Journal of Machine Learning Research 5 (2004) 725–775 Submitted 3/03; Revised 11/03; Published 7/04

Bias-Variance Analysis of Support Vector Machines for the Development of SVM-Based Ensemble Methods

Giorgio Valentini VALENTINI@DSI.UNIMI.IT
DSI - Dipartimento di Scienze dell'Informazione
Università degli Studi di Milano
Via Comelico 39, Milano, Italy

Thomas G. Dietterich TGD@CS.ORST.EDU
Department of Computer Science
Oregon State University
Corvallis, OR 97331, USA

Editor: Nello Cristianini

Abstract

Bias-variance analysis provides a tool to study learning algorithms and can be used to properly design ensemble methods well tuned to the properties of a specific base learner. Indeed the effectiveness of ensemble methods critically depends on accuracy, diversity and learning characteristics of base learners. We present an extended experimental analysis of bias-variance decomposition of the error in Support Vector Machines (SVMs), considering Gaussian, polynomial and dot product kernels. A characterization of the error decomposition is provided, by means of the analysis of the relationships between bias, variance, kernel type and its parameters, offering insights into the way SVMs learn. The results show that the expected trade-off between bias and variance is sometimes observed, but more complex relationships can be detected, especially in Gaussian and polynomial kernels. We show that the bias-variance decomposition offers a rationale to develop ensemble methods using SVMs as base learners, and we outline two directions for developing SVM ensembles, exploiting the SVM bias characteristics and the bias-variance dependence on the kernel parameters.

Keywords: Bias-variance analysis, support vector machines, ensemble methods, multi-classifier systems.

1. Introduction

Ensembles of classifiers represent one of the main research directions in machine learning (Dietterich, 2000a). Empirical studies showed that both in classification and regression problems ensembles are often much more accurate than the individual base learner that make them up (Bauer and Kohavi, 1999; Dietterich, 2000b; Freund and Schapire, 1996), and recently different theoretical explanations have been proposed to justify the effectiveness of some commonly used ensemble methods (Kittler et al., 1998; Schapire, 1999; Kleinberg, 2000; Allwein et al., 2000).

Two main theories are invoked to explain the success of ensemble methods. The first one considers the ensembles in the framework of large margin classifiers (Mason et al., 2000), showing that ensembles enlarge the margins, enhancing the generalization capabilities of learning algorithms (Schapire et al., 1998; Allwein et al., 2000). The second is based on the classical bias-variance decomposition of the error (Geman et al., 1992), and it shows that ensembles can reduce variance (Breiman, 1996b) and also bias (Kong and Dietterich, 1995).

Recently Domingos proved that Schapire's notion of margins (Schapire et al., 1998) can be expressed in terms of bias and variance and vice versa (Domingos, 2000c), and hence Schapire's bounds of ensemble's generalization error can be equivalently expressed in terms of the distribution of the margins or in terms of the bias-variance decomposition of the error, showing the equivalence of margin-based and bias-variance-based approaches.

The effectiveness of ensemble methods depends on the specific characteristics of the base learners; in particular on the relationship between diversity and accuracy of the base learners (Dietterich, 2000a; Kuncheva et al., 2001b; Kuncheva and Whitaker, 2003), on their stability (Breiman, 1996b; Bousquet and Elisseeff, 2002), and on their general geometrical properties (Cohen and Intrator, 2001).

From this standpoint the analysis of the features and properties of the base learners used in ensemble methods is crucial in order to design ensemble methods well tuned to the characteristics of a specific base learner. For instance, considering that the agglomeration of many classifiers into one classification rule reduces variance (Breiman, 1996a), we could apply low-bias base learners to reduce both bias and variance using ensemble methods. To this purpose in this paper we study Support Vector Machines (SVMs), that are "strong" dichotomic classifiers, well founded on Vapnik's statistical learning theory (Vapnik, 1998), in order to establish if and how we can exploit their specific features in the context of ensemble methods. We analyze the learning properties of SVMs using the bias-variance decomposition of the error as a tool to understand the relationships between kernels, kernel parameters, and learning processes in SVM.

Historically, the bias-variance insight was borrowed from the field of regression, using squared-loss as the loss function (Geman et al., 1992). For classification problems, where the $0/1$ loss is the main criterion, several authors proposed bias-variance decompositions related to $0/1$ loss. Kong and Dietterich (1995) proposed a bias-variance decomposition in the context of ECOC ensembles (Dietterich and Bakiri, 1995), but their analysis is extensible to arbitrary classifiers, even if they defined variance simply as a difference between loss and bias.

In Breiman's decomposition (Breiman, 1996b) bias and variance are always non-negative (while Dietterich definition allows a negative variance), but at any input the reducible error (i.e. the total error rate less noise) is assigned entirely to variance if the classification is unbiased, and to bias if biased. Moreover he forced the decomposition to be purely additive, while for the 0/1 loss this is not the case. Kohavi and Wolpert (1996) approach leads to a biased estimation of bias and variance, assigning a non-zero bias to a Bayes classifier, while Tibshirani (1996) did not use directly the notion of variance, decomposing the $0/1$ loss in bias and an unrelated quantity he called "aggregation effect", which is similar to the James' notion of *variance effect* (James, 2003).

Friedman (1997) showed that bias and variance are not purely additive: in some cases increasing variance increases the error, but in other cases can also reduce the error, especially when the prediction is biased.

Heskes (1998) proposed a bias-variance decomposition using as loss function the Kullback-Leibler divergence. By this approach the error between the target and the predicted classifier densities is measured; anyway when he tried to extend this approach to the zero-one function interpreted as the limit case of log-likelihood type error, the resulting decomposition produces a definition of bias that loses his natural interpretation as systematic error committed by the classifier.

As briefly outlined, these decompositions suffer of significant shortcomings: in particular they lose the relationship to the original squared loss decomposition, forcing in most cases bias and variance to be purely additive.

We consider classification problems and the $0/1$ loss function in the Domingos' unified framework of bias-variance decomposition of the error (Domingos, 2000c,b). In this approach bias and variance are defined for an arbitrary loss function, showing that the resulting decomposition specializes to the standard one for squared loss, but it holds also for the $0/1$ loss (Domingos, 2000c).

A similar approach has been proposed by James (2003): he extended the notion of variance and bias for general loss functions, distinguishing also between bias and variance, interpreted respectively as the systematic error and the variability of an estimator, and the actual effect of bias and variance on the error.

Using Domingos' theoretical framework, we tried to answer two main questions:

1. Can we characterize bias and variance in SVMs with respect to the kernel and its parameters?

2. Can the bias-variance decomposition offer guidance for developing ensemble methods using SVMs as base learners?

In order to answer these two questions, we planned and performed an extensive series of experiments on synthetic and real data sets to evaluate bias variance-decomposition of the error with different kernels and different kernel parameters.

The paper is organized as follows. In Section 2, we summarize the main results of Domingos' unified bias-variance decomposition of error. Section 3 outlines how to measure in practice bias and variance decomposition of the error with artificial or large benchmark data sets, or when only a small "real" data set is available. Section 4 outlines the main characteristics of the data sets employed in our experiments and the main experimental tasks performed. Then we present the main results of our experiments about bias-variance decomposition of the error in SVMs, considering separately Gaussian, polynomial and and dot product SVMs, and comparing also the results between different kernels. Section 6 provides a characterization of bias-variance decomposition of the error for Gaussian, polynomial and and dot product SVMs, highlighting the common patterns for each different kernel. Section 7 exploits the knowledge achieved by the bias-variance decomposition of the error to formulate hypotheses about the effectiveness of SVMs as base learners in ensembles of learning machines, and two directions for developing new ensemble models of SVM are proposed. An outline of ongoing and future developments of this work concludes the paper.

2. Bias-Variance Decomposition for the 0/1 Loss Function

The analysis of bias-variance decomposition of the error has been originally developed in the standard regression setting, where the squared error is usually used as loss function. Considering a prediction $y = f(x)$ of an unknown target t, provided by a learner f on input x, with $\mathbf{x} \in \mathbb{R}^d$ and $y \in \mathbb{R}$, the classical decomposition of the error in bias and variance for the squared error loss is (Geman et al., 1992)

$$\begin{aligned} E_{y,t}[(y-t)^2] &= E_t[(t-E[t])^2] + E_y[(y-E[y])^2] + (E[y]-E[t])^2 \\ &= Noise(t) + Var(y) + Bias^2(y). \end{aligned}$$

In words, the expected loss of using y to predict t is the sum of the variances of t (noise) and y plus the squared bias. $E_y[\cdot]$ indicates the expected value with respect to the distribution of the random variable y.

This decomposition cannot be automatically extended to the standard classification setting, as in this context the 0/1 loss function is usually applied, and bias and variance are not purely additive. As we are mainly interested in analyzing bias-variance for classification problems, we introduce the bias-variance decomposition for the 0/1 loss function, according to the Domingos unified bias-variance decomposition of the error (Domingos, 2000b).

2.1 Expected Loss Depends on the Randomness of the Training Set and the Target

Consider a (potentially infinite) population U of labeled training data points, where each point is a pair $(\mathbf{x}_j, t_j)$, $t_j \in \mathcal{C}$, $\mathbf{x}_j \in \mathbb{R}^d$, $d \in \mathbb{N}$, where $\mathcal{C}$ is the set of the class labels. Let $P(\mathbf{x}, t)$ be the joint distribution of the data points in U. Let D be a set of m points drawn identically and independently from U according to P. We think of D as being the training sample that we are given for training a classifier. We can view D as a random variable, and we will let $E_D[\cdot]$ indicate the expected value with respect to the distribution of D.

Let $\mathcal{L}$ be a learning algorithm, and define $f_D = \mathcal{L}(D)$ as the classifier produced by $\mathcal{L}$ applied to a training set D. The model produces a prediction $f_D(\mathbf{x}) = y$. Let $L(t, y)$ be the $0/1$ loss function, that is $L(t, y) = 0$ if $y = t$, and $L(t, y) = 1$ otherwise.

Suppose we consider a fixed point $\mathbf{x} \in \mathbb{R}^d$. This point may appear in many labeled training points in the population. We can view the corresponding labels as being distributed according to the conditional distribution $P(t|\mathbf{x})$. Recall that it is always possible to factor the joint distribution as $P(\mathbf{x}, t) = P(\mathbf{x})P(t|\mathbf{x})$. Let $E_t[\cdot]$ indicate the expectation with respect to t drawn according to $P(t|\mathbf{x})$.

Suppose we consider a *fixed* predicted class y for a given $\mathbf{x}$. This prediction will have an expected loss of $E_t[L(t, y)]$. In general, however, the prediction y is not fixed. Instead, it is computed from a model f_D which is in turn computed from a training sample D.

