Random aggregated and bagged ensembles of SVMs: an empirical bias–variance analysis

Giorgio Valentini

DSI - Dipartimento di Scienze dell'Informazione,
Università degli Studi di Milano, Via Comelico 39, Milano, Italy,
valentini@dsi.unimi.it

Abstract. Bagging can be interpreted as an approximation of random aggregating, an ideal ensemble method by which base learners are trained using data sets randomly drawn according to an unknown probability distribution. An approximate realization of random aggregating can be obtained through subsampled bagging, when large training sets are available. In this paper we perform an experimental bias–variance analysis of bagged and random aggregated ensembles of Support Vector Machines, in order to quantitatively evaluate their theoretical variance reduction properties. Experimental results with small samples show that random aggregating, implemented through subsampled bagging, reduces the variance component of the error by about 90%, while bagging, as expected, achieves a lower reduction. Bias–variance analysis explains also why ensemble methods based on subsampling techniques can be successfully applied to large data mining problems.

1 Introduction

Random aggregating is a process by which base learners, trained on samples drawn accordingly to an unknown probability distribution from the entire universe population, are aggregated through majority voting (classification) or averaging between them (regression).

Random aggregating is only a theoretical ensemble method. When large data sets are available, *subsampled bagging* can simulate random aggregating, using an uniform probability distribution and resampling techniques, assuming that the large available data set and the uniform probability distribution are good approximations respectively of the universe population and of the unknown probability distribution.

Breiman showed that in regression problems, random aggregation of predictors always improves the performance of single predictors, while in classification problems this is not always the case, if poor base predictors are used [1]. The improvement depends on the stability of the base learner: random aggregating and bagging are effective with unstable learning algorithms, that is when small changes in the training set can result in large changes in the predictions of the base learners [3].

Breiman showed also that bootstrap aggregating (*bagging*) techniques improve the accuracy of a single predictor reducing mainly the variance component of the error [1, 2].

Recently bias–variance decomposition of the error has been applied as a tool to study the behaviour of learning algorithms and to develop new ensemble methods well-suited to the bias–variance characteristics of the base learners [4]. In this paper we extend to bagged and random aggregated ensembles of Support Vector Machines (SVMs) the bias–variance analysis performed on single SVMs [5].

The main purpose of this contribution consists in quantitatively evaluating the variance reduction properties of both random aggregating and bagging, through an extended experimental bias–variance decomposition of the error. In this way we can quantitatively verify the theoretical results obtained by Breiman about the variance reduction properties of random aggregating, and we can also understand the extents to which Breiman's results obtained for random aggregating can be extended to bagging.

The experimental results suggest that subsampled bagged ensembles of SVMs could be successfully applied to very large scale data mining problems, as they significantly reduce the variance component of the error with respect to a single Support Vector Machine. In order to verify this hypothesis, we performed some preliminary experiments on a large synthetic data set, comparing the accuracy and the computational time of single SVMs trained on the entire large training set with ensembles of SVMs trained on small subsamples of the available data.

2 Random aggregating, subsampled and classical bagging

There are close relationships between random aggregating, subsampled and classical bagging: bagging can be interpreted as an approximation of random aggregating. Subsampled bagging, in turn, can be interpreted as an approximate implementation of random aggregating, where the universe population is replaced by a large training set $\mathcal{D}$, and subsamples are randomly drawn from $\mathcal{D}$ according to an uniform probability distribution.

2.1 Random aggregating

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 $(\boldsymbol{x}_j, t_j)$, and $P(\boldsymbol{x}, t)$ is the joint distribution of the data points in U, with $\boldsymbol{x} \in \mathbb{R}^d$.

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 $f_D(\boldsymbol{x}) = y$. Suppose that a sequence of learning sets $\{D_k\}$ is given, each i.i.d. from the same underlying distribution P. Breiman proposed to aggregate the f_D trained with different samples drawn from U to get a better predictor $f_A(\boldsymbol{x}, P)$ [1]. For regression problems $t_j \in \mathbb{R}$ and $f_A(\boldsymbol{x}, P) = E_D[f_D(\boldsymbol{x})]$, 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 $f_A(\boldsymbol{x}, P) = \arg\max_j |\{k | f_{D_k}(\boldsymbol{x}) = j\}|$.

