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

Theodoros Evgeniou, (theodoros.evgeniou@insead.edu)
Technology Management,
INSEAD,
Boulevard de Constance, 77305 Fontainebleau, France

Massimiliano Pontil, (pontil@dii.unisi.it)
DII - University of Siena
Via Roma 56, 53100 Siena, Italy

André Elisseeff, (andre.elisseeff@tuebingen.mpg.de)
Max Planck Institute for Biological Cybernetics,
Spemannstrasse 38, 72076 Tübingen, Germany

Abstract. We study the leave-one-out and generalization errors of voting combinations of learning machines. A special case considered is a variant of bagging. We analyze in detail combinations of kernel machines, such as support vector machines, and present theoretical estimates of their leave-one-out error. 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. We report experiments supporting the theoretical findings.

Keywords: Cross-Validation, Bagging, Combinations of Machines, Stability

1. Introduction

Studying the generalization performance of ensembles of learning machines has been the topic of ongoing research in recent years (Breiman, 1996; Schapire et al., 1998; Friedman et al., 1998). There is a lot of experimental work showing that combining learning machines, for example using boosting or bagging methods (Breiman, 1996; Schapire et al., 1998), very often leads to improved generalization performance. A number of theoretical explanations have also been proposed (Schapire et al., 1998; Breiman, 1996), 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 (Bousquet and Elisseeff, 2002; Boucheron et al., 2000). The second, although an older tool (Devroye and Wagner, 1979; Devroye et al., 1996), has become only important recently with the work of (Kearns and Ron, 1999; Bousquet and Elisseeff, 2002).

Stability has been discussed extensively also in the work of Breiman (1996). The theory in Breiman (1996) is that bagging increases performance because

it reduces the variance of the base learning machines, although it does not always increase the bias (Breiman, 1996). The definition of the variance in Breiman (1996) is similar in spirit to that of stability we use in this paper. The key difference is that in Breiman (1996) the variance of a learning machine is defined in an asymptotic way and is not used to derive any non-asymptotic bounds on the generalization error of bagging machines, while here we define stability for finite samples like it is done in (Bousquet and Elisseeff, 2002) and we also derive such non-asymptotic bounds. The intuition given by Breiman (1996) gives interesting insights: the effect of bagging depends on the "stability" of the base classifier. Stability means here changes in the output of the classifier when the training set is perturbed. If the base classifiers are stable, then bagging is not expected to decrease the generalization error. On the other hand, if the base classifier is unstable, such as often occurs with decision trees, the generalization performance is supposed to increase with bagging. Despite experimental evidence, the insights in (Breiman, 1996) had not been supported by a general theory linking stability to the generalization error of bagging, which is what Section 5 below is about.

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 (Breiman, 1996), this paper considers combinations of the real outputs of the classifiers, and each machine is trained on a different and small subset of the initial training set chosen by 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 a 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 observe.

The paper is organized as follows. Section 2 gives the basic notation and background. In Section 3 we present bounds for a 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 (Vapnik, 1998; Schölkopf et al., 1998; Evgeniou et al., 2000). For an account consistent with our notation see (Evgeniou et al., 2000).

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

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

where we use the following notation:

- Let $\mathcal{X} \subset \mathbb{R}^n$ be the input set, the pairs $(\mathbf{x}_i, y_i) \in \mathcal{X} \times \{-1, 1\}$, $i = 1, .., \ell$ are sampled independently and identically according to an unknown probability distribution $P(\mathbf{x}, y)$. The set $D_\ell = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_\ell, y_\ell)\}$ 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 (Vapnik, 1998; Wahba, 1990) 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 algorithm (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 (Evgeniou et al., 2000) the solution of Equation (1) is of the form

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

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

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

where $S(\cdot)$ is a continuous and concave function (strictly concave if matrix $K(\mathbf{x}_i, \mathbf{x}_j)$ is not strictly positive definite) and $C = \frac{1}{2\ell\lambda}$ a constant. Thus, $H(\alpha)$ is strictly concave and the above optimization problem has a unique solution.

Support Vector Machines (SVMs) 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. Notice that the bias term (threshold b in the general case of machines $f(\mathbf{x}) = \sum_{i=1}^{\ell} \alpha_i K(\mathbf{x}_i, \mathbf{x}) + b$) is incorporated in the kernel K, and it is therefore also regularized. Notice also that function $S(\cdot)$ in (3) can take general forms - leading to machines other than SVM - but in the general case the optimization of (3) may be computationally inefficient.

2.1. KERNEL MACHINE ENSEMBLES

Given a learning algorithm - such as a SVM or an ensemble of SVMs - we define f_{D_ℓ} to be the solution of the algorithm when the training set $D_\ell = \{(\mathbf{x}_i, y_i),\ i = 1, \ldots, \ell\}$ 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.

We consider the general case where each of the machines in the ensemble uses a different kernel and different subsets $D_{r,t}$ of the training set D_ℓ where r refers to the size of the subset and $t = 1, .., T$ to the machine that uses it to learn. Let $f_{D_{r,t}}(\mathbf{x})$ be the optimal solution of machine t using a kernel $K^{(t)}$. We 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 decision function of the ensemble is given by

$$F_{r,T}(\mathbf{x}) = \sum_{t=1}^{T} c_t f_{D_{r,t}}(\mathbf{x}) \tag{4}$$

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 $F_{r,T}(\mathbf{x})$. Below for simplicity we will note with capital F the combination $F_{r,T}$. In Section 5 we will consider only the case that $c_t = \frac{1}{T}$ for simplicity.

In the following, the sets $D_{r,t}$ will be identically sampled according to the uniform distribution and without replacement from the training set D_ℓ. We will denote by $\mathbb{E}_{D_r \sim D_\ell}$ the expectation with respect to the subsampling from D_ℓ according to the uniform distribution (without replacement), and sometimes we write $f_{D_{r,t} \sim D_\ell}$ rather than $f_{D_{r,t}}$ to make clear which training

set has been used during learning. The letter r will always refer to the number of elements in $D_{r,t}$.