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

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

where $f_D = \mathcal{L}(D)$ is the classifier learned by $\mathcal{L}$ on training data D. The purpose of the bias-variance analysis is to decompose this expected loss into terms that separate the bias and the variance.

2.2 Optimal and Main Prediction

To derive this decomposition, we must define two things: the *optimal prediction* and the *main prediction*: according to Domingos, bias and variance can be defined in terms of these quantities.

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

$$y_*(\mathbf{x}) = \arg\min_y E_t[L(t, y)]. \tag{1}$$

It is equal to the label t that is observed more often in the universe U of the data points, and corresponds to the prediction provided by the Bayes classifier. The *optimal model* $\hat{f}(\mathbf{x}) = y_*$, $\forall \mathbf{x}$

makes the optimal prediction at each point $\mathbf{x}$, and corresponds to the Bayes classifier; its error rate corresponds to the Bayes error rate.

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

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

Note that in the deterministic case $y_* = t$ and $N(\mathbf{x}) = 0$.

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

$$y_m = \arg\min_{y'} E_D[L(f_D(\mathbf{x}), y')]. \tag{2}$$

This is a value that would give the lowest expected loss if it were the "true label" of $\mathbf{x}$. It expresses the "central tendency" of a learner, that is its systematic prediction, or, in other words, it is the label for $\mathbf{x}$ that the learning algorithm "wishes" were correct. For 0/1 loss, the main prediction is the class predicted most often by the learning algorithm $\mathcal{L}$ when applied to training sets D.

2.3 Bias, Unbiased and Biased Variance.

Given these definitions, the *bias* $B(\mathbf{x})$ (of a learning algorithm $\mathcal{L}$ on training sets of size m) is the loss of the main prediction relative to the optimal prediction:

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

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

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

$$V(\mathbf{x}) = E_D[L(y_m, f_D(\mathbf{x}))]. \tag{3}$$

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

In the case of the 0/1 loss we can also distinguish two opposite effects of variance (and noise) on the error: in the unbiased case variance and noise increase the error, while in the biased case variance and noise decrease the error.

There are three components that determine whether $t = y$:

1. Noise: is $t = y_*$?
2. Bias: is $y_* = y_m$?
3. Variance: is $y_m = y$?

Note that bias is either 0 or 1 because neither y_* nor y_m are random variables. From this standpoint we can consider two different cases: the unbiased and the biased case.

In the unbiased case, $B(\mathbf{x}) = 0$ and hence $y_* = y_m$. In this case we suffer a loss if the prediction y differs from the main prediction y_m (variance) and the optimal prediction y_* is equal to the target t, or y is equal to y_m, but y_* is different from t (noise).

In the biased case, $B(\mathbf{x}) = 1$ and hence $y_* \neq y_m$. In this case we suffer a loss if the prediction y is equal to the main prediction y_m and the optimal prediction y_* is equal to the target t, or if both y is different from y_m (variance), and y_* is different from t (noise). Figure 1 summarizes the different

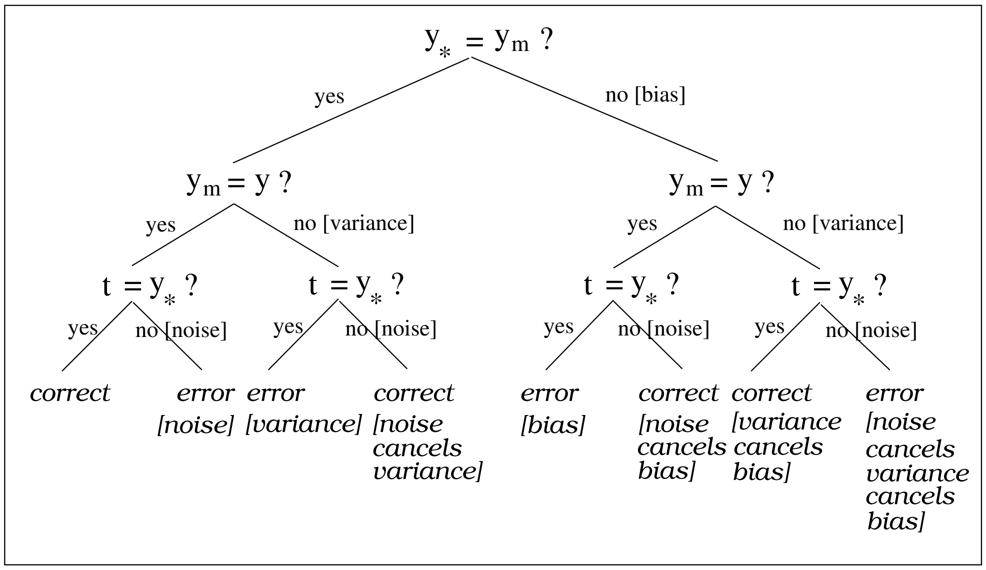

Figure 1: Case analysis of error.

conditions under which an error can arise, considering the combined effect of bias, variance and noise on the learner prediction.

Considering the above case analysis of the error, if we let $P(t \neq y_*) = N(\mathbf{x}) = \tau$ and $P(y_m \neq y) = V(\mathbf{x}) = \sigma$, in the unbiased case we have

$$\begin{aligned} L(t,y) &= \tau(1-\sigma)+\sigma(1-\tau) \\ &= \tau+\sigma-2\tau\sigma \\ &= N(\mathbf{x})+V(\mathbf{x})-2N(\mathbf{x})V(\mathbf{x}), \end{aligned} \tag{4}$$

while in the biased case

$$\begin{aligned} L(t,y) &= \tau\sigma+(1-\tau)(1-\sigma) \\ &= 1-(\tau+\sigma-2\tau\sigma) \\ &= B(\mathbf{x})-(N(\mathbf{x})+V(\mathbf{x})-2N(\mathbf{x})V(\mathbf{x})). \end{aligned} \tag{5}$$

Note that in the unbiased case (Equation 4) the variance is an additive term of the loss function, while in the biased case (Equation 5) the variance is a subtractive term of the loss function. Moreover the interaction terms will usually be small, because, for instance, if both noise and variance term will be both lower than 0.1, the interaction term $2N(\mathbf{x})V(\mathbf{x})$ will be reduced to less than 0.02.

In order to distinguish between these two different effects of the variance on the loss function, Domingos defines the *unbiased variance*, $V_u(\mathbf{x})$, to be the variance when $B(\mathbf{x}) = 0$ and the *biased variance*, $V_b(\mathbf{x})$, to be the variance when $B(\mathbf{x}) = 1$. We can also define the *net variance* $V_n(\mathbf{x})$ to take into account the combined effect of the unbiased and biased variance:

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

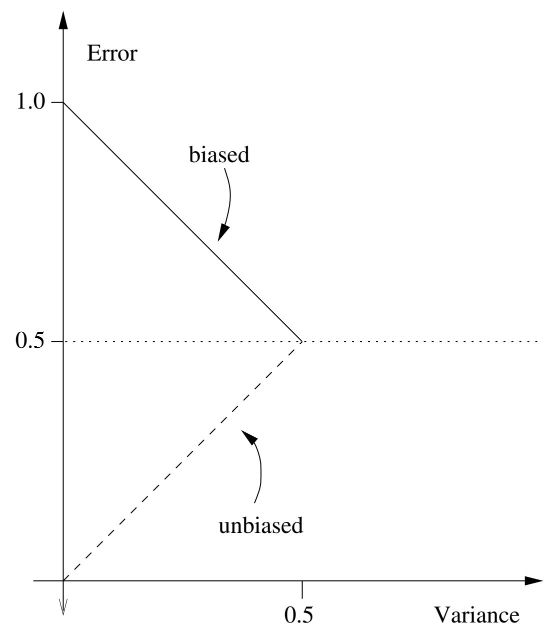

Figure 2: Effects of biased and unbiased variance on the error. The unbiased variance increments, while the biased variance decrements the error.

Figure 2 summarizes in graphic form the opposite effects of biased and unbiased variance on the error.

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

2.4 Domingos' Bias-Variance Decomposition

Domingos (2000a) showed that for a quite general loss function the expected loss is

$$EL(\mathcal{L}, \mathbf{x}) = c_1 N(\mathbf{x}) + B(\mathbf{x}) + c_2 V(\mathbf{x}). \tag{6}$$

For the $0/1$ loss function c_1 is $2P_D(f_D(\mathbf{x}) = y_*) - 1$ and c_2 is +1 if $B(\mathbf{x}) = 0$ and -1 if $B(\mathbf{x}) = 1$. Note that $c_2 V(\mathbf{x}) = V_u(\mathbf{x}) - V_b(\mathbf{x}) = V_n(\mathbf{x})$ (Equation 3), and if we disregard the noise, Equation 6 can be simplified to

$$EL(\mathcal{L}, \mathbf{x}) = B(\mathbf{x}) + V_n(\mathbf{x}). \tag{7}$$

Summarizing, one of the most interesting aspects of Domingos' decomposition is that variance hurts on unbiased points $\mathbf{x}$, but it helps on biased points. Nonetheless, to obtain low overall expected loss, we want the bias to be small, and hence, we see to reduce both the bias and the unbiased variance. A good classifier will have low bias, in which case the expected loss will approximately equal the variance.

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

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

3. Measuring Bias and Variance

The procedures to measure bias and variance depend on the characteristics and on the cardinality of the data sets used.

For synthetic data sets we can generate different sets of training data for each learner to be trained. Then a large synthetic test set can be generated in order to estimate the bias-variance decomposition of the error for a specific learner model.

Similarly, if a large data set is available, we can split it in a large learning set and in a large testing set. Then we can randomly draw subsets of data from the large training set in order to train the learners; bias-variance decomposition of the error is measured on the large independent test set.

However, in practice, for real data we dispose of only one and often small data set. In this case, we can use cross-validation techniques for estimating bias-variance decomposition, but we propose to use out-of-bag (Breiman, 2001) estimation procedures, as they are computationally less expensive.

3.1 Measuring with Artificial or Large Benchmark Data Sets

Consider a set $\mathcal{D} = \{D_i\}_{i=1}^n$ of learning sets $D_i = \{\mathbf{x}_k, t_k\}_{k=1}^m$. The data sets D_i can be generated according to some known probability distribution or can be drawn with replacement from a large data set $\mathcal{D}$ according to an uniform probability distribution. Here we consider only a two-class case, i.e. $t_k \in \mathcal{C} = \{-1, 1\}$, $\mathbf{x}_k \in \mathbf{X}$, for instance $\mathbf{X} = \mathbb{R}^d$, $d \in \mathbb{N}$, but the extension to the multiclass case is straightforward.