As 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 $f_A(\boldsymbol{x}, P)$ as $f_A(\boldsymbol{x})$. Considering regression problems, if T and $\boldsymbol{X}$ are random variables having joint distribution P, the expected squared loss EL for the single predictor $f_D(\boldsymbol{X})$ is:

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

while the expected squared loss EL_A for the aggregated predictor is:

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

Breiman showed [1] that $EL \geq EL_A$. This disequality depends on the instability of the predictions, that is on how unequal the two sides of eq. 3 are:

$$E_D[f_D(\boldsymbol{X})]^2 \leq E_D[f_D^2(\boldsymbol{X})] \tag{3}$$

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

$$\begin{aligned} V(\boldsymbol{X}) &= E_D[(f_D(\boldsymbol{X}) - E_D[f_D(\boldsymbol{X})])^2] \\ &= E_D[f_D^2(\boldsymbol{X})] - E_D[f_D(\boldsymbol{X})]^2 \end{aligned} \tag{4}$$

Comparing eq. 3 and 4 we see that higher the instability of the base classifiers, higher their variance is. In other words the reduction of the error in random aggregating is due to the reduction of the variance component (eq. 4) of the error, and hence if $V(\boldsymbol{X})$ is quite large we may expect a considerable reduction of the error.

Breiman showed also that in classification problems, as in regression, aggregating relatively good predictors can lead to better performances, as long as the base predictor is unstable, whereas, unlike regression, aggregating poor predictors can lower performances.

2.2 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, that is in turn a sample from the population U. On the contrary f_A uses samples drawn directly from U. Second, bootstrap samples are drawn from D through an uniform probability distribution, which is only an approximation of the unknown true distribution P.

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

2.3 Subsampled bagging

While bagging had been successfully applied to different classification and regression problems [2, 6, 7], random aggregating is almost ideal, because in most cases the true distribution P is unknown and often only small data sets are available.

In order to obtain an approximation of random aggregating, we can approximate the universe U with a large data set $\mathcal{D}$ from which subsampled data, that is samples $D \subset \mathcal{D}$ much smaller than $\mathcal{D}$, are randomly drawn with replacement.

In real problems, if we have very large learning sets, or on-line available learning data, we could use *subsampled bagging* in order to overcome the space complexity problem arising from learning too large data sets, or to allow on-line learning [8]. For instance most of the 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, even if SVMs implemented through decomposition methods can achieve a certain reduction of space and time complexity [9]. As an alternative, we could use relatively small data sets randomly drawn from the large available data set, using random subsampling and aggregation techniques [8, 10].

3 Bias–variance decomposition of the error

Bias–variance decomposition of the error has been recently applied as a tool to analyze learning algorithms and ensemble methods [5, 4]. The main purpose of this analysis consists in discovering relationships between properties and parameters of learning algorithms and ensemble methods with respect to their bias–variance characteristics, in order to gain insights into their learning behaviour and to develop ensemble methods well-tuned to the properties of a specific base learner [11].

Here we present only a brief overview of the main concepts of Domingos' bias–variance decomposition of the error theory [12] necessary to understand the bias–variance analysis of random aggregating and bagging. For more details about the application of Domingos theory to the bias–variance analysis of ensemble methods see [11].

Expected loss depends on the randomness of the training set and the target. Let $\mathcal{L}$ be a learning algorithm, and define $f_D = \mathcal{L}(D)$ as the classifier produced by $\mathcal{L}$ applied to a training set D, where D is a set of m points $(\boldsymbol{x}_j, t_j)$, $t_j \in \mathcal{C}$, $\boldsymbol{x}_j \in \mathbb{R}^d$, $d \in \mathbb{N}$, drawn identically and independently from the "universe" population U according to $P(\boldsymbol{x}, t)$, the joint distribution of the data points in U. The model produces a prediction $f_D(\boldsymbol{x}) = y$. 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 $\boldsymbol{x}$ can be written by considering both the randomness due to the choice of the training set D and the randomness in t due to the choice of a particular test point $(\boldsymbol{x}, t)$: $EL(\mathcal{L}, \boldsymbol{x}) = E_D[E_t[L(t, f_D(\boldsymbol{x}))]]$, 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.

Purpose of bias–variance analysis. It consists in decomposing the 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 $\boldsymbol{x}$ minimizes $E_t[L(t,y)] : y_*(\boldsymbol{x}) = \arg\min_y E_t[L(t,y)]$. The *main prediction* y_m at point $\boldsymbol{x}$ is defined as $y_m = \arg\min_{y'} E_D[L(f_D(\boldsymbol{x}), y')]$. For 0/1 loss, the main prediction is the class predicted most often by the learning algorithm $\mathcal{L}$ when applied to training sets D.