2.2. LEAVE-ONE-OUT ERROR

Table I. Notation.

f	Real valued prediction rule of one learning machine, $f : \mathcal{X} \to \mathbb{R}$
$V(f, y)$	Loss function
$P(\mathbf{x}, y)$	Probability distribution underlining the data
D_ℓ	Set of i.i.d examples sampled from $P(\mathbf{x}, y)$, $D_\ell = \{(\mathbf{x}_i, y_i) \in \mathcal{X} \times \{-1, 1\}\}_{i=1}^{\ell}$
D_ℓ^i	The set $D_\ell \backslash \{(\mathbf{x}_i, y_i)\}$
f_{D_ℓ}	Learning machine (e.g. SVM) trained on D_ℓ. Also noted as f
$\text{Loo}_{D_\ell}(f)$	Leave-one-out error of f on the data set D_ℓ
$\pi_\delta(x)$	Soft margin loss, $\pi_\delta(x) = 0$, if $x < -\delta$, 1 if $x > 0$, and $\frac{x}{\delta}$ if $-\delta \leq x \leq 0$
$\text{Loo}_{\delta, D_\ell}(f)$	Leave one out error with soft margin π_δ
β_ℓ	Uniform stability of f
$D_{r,t}$ or $D_{r,t} \sim D_\ell$	Set of r points sampled uniformly from D_ℓ used by machine t, $t = 1, \ldots, T$
$D_r \sim D_\ell$	Set of r points sampled uniformly from D_ℓ
$(D_{r,t} \sim D_\ell)^i$	"Original" $D_{r,t}$ with point $(\mathbf{x}_i, y_i)$ removed
$F_{r,T}$, or just F	Ensemble of T machines, $F_{r,T} = \sum_{t=1}^{T} c_t f_{D_{r,t}}$
$\hat{F}$	Expected combination of machines $\mathbb{E}_{D_r \sim D_\ell}[f_{D_r}]$
$\text{DLoo}_{D_\ell}(F)$	Deterministic leave out out error
$\text{DLoo}_{\delta, D_\ell}(F)$	Deterministic leave out out error with soft margin π_δ

If θ is, as before, the Heavyside function, then the leave-one-out error of f on D_ℓ is defined by

$$\text{Loo}_{D_\ell}(f) = \frac{1}{\ell} \sum_{i=1}^{\ell} \theta(-y_i f_i(\mathbf{x}_i)) \tag{5}$$

Notice that for simplicity there is a small abuse of notation here, since the leave-one-out error typically refers to a learning method while here we use the solution f in the notation. The leave-one-out error provides an estimate of the average generalization performance of a machine. It is known 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 (Vapnik, 1998).

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

$$\mathbb{E}_{D_\ell}\left[\mathbb{E}_{(\mathbf{x},y)}[\theta(-yf_{D_\ell}(\mathbf{x}))]\right] = \mathbb{E}_{D_{\ell+1}}\left[\mathrm{Loo}_{D_{\ell+1}}(f_{D_{\ell+1}})\right]$$

As observed (Kearns and Ron, 1999), this theorem can be extended to general learning algorithms by adding a randomizing preprocessing step. The way the leave-one-out error is computed can however be different depending on the randomness. Consider the previous ensemble of kernel machines (4). The data sets $D_{r,t}$, $t = 1,..,T$ are drawn randomly from the training set D_ℓ. We can then compute a leave-one-out estimate for example in either of the following ways:

1. For $i = 1,..,\ell$, remove $(\mathbf{x}_i, y_i)$ from D_ℓ and sample new data sets $D_{r,t}$, $t = 1,..,T$ from D_ℓ^i. Compute the $f_{D_{r,t}\sim D_\ell^i}$ and average then the error of the resulting ensemble machine computed on $(\mathbf{x}_i, y_i)$. This leads to the classical definition of leave-one-out error and can be computed as:

$$\mathrm{Loo}_{D_\ell}(F) = \frac{1}{\ell}\sum_{i=1}^{\ell}\theta\left(-y_i\frac{1}{T}\sum_{t=1}^{T} f_{D_{r,t}\sim D_\ell^i}(\mathbf{x}_i)\right) \tag{6}$$

2. For $i = 1,..,\ell$, remove $(\mathbf{x}_i, y_i)$ from each $D_{r,t} \sim D_\ell$. Compute the $f_{(D_{r,t}\sim D_\ell)^i}$ and average the error of the resulting ensemble machine computed on $(\mathbf{x}_i, y_i)$. Note that we have used the notation $(D_{r,t} \sim D_\ell)^i$ to denote the set $D_{r,t} \sim D_\ell$ where $(\mathbf{x}_i, y_i)$ has been removed. This leads to what we will call a *deterministic* version of the leave-one-out error, in short det-leave-one-out, or *DLoo*:

$$\mathrm{DLoo}_{D_\ell}(F) = \frac{1}{\ell}\sum_{i=1}^{\ell}\theta\left(-y_i\frac{1}{T}\sum_{t=1}^{T} f_{(D_{r,t}\sim D_\ell)^i}(\mathbf{x}_i)\right) \tag{7}$$

Note that the first computation requires to re-sample new data sets for each "leave-one-out round", while the second computation uses the same subsample data sets for each "leave-one-out round" removing at most one point from each of them. In a sense, the det-leave-one-out error is then more "deterministic" than the classical computation (6). In this paper, we will consider mainly the det-leave-one-out error for which we will derive easy-to-compute bounds and from which we will bound the generalization error of ensemble machines. Finally notice that the size of the subsampling is implicit in the notation $\mathrm{DLoo}_{D_\ell}(F)$: r is fixed in this paper so there is no need to complicate the notation further.

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 (Jaakkola and Haussler, 1998):

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

$$\mathrm{Loo}_{D_\ell}(f) \leq \frac{1}{\ell}\sum_{i=1}^{\ell}\theta\left(\alpha_i K(\mathbf{x}_i,\mathbf{x}_i) - y_i f_{D_\ell}(\mathbf{x}_i)\right) \tag{8}$$

where f_{D_ℓ} is the optimal function found by solving problem (3) on the whole training set.

In the particular case of SVMs where the data are separable the r.h.s of Equation (8) can be bounded by geometric quantities, namely (Vapnik, 1998):

$$\mathrm{Loo}_{D_\ell}(f) \leq \frac{1}{\ell}\sum_{i=1}^{\ell}\theta\left(\alpha_i K(\mathbf{x}_i,\mathbf{x}_i) - y_i f_{D_\ell}(\mathbf{x}_i)\right) \leq \frac{1}{\ell}\frac{d_{sv}^2}{\rho^2} \tag{9}$$

where d_{sv} is the radius of the smallest sphere in the feature space induced by kernel K (Wahba, 1990; Vapnik, 1998) 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 next theorem is a direct application of Theorem 2.1:

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

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

where the expectation $\mathbb{E}$ is taken with respect to the probability of a training set D_ℓ of size ℓ.

Notice that this result shows that the performance of the 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 det-leave-one-out error of such ensembles.