The estimates of the error, bias, unbiased and biased variance are performed on a test set $\mathcal{T}$ separated from the training set $\mathcal{D}$. In particular these estimates with respect to a single example $(\mathbf{x}, t) \in \mathcal{T}$ are performed using the classifiers $f_{D_i} = \mathcal{L}(D_i)$ produced by a learner $\mathcal{L}$ using training sets D_i drawn from $\mathcal{D}$. These classifiers produce a prediction $y \in \mathcal{C}$, that is $f_{D_i}(\mathbf{x}) = y$. The estimates are performed for all the $(\mathbf{x}, t) \in \mathcal{T}$, and the overall loss, bias and variance can be evaluated averaging over the entire test set $\mathcal{T}$.

In presence of noise and with the 0/1 loss, the optimal prediction y_* is equal to the label t that is observed more often in the universe U of data points:

$$y_*(\mathbf{x}) = \arg\max_{t \in \mathcal{C}} P(t|\mathbf{x}).$$

The noise $N(\mathbf{x})$ for the 0/1 loss can be estimated if we can evaluate the probability of the targets for a given example $\mathbf{x}$:

$$N(\mathbf{x}) = \sum_{t \in \mathcal{C}} L(t, y_*) P(t|\mathbf{x}) = \sum_{t \in \mathcal{C}} ||t \neq y_*|| P(t|\mathbf{x}),$$

where $||z|| = 1$ if z is true, 0 otherwise. In practice it is difficult to estimate the noise for "real world" data sets, and to simplify the computation we consider the noise free case. In this situation we have $y_* = t$.

The main prediction is a function of the $y = f_{D_i}(\mathbf{x})$. Considering a $0/1$ loss, we have

$$y_m = \arg\max(p_1, p_{-1}),$$

where $p_1 = P_D(y = 1|\mathbf{x})$ and $p_{-1} = P_D(y = -1|\mathbf{x})$, i.e. the main prediction is the mode. To calculate p_1, having a test set $\mathcal{T} = \{\mathbf{x}_j, t_j\}_{j=1}^r$, it is sufficient to count the number of learners that predict class 1 on a given input $\mathbf{x}$:

$$p_1(\mathbf{x}_j) = \frac{\sum_{i=1}^n \|f_{D_i}(\mathbf{x}_j) = 1\|}{n},$$

where $\|z\| = 1$ if z is true and $\|z\| = 0$ if z is false

The bias can be easily calculated after the evaluation of the main prediction:

$$B(\mathbf{x}) = \begin{cases} 1 & \text{if } y_m \neq t \\ 0 & \text{if } y_m = t \end{cases} = \left| \frac{y_m - t}{2} \right|, \tag{8}$$

or equivalently:

$$B(\mathbf{x}) = \begin{cases} 1 & \text{if } p_{corr}(\mathbf{x}) \leq 0.5 \\ 0 & \text{otherwise}, \end{cases}$$

where p_{corr} is the probability that a prediction is correct, i.e. $p_{corr}(\mathbf{x}) = P(y = t|\mathbf{x}) = P_D(f_D(\mathbf{x}) = t)$.

In order to measure the variance $V(\mathbf{x})$, if we define $y_{D_i} = f_{D_i}(\mathbf{x})$, we have

$$V(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^n L(y_m, y_{D_i}) = \frac{1}{n}\sum_{i=1}^n ||(y_m \neq y_{D_i})||$$

The *unbiased variance* $V_u(\mathbf{x})$ and the *biased variance* $V_b(\mathbf{x})$ can be calculated evaluating if the prediction of each learner differs from the main prediction respectively in the unbiased and in the biased case:

$$V_u(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^n ||(y_m = t) \text{ and } (y_m \neq y_{D_i})||,$$

$$V_b(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^n ||(y_m \neq t) \text{ and } (y_m \neq y_{D_i})||.$$

In the noise-free case, the *average loss on the example* $\mathbf{x}$ $E_D(\mathbf{x})$ is calculated by a simple algebraic sum of bias, unbiased and biased variance:

$$E_D(\mathbf{x}) = B(\mathbf{x}) + V_u(\mathbf{x}) - V_b(\mathbf{x}) = B(\mathbf{x}) + (1 - 2B(\mathbf{x}))V(\mathbf{x}).$$

We can easily calculate the *average bias, variance, unbiased, biased and net variance*, averaging over the entire set of the examples of the test set $\mathcal{T} = \{(\mathbf{x}_j, t_j)\}_{j=1}^r$. In the remaining part of this section the indices j refer to the examples that belong to the test set $\mathcal{T}$, while the indices i refer to the training sets D_i, drawn with replacement from the separated training set $\mathcal{D}$, and used to train the classifiers f_{D_i}.

The average quantities are

Average bias:

$$E_{\mathbf{x}}[B(\mathbf{x})] = \frac{1}{r}\sum_{j=1}^r B(\mathbf{x}_j) = \frac{1}{r}\sum_{j=1}^r \left| \frac{y_m(\mathbf{x}_j) - t_j}{2} \right|,$$

Average variance:

$$\begin{aligned} E_{\mathbf{x}}[V(\mathbf{x})] &= \frac{1}{r}\sum_{j=1}^{r} V(\mathbf{x}_j) \\ &= \frac{1}{nr}\sum_{j=1}^{r}\sum_{i=1}^{n} L(y_m(\mathbf{x}_j), f_{D_i}(\mathbf{x}_j)) \\ &= \frac{1}{nr}\sum_{j=1}^{r}\sum_{i=1}^{n} ||y_m(\mathbf{x}_j) \neq f_{D_i}(\mathbf{x}_j)||, \end{aligned}$$

Average unbiased variance:

$$E_{\mathbf{x}}[V_u(\mathbf{x})] = \frac{1}{r}\sum_{j=1}^{r} V_u(\mathbf{x}_j) = \frac{1}{nr}\sum_{j=1}^{r}\sum_{i=1}^{n} ||(y_m(\mathbf{x}_j) = t_j) \text{ and } (y_m(\mathbf{x}_j) \neq f_{D_i}(\mathbf{x}_j))||,$$

Average biased variance:

$$E_{\mathbf{x}}[V_b(\mathbf{x})] = \frac{1}{r}\sum_{j=1}^{r} V_b(\mathbf{x}_j) = \frac{1}{nr}\sum_{j=1}^{r}\sum_{i=1}^{n} ||(y_m(\mathbf{x}_j) \neq t_j) \text{ and } (y_m(\mathbf{x}_j) \neq f_{D_i}(\mathbf{x}_j))||,$$

and the *Average net variance*:

$$E_{\mathbf{x}}[V_n(\mathbf{x})] = \frac{1}{r}\sum_{j=1}^{r} V_n(\mathbf{x}_j) = \frac{1}{r}\sum_{j=1}^{r} (V_u(\mathbf{x}_j) - V_b(\mathbf{x}_j)).$$

Finally, the average loss on all the examples (with no noise) is the algebraic sum of the average bias, unbiased and biased variance.

$$E_{\mathbf{x}}[L(t,y)] = E_{\mathbf{x}}[B(\mathbf{x})] + E_{\mathbf{x}}[V_u(\mathbf{x})] - E_{\mathbf{x}}[V_b(\mathbf{x})]$$

3.2 Measuring with Small Data Sets

In practice (unlike in theory), we have only one and often small data set $\mathcal{S}$. We can simulate multiple training sets by bootstrap replicates $S_b = \{\mathbf{x}|\mathbf{x} \text{ is drawn at random with replacement from } \mathcal{S}\}$. In order to measure bias and variance we can use out-of-bag points, providing in such a way an unbiased estimate of the error. At first we need to construct B bootstrap replicates of $\mathcal{S}$ (e. g., $B = 200$): $S_1, \ldots, S_B$. Then we apply a learning algorithm $\mathcal{L}$ to each replicate S_b to obtain hypotheses $f_b = \mathcal{L}(S_b)$.

Let $T_b = \mathcal{S} \backslash S_b$ be the data points that do not appear in S_b (out of bag points). We can use these data sets T_b to evaluate the bias-variance decomposition of the error; that is we compute the predicted values $f_b(\mathbf{x})$, $\forall\mathbf{x}$ s.t. $\mathbf{x} \in T_b$. For each data point $\mathbf{x}$, we have now the observed corresponding value t and several predictions $y_1, \ldots, y_K$, where $K = |\{T_b|\mathbf{x} \in T_b, 1 \leq b \leq B\}|$, $K \leq B$, and on the average $K \simeq B/3$, because about $1/3$ of the predictors is not trained on a specific input $\mathbf{x}$. Note that the value of K depends on the specific example $\mathbf{x}$ considered. Moreover if $\mathbf{x} \in T_b$ then $\mathbf{x} \notin S_b$, hence $f_b(\mathbf{x})$ makes a prediction on an unknown example $\mathbf{x}$.

In order to compute the main prediction, for a two-class classification problem, we can define

$$p_1(\mathbf{x}) = \frac{1}{K}\sum_{b=1}^{B} ||(\mathbf{x} \in T_b) \text{ and } (f_b(\mathbf{x}) = 1)||,$$

$$p_{-1}(\mathbf{x}) = \frac{1}{K}\sum_{b=1}^{B} ||(\mathbf{x} \in T_b) \text{ and } (f_b(\mathbf{x}) = -1)||.$$

The main prediction $y_m(\mathbf{x})$ corresponds to the mode:

$$y_m = \arg\max(p_1, p_{-1}).$$

The bias can be calculated as in Equation 8, and the variance $V(\mathbf{x})$ is

$$V(\mathbf{x}) = \frac{1}{K}\sum_{b=1}^{B} ||(\mathbf{x} \in T_b) \text{ and } (y_m \neq f_b(\mathbf{x}))||.$$

Similarly easily computed are the unbiased, biased and net-variance:

$$V_u(\mathbf{x}) = \frac{1}{K}\sum_{b=1}^{B} ||(\mathbf{x} \in T_b) \text{ and } (B(\mathbf{x}) = 0) \text{ and } (y_m \neq f_b(\mathbf{x}))||,$$

$$V_b(\mathbf{x}) = \frac{1}{K}\sum_{b=1}^{B} ||(\mathbf{x} \in T_b) \text{ and } (B(\mathbf{x}) = 1) \text{ and } (y_m \neq f_b(\mathbf{x}))||,$$

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

Average bias, variance, unbiased, biased and net variance, can be easily calculated averaging over all the examples.

4. Bias-Variance Analysis in SVMs

The bias-variance decomposition of the error represents a powerful tool to analyze learning processes in learning machines. According to the procedures described in the previous section, we measured bias and variance in SVMs, in order to study the relationships with different kernel types and their parameters. To accomplish this task we computed bias-variance decomposition of the error on different synthetic and "real" data sets.