Bias and variance. The *bias* $B(\boldsymbol{x})$ is the loss of the main prediction relative to the optimal prediction: $B(\boldsymbol{x}) = L(y_*, y_m)$. For 0/1 loss, the bias is always 0 or 1. We will say that $\mathcal{L}$ is *biased at point* $\boldsymbol{x}$, if $B(\boldsymbol{x}) = 1$. The *variance* $V(\boldsymbol{x})$ is the average loss of the predictions relative to the main prediction: $V(\boldsymbol{x}) = E_D[L(y_m, f_D(\boldsymbol{x}))]$ It captures the extent to which the various predictions $f_D(\boldsymbol{x})$ vary depending on D.

Unbiased and biased variance. Domingos distinguishes between two opposite effects of variance on the error with classification problems: in the unbiased case variance increases the error, while in the biased case variance decreases the error. As a consequence we can define an *unbiased variance*, $V_u(\boldsymbol{x})$, that is the variance when $B(\boldsymbol{x}) = 0$ and a *biased variance*, $V_b(\boldsymbol{x})$, that is the variance when $B(\boldsymbol{x}) = 1$. Finally we can also define the *net variance* $V_n(\boldsymbol{x})$, to take into account the combined effect of the unbiased and biased variance: $V_n(\boldsymbol{x}) = V_u(\boldsymbol{x}) - V_b(\boldsymbol{x})$

Noise-free decomposition of the 0/1 *loss.* The decomposition for a single point $\boldsymbol{x}$ can be generalized to the entire population by defining $E_{\boldsymbol{x}}[\cdot]$ to be the expectation with respect to $P(\boldsymbol{x})$. Then we can define the *average bias* $E_{\boldsymbol{x}}[B(\boldsymbol{x})]$, the *average unbiased variance* $E_{\boldsymbol{x}}[V_u(\boldsymbol{x})]$, and the *average biased variance* $E_{\boldsymbol{x}}[V_b(\boldsymbol{x})]$. In the noise-free case, the expected loss over the entire population is:

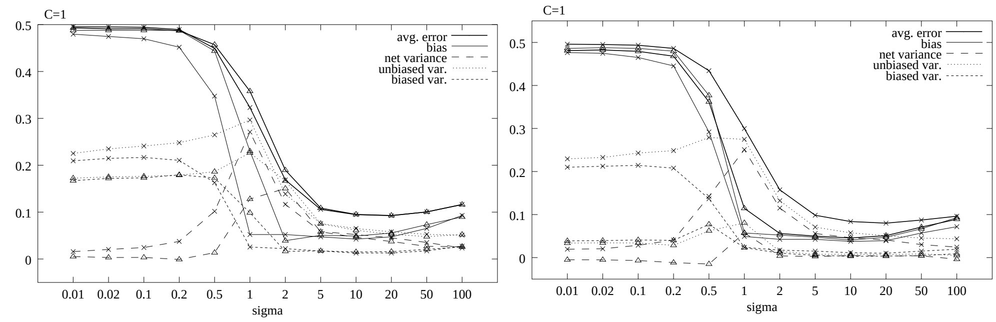
$$E_{\boldsymbol{x}}[EL(\mathcal{L}, \boldsymbol{x})] = E_{\boldsymbol{x}}[B(\boldsymbol{x})] + E_{\boldsymbol{x}}[V_u(\boldsymbol{x})] - E_{\boldsymbol{x}}[V_b(\boldsymbol{x})].$$

4 Experimental bias–variance analysis in bagged and random aggregated ensembles of SVMs

The main purpose of the experiments consists in understanding the effect of bagging and random aggregating on the bias and variance components of the error. We used gaussian, polynomial and dot-product Support Vector Machines (SVMs) as base learners.

4.1 Experimental set–up

For bagging we draw with replacement from a learning set $\mathcal{D}$ $n = 100$ samples D_i of size $s = 100$, according to an uniform probability distribution. From each D_i, $1 \leq i \leq n$, we generate by bootstrap $m = 60$ replicates D_{ij}, collecting them in n different sets $\bar{D}_i = \{D_{ij}\}_{j=1}^m$.



