

Leave one out error, stability, and generalization of voting combinations of classifiers

Theodoros Evgeniou

Technology Management Department,
INSEAD,
Boulevard de Constance, 77305 Fontainebleau, France
`theodoros.evgeniou@insead.fr`

Massimiliano Pontil

DII - University of Siena
Via Roma 56, 53100 Siena, Italy
`massi@dii.unisi.it`

André Elisseeff

Biowulf Technologies, Savannah, GA
`andre@barnhilltechnologies.com`

Abstract

We study the generalization error of voting combinations of learning machines. A special case considered is bagging. We analyze in detail combinations of kernel machines, such as support vector machines, and present theoretical bounds on their generalization error using leave one out error estimates. We also derive novel bounds on the stability of combinations of any classifiers. These bounds can be used to formally show that, for example, bagging increases the stability of unstable learning machines. As a special case we study the stability and generalization of bagging kernel machines and report experiments validating the theoretical findings.

1 Introduction

Studying the generalization performance of ensembles of learning machines has been the topic of ongoing research in recent years [3, 16, 9]. There is a lot of experimental work showing that combining learning machines, for example using boosting or bagging methods [3, 16], very often leads to improved generalization performance. A number of theoretical explanations have also been proposed [16, 3], but more work on this aspect is still needed. Two important theoretical tools for studying the generalization performance of learning machines are the leave-one-out (or cross validation) error of the machines, and the stability of the machines [2, 1]. The second, although an older tool [7, 6], has become only important recently with the work of [11, 2].

In this paper we study the generalization performance of ensembles of kernel machines using both leave-one-out and stability arguments. We consider the general case where each of the machines in the ensemble uses a different kernel and different subsets of the training set. The ensemble is a convex combination of the individual machines. A particular case of this scheme is that of bagging kernel machines. Unlike “standard” bagging [3], this paper considers combinations of the real outputs of the classifiers, and each machine is trained on a different (possibly small) subset of the initial training set. This subset can be chosen by deterministically or randomly subsampling from the initial training set. Each machine in the ensemble uses in general a different kernel. As a special case, appropriate choices of these kernels lead to machines that may use different subsets of the initial input features, or different input representations in general.

We derive theoretical bounds for the generalization error of the ensembles based on the leave-one-out error estimate. We also present results on the stability of combinations of classifiers, which we apply to the case of bagging kernel machines. They can also be applied to bagging learning machines other than kernel machines, showing formally that bagging can increase the stability of the learning machines when these are not stable, and decrease it otherwise. An implication of this result is that it can be easier to control the generalization error of bagging machines: for example the leave one out error is a better estimate of their test error, something that we experimentally validate.

The paper is organized as follows. Section 2 gives the basic notation and background. In section 3 we present bounds for the leave-one-out error of kernel machine ensembles. These bounds are used for model selection experiments in section 4. In section 5 we discuss the algorithmic stability of ensembles, and present a formal analysis of how bagging influences the stability of learning machines. The results can also provide a justification of the experimental findings of section 4. Section 6 discusses other ways of combining learning machines.

2 Background and Notations

In this section we recall the main features of kernel machines. For a more detailed account see [17, 15, 8, 5]. For an account consistent with our notation see [8].

Kernel machine classifiers are the minimizers of functionals of the form:

$$H[f] = \frac{1}{\ell} \sum_{i=1}^{\ell} V(y_i, f(\mathbf{x}_i)) + \lambda \|f\|_K^2 \quad (1)$$

where we use the following notation:

- The pair $(\mathbf{x}_i, y_i) \in \mathbb{R}^n \times \{-1, 1\}$ is sampled according to an unknown probability distribution $P(\mathbf{x}, y)$. The set $D_\ell = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_\ell, y_\ell)\}$ is the training set.
- f is a function $\mathbb{R}^n \rightarrow \mathbb{R}$ belonging to a Reproducing Kernel Hilbert Space (RKHS) \mathcal{H} defined by kernel K , and $\|f\|_K^2$ is the norm of f in this space. See [17, 18] for a number of kernels. The classification is done by taking the sign of this function.
- $V(y, f(\mathbf{x}))$ is the loss function. The choice of this function determines different learning techniques, each leading to a different learning algorithms (for computing the coefficients α_i - see below).
- λ is called the regularization parameter and is a positive constant.

Machines of this form have been motivated in the framework of statistical learning theory. Under rather general conditions the solution of equation (1) is of the form

$$f(\mathbf{x}) = \sum_{i=1}^{\ell} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}), \quad (2)$$

The coefficients α_i in eq. (2) are learned by solving the following optimization problem:

$$\begin{aligned} \max_{\alpha} H(\alpha) &= \sum_{i=1}^{\ell} S(\alpha_i) - \frac{1}{2} \sum_{i,j=1}^{\ell} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \\ \text{subject to : } &0 \leq \alpha_i \leq C, \quad i = 1, \dots, \ell, \end{aligned} \quad (3)$$

where $S(\cdot)$ is a convex function and $C = \frac{1}{2\ell\lambda}$ a constant. SVM are a particular case of these machines for $S(\alpha) = \alpha$. This corresponds to a loss function V in (1) that is of the form $\theta(1 - yf(\mathbf{x}))(1 + yf(\mathbf{x}))$, where θ is the Heavyside function: $\theta(x) = 1$ if $x > 0$, and zero otherwise. The points for which $\alpha_i > 0$ are called support vectors.

2.1 Kernel Machine Ensembles

We consider the general case where each of the machines in the ensemble uses a different kernel and different subsets of the training set. Let T be the number of machines, $K^{(t)}$ the kernel used by machine t , and $f^{(t)}(\mathbf{x})$ the optimal solution of machine t . We also denote by $\alpha_i^{(t)}$ the optimal weight that machine t assigns to point (\mathbf{x}_i, y_i) (after solving - optimizing - problem (3)). We consider ensembles that are convex combinations of the individual machines: The separating surface of the ensemble is given by

$$F(\mathbf{x}) = \sum_{t=1}^T c_t f^{(t)}(\mathbf{x}) \quad (4)$$

with $c_t \geq 0$, and $\sum_{t=1}^T c_t = 1$ (for scaling reasons). The coefficients c_t are not learned and all parameters (C 's and kernels) are fixed before training. The classification is done by taking the sign of $F(\mathbf{x})$.

2.2 Leave-One-Out Error

Given a learning algorithm - such as SVM or an ensemble of SVM - we define f_{D_ℓ} to be the solution of the algorithm when the training set $D_\ell = \{(\mathbf{x}_i, y_i), i = 1, \dots, \ell\}$ is used. We denote by D_ℓ^i the training set obtained by removing point (\mathbf{x}_i, y_i) from D_ℓ , that is the set $D_\ell \setminus \{(\mathbf{x}_i, y_i)\}$. When it is clear in the text we will denote f_{D_ℓ} by f and $f_{D_\ell^i}$ by f_i . If θ is, as before, the Heavyside function, then the leave-one-out error is defined by

$$Loo(D_\ell) = \frac{1}{\ell} \sum_{i=1}^{\ell} \theta(-y_i f_i(x_i)).$$

The leave-one-out error provides an estimate of the average generalization performance of a machine. The following theorem says that the expectation of the generalization error of a machine trained using ℓ points is equal to the expectation of the Loo -error of a machine trained on $\ell + 1$ points. This is summarized by the following theorem, originally due to Luntz and Brailovsky (see [17]).

Theorem 2.1 *Suppose f_{D_ℓ} is the outcome of a determinist learning algorithm. Then*

$$E_{D_\ell} [E_{x,y}[\theta(-y f_{D_\ell}(x))]] = E_{D_{\ell+1}} [Loo(D_{\ell+1})]$$

As observed [11], this theorem can be extended to general learning algorithms by adding a randomizing preprocessing step.

3 Leave-One-Out Error Estimates of Kernel Machine Ensembles

We begin with some known results about the leave-one-out error of kernel machines. The following theorem is from [10]:

Theorem 3.1 *The leave-one-out error of a kernel machine (3) is upper bounded as:*

$$Loo(D_\ell) \leq \frac{1}{\ell} \sum_{i=1}^{\ell} \theta(\alpha_i K(\mathbf{x}_i, \mathbf{x}_i) - y_i f(\mathbf{x}_i)) \quad (5)$$

where f is the optimal function found by solving maximization problem (3) on the whole training set.

