error reduction from 10 to 70 % (at least for relatively small samples), due to the reduction of the unbiased variance to more than 90 %, while in bagging error reduction is limited to about 15 %, as a smaller reduction of the unbiased variance is achieved.

The characterization of bias–variance decomposition of error presented in [20] for single SVMs, also holds for bagged and RA ensembles of SVMs: the main characteristics are maintained, with an overall reduction of the variance component.

Enlarging the sample size, the absolute values of error, bias and variance tend to converge to the same values in bagged and RA SVMs. On the other hand, if we consider the relative reduction of error, bias and variance of RA ensembles with respect to single SVMs, unbiased and net-variance reduction remain constant and very large, independently of the sample size, according to Breiman’s theory. On the contrary, with bagging the relative variance reduction depends on the size of the samples and it is in general lower w.r.t. RA ensembles.

Our experiments with RA SVMs show also that ensembles built on small samples work reducing variance, and suggest new research directions to improve Lobag.

Acknowledgments

This work has been developed in the context of the *CIMAINA* Center of Excellence. We would like to thank the anonymous reviewers for their valuable comments and suggestions. We would also like to thank Philip Grew for proofreading the manuscript.

Appendix I
Supplementary material available on the web

The full experimental results are subdivided into three downloadable papers:

1) Bias–Variance decomposition of error in Random aggregated SVM ensembles: results and graphics:
`http://homes.dsi.unimi.it/∼valenti/papers/BV/bv-svm-RA.pdf`
2) Bias–Variance decomposition of error in bagged SVM ensembles: results and graphics.
`http://homes.dsi.unimi.it/∼valenti/papers/BV/bv-svm-bagging.pdf`
3) Bias–Variance decomposition of error in bagged and random aggregated ensemble of SVMs, while varying the cardinality of the data: results and graphics:
`http://homes.dsi.unimi.it/∼valenti/papers/BV/bv-card.pdf`

References

[1] J. Kittler, M. Hatef, R. Duin, and J. Matas, “On combining classifiers,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 226–239, 1998.

[2] T. Dietterich, “Ensemble methods in machine learning,” in *Multiple Classifier Systems. First International Workshop, MCS 2000, Cagliari, Italy*, ser. Lecture Notes in Computer Science, J. Kittler and F. Roli, Eds., vol. 1857. Springer-Verlag, 2000, pp. 1–15.

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

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

[5] E. Kleinberg, “On the Algorithmic Implementation of Stochastic Discrimination,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 5, pp. 473–490, 2000.

[6] L. Breiman, “Bias, variance and arcing classifiers,” Statistics Department, University of California, Berkeley, CA, Tech. Rep. TR 460, 1996.

[7] J. Friedman and P. Hall, “On Bagging and Nonlinear Estimation,” Statistics Department, University of Stanford, CA, Tech. Rep. Tech. Report, 2000.

[8] S. Geman, E. Bienenstock, and R. Doursat, “Neural networks and the bias-variance dilemma,” *Neural Computation*, vol. 4, no. 1, pp. 1–58, 1992.

[9] J. Friedman, “On bias, variance, 0/1 loss and the curse of dimensionality,” *Data Mining and Knowledge Discovery*, vol. 1, pp. 55–77, 1997.

[10] T. Heskes, “Bias/Variance Decompostion for Likelihood-Based Estimators,” *Neural Computation*, vol. 10, pp. 1425–1433, 1998.

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

[12] G. James, “Variance and bias for general loss function,” *Machine Learning*, no. 2, pp. 115–135, 2003.

[13] K. Tumer and J. Ghosh, "Error correlation and error reduction in ensemble classifiers," *Connection Science*, vol. 8, no. 3/4, pp. 385–404, 1996.

[14] Y. Liu and X. Yao, "Simultaneous training of negatively correlated neural networks in an ensemble," *IEEE Transactions on Systems, Man and Cybernetics, Part B*, vol. 29, no. 6, pp. 716–725, 1999.

[15] E. Bauer and R. Kohavi, "An empirical comparison of voting classification algorithms: Bagging, Boosting and variants," *Machine Learning*, vol. 36, no. 1/2, pp. 105–139, 1999.

[16] Z. Zhou, J. Wu, and W. Tang, "Ensembling nerual networks: Many could be better than all," *Artificial Intelligence*, vol. 137, no. 1/2, pp. 239–263, 2002.

[17] G. Valentini and T. Dietterich, "Bias–variance analysis and ensembles of SVM," in *Multiple Classifier Systems. Third International Workshop, MCS2002, Cagliari, Italy*, ser. Lecture Notes in Computer Science, vol. 2364. Springer-Verlag, 2002, pp. 222–231.

[18] ——, "Low Bias Bagged Support Vector Machines," in *Machine Learning, Proceedings of the Twentieth International Conference (ICML 2003)*, T. Fawcett and N. Mishra, Eds. Washington D.C., USA: AAAI Press, 2003, pp. 752–759.

[19] C. Cortes and V. Vapnik, "Support vector networks," *Machine Learning*, vol. 20, pp. 273–297, 1995.

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

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

[22] H. Kim, S. Pang, H. Je, D. Kim, and S. Bang, "Pattern Classification Using Support Vector Machine Ensemble," in *Proc. of ICPR'02*, vol. 2. IEEE, 2002, pp. 20 160–20 163.

[23] R. Collobert, S. Bengio, and Y. Bengio, "A Parallel Mixture of SVMs for Very Large Scale Problems," *Neural Computation*, vol. 14, no. 5, pp. 1105–1114, 2002.