(a) (b)

Fig. 1. Comparison of bias-variance decomposition between single gaussian SVMs (lines labeled with crosses) and gaussian SVM ensembles (lines labeled with triangles), while varying σ and for a fixed values of the regularization parameter ($C = 1$) with the Letter-Two data set: (a) Bagged ensemble (b) Random aggregated ensemble.

We approximated random aggregating through subsampled bagging. In particular we draw with replacement from $\mathcal{D}$ $n = 100$ sets of samples $\bar{D}_i$, according to an uniform probability distribution. Each set of samples $\bar{D}_i = \{D_{ij}\}_{j=1}^{m}$ is composed by $m = 60$ samples D_{ij} drawn with replacement from $\mathcal{D}$, using an uniform probability distribution; each D_{ij} is composed by 100 examples. Note that in this case each sample D_{ij} is directly drawn from $\mathcal{D}$ and not from the samples $D_i \subset \mathcal{D}$. In order to evaluate the bias–variance decomposition of the error for single SVMs, we used 100 samples D_i of size 100.

We computed the error and its decomposition in bias, net–variance, unbiased and biased variance with respect to a separated large testing set $\mathcal{T}$, comparing the bias–variance decomposition of the error in random aggregated, bagged and single SVMs. To perform this computationally intensive analysis (more than 10 millions of single SVMs were trained and tested) we used a cluster of workstations, and we developed new classes and specific C++ applications, using the *NEURObjects* software library [13] and the *SVMlight* applications [9].

4.2 Results

In particular we analyzed the relationships of the components of the error with the kernels and kernel parameters, using data sets from UCI [14] (Waveform, Grey-Landsat, Letter-Two, Letter-Two with added noise, Spam, Musk) and the P2 synthetic data set [1]. We achieved a characterization of the bias–variance decomposition of the error in bagged and random aggregated ensembles that resembles the one obtained for single SVMs [5] (Fig. 1. For more details see [11]). The results show that bagging reduces the error with respect to single SVMs, but

[1] The application `gensimple` for the automatic generation of the P2 data set is available at `ftp://ftp.disi.unige.it/person/ValentiniG/BV/gensimple`.

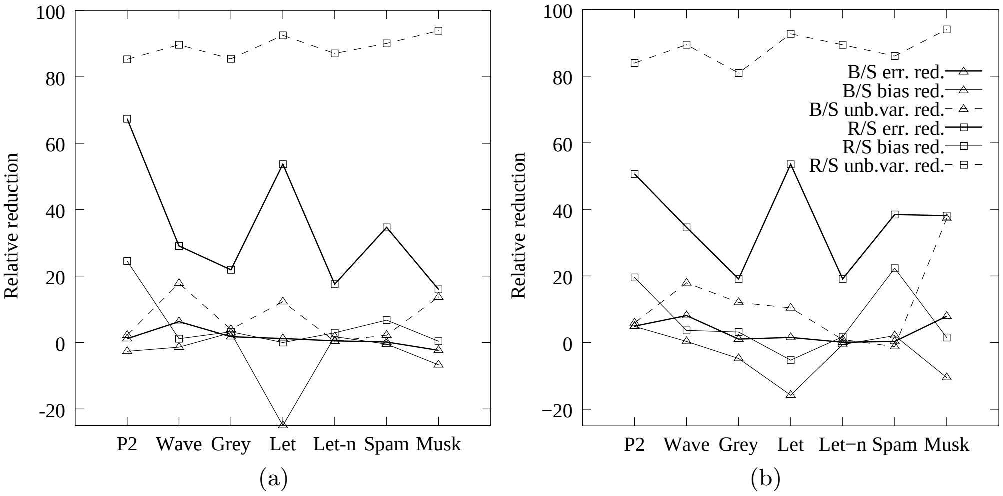


Fig. 2. Comparison of the relative error, bias and unbiased variance reduction between bagged and single SVMs (lines labeled with triangles), and between random aggregated and single SVMs (lines labeled with squares). B/S stands for Bagged versus Single SVMs, and R/S for random aggregated versus Single SVMs. Results refers to 7 different data sets. (a) Gaussian kernels (b) Polynomial kernels.

not so largely as random aggregating. Fig. 2 summarizes the main outcomes of our experiments: bagged ensembles of SVMs achieved a reduction of the relative average error from 0 to 20 % with 35% decrement of the net-variance, while random aggregated ensembles obtained about a 90 % net-variance decrement, with a relative reduction of the error from 20 to 70 %. Note that a negative sign in the relative bias reduction means that bagging with some data sets may increment the bias with respect to single SVMs (Fig. 2).