In the particular case of SVM where the data are separable the r.h.s of equation (5) can be bounded by geometric quantities, namely [17]:

$$Loo(D_\ell) \leq \frac{1}{\ell} \sum_{i=1}^{\ell} \theta(\alpha_i K(\mathbf{x}_i, \mathbf{x}_i) - y_i f(\mathbf{x}_i)) \leq \frac{1}{\ell} \frac{d_{sv}^2}{\rho^2} \quad (6)$$

where d_{sv} is the radius of the smallest sphere in the feature space induced by kernel K [18, 17] centered at the origin containing the support vectors, that is $d_{sv} = \max_{i:\alpha_i>0} K(\mathbf{x}_i, \mathbf{x}_i)$, and ρ is the margin ($\rho^2 = \frac{1}{\|f\|_K^2}$) of the SVM.

Using this result, the following theorem is a direct application of the Luntz and Brailovsky theorem [17]:

Theorem 3.2 *Suppose that the data is separable by the SVM. Then, the average generalization error of an SVM trained on ℓ points is upper bounded by*

$$\frac{1}{\ell + 1} E \left(\frac{d_{sv}^2(\ell)}{\rho^2(\ell)} \right),$$

where the expectation E is taken with respect to the probability of a training set of size ℓ .

Notice that this result shows that the performance of SVM does not depend only on the margin, but also on other geometric quantities, namely the radius d_{sv} .

We now extend these results to the case of ensembles of kernel machines. In the particular case of bagging, the subsampling of the training data should be deterministic. By this we mean that when the bounds on the leave one out error are used for model (parameter) selection, for each model the same subsample sets of the data need to be used. These subsamples, however, are still random ones. We believe that the results presented below also hold (with minor modifications) in the general case that the subsampling is always random. We now consider the leave-one-out error of such ensembles.

Theorem 3.3 *The leave-one-out error of a kernel machines ensemble is upper bounded by:*

$$Loo(D_\ell) \leq \frac{1}{\ell} \sum_{i=1}^{\ell} \theta \left(\sum_{t=1}^T c_t \alpha_i^{(t)} K^{(t)}(\mathbf{x}_i, \mathbf{x}_i) - y_i F(\mathbf{x}_i) \right). \quad (7)$$

The proof of this theorem is based on the following lemma shown in [17, 10]:

Lemma 3.1 *Let α_i be the coefficient of the solution $f(\mathbf{x})$ of machine (3) corresponding to point (\mathbf{x}_i, y_i) , $\alpha_i > 0$. Let $f_i(\mathbf{x})$ be the solution of machine (3) found when the data point (\mathbf{x}_i, y_i) is removed from the training set. Then: $y_i f_i(\mathbf{x}_i) \geq y_i f(\mathbf{x}_i) - \alpha_i K(\mathbf{x}_i, \mathbf{x}_i)$.*

Using lemma 3.1 we can now prove theorem 3.3.

Proof of theorem 3.3: Let $F_i(\mathbf{x}) = \sum_{t=1}^T c_t f_i^{(t)}(\mathbf{x})$ be the ensemble machine trained with all initial training data except (\mathbf{x}_i, y_i) . Lemma 3.1 gives that

$$\begin{aligned} y_i F_i(\mathbf{x}_i) &= y_i \sum_{t=1}^T c_t f_i^{(t)}(\mathbf{x}_i) \geq \sum_{t=1}^T c_t \left[y_i f^{(t)}(\mathbf{x}_i) - \alpha_i^{(t)} K^{(t)}(\mathbf{x}_i, \mathbf{x}_i) \right] = \\ &= y_i F(\mathbf{x}_i) - \sum_{t=1}^T c_t \alpha_i^{(t)} K^{(t)}(\mathbf{x}_i, \mathbf{x}_i) \Rightarrow \theta(-y_i F_i(\mathbf{x}_i)) \leq \theta \left(\sum_{t=1}^T c_t \alpha_i^{(t)} K^{(t)}(\mathbf{x}_i, \mathbf{x}_i) - y_i F(\mathbf{x}_i) \right), \end{aligned}$$

therefore the leave one out error $\sum_{i=1}^{\ell} \theta(-y_i F_i(\mathbf{x}_i))$ is not more than

$$\sum_{i=1}^{\ell} \theta\left(\sum_{t=1}^T c_t \alpha_i^{(t)} K^{(t)}(\mathbf{x}_i, \mathbf{x}_i) - y_i F(\mathbf{x}_i)\right),$$

which proves the theorem. \square

Notice that the bound has the same form as the bound in eq. (5): for each point (\mathbf{x}_i, y_i) we only need to take into account its corresponding parameter $\alpha_i^{(t)}$ and “remove” the effects of $\alpha_i^{(t)}$ from the value of $F(\mathbf{x}_i)$.

The leave-one-out error can also be bounded using geometric quantities. To this purpose we introduce one more parameter that we call the *ensemble margin* (in contrast to the margin of a single SVM). For each point (\mathbf{x}_i, y_i) we define its ensemble margin to be $y_i F(\mathbf{x}_i)$. This is exactly the definition of margin in [16]. For any given $\delta > 0$ we define E_δ to be the empirical error with ensemble margin less than δ :

$$E_\delta = \frac{1}{\ell} \sum_{i=1}^{\ell} \theta(-y_i F(\mathbf{x}_i) + \delta).$$

and by N_δ the set of the remaining training points - the ones with ensemble margin $\geq \delta$. Finally, we note by $d_{t(\delta)}$ the radius of the smallest sphere in the feature space induced by kernel $K^{(t)}$ centered at the origin which contains the points of machine t with $\alpha_i^{(t)} > 0$ and ensemble margin larger than δ ¹.

Corollary 3.1 *For any $\delta > 0$ the leave-one-out error of a kernel machines ensemble is upper bounded by:*

$$Loo(D_\ell) \leq E_\delta + \frac{1}{\ell} \left(\frac{1}{\delta} \sum_{t=1}^T c_t d_{t(\delta)}^2 \left(\sum_{i \in N_\delta} \alpha_i^{(t)} \right) \right) \quad (8)$$

Proof: For each training point (\mathbf{x}_i, y_i) with ensemble margin $y_i F(\mathbf{x}_i) < \delta$ we upper bound $\theta(\sum_{t=1}^T c_t \alpha_i^{(t)} K^{(t)}(\mathbf{x}_i, \mathbf{x}_i) - y_i F(\mathbf{x}_i))$ with 1 (this is a trivial bound). For the remaining points (the points in N_δ) we show that:

$$\theta\left(\sum_{t=1}^T c_t \alpha_i^{(t)} K^{(t)}(\mathbf{x}_i, \mathbf{x}_i) - y_i F(\mathbf{x}_i)\right) \leq \frac{1}{\delta} \left(\sum_{t=1}^T c_t \alpha_i^{(t)} K^{(t)}(\mathbf{x}_i, \mathbf{x}_i) \right). \quad (9)$$

In the case that $\sum_{t=1}^T c_t \alpha_i^{(t)} K^{(t)}(\mathbf{x}_i, \mathbf{x}_i) - y_i F(\mathbf{x}_i) < 0$, equation (9) is trivially satisfied. If $\sum_{t=1}^T c_t \alpha_i^{(t)} K^{(t)}(\mathbf{x}_i, \mathbf{x}_i) - y_i F(\mathbf{x}_i) \geq 0$, then

$$\theta\left(\sum_{t=1}^T c_t \alpha_i^{(t)} K^{(t)}(\mathbf{x}_i, \mathbf{x}_i) - y_i F(\mathbf{x}_i)\right) = 1,$$

¹In the case of SVM, these are the support vectors of machine t with ensemble margin larger than δ .

while

$$\sum_{t=1}^T c_t \alpha_i^{(t)} K^{(t)}(\mathbf{x}_i, \mathbf{x}_i) \geq y_i F(\mathbf{x}_i) \geq \delta \Rightarrow \frac{1}{\delta} \sum_{t=1}^T c_t \alpha_i^{(t)} K^{(t)}(\mathbf{x}_i, \mathbf{x}_i) \geq 1.$$

So in both cases inequality (9) holds. Therefore:

$$\begin{aligned} \sum_{i=1}^{\ell} \theta \left(\sum_{t=1}^T c_t \alpha_i^{(t)} K^{(t)}(\mathbf{x}_i, \mathbf{x}_i) - y_i F(\mathbf{x}_i) \right) &\leq \ell E_\delta + \frac{1}{\delta} \left(\sum_{i \in N_\delta} \sum_{t=1}^T c_t K^{(t)}(\mathbf{x}_i, \mathbf{x}_i) \alpha_i^{(t)} \right) \leq \\ &\ell E_\delta + \frac{1}{\delta} \left(\sum_{t=1}^T c_t d_{t(\delta)}^2 \left(\sum_{i \in N_\delta} \alpha_i^{(t)} \right) \right). \end{aligned}$$

The statement of the corollary follows by applying theorem 3.3. \square

Notice that equation (8) holds for any $\delta > 0$, so the best bound is obtained for the minimum of the right hand side with respect to $\delta > 0$. Using the Luntz and Brailovsky theorem, theorems 3.3 and 3.1 provide bounds on the average generalization performance of general kernel machines ensembles like that of theorem 3.2.