THEOREM 3.3. *The det-leave-one-out error of a kernel machine ensemble is upper bounded by:*

$$\text{DLoo}_{D_\ell}(F) \leq \frac{1}{\ell}\sum_{i=1}^{\ell}\theta\left(\sum_{t=1}^{T} c_t\alpha_i^{(t)}K^{(t)}(\mathbf{x}_i,\mathbf{x}_i) - y_iF(\mathbf{x}_i)\right). \qquad (10)$$

The proof of this Theorem is based on the following lemma shown in (Vapnik, 1998; Jaakkola and Haussler, 1998):

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 $y_if_i(\mathbf{x}_i) \geq y_if(\mathbf{x}_i) - \alpha_iK(\mathbf{x}_i,\mathbf{x}_i)$.*

Using lemma 3.1 we can now prove Theorem 3.3.

Proof of Theorem 3.3: Let $F_i(\mathbf{x}) = \sum_{t=1}^{T} c_tf_i^{(t)}(\mathbf{x})$ be the ensemble machine trained with all initial training data except $(\mathbf{x}_i, y_i)$ (subsets $D_{r,t}$ are the "original" ones - only $(\mathbf{x}_i, y_i)$ is removed from them). Lemma 3.1 gives that

$$\begin{aligned} y_iF_i(\mathbf{x}_i) &= y_i\sum_{t=1}^{T} c_tf_i^{(t)}(\mathbf{x}_i) \geq \sum_{t=1}^{T} c_t\left[y_if^{(t)}(\mathbf{x}_i) - \alpha_i^{(t)}K^{(t)}(\mathbf{x}_i,\mathbf{x}_i)\right] \\ &= y_iF(\mathbf{x}_i) - \sum_{t=1}^{T} c_t\alpha_i^{(t)}K^{(t)}(\mathbf{x}_i,\mathbf{x}_i) \end{aligned}$$

from which it follows that:

$$\theta\left(-y_iF_i(\mathbf{x}_i)\right) \leq \theta\left(\sum_{t=1}^{T} c_t\alpha_i^{(t)}K^{(t)}(\mathbf{x}_i,\mathbf{x}_i) - y_iF(\mathbf{x}_i)\right).$$

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

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

which proves the Theorem. □

Notice that the bound has the same form as the bound in Equation (8): 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 det-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 (Schapire et al., 1998). For any given $\delta > 0$ we define Err_δ to be the empirical error with ensemble margin less than δ,

$$\text{Err}_\delta(F) = \frac{1}{\ell} \sum_{i=1}^{\ell} \theta \left(-y_i F(\mathbf{x}_i) + \delta \right).$$

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 det-leave-one-out error of a kernel machine ensemble is upper bounded by:*

$$\text{DLoo}_{D_\ell}(F) \leq \text{Err}_\delta(F) + \frac{1}{\ell} \left(\frac{1}{\delta} \sum_{t=1}^{T} c_t d_{t(\delta)}^2 (\sum_{i \in N_\delta} \alpha_i^{(t)}) \right) \tag{11}$$

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:

$$\theta \left(\sum_{t=1}^{T} c_t \alpha_i^{(t)} K^{(t)}(\mathbf{x}_i, \mathbf{x}_i) - y_i F(\mathbf{x}_i) \right) \leq \frac{1}{\delta} \sum_{t=1}^{T} c_t \alpha_i^{(t)} K^{(t)}(\mathbf{x}_i, \mathbf{x}_i). \tag{12}$$

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 (12) 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

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

while

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

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

So in both cases inequality (12) holds. Therefore:

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

$$\ell \text{Err}_\delta + \frac{1}{\delta} \sum_{i \in N_\delta} \sum_{t=1}^{T} c_t K^{(t)}(\mathbf{x}_i, \mathbf{x}_i) \alpha_i^{(t)} \leq \ell \text{Err}_\delta + \frac{1}{\delta} \sum_{t=1}^{T} c_t d_{t(\delta)}^2 (\sum_{i \in N_\delta} \alpha_i^{(t)}).$$

The statement of the corollary follows by applying Theorem 3.3. □

Notice that Equation (11) 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 Theorem 2.1, 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 det-leave-one-out error of an ensemble of SVMs is upper bounded by:*

$$\text{DLoo}_{D_\ell}(F) \leq \text{Err}_1(F) + \frac{1}{\ell} \sum_{t=1}^{T} c_t \frac{d_t^2}{\rho_t^2} \tag{13}$$

where Err_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 the $t-$th SVM .

Proof: We chose $\delta = 1$ in (11). 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 (Vapnik, 1998) for a proof of this equality). □

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 - which of course also depends on the choice of C for each of the machines. 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 (as discussed in Section 2 this means that the separating function includes a bias b, so it is $f(\mathbf{x}) = \sum_{i=1}^{\ell} \alpha_i K(\mathbf{x}_i, \mathbf{x}) + b$), which is common in practice. Recent work in (Chapelle and Vapnik, 1999) may be used to extend our results to an ensemble of kernel machines with the bias not regularized: whether this can be done is an open question.

4. Experiments

To test how tight the bounds we presented are, we conducted a number of experiments using datasets from UCI[2], as well as the US Postal Service (USPS) dataset (LeCun et al., 1990). 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-out bound given by Theorem 3.1, and then compute the test performance using the test set.

2. We repeated (1) this time 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

[2] Available from `http://www.ics.uci.edu/ mlearn/MLRepository.html`

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, (selected by hand so that it is less than 1 in Figures 1-5, and more than 1 in Figure 6, for reasons explained below - for simplicity we used the same value of C in Figures 1-5, $C = 0.5$, but we found the same trend for other small values of C, $C < 1$). 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 $b \neq 0$; 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 3.1 and 3.3. The dotted line is the validation set error discussed below. The horizontal axis shows the logarithm of the σ of the Gaussian kernel used. For simplicity, only one error bar (standard deviation over the 50 training-testing splits) is shown (the others were similar). Notice that even for training-testing splits for which the error is one standard deviation away from the mean over the 50 runs (i.e. instead of plotting the graphs through the center of the error bars, we plot them at the end of the error bars) the bounds for combinations of machines are still tighter than for single machines in Figures 3 to 5. 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.*

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 ($C = 100$). 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 (Vapnik, 1998) 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. $yf(\mathbf{x}) = 1$). 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