4.1 Experimental Setup

In the experiments we employed seven different data sets, both synthetic and "real". *P2* is a synthetic bidimensional two-class data set;[1] each region is delimited by one or more of four simple polynomial and trigonometric functions (Figure 3). The synthetic data set *Waveform* is generated from a combination of two of three "base" waves; we reduced the original three classes of *Waveform* to two, deleting all samples pertaining to class 0. The other data sets are all from the UCI repository (Merz and Murphy, 1998). Table 4.1 summarizes the main features of the data sets used in the experiments.

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

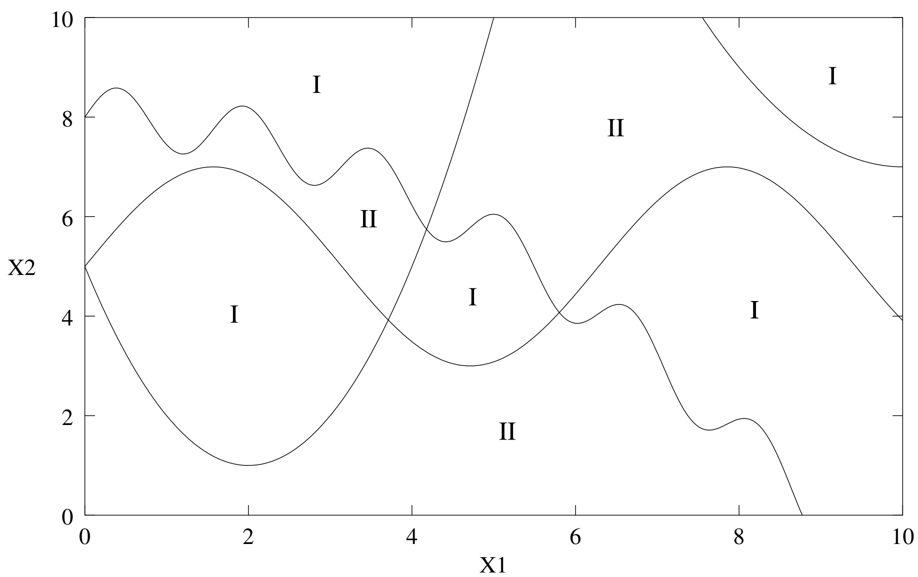

Figure 3: P2 data set, a bidimensional two class synthetic data set. Roman numbers label the regions belonging to the two classes.

Data set	# of attr.	# of tr. samples	# of tr. sets	# base tr. set	# of test samples
P2	2	100	400	synthetic	10000
Waveform	21	100	200	synthetic	10000
Grey-Landsat	36	100	200	4425	2000
Letter	16	100	200	614	613
Letter w. noise	16	100	200	614	613
Spam	57	100	200	2301	2300
Musk	166	100	200	3299	3299

Table 1: Data sets used in the experiments.

In order to perform a reliable evaluation of bias and variance we used small training set and large test sets. For synthetic data we generated the desired number of samples. For real data sets we used bootstrapping to replicate the data. In both cases we computed the main prediction, bias, unbiased and biased variance, net-variance according to the procedures explained in Section 3.1. In our experiments, the computation of James' variance and systematic effect (James, 2003) is reduced to the measurements of the net-variance and bias, and hence we did not explicitly compute these quantities (see Appendix *A* for details).

With synthetic data sets, we generated small training sets of about 100 examples and reasonably large test sets using computer programs. In fact small samples show bias and variance more clearly than having larger samples. We produced 400 different training sets for *P2* and 200 training sets for

Waveform. The test sets were chosen reasonably large (10000 examples) to obtain reliable estimates of bias and variance.

For real data sets we first divided the data into a training $\mathcal{D}$ and a test $\mathcal{T}$ sets. If the data sets had a predefined training and test sets reasonably large, we used them (as in *Grey-Landsat* and *Spam*), otherwise we split them in a training and test set of equal size. Then we drew from $\mathcal{D}$ bootstrap samples. We chose bootstrap samples much smaller than $|\mathcal{T}|$ (100 examples). More precisely we drew 200 data sets from $\mathcal{D}$, each one consisting of 100 examples uniformly drawn with replacement.

Summarizing, both with synthetic and real data sets we generated small training sets for each data set and a much larger test set. Then all the data were normalized in such a way that for each attribute the mean was 0 and the standard deviation 1. In all our experiments we used *NEURObjects* (Valentini and Masulli, 2002),[2] a C++ library for the development of neural networks and machine learning applications, and *SVM-light* (Joachims, 1999), a set of C applications for training and testing SVMs.

We developed and used the C++ application `analyze_BV`, to perform bias-variance decomposition of the error.[3] This application analyzes the output of a generic learning machine model and computes the main prediction, error, bias, net-variance, unbiased and biased variance using the $0/1$ loss function. Other C++ applications have been developed to process and analyze the results, using also Cshell scripts to train, test and analyze bias-variance decomposition of all the SVM models for each specific data set.

4.2 Experimental Tasks

To evaluate bias and variance in SVMs we conducted experiments with different kernels (Gaussian, polynomial and dot product) and different kernel parameters. For each kernel we considered the same set of values for the parameter C that controls the trade-off between training error and margin, ranging from $C = 0.01$ to $C = 1000$.

1. **Gaussian kernels**. We evaluated bias-variance decomposition varying the parameters σ of the kernel and the C parameter. In particular we analyzed:

   (a) The relationships between average error, bias, net-variance, unbiased and biased variance, the σ parameter of the kernel and the C parameter.

   (b) The relationships between generalization error, training error, number of support vectors and capacity with respect to σ.

   We trained RBF-SVM with all the combinations of the parameters σ and C, using the a set of values for σ ranging from $\sigma = 0.01$ to $\sigma = 1000$. We evaluated about 200 different RBF-SVM models for each data set.

2. **Polynomial kernels**. We evaluated bias-variance decomposition varying the degree of the kernel and the C parameter. In particular we analyzed the relationships between average error, bias, net-variance, unbiased and biased variance, the degree of the kernel and the C parameter.

2. This library may be downloaded from the web at `http://www.disi.unige.it/person/ValentiniG/NEURObjects`.
3. The source code is available at `ftp://ftp.disi.unige.it/person/ValentiniG/BV`. Moreover C++ classes for bias-variance analysis have been developed as part of the *NEURObjects* library.

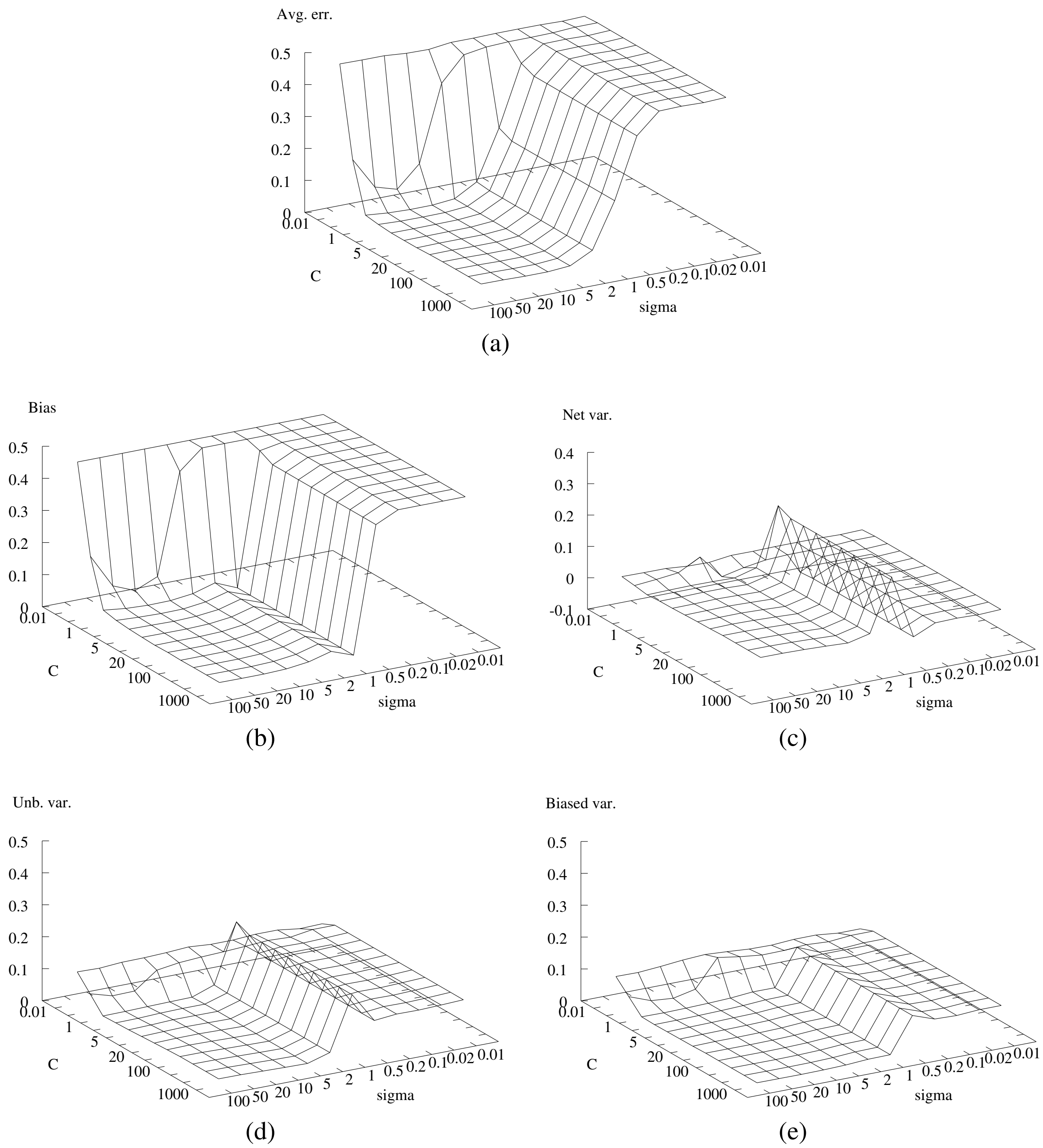

Figure 4: Grey-Landsat data set. Error (a) and its decomposition in bias (b), net variance (c), unbiased variance (d), and biased variance (e) in SVM RBF, varying both C and σ.

We trained polynomial-SVM with several combinations of the degree parameter of the kernel and C values, using all the polynomial degrees between 1 and 10, evaluating in such a way about 120 different polynomial-SVM models for each data set. Following the heuristic of Jakkola, the dot product of polynomial kernel was divided by the dimension of the input data, to "normalize" the dot product before to raise to the degree of the polynomial.