We now consider the particular case of SVM ensembles. In this case we have the following

Corollary 3.2 *Suppose that each SVM in the ensembles separated the data set used during training. Then, the leave-one-out error of an ensemble of SVM is upper bounded by:*

$$Loo(D_\ell) \leq E_1 + \frac{1}{\ell} \sum_{t=1}^T c_t \frac{d_t^2}{\rho_t^2} \quad (10)$$

where E_1 is the margin empirical error with ensemble margin 1, d_t is the radius of the smallest sphere centered at the origin, in the feature space induced by kernel $K^{(t)}$, containing the support vectors of machine t , and ρ_t is the margin of SVM t .

Proof: We chose $\delta = 1$ in (8). Clearly we have that $d_t \geq d_{t(\delta)}$ for any δ , and $\sum_{i \in N_\delta} \alpha_i^{(t)} \leq \sum_{i=1}^{\ell} \alpha_i^{(t)} = \frac{1}{\rho_t^2}$ (see [17] for a proof of this equality). \square

Notice that the average generalization performance of the SVM ensemble now depends on the “average” (convex combination of) $\frac{D^2}{\rho^2}$ of the individual machines. In some cases this may be smaller than the $\frac{D^2}{\rho^2}$ of a single SVM. For example, suppose we train many SVMs on different sub-samples of the training points and we want to compare such an ensemble with a single SVM using all the points. If all SVMs (the single one, as well as the individual ones of the ensemble) have most of their training points as support vectors, then clearly the D^2 of each SVM in the ensemble is smaller than that of the single SVM. Moreover the margin of each SVM in the ensemble is expected to be larger than that of the single SVM using all the points. So the “average” $\frac{D^2}{\rho^2}$ in this case is expected to be smaller than that of the single SVM. Another case where an ensemble of SVMs may be better than a single SVM is the one where there are outliers among the training data: if the individual SVMs are trained on subsamples of the training data, some of the machines may have smaller $\frac{D^2}{\rho^2}$ because they do not use some outliers. In general it is not clear when ensembles of kernel machines are

better than single machines. The bounds in this section may provide some insight to this question.

Finally, we remark that all the results discussed hold for the case that there is no bias (threshold b), or the case where the bias is included in the kernel (as discussed in the introduction). In the experiments discussed below we use the results also in the case that the bias is not regularized, which is common in practice. Recent work in [4] may be used to extend our results to ensemble of kernel machines with the bias not regularized.

4 Experiments

To test how tight the bounds we presented are, we conducted a number of experiments using datasets from UCI², as well as the US Postal Service (USPS) dataset [12]. We show results for some of the sets in figures 1-5. For each dataset we split the overall set in training and testing (the sizes are shown in the figures) in 50 different (random) ways, and for each split:

1. We trained one SVM with $b = 0$ using all training data, computed the leave-one-bound given by theorem 3.1, and then compute the test performance using the test set.
2. We repeated (1) this time with with $b \neq 0$.
3. We trained 30 SVMs with $b = 0$ each using a random subsample of size 40% of the training data (bagging), computed the leave-one-out bound given by theorem 3.3 using $c_t = \frac{1}{30}$, and then compute the test performance using the test set.
4. We repeated (3) this time with with $b \neq 0$.

We then averaged over the 50 training-testing splits the test performances and the leave-one-out bounds found, and computed the standard deviations. All machines were trained using a Gaussian kernel, and we repeated the procedure for a number of different σ 's of the Gaussian, and for a *fixed* value of the parameter C . We show the averages and standard deviations of the results in figures 1 to 5. In all figures we use the following notation: top left figure: bagging with $b = 0$; top right figure: single SVM with $b = 0$; bottom left figure: bagging with $b \neq 0$; and bottom right figure: single SVM with $b \neq 0$. In each plot the solid line is the mean test performance and the dashed line is the error bound computed using the leave-one-out theorems (theorems 3.1 and 3.3). The dotted line is the validation set error discussed below. For simplicity, only one error bar (standard deviation over the 50 training-testing splits) is shown (the others were similar). The cost parameter C used is given in each of the figures. The horizontal axis is the natural logarithm of the σ of the Gaussian kernel used, while the vertical axis is the error.

An interesting observation is that *the bounds are always tighter for the case of bagging than they are for the case of a single SVM*. This is an interesting experimental finding for which we provide a possible theoretical explanation in the next section. *This finding can practically justify the use of ensembles of machines for model selection: parameter selection using the leave-one-out bounds presented in this paper is easier for ensembles of machines than it is for single machines.*

²Available from <http://www.ics.uci.edu/mllearn/MLRepository.html>

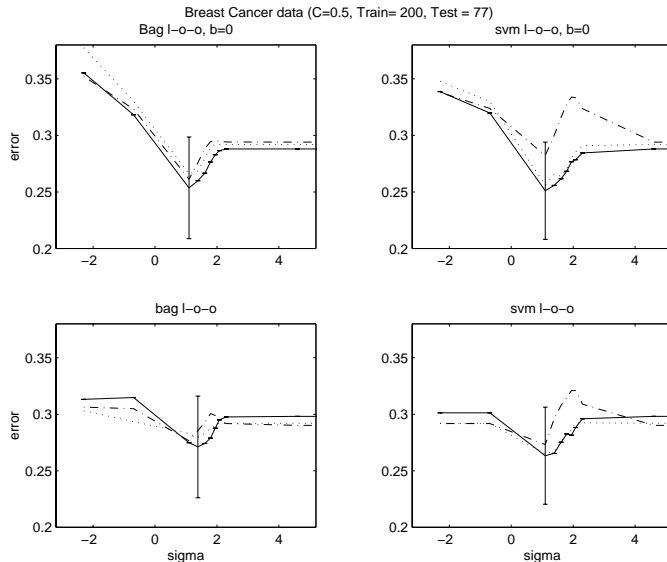


Figure 1: Breast cancer data: see text for description.

Another interesting observation is that the bounds seem to work similarly in the case that the bias b is not 0. In this case, as before, the bounds are tighter for ensembles of machines than they are for single machines.

Experimentally we found that the bounds presented here do not work well in the case that the C parameter used is large. An example is shown in figure 6. Consider the leave-one-out bound for a single SVM given by theorem 3.1. Let (\mathbf{x}_i, y_i) be a support vector for which $y_i f(\mathbf{x}_i) < 1$. It is known [17] that for these support vectors the coefficient α_i is C . If C is such that $CK(\mathbf{x}_i, \mathbf{x}_i) > 1$ (for example consider Gaussian kernel with $K(\mathbf{x}, \mathbf{x}) = 1$ and any $C > 1$), then clearly $\theta(CK(\mathbf{x}_i, \mathbf{x}_i) - y_i f(\mathbf{x}_i)) = 1$. In this case the bound of theorem 3.1 effectively counts *all support vectors outside the margin* (plus some of the ones *on* the margin, i.e. $y f(\mathbf{x}) = 1$). This means that for “large” C (in the case of Gaussian kernels this can be for example for any $C > 1$), the bounds of this paper effectively are similar (not larger than) to another known leave-one-out bound for SVMs, namely one that uses the number of all support vectors to bound generalization performance [17]. So effectively our experimental results show that *the number of support vectors does not provide a good estimate of the generalization performance of the SVMs and their ensembles*.