[24] G. Valentini, M. Muselli, and F. Ruffino, "Bagged Ensembles of SVMs for Gene Expression Data Analysis," in *IJCNN2003, The IEEE-INNS-ENNS International Joint Conference on Neural Networks*. Portland, USA: IEEE, 2003, pp. 1844–49.

[25] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

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

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

[28] N. Chawla, L. Hall, K. Bowyer, T. Moore, and W. Kegelmeyer, "Distributed pasting of small votes," in *Multiple Classifier Systems. Third International Workshop, MCS2002, Cagliari, Italy*, ser. Lecture Notes in Computer Science, vol. 2364. Springer-Verlag, 2002, pp. 52–61.

[29] T. Evgeniou, L. Perez-Breva, M. Pontil, and T. Poggio, "Bounds on the Generalization Performance of Kernel Machine Ensembles," in *Proc. of the Seventeenth International Conference on Machine Learning (ICML 2000)*, P. Langley, Ed. Morgan Kaufmann, 2000, pp. 271–278.

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

[31] L. Breiman, "Arcing classifiers," *The Annals of Statistics*, vol. 26, no. 3, pp. 801–849, 1998.

[32] G. Valentini, "Ensemble methods based on bias–variance analysis," Ph.D. dissertation, DISI, Dipartimento di Informatica e Scienze dell' Informazione, Università di Genova, Genova, Italy, 2003, ftp://ftp.disi.unige.it/person/ValentiniG/Tesi/finalversion/vale-th-2003-04.pdf.

[33] E. Kong and T. Dietterich, "Error - correcting output coding correct bias and variance," in *The XII International Conference on Machine Learning*. San Francisco, CA: Morgan Kauffman, 1995, pp. 313–321.

[34] R. Kohavi and D. Wolpert, "Bias plus variance decomposition for zero-one loss functions," in *Proc. of the Thirteenth International Conference on Machine Learning*. San Mateo, CA: Morgan Kaufmann, 1996, pp. 275–283.

[35] R. Tibshirani, "Bias, variance and prediction error for classification rules," Department of Preventive Medicine and Biostatistics and Department od Statistics, University of Toronto, Toronto, Canada, Tech. Rep., 1996.

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

[37] G. Valentini and F. Masulli, "NEURObjects: an object-oriented library for neural network development," *Neurocomputing*, vol. 48, no. 1–4, pp. 623–646, 2002.

[38] C. Merz and P. Murphy, "UCI repository of machine learning databases," Irvine, CA, 1998, www.ics.uci.edu/mlearn/MLRepository.html.

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

[40] T. Joachims, “Making large scale SVM learning practical,” in *Advances in Kernel Methods - Support Vector Learning*, S. A. Scholkopf B., Burges C., Ed. Cambridge, MA: MIT Press, 1999, pp. 169–184.

[41] F. Girosi, M. Jones, and T. Poggio, “Regularization theory and neural networks architecture,” *Neural Computation*, no. 7, pp. 219–269, 1995.

[42] D. Marginenatu and T. Dietterich, “Prununig adaptive boosting,” in *Proc. of the 14th International Conference on Machine Learning*. San Francisco, USA: Morgan Kaufmann, 1997, pp. 378–387.

[43] G. Giacinto and F. Roli, “Design of effective neural network ensembles for image classification processes,” *Image Vision and Computing Journal*, vol. 19, no. 9/10, pp. 699–707, 2001.

[44] F. Roli, G. Giacinto, and G. Vernazza, “Methods for Designing Multiple Classifier Systems,” in *Multiple Classifier Systems. Second International Workshop, MCS 2001, Cambridge, UK*, ser. Lecture Notes in Computer Science, J. Kittler and F. Roli, Eds., vol. 2096. Springer-Verlag, 2001, pp. 78–87.

[45] R. Banfield, L. Hall, K. Bowyer, and W. Kegelmeyer, “A new ensemble diversity measure applied to thinning ensemble,” in *Multiple Classifier Systems. Fourth International Workshop, MCS 2003, Guilford, UK*, ser. Lecture Notes in Computer Science, T. Windeatt and F. Roli, Eds., vol. 2709. Springer-Verlag, 2003, pp. 306–316.

[46] N. Chawla, T. Moore, K. Bowyer, L. Hall, C. Springer, and W. Kegelmeyer, “Bagging is a small-data-set phenomenon,” in *Proc. of the 2001 IEEE Conference on Computer Vision and Pattern recognition, CVPR 2001*, vol. 2, 2001, pp. 684–689.

Giorgio Valentini received the “laurea” degree in Biological Sciences and in Computer Science from the University of Genova, Italy in 1981 and 1999, respectively, and the Ph.D. degree in Computer Science from the DISI, Computer Science Department of the University of Genova in 2003. He is currently Assistant Professor with the DSI, Computer Science Department of the University of Milano, Italy, where he attends to both teaching and research. His research interests include machine learning, in particular ensembles of learning machines and bioinformatics. He is member of the International Neural Network Society and of the International Society of Computational Biology.

LIST OF FIGURES

LIST OF TABLES

List of footnotes

1) The extended version of the *NEURObjects* library is available at: `http://homes.dsi.unimi.it/∼valenti/sw/NEURObjects`
2) The application `gensimple`, that we developed to generate the data, is available on line at: `ftp://ftp.disi.unige.it/person/ValentiniG/BV/gensimple`.
3) These refinements of Lobag have been originally suggested by Tom Dietterich (personal communication), before that our experimental results confirmed the feasibility of this approach.