3. **Dot product kernels**. We evaluated bias-variance decomposition varying the C parameter. We analyzed the relationships between average error, bias, net-variance, unbiased and biased variance and the parameter C (the regularization factor) of the kernel. We trained dot-product-SVM considering different values for the C parameter, evaluating in such a way 12 different dot-product-SVM models for each data set.

Each SVM model required the training of 200 different SVMs, one for each synthesized or bootstrapped data set, for a total of $(204+120+12) \times 200 = 67200$ trained SVM for each data set (134400 for the data set *P2*, as for this data set we used 400 data sets for each model). The experiments required the training of more than half million of SVMs, considering all the data sets and of course the testing of all the SVM previously trained in order to evaluate the bias-variance decomposition of the error of the different SVM models. For each SVM model we computed the main prediction, bias, net-variance, biased and unbiased variance and the error on each example of the test set, and the corresponding average quantities on the overall test set.

5. Results

In this section we present the results of the experiments. We analyzed bias-variance decomposition with respect to the kernel parameters considering separately Gaussian, polynomial and dot product SVMs, comparing also the results among different kernels. Here we present the main results. Full results, data and graphics are available by anonymous ftp at `ftp://ftp.disi.unige.it/person/ValentiniG/papers/bv-svm.ps.gz`.

5.1 Gaussian Kernels

Figure 4 depicts the average loss, bias net-variance, unbiased and biased variance varying the values of σ and the regularization parameter C in *RBF-SVM* on the *Grey-Landsat* data set. We note that σ is the most important parameter: although for very low values of C the SVM cannot learn, independently of the values of σ, (Figure 4 a), the error, the bias, and the net-variance depend mostly on the σ parameter. In particular for low values of σ, bias is very high (Figure 4 b) and net-variance is 0, as biased and unbiased variance are about equal (Figure 4d and 4e). Then the bias suddenly drops (Figure 4b), lowering the average loss (Figure 4a), and then stabilizes for higher values of σ. Interestingly enough, in this data set (but also in others, data not shown), we note an increment followed by a decrement of the net-variance, resulting in a sort of "wave shape" of the net variance graph (Figure 4c).

Figure 5 shows the bias-variance decomposition on different data sets, varying σ, and for a fixed value of C, that is a sort of "slice" along the σ axis of the Figure 4. The plots show that average loss, bias, and variance depend significantly on σ for all the considered data sets, confirming the existence of a "high biased region" for low values of σ. In this region, biased and unbiased variance are about equal (net-variance $V_n = V_u - V_b$ is low). Then unbiased variance increases while biased variance decreases (Figure 5 a,b,c and d), and finally both stabilize for relatively high values of σ. Interestingly, the average loss and the bias do not increase for high values of σ, especially if C is high.

Bias and average loss increases with σ only for very small C values. Note that net-variance and bias show opposite trends only for small values of C (Figure 5 c). For larger C values the symmetric trend is limited only to $\sigma \leq 1$ (Figure 5 d), otherwise bias stabilizes and net-variance slowly decreases. Figure 6 shows more in detail the effect of the C parameter on bias-variance decomposition. For $C \geq 1$ there are no variations of the average error, bias and variance for a fixed value of σ. Note that for very low values of σ (Figure 6a and b) there is no learning. In the Letter-Two data set, as in other data sets (figures not shown), only for small C values we have variations in bias and variance values (Figure 6).

5.1.1 DISCRIMINANT FUNCTION COMPUTED BY THE SVM-RBF CLASSIFIER

In order to get insights into the behaviour of the SVM learning algorithm with Gaussian kernels we plotted the real-valued functions computed without considering the discretization step performed through the sign function. The real valued function computed by a Gaussian SVM is

$$f(\mathbf{x}, \alpha, b) = \sum_{i \in SV} y_i \alpha_i \exp(-\|\mathbf{x_i} - \mathbf{x}\|^2/\sigma^2) + b,$$

where the α_i are the Lagrange multipliers found by the solution of the dual optimization problem, the $\mathbf{x_i} \in SV$ are the support vectors, that is the points for which $\alpha_i > 0$.

We plotted the surface computed by the Gaussian SVM with the synthetic data set *P2*. Indeed it is the only surface that can be easily visualized, as the data are bidimensional and the resulting real valued function can be easily represented through a wireframe three-dimensional surface. The SVMs are trained with exactly the same training set composed by 100 examples. The outputs are referred to a test set of 10000 examples, selected in an uniform way through all the data domain. In particular we considered a grid of equi-spaced data at 0.1 interval in a two dimensional 10×10 input space. If $f(\mathbf{x}, \alpha, b) > 0$ then the SVM matches up the example $\mathbf{x}$ with class 1, otherwise with class 2.

With small values of σ we have "spiky" functions: the response is high around the support vectors, and is close to 0 in all the other regions of the input domain (Figure 7). In this case we have overfitting: a large error on the test set (about 46 % with $\sigma = 0.01$ and 42.5 % with $\sigma = 0.02$), and a training error near to 0.

If we enlarge the values of σ we obtain a wider response on the input domain and the error decreases (with $\sigma = 0.1$ the error is about 37 %). With $\sigma = 1$ we have a smooth function that fits quite well the data (Figure 8). In this case the error drops down to about 13 %.

Enlarging too much σ we have a too smooth function (Figure 9 (a)), and the error increases to about 37 %: in this case the high bias is due to an excessive smoothing of the function. Increasing the values of the regularization parameter C (in order to better fit the data), we can diminish the error to about 15 %: the shape of the function now is less smooth (Figure 9 (b)).

As noted in Scholkopf and Smola (2002), using very large values of sigma, we have a very smooth discriminant function (in practice a plane), and increasing it even further does not change anything. Indeed, enlarging σ to 500 we obtain a plane (Fig 9 (c)), and a very biased function (error about 45 %), and even if we increment C, we can obtain better results, but always with a large error (about 35 %, Fig 9 (d)).

5.1.2 BEHAVIOR OF SVMS WITH LARGE σ VALUES

Fig 4 and 5 show that σ parameter plays a sort of smoothing effect, as the value of σ increases. In particular with large values of σ we did not observe any increment of bias nor decrement of variance. In order to get insights into this counter-intuitive behaviour we tried to answer these two questions:

1. Does the bias increase while variance decrease with large values of σ, and what is the combined effect of bias-variance on the error?

2. In this situation (large values for σ), what is the effect of the C parameter?

In Figure 5 we do not observe an increment of bias with large values of σ, but we limited our experiments to values of $\sigma \leq 100$. Here we investigate the effect for larger values of σ (from 100 to 1000).

In most cases, also increasing the values of σ right to 1000 we do not observe an increment of the bias and a substantial decrement of the variance. Only for low values of C, that is $C < 1$, the bias and the error increase with large values of σ (Figure 10). With the P2 data set the situation is different: in this case we observe an increment of the bias and the error with large values of σ, even if with large values of C the increment rate is lower (Figure 11 a and b).

Also with the musk data set we note an increment of the error with very large values of σ, but surprisingly this is due to an increment of the unbiased variance, while the bias is quite stable, at least for values of $C > 1$, (Figure 11 c and d).

Larger values of C counter-balance the bias introduced by large values of σ. But with some distributions of the data too large values of σ produce too smooth functions, and also incrementing C it is very difficult to fit the data. Indeed, the discriminant function computed by the RBF-SVM with the P2 data set (that is the function computed without considering the sign function) is too smooth for large values of σ: for $\sigma = 20$, the error is about 37%, due almost entirely to the large bias, (Figure 9 a), and for $\sigma = 500$ the error is about 45 % and also incrementing the C value to 1000, we obtain a surface that fits the data better, but with an error that remains large (about 35%). Indeed with very large values of σ the Gaussian kernel becomes nearly linear (Scholkopf and Smola, 2002) and if the data set is very far from being linearly separable, as with the *P2* data set (Figure 3), the error increases, especially in the bias component (Figure 11 (a) and (b)). Summarizing with large σ values bias can increment, while net-variance tends to stabilize, but this effect can be counter-balanced by larger C values.

5.1.3 RELATIONSHIPS BETWEEN GENERALIZATION ERROR, TRAINING ERROR, NUMBER OF SUPPORT VECTORS AND CAPACITY

Looking at Figure 4 and 5, we see that SVMs do not learn for small values of σ. Moreover the low error region is relatively large with respect to σ and C.

In this section we evaluate the relationships between the estimated generalization error, the bias, the training error, the number of support vectors and the estimated Vapnik Chervonenkis dimension, in order to answer the following questions:

1. Why SVMs do not learn for small values of σ?

2. Why we have a so large bias for small values of σ?

3. Can we use the variation of the number of support vectors to predict the "low error" region?

4. Is there any relationship between the bias, variance and VC dimension, and can we use this last one to individuate the "low error" region?

The generalization error, bias, training error, number of support vectors and the Vapnik Chervonenkis dimension are estimated averaging with respect to 400 SVMs (*P2* data set) or 200 SVMs (other data sets) trained with different bootstrapped training sets composed by 100 examples each one. The test error and the bias are estimated with respect to an independent and sufficiently large data set.

The *VC dimension* is estimated using the Vapnik's bound based on the radius R of the sphere that contains all the data (in the feature space), approximated through the sphere centered in the origin, and on the norm of the weights in the feature space (Vapnik, 1998). In this way the VC dimension is overestimated but it is easy to compute and we are interested mainly in the comparison of the VC dimension of different SVM models:

$$VC \leq R^2 \cdot \|\mathbf{w}\|^2 + 1,$$

where

$$\|\mathbf{w}\|^2 = \sum_{i \in SV} \sum_{j \in SV} \alpha_i \alpha_j K(\mathbf{x_i}, \mathbf{x_j}) y_i y_j$$

and

$$R^2 = \max_i K(\mathbf{x_i}, \mathbf{x_i}).$$

The number of support vectors is expressed as the halved ratio of the number ($\%\ SV$) of support vectors with respect to the total number of training data:

$$\%SV = \frac{\#SV}{\#trainingdata \cdot 2}.$$

In the graphs shown in Figure 12 and Figure 13, on the left y axis is represented the error, training error and bias, and the halved ratio of support vectors. On the right y axis is reported the estimated Vapnik Chervonenkis dimension.