5 Stability of ensemble methods

We now present a theoretical explanation of the experimental finding that the leave-one-out bound is tighter for the case of ensemble machines than it is for single machines. The analysis is done within the framework of stability and learning [2]. It has been proposed in the past that bagging increases the “stability” of the learning methods [3]. Here we provide a formal argument for this. As before, we denote by D_ℓ^i the training set D_ℓ without example point (\mathbf{x}_i, y_i) . We use the following notion of stability defined in [2]:

Definition: We say that a learning method is β_ℓ -stable with respect to a loss function V

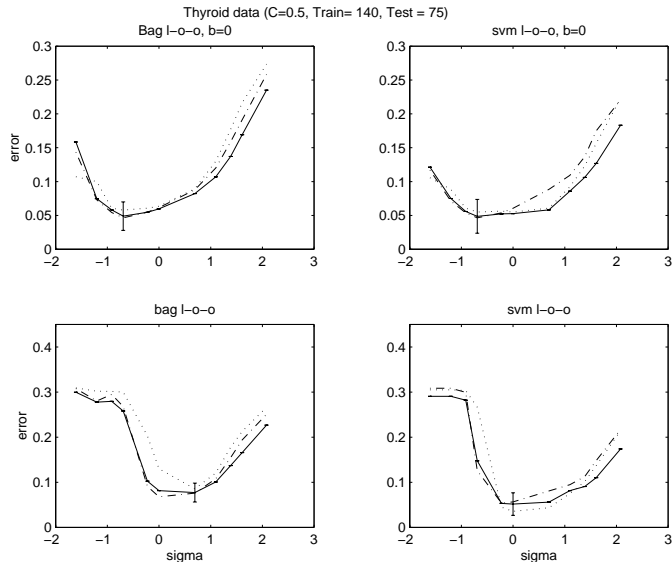


Figure 2: Thyroid data: see text for description.

and training sets of size ℓ if the following holds:

$$\forall i \in \{1, \dots, \ell\}, \forall D_\ell, \forall (\mathbf{x}, y) : |V(f_{D_\ell}(\mathbf{x}), y) - V(f_{D_\ell^i}(\mathbf{x}), y)| \leq \beta_\ell$$

Roughly speaking the cost of a learning machine on a new (test) point (\mathbf{x}, y) should not change more than β_ℓ when we train the machine with any training set of size ℓ and when we train the machine with the same training set but one training point (any point) removed. Notice that this definition is useful mainly for real-valued loss functions V . To use it for classification machines we need to start with the real valued output (2) before thresholding for classification. We define for any given constant δ the leave-one-out error Loo_δ on a training set D_ℓ to be:

$$Loo_\delta(D_\ell) = \frac{1}{\ell} \sum_{i=1}^{\ell} \pi_\delta(-y_i f_{D_\ell^i}(\mathbf{x}_i))$$

where the function $\pi_\delta(x)$ is 0 for $x < -\delta$, 1 for $x > 0$, and $\frac{x}{\delta} + 1$ for $-\delta \leq x \leq 0$ (a soft margin function). For $\delta \rightarrow 0$ we get the standard leave one out error and clearly $Loo^0(D_\ell) \leq Loo^\delta(D_\ell)$ for all $\delta > 0$.

Let β_ℓ be the stability of the kernel machine for the real valued output wrt. the ℓ_1 norm, that is:

$$\forall i \in \{1, \dots, \ell\}, \forall D_\ell, \forall \mathbf{x} : |f_{D_\ell}(\mathbf{x}) - f_{D_\ell^i}(\mathbf{x})| \leq \beta_\ell$$

For SVM it is known [2] that $\beta_\ell \leq \frac{C \cdot K}{2}$ where K is an upper bound that the kernel $K(\mathbf{x}, \mathbf{x})$ can take on the input space. The bound on the stability of SVM is not explicitly dependent of the size of the training set ℓ . However, the value of C is often chosen such that C is small for large ℓ . In the former experiments, C is fixed for all machines which are trained on learning sets of same sizes. This means that they have all the same stability for the ℓ_1 norm.

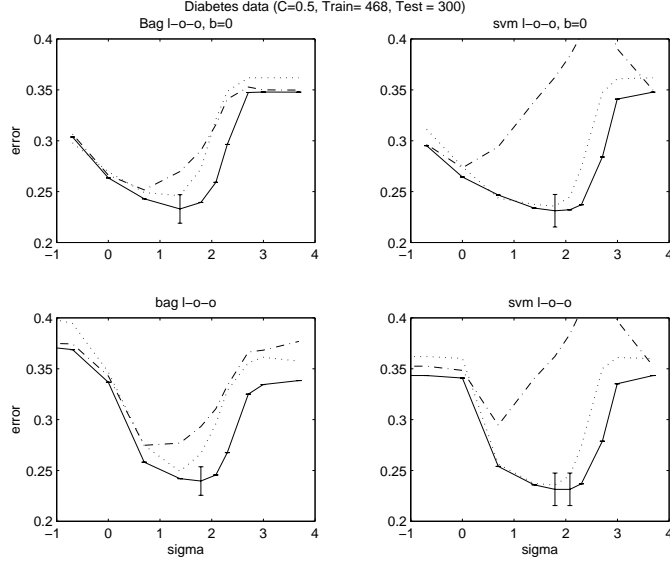


Figure 3: Diabetes data: see text for description.

We first state a bound on the expected error of the kernel machine in terms of its Loo_δ error. The following theorem is from [2]. The proof is detailed here for the sake of completeness.

Theorem 5.1 *For any given δ , with probability $1 - \eta$ the generalization misclassification error of an algorithm that is β_ℓ stable wrt. the ℓ_1 norm is bounded as:*

$$E_{(x,y)} [\theta(-yf_{D_\ell}(\mathbf{x}))] \leq Loo_\delta(D_\ell) + \beta_\ell + \sqrt{\frac{\ell}{2} \left(2\frac{\beta_{\ell-1}}{\delta} + \frac{1}{\ell} \right)^2 \ln\left(\frac{1}{\eta}\right)}$$

where β_ℓ is assumed to be a non-increasing function of ℓ .

Proof The idea is to study the random variable

$$X = Loo_\delta(D_\ell) - E_{(x,y)} [\pi_\delta(yf_{D_\ell}(x))]$$

where the expectation is taken wrt. (x, y) . We will show that X is close to its expectation $E_{D_\ell}[X]$.

Let us focus first on the Loo_δ part: since the function π_δ is $\frac{1}{\delta}$ -lipschitzian, we have for all $i = 1, \dots, \ell$ (denoting by $D_\ell^{i,j}$ the learning set D_ℓ where the i^{th} and j^{th} element have been removed):

$$\begin{aligned} \left| Loo_\delta(D_\ell) - \frac{\ell-1}{\ell} Loo_\delta(D_\ell^i) \right| &\leq \frac{1}{\ell\delta} \sum_{j=1, j \neq i}^{\ell} \underbrace{\left| -y_j (f_{D_\ell^j} - f_{D_\ell^{i,j}})(\mathbf{x}_j) \right|}_{\leq \beta_{\ell-1}} + \frac{1}{\ell} \\ &\leq \frac{(\ell-1)\beta_{\ell-1}}{\ell\delta} + \frac{1}{\ell} \\ &\leq \frac{\beta_{\ell-1}}{\delta} + \frac{\delta - \beta_{\ell-1}}{\ell\delta} \end{aligned}$$

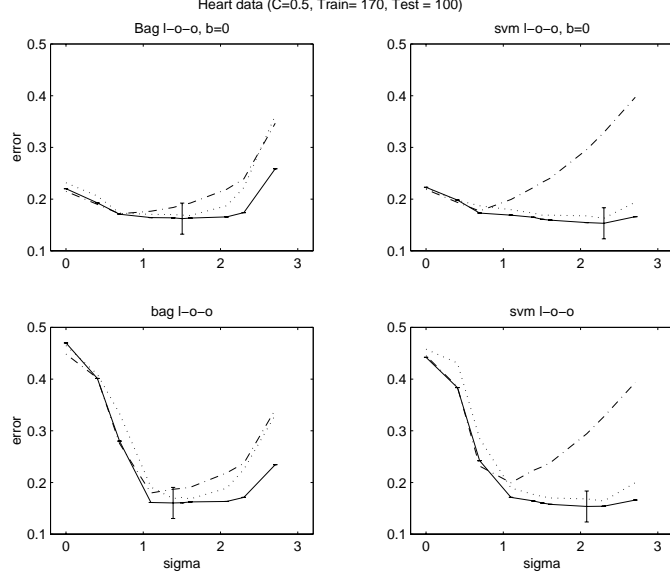


Figure 4: Heart data: see text for description.