These results are not surprising: bagging is a variance reduction method, but we cannot expect as a large decrement of the variance as in random aggregating. Indeed with random aggregating the base learners use more variable training sets drawn from U. In this way random aggregating exploits information from the entire population U, while bagging can exploit only the information from a single data set D drawn from U, through bootstrap replicates of the data drawn from D. Note that these results do not show that random aggregating is "better" than bagging, as they are referred to differently sized data sets, but rather they confirm the theoretical variance reduction properties of random aggregating and bagging. Moreover they show that with well-tuned and accurate base learners a very large variance reduction is obtained only with random aggregating (Fig. 2). Anyway note that in our experiments we used small samples: we could expect that with larger samples the differences between random aggregating and bagging will be smaller.

Table 1. Comparison of the results between single and subsampled bagged SVMs (see text).

	$N.samples$	$Error$	$Time$	$\Delta Error$	$Speed-up$
S-SVM	100000	0.0023	5474.68	——	——
SB-SVM	10000	0.0032	4089.82	0.39	1.3
SB-SVM	5000	0.0043	855.85	0.87	6.4
SB-SVM	2000	0.0076	124.93	2.30	43.8
SB-SVM	1000	0.0127	38.49	4.52	142.2
SB-SVM	200	0.0358	2.92	14.57	1874.9
SB-SVM	100	0.0539	1.69	22.43	3239.4

5 Subsampled bagged ensembles of SVMs for large scale classification problems

The bias–variance analysis of random aggregated ensembles shows that the variance component of the error is strongly reduced, while the bias remains unchanged or it is lowered (Fig. 2). Breiman proposed random subsampling techniques for classification in large databases, using decision trees as base learners [8], and these techniques have been also successfully applied in distributed environments [10].

These facts suggest to apply subsampled bagging to large scale classification problems, using SVMs as base learners, considering also that the SVM algorithm, as well as other learning algorithms, does not scale too well when very large samples are available [9].

To get some insights into this hypothesis, we performed a preliminary experiment with the synthetic data set *P2*, using a quite large learning set of 10^5 examples, and comparing the results of single and subsampled bagged SVMs on a separated large testing set. Tab. 1 summarizes the results of the experiments: S-SVM stands for single gaussian SVMs trained on the entire available learning set; SB-SVM stands for Subsampled Bagged SVMs trained on subsamples of the available training set, whose cardinality are shown in the column $N.samples$. The column $Error$ shows the error on the separated test set (composed by 100000 examples); the column $Time$ shows the the training time in seconds, using an AMD Athlon 2000+ processor (1.8 GHz) with 512 Mb RAM. $\Delta Error$ is the relative error increment using an ensemble instead of a single SVM, and the last column represents the speed-up achieved.

The results show that with subsampled bagging we can obtain a consistent speed-up, but with a slight decrement in the accuracy. For instance, we need two minutes for training ensembles with 1/50 of the available data against one hour and a half for a single SVM trained on the entire data set (Tab. 1), with an increment of the error from 0.23 % to 0.76 % (a small absolute difference,but a high relative error increment). Note that if accuracy is not the main goal, we achieve most of the error reduction with about 30 base learners (Fig. 3), and

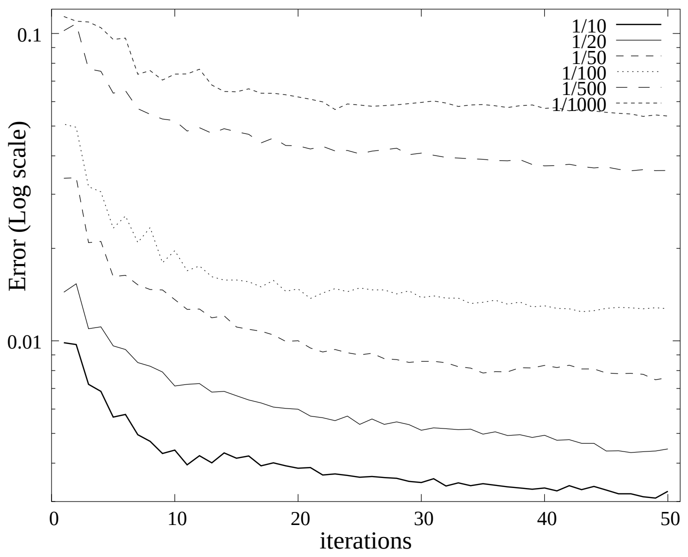


Fig. 3. Error (Log scale) as a function of the number of base learners (gaussian SVM) employed. The different curves refer to ensembles with base learners trained with different fractions of the learning set.

using a parallel implementation we may expect a further speed-up linear in the number of the base learners.