For very small values of σ the training error is very small (about 0), while the number of support vectors is very high, and high is also the error and the bias (Figure12 and 13). These facts support the hypothesis of overfitting problems with small values of σ. Indeed the real-valued function computed by the SVM (that is the function computed without considering the sign function, see Section 5.1.1) is very spiky with small values of σ (Figure 7). The response of the SVM is high only in small areas around the support vectors, while in all the other areas "not covered" by the support vectors the response is very low (about 0), that is the SVM is not able to get a decision, with a consequently very high bias. In the same region (small values for σ) the net variance is usually very small, for either one of these reasons: 1) biased and unbiased variance are almost equal because the SVM performs a sort of random guessing for the most part of the unknown data; 2) both biased and unbiased variance are about 0, showing that all the SVMs tend to answer in the same way independently of a particular instance of the training set (Figure 5 a, b and f). Enlarging σ we obtain a wider response on the input domain: the real-valued function computed by the SVM becomes smoother (Figure 8), as the "bumps" around the support vectors become wider and the

SVM can decide also on unknown examples. At the same time the number of support vectors decreases (Figure 12 and 13).

Considering the variation of the ratio of the support vectors with σ, in all data sets the trend of the rate of support vectors follows the error, with a sigmoid shape that sometimes becomes an U shape for small values of C (Figure12 and 13). This is not surprising because it is known that the support vector ratio offers an approximation of the generalization error of the SVMs (Vapnik, 1998). Moreover, on all the data sets the %SV decreases in the "stabilized" region, while in the transition region remains high. As a consequence the decrement in the number of support vectors shows that we are entering the "low error" region, and in principle we can use this information to detect this region.

In our experiments, an inspection of the support vectors relative to the *Grey-Landsat* and *Waveform* data sets found that most of the support vectors are shared in polynomial and Gaussian kernels with respectively the best degree and σ parameters. Even if these results confirmed the ones found by other authors (see e.g. Vapnik (1998)), it is worth noting that we did not perform a systematic study on this topic: we considered only two data sets and we compared only few hundreds of different SVMs.

In order to analyze the role of the VC dimension on the generalization ability of learning machines, we know from statistical learning theory that the form of the bounds of the generalization error E of SVMs is

$$E(f(\sigma,C)_n^k)) \leq E_{emp}(f(\sigma,C)_n^k)) + \Phi(\frac{h_k}{n}), \tag{9}$$

where $f(\sigma,C)_n^k$ represents the set of functions computed by an RBF-SVM trained with n examples and with parameters (σ_k, C_k) taken from a set of parameters $S = \{(\sigma_i, C_i), i \in \mathbb{N}\}$, E_{emp} represents the empirical error and Φ the confidence interval that depends on the cardinality n of the data set and on the VC dimension h_k of the set of functions identified by the actual selection of the parameters (σ_k, C_k). In order to obtain good generalization capabilities we need to minimize both the empirical risk and the confidence interval. According to Vapnik's bounds (Equation 9), in Figure 12 and 13 the lowest generalization error is obtained for a small empirical risk and a small estimated VC dimension.

But sometimes with relatively small values of VC we may have a very large error, as the training error and the number of support vectors increase with very large values of σ (Figure 12 a and 13 a). Moreover with a very large estimate of the VC dimension and low empirical error (Figure 12 and 13) we may have a relatively low generalization error. In conclusion it seems very difficult to use in practice these estimate of the VC dimension to infer the generalization abilities of the SVM. In particular it seems unreliable to use the VC dimension to infer the "low error" region of the RBF-SVM.

5.2 Polynomial and Dot Product Kernels

In this section we analyze the characteristics of bias-variance decomposition of the error in polynomial SVMs, varying the degree of the kernel and the regularization parameter C.

Error shows a U shape with respect to the degree. This shape depends on unbiased variance (Figure 14 a and b), or both by bias and unbiased variance (Figure 14 c and d). The U shape of the error with respect to the degree tends to be more flat for increasing values of C, and net-variance and bias show often opposite trends (Figure 15).

Average error and bias tends to be higher for low C and degree values, but, incrementing the degree, the error is less sensitive to C values (Figure 16).

Bias is flat (Figure 17 a) or decreasing with respect to the degree (Figure 15 b), or it can be constant or decreasing, depending on C (Figure 17 b). Unbiased variance shows an U shape (Figure 14 a and b) or it increases (Figure 14 c) with respect to the degree, and the net-variance follows the shape of the unbiased variance. Note that in the P2 data set (Figure 15) bias and net-variance follow the classical opposite trends with respect to the degree. This is not the case with other data sets (see, e.g. Figure 14).

For large values of C bias and net-variance tend to converge, as a result of the bias reduction and net-variance increment (Figure 18), or because both stabilize at similar values (Figure 16).

In dot product SVMs bias and net-variance show opposite trends: bias decreases, while net-variance and unbiased variance tend to increase with C (Figure 19). On the data set P2 this trend is not observed, as in this task the bias is very high and the SVM does not perform better than random guessing (Figure 19a). The minimum of the average loss for relatively low values of C is the result of the decrement of the bias and the increment of the net-variance: it is achieved usually before the crossover of bias and net-variance curves and before the stabilization of the bias and the net-variance for large values of C. The biased variance remains small independently of C.

5.3 Comparing Kernels

In this section we compare the bias-variance decomposition of the error with respect to the C parameter, considering Gaussian, polynomial and dot product kernels. For each kernel and for each data set the best results are selected. Table 5.3 shows the best results achieved by the SVM, considering each kernel and each data set used in the experiments. Interestingly enough in 3 data sets (Waveform, Letter-Two with added noise and Spam) there are not significant differences in the error between the kernels, but there are differences in bias, net-variance, unbiased or biased variance. In the other data sets Gaussian kernels outperform polynomial and dot product kernels, lowering bias or net-variance or both. Considering bias and net-variance, in some cases they are lower for polynomial or dot product kernel, showing that different kernels learn in different ways with different data.

Considering the data set *P2* (Figure 20 a, c, e), RBF-SVMs achieve the best results, as bias is lower. Unbiased variance is comparable between polynomial and Gaussian kernel, while net-variance is lower, as biased variance is higher for polynomial-SVM. In this task the bias of dot product SVM is very high. Also in the data set *Musk* (Figure 20 b, d, f) RBF-SVM obtains the best results, but in this case unbiased variance is responsible for this fact, while bias is similar. With the other data sets the bias is similar between RBF-SVM and polynomial-SVM, but for dot product SVM often the bias is higher (Figure 21 b, d, f). Interestingly enough RBF-SVM seems to be more sensible to the C value with respect to both polynomial and dot product SVM: for $C < 0.1$ in some data sets the bias is much higher (Figure 21 a, c, e). With respect to C bias and unbiased variance show sometimes opposite trends, independently of the kernel: bias decreases, while unbiased variance increases, but this does not occur in some data sets. We outline also that the shape of the error, bias and variance curves is similar between different kernels in all the considered data sets: that is, well tuned SVMs having different kernels tend to show similar trends of the bias and variance curves with respect to the C parameter.

	Parameters	**Avg. Error**	**Bias**	**Var. unb.**	**Var. bias.**	**Net Var.**
Data set *P2*						
RBF-SVM	$C = 20, \quad \sigma = 2$	0.1516	0.0500	0.1221	0.0205	0.1016
Poly-SVM	$C = 10, \quad degree = 5$	0.2108	0.1309	0.1261	0.0461	0.0799
D-prod SVM	$C = 500$	0.4711	0.4504	0.1317	0.1109	0.0207
Data set *Waveform*						
RBF-SVM	$C = 1, \quad \sigma = 50$	0.0706	0.0508	0.0356	0.0157	0.0198
Poly-SVM	$C = 1, \quad degree = 1$	0.0760	0.0509	0.0417	0.0165	0.0251
D-prod SVM	$C = 0.1$	0.0746	0.0512	0.0397	0.0163	0.0234
Data set *Grey-Landsat*						
RBF-SVM	$C = 2, \quad \sigma = 20$	0.0382	0.0315	0.0137	0.0069	0.0068
Poly-SVM	$C = 0.1, degree = 5$	0.0402	0.0355	0.0116	0.0069	0.0047
D-prod SVM	$C = 0.1$	0.0450	0.0415	0.0113	0.0078	0.0035
Data set *Letter-Two*						
RBF-SVM	$C = 5, \quad \sigma = 20$	0.0743	0.0359	0.0483	0.0098	0.0384
Poly-SVM	$C = 2, \quad degree = 2$	0.0745	0.0391	0.0465	0.0111	0.0353
D-prod SVM	$C = 0.1$	0.0908	0.0767	0.0347	0.0205	0.0142
Data set *Letter-Two with added noise*						
RBF-SVM	$C = 10, \quad \sigma = 100$	0.3362	0.2799	0.0988	0.0425	0.0563
Poly-SVM	$C = 1, \quad degree = 2$	0.3432	0.2799	0.1094	0.0461	0.0633
D-prod SVM	$C = 0.1$	0.3410	0.3109	0.0828	0.0527	0.0301
Data set *Spam*						
RBF-SVM	$C = 5, \quad \sigma = 100$	0.1263	0.0987	0.0488	0.0213	0.0275
Poly-SVM	$C = 2, \quad degree = 2$	0.1292	0.0969	0.0510	0.0188	0.0323
D-prod SVM	$C = 0.1$	0.1306	0.0965	0.0547	0.0205	0.0341
Data set *Musk*						
RBF-SVM	$C = 2, \quad \sigma = 100$	0.0884	0.0800	0.0217	0.0133	0.0084
Poly-SVM	$C = 2, \quad degree = 2$	0.1163	0.0785	0.0553	0.0175	0.0378
D-prod SVM	$C = 0.01$	0.1229	0.1118	0.0264	0.0154	0.0110

Table 2: Compared best results with different kernels and data sets. RBF-SVM stands for SVM with Gaussian kernel; Poly-SVM for SVM with polynomial kernel and D-prod SVM for SVM with dot product kernel. Var unb. and Var. bias. stand for unbiased and biased variance.