According to the definition of the stability β_ℓ and since π_δ is $\frac{1}{\delta}$ -lipschitz, we have for all $i = 1, \dots, \ell$:

$$\left| E_{(x,y)} [\pi_\delta(yf_{D_\ell}(x))] - E_{(x,y)} [\pi_\delta(yf_{D_\ell^i}(x))] \right| \leq \frac{\beta_\ell}{\delta}$$

Defining X^i as to be:

$$X^i = \frac{\ell - 1}{\ell} \text{LoO}_\delta(D_\ell^i) - E_{(x,y)} [\pi_\delta(yf_{D_\ell^i}(x))]$$

we thus have:

$$|X - X^i| \leq \frac{\beta_\ell}{\delta} + \frac{\beta_{\ell-1}}{\delta} + \frac{\delta - \beta_{\ell-1}}{\ell\delta} \leq \frac{2\beta_{\ell-1}}{\delta} + \frac{1}{\ell}$$

if we assume that β_ℓ is a non-increasing function of ℓ . We then apply Mc-Diarmid's inequality:

Theorem 5.2 *Let Y_1, \dots, Y_ℓ be ℓ i.i.d. random variables taking values in a set A , and let $X : A^\ell \rightarrow \mathbf{R}$ be a function such that there exist for each $1 \leq i \leq \ell$, a function $X^i : A^{\ell-1} \rightarrow \mathbf{R}$ verifying*

$$\sup_{y_1, \dots, y_\ell \in A} |X(y_1, \dots, y_\ell) - X^i(y_1, \dots, y_{i-1}, y_{i+1}, \dots, y_\ell)| \leq c_i$$

then

$$\mathbf{P} [E[X(Y_1, \dots, Y_\ell)] - X(Y_1, \dots, Y_\ell) > \epsilon] \leq e^{-2\epsilon^2 / \sum_{i=1}^{\ell} c_i^2}$$

which yields the following inequality:

$$\mathbf{P} [E_{D_\ell}[X] - X > \epsilon] \leq e^{-2\epsilon^2 / (\ell c^2)}$$

where $c = \frac{2\beta_{\ell-1}}{\delta} + \frac{1}{\ell}$.

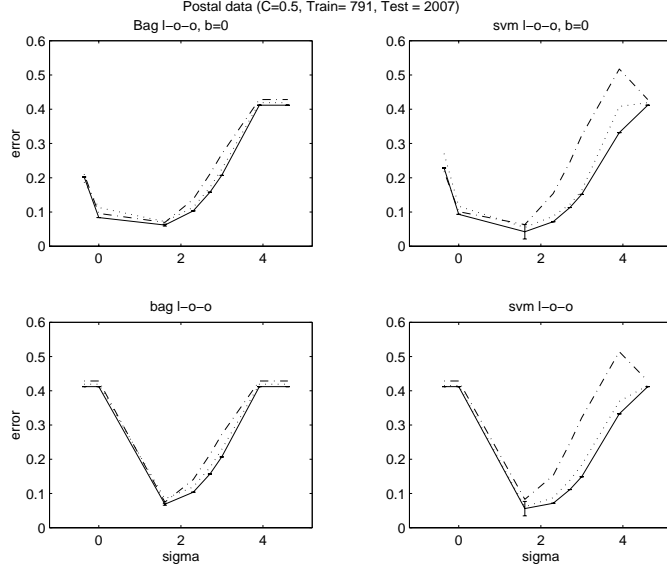


Figure 5: USPS data: see text for description.

Let us compute the expectation of X :

$$\begin{aligned}
|E_{D_\ell}[X]| &= |E_{D_\ell}[Loo_\delta(D_\ell)] - E_{D_\ell}[E_{(x,y)}[\pi_\delta(yf_{D_\ell}(x))]]| \\
&= \left| E_{D_\ell} \left[\frac{1}{\ell} \sum_{i=1}^{\ell} \pi_\delta(y_i f_{D_\ell^i}(x_i)) \right] + \frac{1}{\ell} \sum_{i=1}^{\ell} E_{D_\ell} [E_{(x,y)}[\pi_\delta(yf_{D_\ell}(x))]] \right| \\
&\leq \left| E_{D_\ell} \left[\frac{1}{\ell} \left(\sum_{i=1}^{\ell} E_{(x_i, y_i)} [\pi_\delta(y_i f_{D_\ell^i}(x_i))] - E_{(x,y)} [\pi_\delta(yf_{D_\ell}(x))]] \right) \right] \right| + \beta_\ell
\end{aligned}$$

The last equation has been derived from the former one by using the fact that the algorithm is stable. We can now change the name of the variable (x, y) as (x_i, y_i) since the latter does not appear in D_ℓ^i . This change shows that the term $E_{(x_i, y_i)} [\pi_\delta(y_i f_{D_\ell^i}(x_i))] - E_{(x,y)} [\pi_\delta(yf_{D_\ell}(x))]$ is zero and we have:

$$|E_{D_\ell}[X]| \leq \beta_\ell$$

Plugging this result into the Mc-Diarmid's inequality yields:

$$\mathbf{P} [E_{(x,y)} [\pi_\delta(yf_{D_\ell}(x))] > Loo_\delta + \beta_\ell + \epsilon] \leq e^{-2\epsilon^2/(\ell c^2)}$$

Forcing the right-hand side of this inequality to be lower than η gives at last the result of the theorem. \square

Notice that the bound holds for a given constant δ . One can derive a bound that holds uniformly for all δ and therefore use the “best” δ (i.e. the empirical margin of the classifier) [2]. For a SVM, the value of β_ℓ is equal to $\frac{CK}{2}$. Theorem 5.1 provides the following bound:

$$E_{(x,y)} [\theta(-yf_{D_\ell}(\mathbf{x}))] \leq Loo_\delta(D_\ell) + \frac{CK}{2} + \sqrt{\frac{\ell}{2} \left(\frac{CK}{\delta} + \frac{1}{\ell} \right)^2 \ln\left(\frac{1}{\eta}\right)} \quad (11)$$

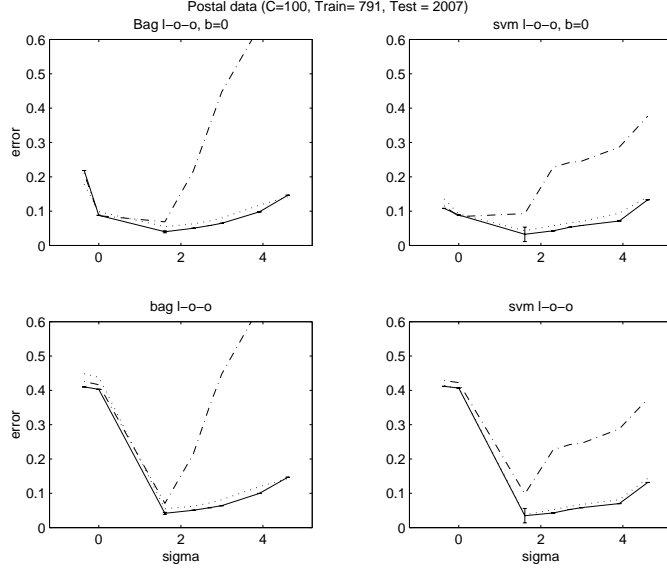


Figure 6: USPS data: using a large C ($C=50$). In this case the bounds do not work - see text for an explanation.

The value of C is often a function of ℓ . Depending on the way C decreases with ℓ , this bound can be tight or loose.