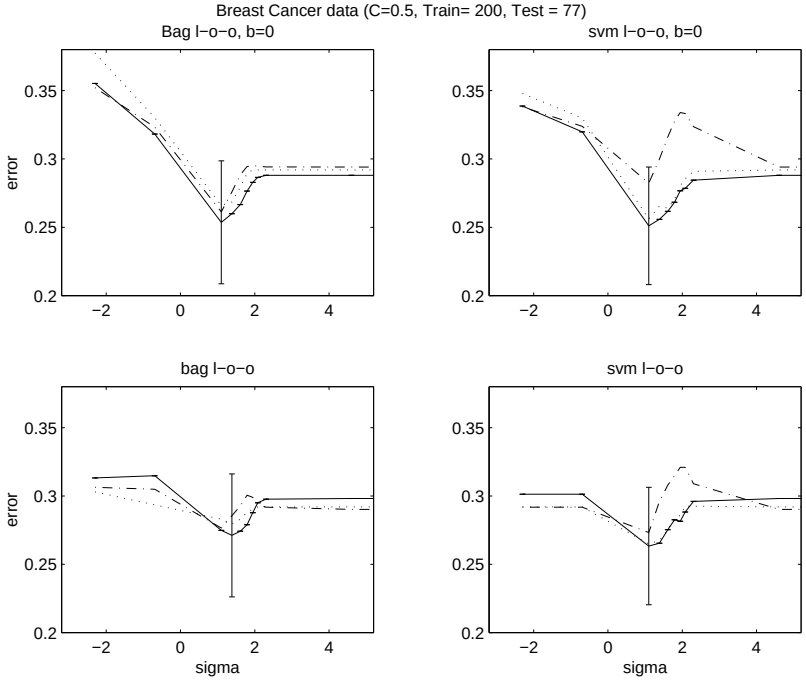

Figure 1. Breast cancer data: Top left figure: bagging with $b = 0$; Top right figure: single SVM with $b = 0$; Bottom left figure: bagging with $b \neq 0$; Bottom right figure: single SVM with $b \neq 0$. In each plot the solid line is the mean test performance and the dashed line is the error bound computed using the leave-one-out Theorems 3.1 and 3.3. The dotted line is the validation set error discussed below. The horizontal axis shows the logarithm of the σ of the Gaussian kernel used.

than) to another known leave-one-out bound for SVMs, namely one that uses the number of all support vectors to bound generalization performance (Vapnik, 1998). 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 (Bousquet and Elisseeff, 2002). It has been proposed in the past that bagging increases the "stability" of the learning methods (Breiman, 1996). Here we provide a formal argument for this. As before, we denote by D_ℓ^i the training set D_ℓ without example point $(\mathbf{x}_i, y_i)$.

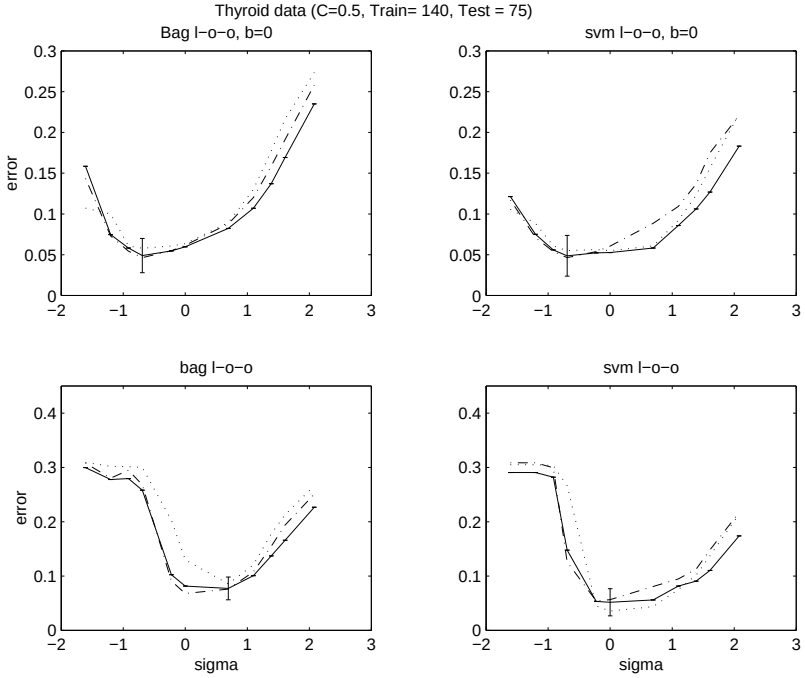

Figure 2. Thyroid data: Notation like in Figure 1.

We use the following notion of stability defined in (Bousquet and Elisseeff, 2002)

Definition (Uniform Stability): We say that a learning method is $\beta_\ell-$stable with respect to a loss function V and training sets of size ℓ if the following holds:

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

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. We define for any given constant δ the leave-one-out error Loo_δ on a training set D_ℓ to be:

$$\mathrm{Loo}_{\delta, D_\ell}(f) = \frac{1}{\ell} \sum_{i=1}^{\ell} \pi_\delta \left(-y_i f_{D_\ell^i}(\mathbf{x}_i) \right),$$

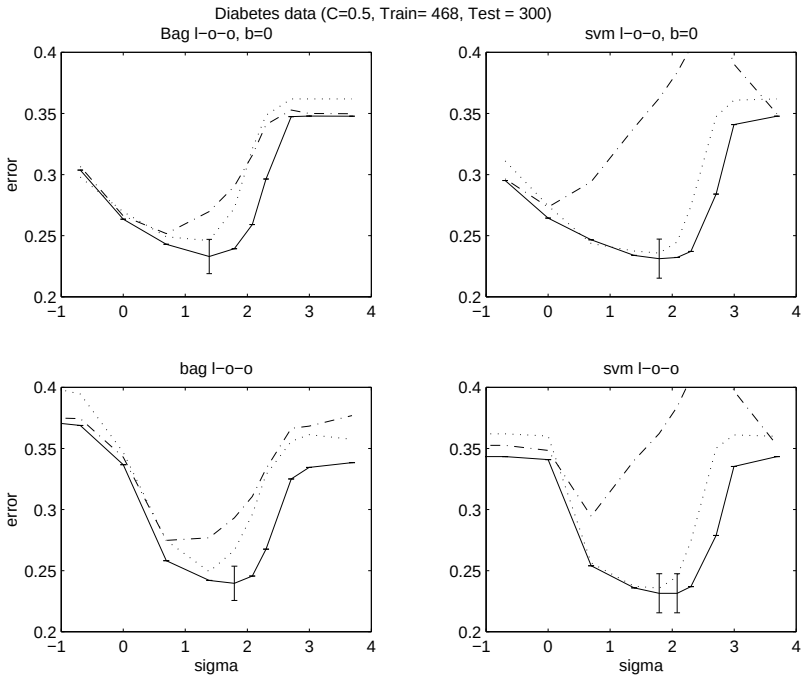

Figure 3. Diabetes data: Notation like in Figure 1.