These results confirm similar ones obtained in [15]. Moreover the bias–variance analysis explains the behaviour of subsampled bagging when small learning sets are used: the increment of the bias due to the reduction of the size of the samples is partially counterbalanced by the reduction of the net–variance (data not shown), and on the other hand a substantial speed-up is obtained.

6 Conclusions

The extensive experimental analysis of bias–variance decomposition of the error in random aggregated and bagged ensembles of SVMs shows a consistent reduction of the net-variance with respect to single SVMs. This behaviour is due primarily to the unbiased variance reduction, while the bias remains substantially unchanged.

In our experiments random aggregating always reduces the variance close to 0, while classical bagging reduces the variance only of about 1/3, confirming the theoretical results of Breiman, and providing a quantitative estimate of the variance reduction both in bagged and random aggregated ensembles of SVMs.

Preliminary results show also that subsampled bagging, as an approximation of random aggregating, can be in practice applied to classification problems when single SVMs cannot comfortably manage very large data sets, if accuracy is not the main concern.

Acknowledgments

This work has been funded by the italian COFIN project *Linguaggi formali ed automi: metodi, modelli ed applicazioni.*

References

1. Breiman, L.: Bagging predictors. Machine Learning **24** (1996) 123–140
2. Dietterich, T.: An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting and randomization. Machine Learning **40** (2000) 139–158
3. Bousquet, O., Elisseeff, A.: Stability and Generalization. Journal of Machine Learning Research **2** (2002) 499–526
4. Valentini, G., Dietterich, T.G.: Bias–variance analysis of Support Vector Machines for the development of SVM-based ensemble methods. Journal of Machine Learning Research (accepted for publication)
5. Valentini, G., Dietterich, T.: Bias–variance analysis and ensembles of SVM. In: MCS2002, Cagliari, Italy. Vol. 2364 of Lecture Notes in Computer Science., Springer-Verlag (2002) 222–231
6. Andersen, T., Rimer, M., Martinez, T.R.: Optimal artificial neural network architecture selection for voting. In: Proc. of the IEEE International Joint Conference on Neural Networks IJCNN'01, IEEE (2001) 790–795
7. Kim, H., Pang, S., Je, H., Kim, D., Bang, S.: Pattern Classification Using Support Vector Machine Ensemble. In: Proc. of ICPR'02. Vol. 2., IEEE (2002) 20160–20163
8. Breiman, L.: Pasting Small Votes for Classification in Large Databases and On-Line. Machine Learning **36** (1999) 85–103
9. Joachims, T.: Making large scale SVM learning practical. In Scholkopf B., Burges C., S.A., ed.: Advances in Kernel Methods - Support Vector Learning. MIT Press, Cambridge, MA (1999) 169–184
10. Chawla, N., Hall, L., Bowyer, K., Moore, T., Kegelmeyer, W.: Distributed pasting of small votes. In: MCS2002, Cagliari, Italy. Vol. 2364 of Lecture Notes in Computer Science., Springer-Verlag (2002) 52–61
11. Valentini, G.: Ensemble methods based on bias–variance analysis. PhD thesis, DISI, Università di Genova, Italy (2003), ftp://ftp.disi.unige.it/person/ValentiniG/Tesi/finalversion/vale-th-2003-04.pdf.
12. Domingos, P.: A Unified Bias-Variance Decomposition for Zero-One and Squared Loss. In: Proc. of the Seventeenth National Conference on Artificial Intelligence, Austin, TX, AAAI Press (2000) 564–569
13. Valentini, G., Masulli, F.: NEURObjects: an object-oriented library for neural network development. Neurocomputing **48** (2002) 623–646
14. Merz, C., Murphy, P.: UCI repository of machine learning databases (1998) www.ics.uci.edu/mlearn/MLRepository.html.
15. Evgeniou, T., Perez-Breva, L., Pontil, M., Poggio, T.: Bounds on the Generalization Performance of Kernel Machine Ensembles. In Langley, P., ed.: Proc. of the Seventeenth International Conference on Machine Learning (ICML 2000), Morgan Kaufmann (2000) 271–278