We now study a similar generalization bound for an ensemble of machines where each machine uses only α points drawn randomly with the uniform distribution from the training set. Such an ensemble is very close to the original idea of bagging despite some differences - namely that in standard bagging each machine uses a training set of size equal to the size of the original set created by random subsampling with replacement, instead of using only α points. To study such a system, let us introduce more notation that incorporate the “random subsampling” effect. We define $Loo_{\delta}^{\alpha}(D_{\ell})$: the leave-one-out error with the cost function π_{δ} of the expected combination $F = E_{D_t}[f^{(t)}]$ where the expectation is taken with respect to the training data D_t of size α drawn uniformly from D_{ℓ} . So, as an extreme case when $T \rightarrow \infty$:

$$Loo_{\delta}^{\alpha}(D_{\ell}) \approx Loo_{\delta}(D_{\ell}) \text{ for the classifier built from } \frac{1}{T} \sum_{t=1}^T f^{(t)}$$

Here, we assume that all functions are measurable and that all the sets are countable. By doing so, we avoid the measurability discussions and we assume that all the quantities we consider are integrable. We then have the following bound on the expected error of ensemble combinations:

Theorem 5.3 *For any given δ , with probability $1 - \eta$ the generalization misclassification error of a combination of classifiers each using a subsample of size α of the training set and each having a stability β_{α} wrt. the ℓ_1 norm is bounded as:*

$$E_{(x,y)} [\theta(-yF(\mathbf{x}))] \leq Loo_{\delta}^{\alpha}(D_{\ell}) + \frac{2\alpha}{\ell} \beta_{\alpha} + \sqrt{\frac{\alpha^2}{2\ell} \left(\frac{4\beta_{\alpha-1}}{\delta} + \frac{1}{\alpha} \right)^2 \ln\left(\frac{1}{\eta}\right)}$$

Proof

To prove this theorem, it is sufficient to show that the expected combination $F = E_{D_t}[f^{(t)}]$ is $2\frac{\alpha\beta_\alpha}{\ell}$ stable wrt. the ℓ_1 norm and to apply the previous theorem. We have:

$$|F - F^i| = |E_{D_t}[f^{(t)}] - E_{D_t^i}[f^{(t)}]|$$

where D_t (resp. D_t^i) is the set of size α used to learn the machine $f^{(t)}$ from the original learning set D_ℓ (resp. D_ℓ^i): $f_{D_t} = f^{(t)}$. We have by definition:

$$|F - F^i| = \left| \int f^{(t)} dD_t - \int f^{(t)} dD_t^i \right|$$

Defining the function $\mathbf{1}_A$ of the set A as to be: $\mathbf{1}_A(z) = 1$ iff $z \in A$, we decompose each of the integral as follows :

$$\begin{aligned} |F - F^i| = & \left| \int f^{(t)} \mathbf{1}_{(x_i, y_i) \in D_t} dD_t + \int f^{(t)} \mathbf{1}_{(x_i, y_i) \notin D_t} dD_t - \dots \right. \\ & \left. \dots \int f^{(t)} \mathbf{1}_{(x_i, y_i) \in D_t} dD_t^i - \int f^{(t)} \mathbf{1}_{(x_i, y_i) \notin D_t} dD_t^i \right| \end{aligned}$$

But, since $\mathbf{1}_{(x_i, y_i) \notin D_t} dD_t = \mathbf{1}_{(x_i, y_i) \notin D_t} dD_t^i$, we have:

$$\begin{aligned} |F - F^i| &= \left| \int f^{(t)} \mathbf{1}_{(x_i, y_i) \in D_t} dD_t - \int f^{(t)} \mathbf{1}_{(x_i, y_i) \in D_t} dD_t^i \right| \\ &\leq \underbrace{\left| \int (f^{(t)})_i \mathbf{1}_{(x_i, y_i) \in D_t} dD_t - \int (f^{(t)})_i \mathbf{1}_{(x_i, y_i) \in D_t} dD_t^i \right|}_{=0} + 2 \left| \int \beta_\alpha \mathbf{1}_{(x_i, y_i) \in D_t} dD_t \right| \end{aligned}$$

D_t is drawn with the uniform distribution and (x_i, y_i) is not in all the D_t . Actually, the average number of times (x_i, y_i) is in D_t is $\frac{\alpha}{\ell}$. The last integral can then be bounded by:

$$\left| \int \beta_\alpha \mathbf{1}_{(x_i, y_i) \in D_t} dD_t \right| \leq \frac{\alpha\beta_\alpha}{\ell} \quad (12)$$

And this gives a bound on the stability of $F = E_{D_t}[f^{(t)}]$. This result plugged into the previous theorem gives the final bound. \square

This theorem holds for ensemble combinations that are theoretically defined from the expectation $E_{D_t}[f^{(t)}]$. Notice that the hypothesis do not require that the combination is formed by only the same type of machines. In particular, one can imagine an ensemble of different kernel machines with different kernels. We formalize this remark in the following theorem:

Theorem 5.4 *Let F be a finite combination of SVM f_s , $s = 1, \dots, S$ with different kernels K^1, \dots, K^S :*

$$F = \frac{1}{S} \sum_{s=1}^S E_{D_t} [f_s^{(t)}] \quad (13)$$

where $f_s^{(t)}$ is a SVM with kernel K^s learned on D_t . Denote by $Loo_\delta^{\alpha,S}(D_\ell)$ the leave-one-out error of F computed with the function π_δ . Assume that each of the $f_s^{(t)}$ are learned with the same C on a subset D_t of size α drawn from D_ℓ with a uniform distribution. For any given δ , with probability $1 - \eta$, the generalization misclassification error is bounded as:

$$E_{(x,y)} [\theta(-yF(\mathbf{x}))] \leq Loo_\delta^{\alpha,S}(D_\ell) + \frac{\alpha}{\ell}(CK) + \sqrt{\frac{\alpha^2}{2\ell} \left(\frac{2CK}{\delta} + \frac{1}{\alpha} \right)^2 \ln\left(\frac{1}{\eta}\right)}$$

where $K = \frac{1}{S} \sum_{s=1}^S \sup_x K^s(x, x)$.

Proof As before, we study

$$F - F^i = \frac{1}{S} \sum_{s=1}^S E_{D_t} [f_s^{(t)}] - E_{D_t^i} [f_s^{(t)}]$$

Following the same calculations as in the previous theorem for each of the summand, we have:

$$|F - F^i| \leq \frac{2}{S} \left| \int \sum_{s=1}^S \beta_{\alpha,s} \mathbf{1}_{(x_i, y_i) \in D_t} dD_t \right|$$

where $\beta_{\alpha,s}$ denotes the stability of a SVM with kernel K^s on a set of size α . As before, since (x_i, y_i) appears in D_t only $\frac{\alpha}{\ell}$ times in average, we have the following bound:

$$|F - F^i| \leq \frac{2}{S} \sum_{s=1}^S \frac{\beta_{\alpha,s} \alpha}{\ell}$$

Replacing $\beta_{\alpha,s}$ by its value for the case of SVMs yields the theorem. \square

Notice that theorem 5.4 holds for combinations of kernel machines where for each kernel we use many machines trained on subsamples of the training set. So it is an “ensemble of ensembles” (see equation (13)).

Compared to what has been derived for a single SVM, combining SVMs provides a tighter bound on the generalization error. This result can then be interpreted as an explanation of the better estimation of the test error by the leave-one-out error for ensemble method. The bounds given by the previous theorems have the form:

$$E_{(x,y)} [\theta(-yF(\mathbf{x}))] \leq Loo_\delta^\alpha(D_\ell) + O \left(\frac{\alpha}{\sqrt{\ell}} C_\alpha K \sqrt{\frac{\ln(\frac{1}{\eta})}{\delta^2}} \right)$$

although the bound for a single SVM is:

$$E_{(x,y)} [\theta(-yf(\mathbf{x}))] \leq Loo_\delta(D_\ell) + O \left(\sqrt{\ell} C_\ell K \sqrt{\frac{\ln(\frac{1}{\eta})}{\delta^2}} \right)$$

we have indexed the coefficients C with an index that indicates that SVMs are not learned with the same training set size in the first and in the second case. In the experiments, the same C was used for all SVMs ($C_\ell = C_\alpha$). The bound derived for combination of SVMs is then tighter than for single SVM with a factor of α/ℓ . This improvement comes from the stability of the combination of SVMs that is better than the stability of a single SVM. This statement is true if we assume that both SVMs are trained with the same C but the discussion becomes more tricky if different C 's are used during learning.