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) [3]. For ensemble machines, we will consider again a definition similar to (7):

$$\mathrm{DLoo}_{\delta,D_\ell}(F) = \frac{1}{\ell}\sum_{i=1}^{\ell} \pi_\delta \left(-y_i \frac{1}{T}\sum_{t=1}^{T} f_{(D_{r,t}\sim D_\ell)^i}(\mathbf{x}_i) \right),$$

Notice that for $\delta \to 0$ we get the leave one out errors that we defined in Section 2, namely equations (5) and (7), and clearly $\mathrm{DLoo}_{0,D_\ell}(F) \leq \mathrm{DLoo}_{\delta,D_\ell}(F)$ 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:

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

For SVMs it is known (Bousquet and Elisseeff, 2002) that β_ℓ is upper bounded by $\frac{C\cdot\kappa}{2}$ where $\kappa = \sup_{\mathbf{x}\in\mathcal{X}} K(\mathbf{x},\mathbf{x})$ is assumed to be finite. The bound on the stability of a 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

[3] We define π_0 to be the Heavyside function θ.

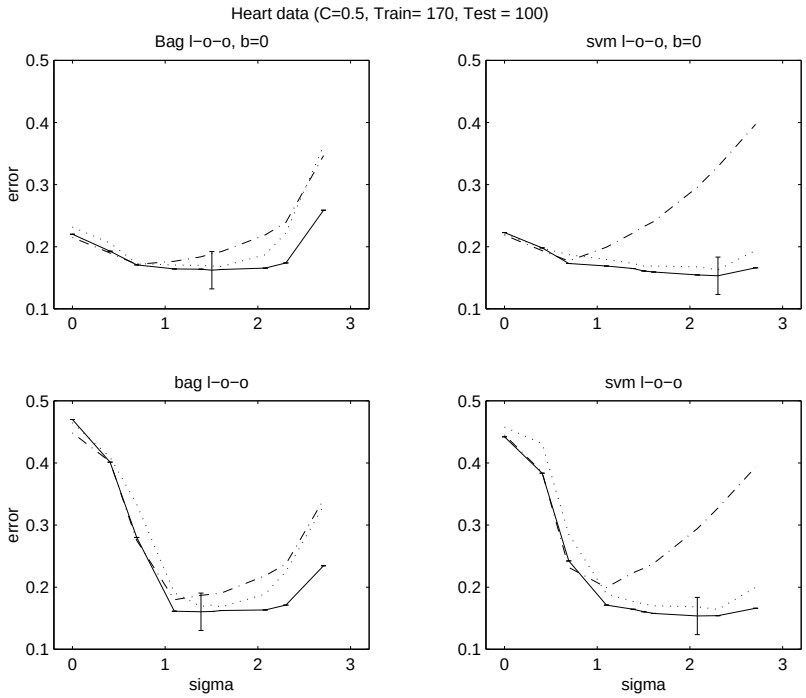

Figure 4. Heart data: Notation like in Figure 1.

learning sets of same sizes. This means that they have all the same stability for the ℓ_1 norm.

We first state a bound on the expected error of a single kernel machine in terms of its Loo_δ error. The following theorem is from (Bousquet and Elisseeff, 2002).

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

$$\mathbb{E}_{(\mathbf{x},y)}\left[\theta(-yf_{D_\ell}(\mathbf{x}))\right] \leq \text{Loo}_{\delta,D_\ell}(f_{D_\ell}) + \beta_\ell + \sqrt{\frac{\ell}{2}\left(2\frac{\beta_{\ell-1}}{\delta}+\frac{1}{\ell}\right)^2 \ln(\frac{1}{\eta})},$$

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

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) (Bousquet and Elisseeff, 2002). For a SVM, the value of β_ℓ is equal to $\frac{C\kappa}{2}$. Theorem 5.1 provides the following bound:

$$\mathbb{E}_{(\mathbf{x},y)}\left[\theta(-yf_{D_\ell}(\mathbf{x}))\right] \leq \text{Loo}_{\delta,D_\ell}(f_{D_\ell}) + \frac{C\kappa}{2} + \sqrt{\frac{\ell}{2}\left(\frac{C\kappa}{\delta}+\frac{1}{\ell}\right)^2 \ln(\frac{1}{\eta})}$$

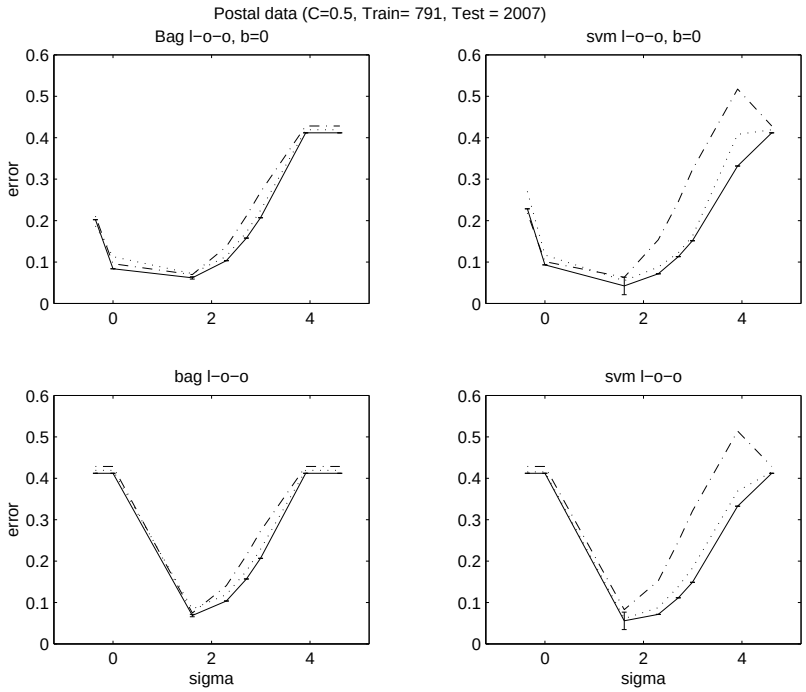

Figure 5. USPS data: Notation like in Figure 1.

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 r points drawn randomly with the uniform distribution from the training set. We consider only the case where the coefficients c_t of (4) are all $\frac{1}{T}$ (so taking the average machine like in standard bagging (Breiman, 1996)). 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 r points.

We will consider the expected combination $\hat{F}$ defined as[4]:

$$\hat{F}(\mathbf{x}) = \mathbb{E}_{D_r \sim D_\ell}[f_{D_r}(\mathbf{x})]$$

where the expectation is taken with respect to the training data D_r of size r drawn uniformly from D_ℓ. The stabilility bounds we present below hold for this expected combination and *not* for the finite combination considered so far - as mentioned below how close these two are is an open question. The

[4] 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.