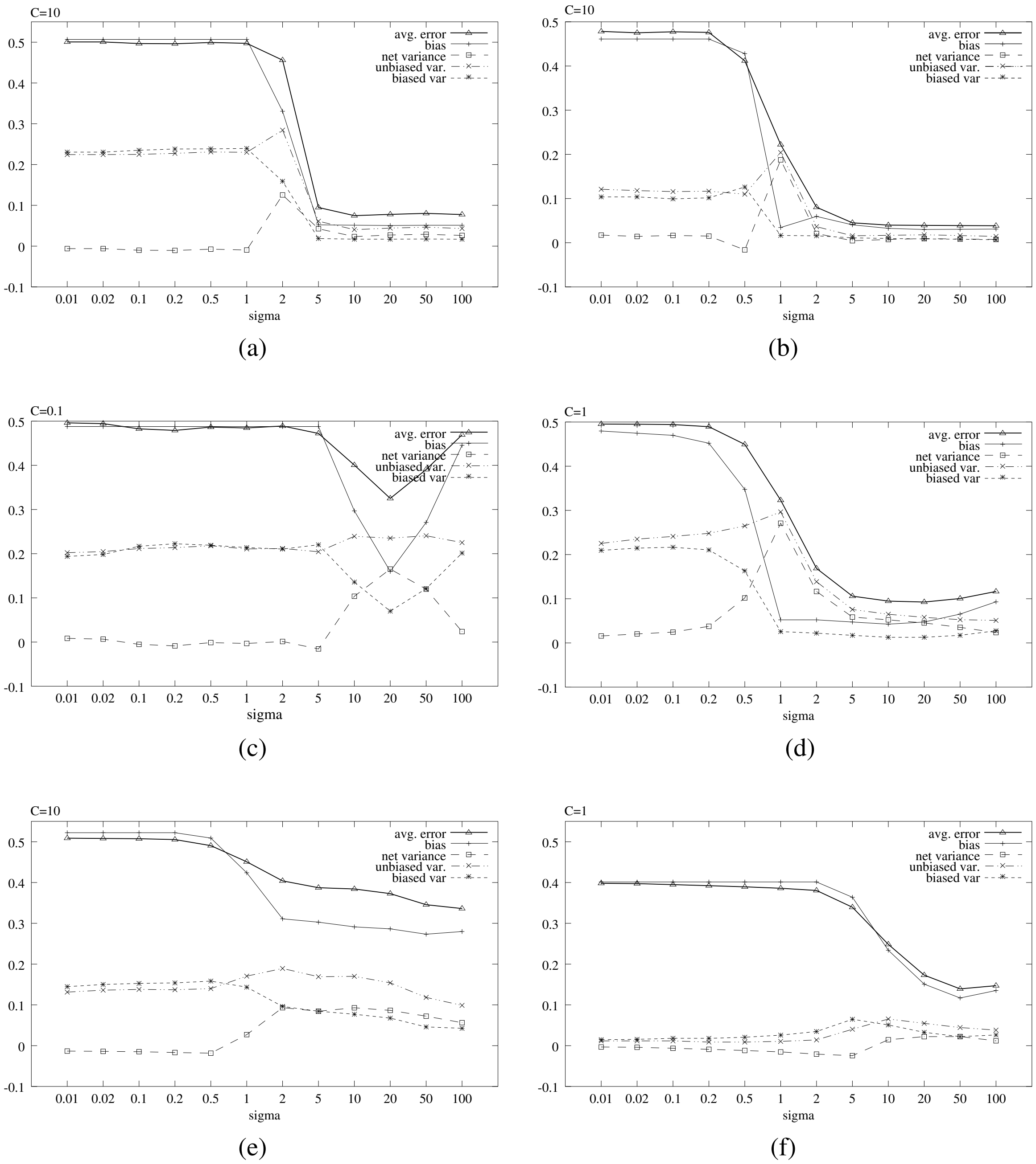

Figure 5: Bias-variance decomposition of error in bias, net variance, unbiased and biased variance in SVM RBF, varying σ and for fixed C values: (a) Waveform, (b) Grey-Landsat, (c) Letter-Two with $C = 0.1$, (c) Letter-Two with $C = 1$, (e) Letter-Two with added noise and (f) Spam.

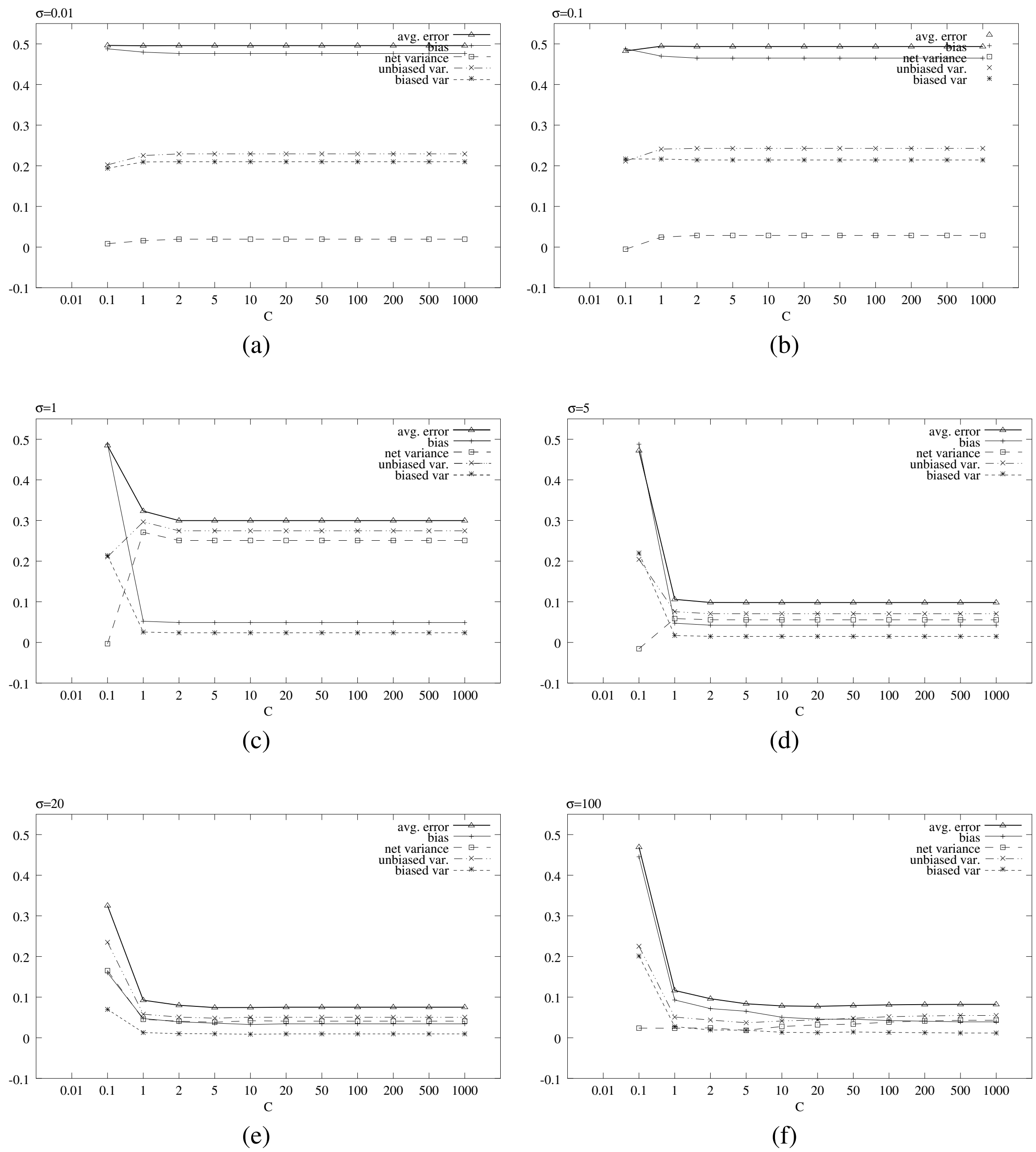

Figure 6: Letter-Two data set. Bias-variance decomposition of the error in bias, net variance, unbiased and biased variance in SVM RBF, while varying C and for some fixed values of σ: (a) $\sigma = 0.01$, (b) $\sigma = 0.1$, (c) $\sigma = 1$, (d) $\sigma = 5$, (e) $\sigma = 20$, (f) $\sigma = 100$.

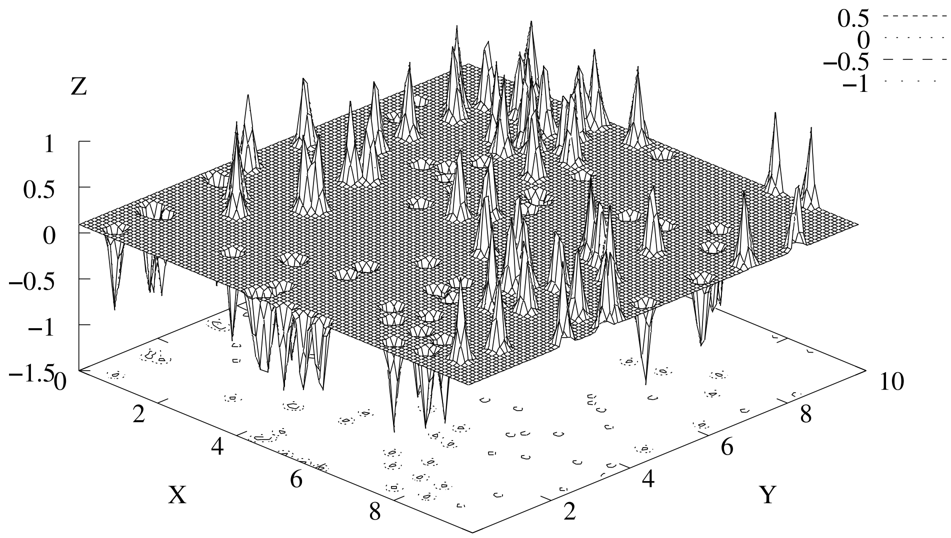

Figure 7: The real valued function computed by the SVM on the P2 data set with $\sigma = 0.01$, $C = 1$.

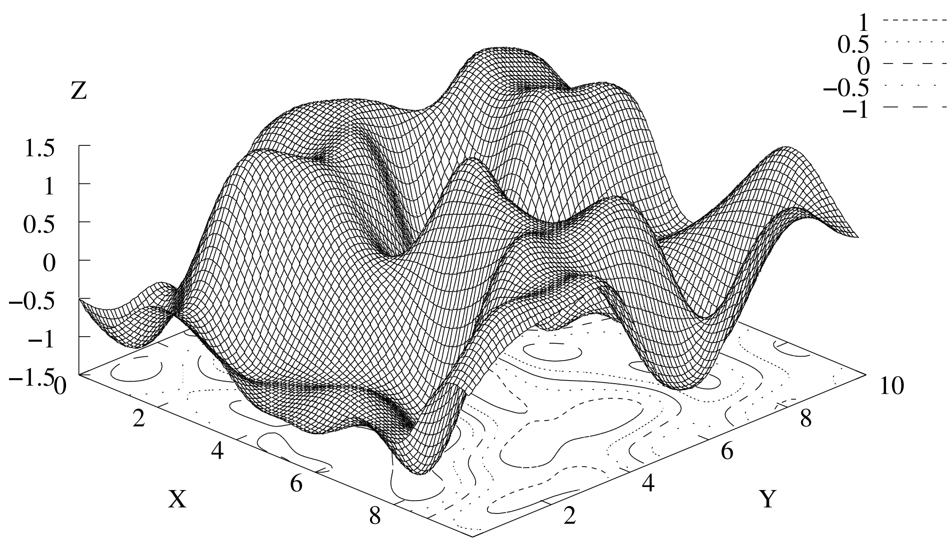

Figure 8: The real valued function computed by the SVM on the P2 data set, with $\sigma = 1$, $C = 1$.