The stability of SVMs depends indeed on the way the value of C is determined. For single SVM, C is generally a function of ℓ , and for combination of SVMs, C also depends on the size of the subsampled learning sets D_t . Let us assume for the discussion that these sets have a size of α . In theorem 5.3, we have seen that the stability of the combination of machines was smaller than $\frac{2\alpha\beta_\alpha}{\ell}$ where β_α is equal to $\frac{CK}{2}$ for SVMs (see eq. (12)). If this stability is better than the stability of a single machine, then combining the functions $f^{(t)}$ provides a better bound. However, in the other case, the bound is worse and combining machines should be avoided. We have the following corollary:

Corollary 5.1 *If a learning system is β_ℓ stable and $\frac{\beta_\ell}{\beta_\alpha} < \frac{2\alpha}{\ell}$, then combining these learning systems does not provide a better bound on the difference between the test error and the leave-one-out error. Conversely, if $\frac{\beta_\ell}{\beta_\alpha} > \frac{2\alpha}{\ell}$, then combining these learning systems leads to a better bound on the difference between the test error and the leave-one-out error.*

Proof Let us look at the stability of the ensemble machine:

$$2\frac{\alpha\beta_\alpha}{\ell} = \left(\frac{2\alpha\beta_\alpha}{\ell\beta_\ell}\right)\beta_\ell$$

but,

$$\frac{\beta_\ell}{\beta_\alpha} < \frac{2\alpha}{\ell} \Leftrightarrow \frac{2\alpha\beta_\alpha}{\ell\beta_\ell} < 1$$

Thus, the combination is less stable than a single learning system. This result plugged into the previous theorems shows that the bounds on the differences between the leave-one-out and the test error are less tight for the combinations of learning systems when the stability of the single machine satisfies: $\frac{\beta_\ell}{\beta_\alpha} < \frac{2\alpha}{\ell}$. The converse is similarly proved. \square

The consequence of this corollary is that combining machines should not be used if the stability of the single machine is very good. However, it is not often the case to have a highly stable single machine therefore typically bagging improves stability. In such a situation, the bounds presented in this paper show that we have better control of the generalization error for combination of SVMs in the sense that the leave one out and the empirical errors are closer to the test error. The bounds presented *do not* necessarily imply that the generalization error of bagging is less than that of single machines. Similar remarks have already been made by Breiman for bagging [3] where similar considerations of stability are experimentally discussed. Another remark that can be made from the work of Breiman is that bagging does not improve performances after a certain number of bagged predictors. On the other hand, it does not decrease performances either. This experimentally derived statement can be translated in our framework as: when T increases the stability of the combined learning

system tends to the stability of the expectation $E_{D_t} [f^{(t)}]$ which does not improve after T has passed a certain value. This value may correspond to the convergence of the finite sum $\frac{1}{T} \sum_{t=1}^T f^{(t)}$ to its expectation wrt. D_t .

At last, it is worthwhile noticing that the stability analysis of this section holds also for the empirical error. Indeed, for a β_ℓ stable algorithm, as it is underlined in [2], the leave-one-out and the empirical error are related by:

$$Loo^\delta(D_\ell) \leq E_0(D_\ell) + \beta_\ell$$

where $E_0(D_\ell)$ is the empirical error on the learning set D_ℓ . Using this inequality in theorems 5.1, 5.3, and 5.4, we can bound the generalization error in terms of the empirical error and the stability of the machines.

6 Other Ensembles and Error Estimates

6.1 Validation Set for Model Selection

Instead of using bounds on the generalization performance of learning machines like the ones discussed above, an alternative approach for model selection is to use a validation set to choose the parameters of the machines. We consider first the simple case where we have N machines and we choose the “best” one based on the error they make on a fixed validation set of size V . This can be thought of as a special case where we consider as our hypothesis space to be the set of the N machines, and then we “train” by simply picking the machine with the smallest “empirical” error (in this case this is the validation error). It is known that if VE_i is the validation error of machine i and TE_i is its true test error, then for all N machines simultaneously the following bound holds with probability $1 - \eta$ [6, 17]:

$$TE_i \leq VE_i + \sqrt{\frac{\log(N) - \log(\frac{\eta}{4})}{V}} \quad (14)$$

So how “accurately” we pick the best machine using the validation set depends, as expected, on the number of machines N and on the size V of the validation set. The bound suggests that a validation set can be used to accurately estimate the generalization performance of a relatively small number of machines (i.e. small number of parameter values examined), as done often in practice.

We used this observation for parameter selection for SVM and for their ensembles. Experimentally we followed a slightly different procedure from what is suggested by bound (14): for each machine (that is, for each σ of the Gaussian kernel in our case, both for a single SVM and for an ensemble of machines) we split the training set (for each training-testing split of the overall dataset as described above) into a smaller training set and a validation set (70-30% respectively). We trained each machine using the new, smaller training set, and measured the performance of the machine on the validation set. Unlike what bound (14) suggests, instead of comparing the validation performance found with the generalization performance of the machines trained on the smaller training set (which is the case for which bound (14) holds), we compared the validation performance with the test performance of the machine trained using *all* the initial (larger) training set. This way *we did not have to*

use less points for training the machines, which is a typical drawback of using a validation set, and we could compare the validation performance with the leave-one-out bounds and the test performance of the *exact same* machines we used in the previous section.

We show the results of these experiments in figures 1-5: see the dotted lines in the plots. We observe that *although the validation error is that of a machine trained on a smaller training set, it still provides a very good estimate of the test performance of the machines trained on the whole training set.* In all cases, including the case of $C > 1$ for which the leave-one-out bounds discussed above did not work well, the validation set error provided a very good estimate of the test performance of the machines.

6.2 Adaptive Combinations of Learning Machines

The ensemble kernel machines (4) considered so far are voting combinations where the coefficients c_t in (4) of the linear combination of the machines are fixed. We now consider the case where these coefficients are also learned. In particular we consider the following two-layer architecture:

1. A number T of kernel machines is trained as before (for example using different training data, or different parameters). Let $f^t(x), t = 1, \dots, T$ be the machines.
2. The T outputs (real valued in our experiments, but could also be thresholded - binary) of the machines at each of the training points are computed.
3. A linear machine (i.e. linear SVM) is trained using as inputs the outputs of the T machines on the training data, and as labels the original training labels. The solution is used as the coefficients c_t of the linear combination of the T machines.

In this case the ensemble machine $F(x)$ is a kernel machine itself which is trained using as kernel the function:

$$\mathcal{K}(\mathbf{x}, \mathbf{t}) = \sum_{t=1}^T f^t(\mathbf{x})f^t(\mathbf{t}).$$

Notice that since each of the machines $f^t(x)$ depend of the data, also the kernel \mathcal{K} is data dependent. Therefore the stability parameter of the ensemble machine is more difficult to compute (when a data point is left out the kernel \mathcal{K} changes). Likewise the leave-one-out error bound of theorem 3.3 does not hold since the theorem assumes fixed coefficients c_t ³.

On the other hand, an important characteristic of this type of ensembles is that independent of what kernels/parameters each of the individual machines of the ensemble use, the “second layer” machine (which finds coefficients c_t) uses always a linear kernel. This may imply that *the overall architecture may not be very sensitive to the kernel/parameters of the machines of the ensemble.* We tested this hypothesis experimentally by comparing how the test performance of this type of machines changes with the σ of the Gaussian kernel used from the individual machines of the ensemble, and compared the behavior with that of single machines and ensembles of machines with fixed c_t . In figure 7 we show two example. In our experiments, for all datasets except from one, learning the coefficients c_t of the combination

³A validation set can still be used for model selection for these machines.