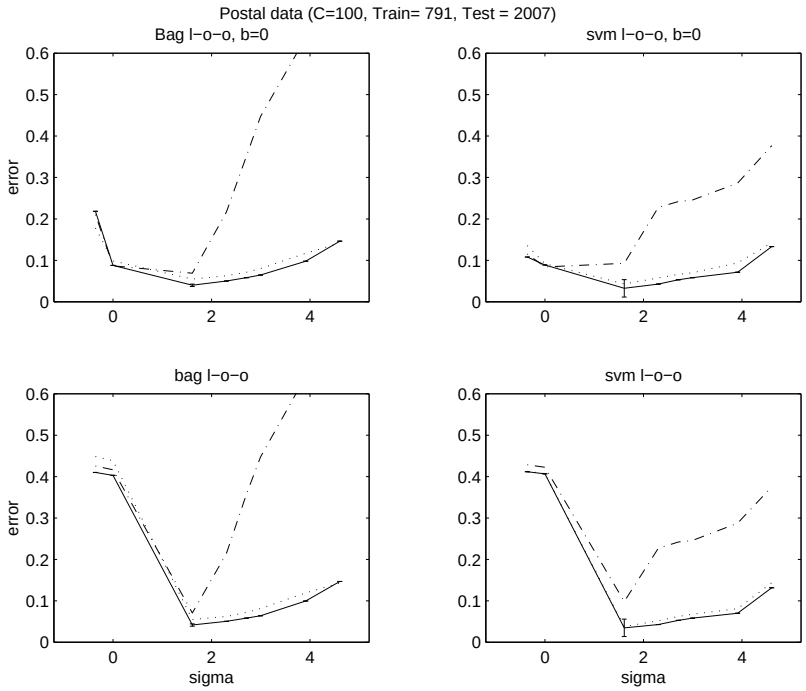

Figure 6. USPS data: Using a large C (C=50). In this case the bounds do not work - see text for an explanation. Notation like in Figure 1.

leave-one-out error we define for this expectation is again like in (7) (as in equation (7) the size r of the subsamples for simplicity is not included in the notation):

$$\mathrm{DLoo}_{\delta,D_\ell}(\hat{F}) = \frac{1}{\ell}\sum_{i=1}^{\ell}\pi_\delta\left(-y_i\mathbb{E}_{D_r\sim D_\ell}\left[f_{D_r^i}(\mathbf{x}_i)\right]\right)$$

which is different from the "standard" leave-one-out error:

$$\frac{1}{\ell}\sum_{i=1}^{\ell}\pi_\delta\left(-y_i\mathbb{E}_{D_r\sim D_\ell^i}\left[f_{D_r}(\mathbf{x}_i)\right]\right)$$

which corresponds to (6). As an extreme case when $T\to\infty$:

$$\mathrm{DLoo}_{\delta,D_\ell}\left(\frac{1}{T}\sum_{t=1}^{T}f_{D_{r,t}}\right)\to\mathrm{DLoo}_{\delta,D_\ell}(\hat{F}) \qquad (14)$$

This relation motivates the choice of our method of calculation for the leave one out estimate in Section 3. Indeed the right hand side of the equation corresponds to the quantity that we have bounded in Sections 3 and 4 and

that ultimately we would like to relate to the stability of the base machine. It is an open question to measure how fast the convergence Eq. (14) is. As we discuss below and as also mentioned in (Breiman, 1996), increasing T beyond a certain value (typically small, i.e. 100) does not influence the performance of bagging, which may imply that the convergence (14) is fast.

We then have the following bound on the expected error of ensemble combinations:

THEOREM 5.2. *For any given δ, with probability $1-\eta$ the generalization misclassification error of the expected combination of classifiers $\hat{F}$ each using a subsample of size r of the training set and each having a stability β_r wrt. the ℓ_1 norm is bounded as:*

$$\mathbb{E}_{(\mathbf{x},y)}\left[\theta(-y\hat{F}(\mathbf{x}))\right] \leq \text{DLoo}_{\delta,D_\ell}(\hat{F}) + \frac{r}{\ell}\beta_r + \sqrt{\frac{r^2}{2\ell}\left(\frac{2\beta_{r-1}}{\delta}+\frac{1}{r}\right)^2 \ln(\frac{1}{\eta})}$$

Proof: We will apply the stability theorem 5.1 to the following algorithm:

- On a set of size ℓ, the algorithm is the same as the expected ensemble machine we consider.
- On a training set of size $\ell-1$, it adds a dummy input pair $(\mathbf{x}_0, y_0)$ and uses the same sampling scheme as the one used with D_ℓ. That is, D_r is sampled from $D_\ell^i \cup \{(x_0,y_0)\}$ with the same distribution as it is sampled from D_ℓ in the definition of $\hat{F}$. When $(\mathbf{x}_0,y_0)$ is drawn in D_r, it is not used in training so that f_{D_r} is replaced by $f_{D_r\setminus\{(\mathbf{x}_0,y_0)\}}$.

The new algorithm that we will call G can then be expressed as: $G(\mathbf{x}) = \mathbb{E}_{D_r\sim D_\ell}[f_{D_r}(\mathbf{x})]$ and G^i, its outcome on the set D_ℓ^i is equal to $G^i(\mathbf{x}) = \mathbb{E}_{D_r\sim D_\ell}[f_{D_r^i}(\mathbf{x})]$ where $(\mathbf{x}_i,y_i)$ plays the role of the dummy pair $(\mathbf{x}_0,y_0)$ previously mentioned. The resulting algorithm has then the same behavior on training sets of size ℓ as the ensemble machine we consider, and the classical leave-one-out error for G corresponds to the det-leave-one-out error we have defined previously for $\hat{F}$.