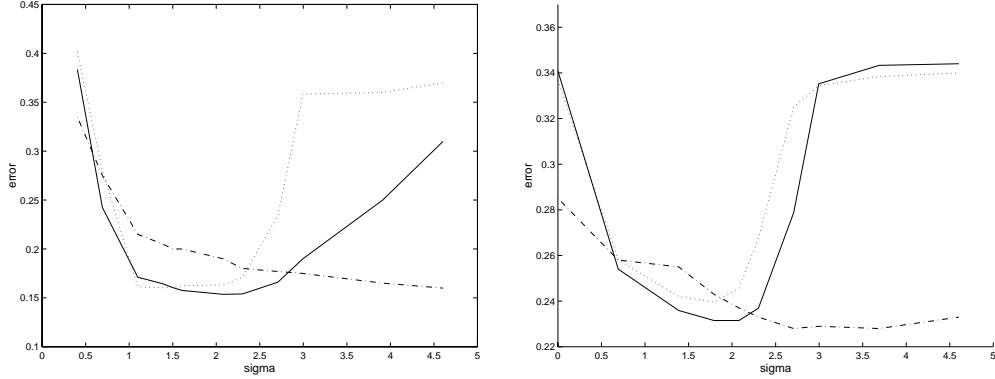


Figure 7: When the coefficients of the second layer are learned using a linear SVM the system is less sensitive to changes of the σ of the Gaussian kernel used by the individual machines of the ensemble. Solid line is one SVM, dotted is ensemble of 30 SVMs with fixed $c_t = \frac{1}{30}$, and dashed line is ensemble of 30 SVMs with the coefficients c_t learned. The horizontal axis shows the natural logarithm of the σ of the Gaussian kernel. Left is the heart dataset, and right is the diabetes one. The threshold b is non-zero for these experiments.

of the machines using a linear machine (we used a linear SVM) made the overall machine *less sensitive* to changes of the parameters of the individual machines (σ of the Gaussian kernel). This can be practically a useful characteristic of the architecture outlined in this section: for example the choice of the kernel parameters of the machines of the ensembles need not be tuned accurately.

6.3 Ensembles Versus Single Machines

So far we concentrated on the theoretical and experimental characteristics of ensembles of kernel machines. We now discuss how ensembles compare with single machines.

Table 1 shows the test performance of one SVM compared with that of an ensemble of 30 SVMs combined with $c_t = \frac{1}{30}$ and an ensemble of 30 SVMs combined using a linear SVM for some UCI datasets (characteristic results). For the tables of this section we use, for convenience, the following notation:

- VCC stands for “Voting Combinations of Classifiers”, meaning that the coefficients c_t of the combination of the machines are fixed.
- ACC stands for “Adaptive Combinations of Classifiers”, meaning that the coefficients c_t of the combination of the machines are learned-adapted.

We only consider SVM and ensembles of SVMs with the threshold b . The table shows mean test errors and standard deviations for the best (decided using the validation set performance in this case) parameters of the machines (σ 's of Gaussians *and* parameter C - hence different from figures 1-5 which where for a given C). As the results show, the best SVM and the best ensembles we found have about the same test performance. Therefore, with appropriate tuning of the parameters of the machines, combining SVM's does not lead to performance improvement compared to a single SVM.

Dataset	SVM	VCC	ACC
Breast	25.5 ± 4.3	25.6 ± 4.5	25 ± 4
thyroid	5.1 ± 2.5	5.1 ± 2.1	4.6 ± 2.7
diabetes	23 ± 1.6	23.1 ± 1.4	23 ± 1.8
heart	15.4 ± 3	15.9 ± 3	15.9 ± 3.2

Table 1: Average errors and standard deviations (percentages) of the “best” machines (best σ of the Gaussian kernel and best C) - chosen according to the validation set performances. The performances of the machines are about the same. VCC and ACC use 30 SVM classifiers.

Although the “best” SVM and the “best” ensemble (that is, after accurate parameter tuning) perform similarly, an important difference of the ensembles compared to a single machine is that the training of the ensemble consists of a large number of (parallelizable) small-training-set kernel machines - in the case of bagging. This implies that one can gain performance similar to that of a single machine by training many faster machines using smaller training sets. This can be an important practical advantage of ensembles of machines especially in the case of large datasets. Table 2 compares the test performance of a single SVM with that of an ensemble of SVM each trained with as low as 1% of the initial training set (for one dataset). For fixed c_t the performance decreases only slightly in all cases (thyroid, that we show, was the only dataset we found in our experiments for which the change was significant for the case of VCC), while in the case of the architecture of section 5 even with 1% training data the performance does not decrease: this is because the linear machine used to learn coefficients c_t uses all the training data. Even in this last case the overall machine can still be faster than a single machine, since the second layer learning machine is a linear one, and fast training methods for the particular case of linear machines exist [14].

Dataset	VCC 10%	VCC 5%	VCC 1%	ACC 10%	ACC 5%	ACC 1%	SVM
Diabetes	23.9	26.2	-	24.9	24.5	-	23 ± 1.6
Thyroid	6.5	22.2	-	4.6	4.6	-	5.1 ± 2.5
Faces	.2	.2	.5	.1	.2	.2	.1

Table 2: Comparison between error rates of a single SVM v.s. error rates of VCC and ACC of 100 SVMs for different percentages of subsampled data. The last dataset is from [13].

7 Conclusions

We presented theoretical bounds on the generalization error of ensembles of kernel machines such as SVM. Our results apply to the general case where each of the machines in the ensemble is trained on different subsets of the training data and/or uses different kernels or input features. A special case of ensembles is that of bagging. The bounds were derived within the frameworks of cross validation error and stability and learning. They involve two main quantities: the leave-one-out error estimate and the stability parameter of the ensembles.

We have shown that the leave-one-error of the ensemble can be bounded with a function of the solution's parameters (c_t and α_i^t 's in equation 4) which can be computed efficiently. In the case of bagging of SVM, this bound is experimentally found to be tighter, i.e. closer to the test error, than the equivalent one for single kernel machine. This experimental finding could be justified by the stability analysis.

In the case of ensembles of kernel machines, each trained with the same regularization parameter C , the stability parameter is a linearly increasing function of the number of points used by each machine. This indicates that ensembles of kernel machines are more stable learning algorithms than the equivalent single kernel machine. This implies that the difference between empirical or leave one out errors and generalization error is smaller for bagging than for single kernel machines - which is experimentally validated. This can be important for example for model selection. It does not necessarily imply that the generalization error of bagging is smaller than that of single machines - as also shown by the experiments.

A main research direction which emerges from the paper is that the theoretical framework presented here can be applied to bagging of any learning machine other than kernel machines, showing formally for which machines bagging increases the stability. An important open problem is how to extend the bounds of section 3 and 5 to the type of machines discussed in section 6.2.

References

- [1] S. Boucheron, G. Lugosi, and P. Massart. A sharp concentration inequality with applications. *Random Structures and Algorithms*, 16:277–292, 2000.
- [2] O. Bousquet and A. Elisseeff. Stability and generalization. Technical report, Centre de Mathématiques Appliquées, Ecole Polytechnique, 2001.
- [3] L. Breiman. Bagging predictors. *Machine Learning*, 26(2):123–140, 1996.
- [4] O. Chapelle and V. Vapnik. Model selection for support vector machines. In *Advances in Neural Information Processing Systems*, 1999.
- [5] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambridge University Press, 2000.
- [6] L. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Number 31 in Applications of mathematics. Springer, New York, 1996.
- [7] L. Devroye and T.J. Wagner. Distribution-free performance bounds for potential function rules. *IEEE Trans. on Information Theory*, 25(5):601–604, 1979.
- [8] T. Evgeniou, M. Pontil, and T. Poggio. A unified framework for regularization networks and support vector machines. A.I. Memo No. 1654, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1999.

- [9] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. Technical report, Technical Report, Department of Statistics, Stanford University., 1998.
- [10] T. Jaakkola and D. Haussler. Exploiting generative models in discriminative classifiers. In *Proc. of Neural Information Processing Conference*, 1998.
- [11] M. Kearns and D. Ron. Algorithmic stability and sanity check bounds for leave-one-out cross validation bounds. *Neural Computation*, 11(6):1427–1453, 1999.
- [12] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L.J. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989.
- [13] E. Osuna, R. Freund, and F. Girosi. Support vector machines: Training and applications. A.I. Memo 1602, MIT A. I. Lab., 1997.
- [14] J. C. Platt. Fast training of support vector machines using sequential minimal optimization. In C. Burges B. Scholkopf, editor, *Advances in Kernel Methods–Support Vector Learning*. MIT press, 1998.
- [15] B. Scholkopf, C. Burges, and A. Smola. *Advances in Kernel Methods – Support Vector Learning*. MIT Press, 1998.
- [16] R. Shapire, Y. Freund, P. Bartlett, and W.S. Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *The Annals of Statistics*, 1998. to appear.
- [17] V. N. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.
- [18] G. Wahba. *Splines Models for Observational Data*. Series in Applied Mathematics, Vol. 59, SIAM, Philadelphia, 1990.