From that perspective, it is sufficient to show that G is $\frac{r\beta_r}{\ell}$ stable wrt. the ℓ_1 norm and to apply theorem 5.1. We have:

$$\left|G-G^i\right| \;=\; \left|\mathbb{E}_{D_r\sim D_\ell}[f_{D_r}] - \mathbb{E}_{D_r\sim D_\ell}[f_{D_r^i}]\right|$$

where $D_r^i = D_r\setminus(\mathbf{x}_i,y_i)$. We have by definition:

$$\left|G-G^i\right| \;=\; \left|\int f_{D_r}dP - \int f_{D_r^i}dP\right|$$

where P denotes here the distribution over the sampling of D_r from D_ℓ. 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 :

$$\left|G - G^i\right| = \left| \int f_{D_r} \mathbf{1}_{(\mathbf{x}_i,y_i)\in D_r} dP + \int f_{D_r} \mathbf{1}_{(\mathbf{x}_i,y_i)\notin D_r} dP - \int f_{D_r^i} \mathbf{1}_{(\mathbf{x}_i,y_i)\in D_r} dP - \int f_{D_r^i} \mathbf{1}_{(\mathbf{x}_i,y_i)\notin D_r} dP \right|$$

Clearly, if $(\mathbf{x}_i, y_i) \notin D_r$, $D_r = D_r^i$, so that:

$$\begin{aligned} \left|G - G^i\right| &= \left| \int f_{D_r} \mathbf{1}_{(\mathbf{x}_i,y_i)\in D_r} dP - \int f_{D_r^i} \mathbf{1}_{(\mathbf{x}_i,y_i)\in D_r} dP \right| \\ &\leq \int \beta_r \mathbf{1}_{(\mathbf{x}_i,y_i)\in D_r} dP \\ &\leq \beta_r P\left[(\mathbf{x}_i, y_i) \in D_r\right] \end{aligned}$$

where the probability is taken with respect to the random subsampling of the data set D_r from D_ℓ. Since this subsampling is done without replacement, such a probability is equal to $\frac{r}{\ell}$ which finally gives a bound on the stability of $G = \mathbb{E}_{D_r\sim D_\ell}[f_{D_r}]$. This result plugged into the previous theorem gives the final bound. □

This theorem holds for ensemble combinations that are theoretically defined from the expectation $\mathbb{E}_{D_r\sim D_\ell}[f_{D_r}]$. 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 5.3. *Let $\hat{F}^S$ be a finite combination of SVMs f_s, $s = 1, .., S$ with different kernels $K^1, \ldots, K^S$:*

$$\hat{F}^S = \frac{1}{S} \sum_{s=1}^{S} \mathbb{E}_{D_r\sim D_\ell} \left[f^s_{D_r}\right] \tag{15}$$

where $f^s_{D_r\sim D_\ell}$ is a SVM with kernel K^s learned on D_r. Denote as before by $\mathrm{DLoo}_{\delta,D_\ell}(\hat{F}^S)$ the det-leave-one-out error of $\hat{F}^S$ computed with the function π_δ. Assume that each of the $f^s_{D_r\sim D_\ell}$ are learned with the same C on a subset D_r of size r drawn from D_ℓ with a uniform distribution. For any given δ, with probability $1 - \eta$, the generalization misclassification error is bounded as:

$$\mathbb{E}_{(\mathbf{x},y)}\left[\theta(-y\hat{F}^S(\mathbf{x}))\right] \leq \mathrm{DLoo}_{\delta,D_\ell}(\hat{F}^S) + \frac{r}{2\ell}(C\kappa) + \sqrt{\frac{r^2}{2\ell}\left(\frac{C\kappa}{\delta} + \frac{1}{r}\right)^2 \ln(\frac{1}{\eta})},$$

where $\kappa = \frac{1}{S}\sum_{s=1}^{S} \sup_{\mathbf{x}\in\mathcal{X}} K^s(\mathbf{x}, \mathbf{x})$.

Proof: As before, we study

$$G - G^i = \frac{1}{S}\sum_{s=1}^{S} \mathbb{E}_{D_r \sim D_\ell}\left[f^s_{D_r}\right] - E_{D_r \sim D_\ell}\left[f^s_{D^i_r}\right]$$

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

$$\left|G - G^i\right| \leq \frac{1}{S}\int \sum_{s=1}^{S} \beta_{r,s} \mathbf{1}_{(\mathbf{x}_i, y_i) \in D_r} dP,$$

where $\beta_{r,s}$ denotes the stability of a SVM with kernel K^s on a set of size r, and P is the distribution over the sampling of D_r from D_ℓ. As before, since $(\mathbf{x}_i, y_i)$ appears in D_r only $\frac{r}{\ell}$ times in average, we have the following bound:

$$\left|G - G^i\right| \leq \frac{1}{S}\sum_{s=1}^{S} \frac{\beta_{r,s} r}{\ell}.$$

Replacing $\beta_{r,s}$ by its value for the case of SVMs yields a bound on the generalization error of G in terms of its leave-one-out error. This translates for F as a bound on its generalization error in terms of its det-leave-one-out error which is the statement of the theorem. □

Notice that Theorem 5.3 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 (15)).

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 det-leave-one-out error for ensemble methods. The bounds given by the previous theorems have the form:

$$\mathbb{E}_{(\mathbf{x},y)}\left[\theta(-yF(\mathbf{x}))\right] \leq \mathrm{DLoo}_{\delta, D_\ell}(F) + O\left(\frac{r}{\sqrt{\ell}} C_r \kappa \sqrt{\frac{\ln(\frac{1}{\eta})}{\delta^2}}\right)$$

although the bound for a single SVM is:

$$\mathbb{E}_{(\mathbf{x},y)}\left[\theta(-yf(\mathbf{x}))\right] \leq \mathrm{Loo}_{\delta, D_\ell}(f) + O\left(\sqrt{\ell} C_\ell \kappa \sqrt{\frac{\ln(\frac{1}{\eta})}{\delta^2}}\right)$$

We have indexed the parameters C with an index that indicates that the 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 ($C_\ell = C_r$). The

bound derived for a combination of SVMs is then tighter than for a single SVM by a factor of r/ℓ. The improvement is because the stability of the combination of SVMs is better than the stability of a single SVM. This 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 a single SVM, C is is generally a function of ℓ, and for combination of SVMs, C also depends on the size of the subsampled learning sets D_t. In Theorem 5.2, we have seen that the stability of the combination of machines was smaller than $\frac{r\beta_r}{\ell}$ where β_r is equal to $\frac{C\kappa}{2}$ for SVMs. If this stability is better than the stability of a single machine, then combining the functions $f_{D_{r,t}}$ provides a better bound. However, in the other case, the bound gets worse. We have the following corollary whose proof is direct:

COROLLARY 5.1. *If a learning system is β_ℓ stable and $\frac{\beta_\ell}{\beta_r} < \frac{r}{\ell}$, 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 $\frac{\beta_\ell}{\beta_r} > \frac{r}{\ell}$, then combining these learning systems leads to a better bound on the difference between the test error and the leave-one-out error.*

This corollary gives an indication that combining machines should not be used if the stability of the single machine is very good. Notice that the corollary is about bounds, and not about whether the generalization error for bagging or the actual difference between the test and leave one out error is always smaller for unstable machines (and larger for stable ones) - this depends on how tight the bounds are in every case.

However, it is not often the case that we have a highly stable single machine and 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 (1996) for bagging 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 reduce 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 $\mathbb{E}_{D_r \sim D_\ell}[f_{D_r}]$ which does not improve after T has passed a certain value. This value may correspond to the convergence of the finite sum $\frac{1}{T}\sum_{t=1}^{T} f_{D_{r,t}}$ to its expectation wrt. $D_{r,t} \sim D_\ell$.

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 (Bousquet and Elisseeff, 2002), the leave-one-out and the empirical error are related by:

$$\text{Loo}_{\delta,D_\ell}(f) \leq \text{Err}_0(f_{D_\ell}) + \beta_\ell,$$

where $\text{Err}_0(f_{D_\ell})$ is the empirical error on the learning set D_ℓ. Using this inequality in Theorems 5.2, and 5.3 for the algorithm G, we can bound the generalization error of F 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 hypothesis space 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 $1 - \eta$ (Devroye et al., 1996; Vapnik, 1998):

$$TE_i \leq VE_i + \sqrt{\frac{\log(N) - \log(\frac{\eta}{4})}{V}}. \tag{16}$$

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 SVMs and for their ensembles. Experimentally we followed a slightly different procedure from what is suggested by bound (16). 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 (16) 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 (16) 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 Section 4.

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 ensembles of 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 $f^t(x), t = 1, \ldots, T$ 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[5].

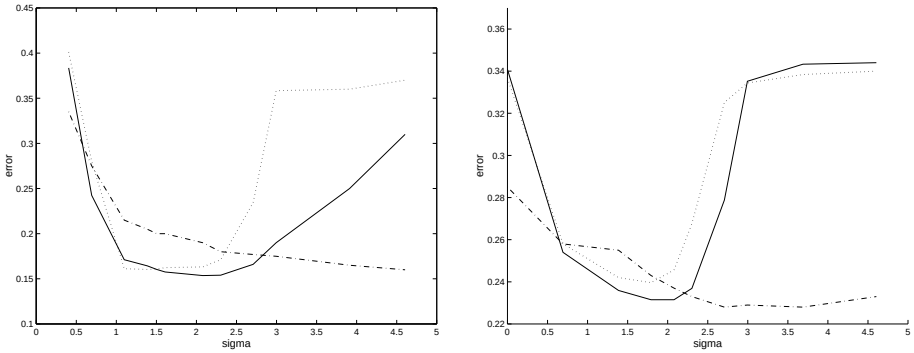

Figure 7. When the coefficients of the second layer are learned using a linear SVM the system is less sensitive to changes of the σ of the Gaussian kernel used by the individual machines of the ensemble. Solid line is one SVM, dotted is ensemble of 30 SVMs with fixed $c_t = \frac{1}{30}$, and dashed 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.

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) always uses a linear kernel. This may imply that *the overall architecture is less 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 examples. In our experiments, for all datasets except from one, learning the coefficients c_t of the combination 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 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.

[5] A validation set can still be used for model selection for these machines.

Table 6.3 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 SVMs 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 SVMs does not lead to performance improvement compared to a single SVM.*

Table II. 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 SVMs.

Dataset	**SVM**	**VCC**	**ACC**
Breast	25.5 ± 4.3	25.6 ± 4.5	25.0 ± 4.0
Thyroid	5.1 ± 2.5	5.1 ± 2.1	4.6 ± 2.7
Diabetes	23.0 ± 1.6	23.1 ± 1.4	23.0 ± 1.8
Heart	15.4 ± 3.0	15.9 ± 3.0	15.9 ± 3.2

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 - although the actual testing may be slower since the size of the union of support vectors of the combination of machines is expected to be larger

than the number of support vectors of a single machine using all the training data. This can be an important practical advantage of ensembles of machines especially in the case of large datasets. Table 6.3 compares the test performance of a single SVM with that of an ensemble of SVMs each trained with as low as 1% of the initial training set (for one dataset - for the other ones we could not use 1% because the size of the original dataset was small so 1% of it was only a couple of points). 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 (Platt, 1998).

Table III. 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 (Osuna et al., 1997).

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	0.2	0.2	0.5	0.1	0.2	0.2	0.1

7. Conclusions

We presented theoretical bounds on the generalization error of ensembles of kernel machines such as SVMs. 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 det-leave-one-out error estimate and the stability parameter of the ensembles.

We have shown that the det-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 SVMs, 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. Then ensembles of kernel machines are more stable learning algorithms than the equivalent single kernel machine. The derived bound on the difference between empirical or leave-one-out estimates and generalization error is tighter for bagging than for single kernel machines - which is experimentally observed. 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. Another important open problem is how to extend the bounds of Section 3 and 5 to the type of machines discussed in Section 6.2, or to the case of boosting (Schapire et al., 1998). As discussed above the theoretical results presented in this paper do not hold when the coefficients of the linear combination of the machines are not fixed a priori.

Acknowledgements

We would like to thank Sayan Mukherjee, Tommy Poggio, and Ryan Rifkin for their helpful feedback and ideas and Luis Perez-Breva for helping with some of the experiments. We would also like to thank the three anonymous referees for their helpful comments.

References

S. Boucheron, G. Lugosi, and P. Massart. A sharp concentration inequality with applications. *Random Structures and Algorithms*, 16:277–292, 2000.

O. Bousquet and A. Elisseeff. Stability and generalization. To appear in *Journal of Machine Learning Research*, 2002.

L. Breiman. Bagging predictors. *Machine Learning*, 26(2):123–140, 1996.

O. Chapelle and V. Vapnik. Model selection for support vector machines. In *Advances in Neural Information Processing Systems*, 1999.

L. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Number 31 in Applications of mathematics. Springer, New York, 1996.

L. Devroye and T.J. Wagner. Distribution-free performance bounds for potential function rules. *IEEE Trans. on Information Theory*, 25(5):601–604, 1979.

T. Evgeniou, M. Pontil, and T. Poggio. Regularization networks and support vector machines. Advances in Computational Mathematics 13, pp. 1–50, 2000.

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.

T. Jaakkola and D. Haussler. Exploiting generative models in discriminative classifiers. In *Proc. of Neural Information Processing Conference*, 1998.

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.

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.

E. Osuna, R. Freund, and F. Girosi. Support vector machines: Training and applications. A.I. Memo 1602, MIT A. I. Lab., 1997.

J. C. Platt. Fast training of support vector machines using sequential minimal optimization. In C. Burges, B. Schölkopf, editor, *Advances in Kernel Methods–Support Vector Learning*. MIT press, 1998.

B. Schölkopf, C. Burges, and A. Smola. *Advances in Kernel Methods – Support Vector Learning*. MIT Press, 1998.

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.

V. N. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.

G. Wahba. *Splines Models for Observational Data*. Series in Applied Mathematics, Vol. 59, SIAM, Philadelphia, 1990.