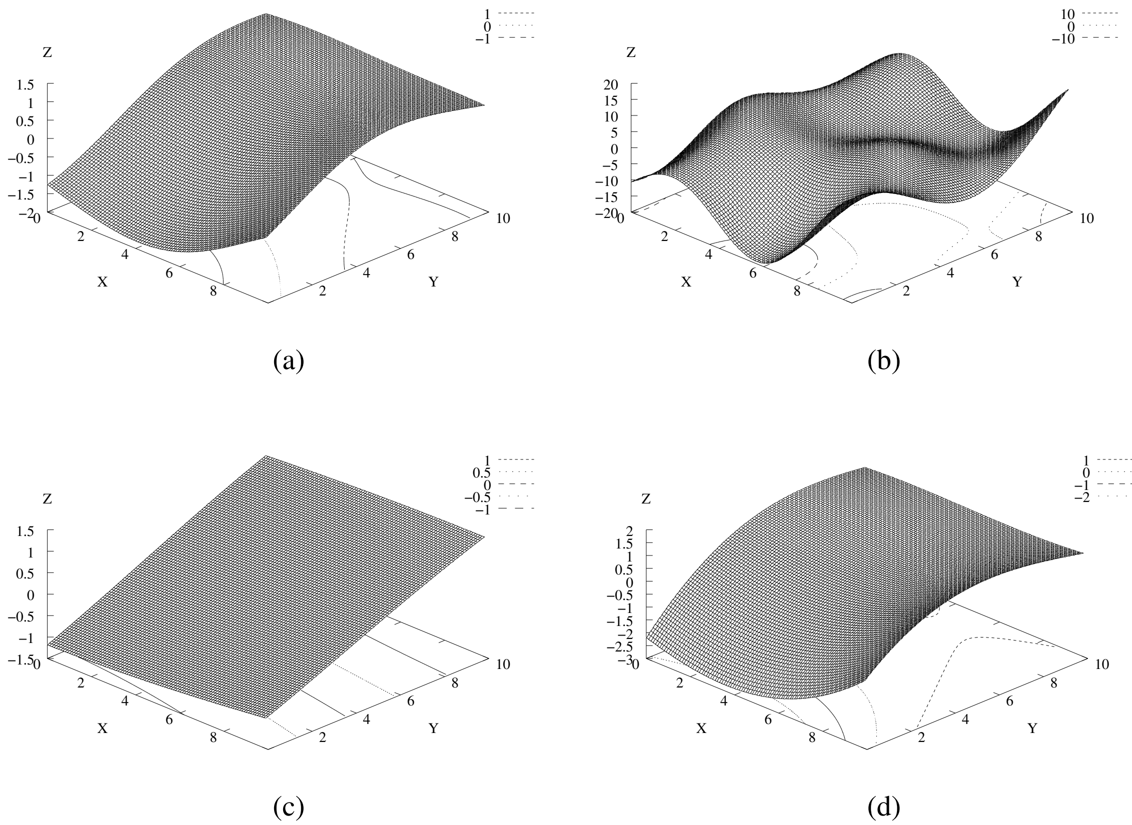

Figure 9: The real valued function computed by the SVM on the P2 data set. (a) $\sigma = 20$ $C = 1$, (b) $\sigma = 20$ $C = 1000$, (c) $\sigma = 500$ $C = 1$, (d) $\sigma = 500$ $C = 1000$.

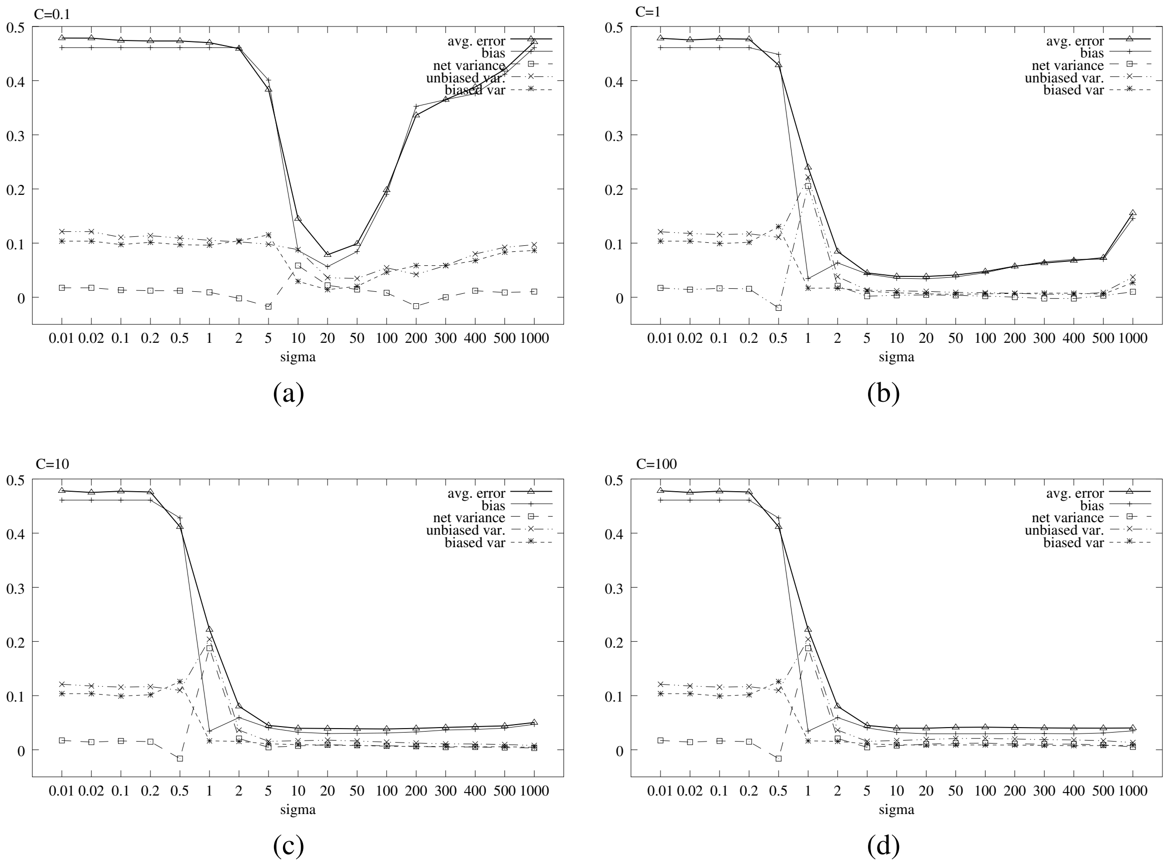

Figure 10: Grey-Landsat data set. Bias-variance decomposition of error in bias, net variance, unbiased and biased variance in SVM RBF, while varying σ and for some fixed values of C: (a) $C = 0.1$, (b) $C = 1$, (c) $C = 10$, (d) $C = 100$.

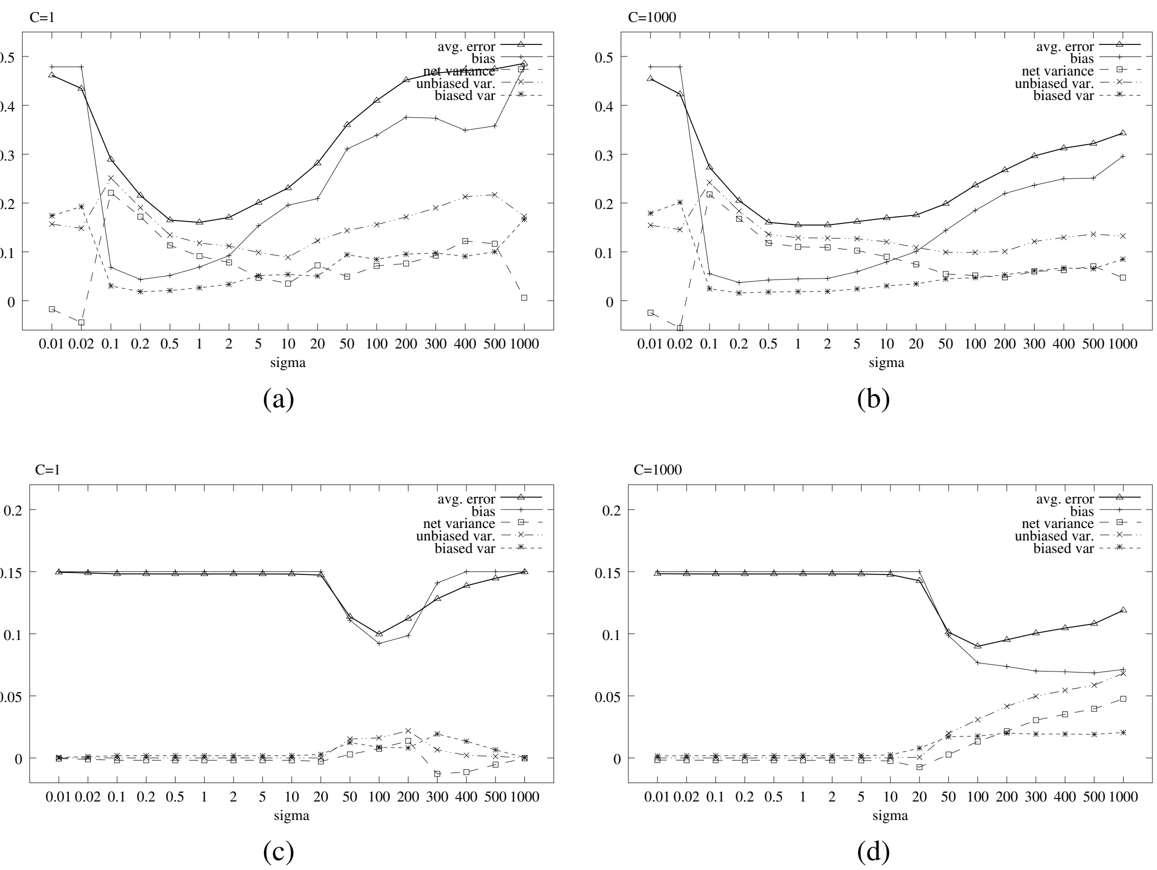

Figure 11: Bias-variance decomposition of the error in bias, net variance, unbiased and biased variance in SVM RBF, while varying σ and for some fixed values of C: (a) P2, with $C = 1$, (b) P2, with $C = 1000$, (c) Musk, with $C = 1$, (d) Musk, with $C = 1000$.

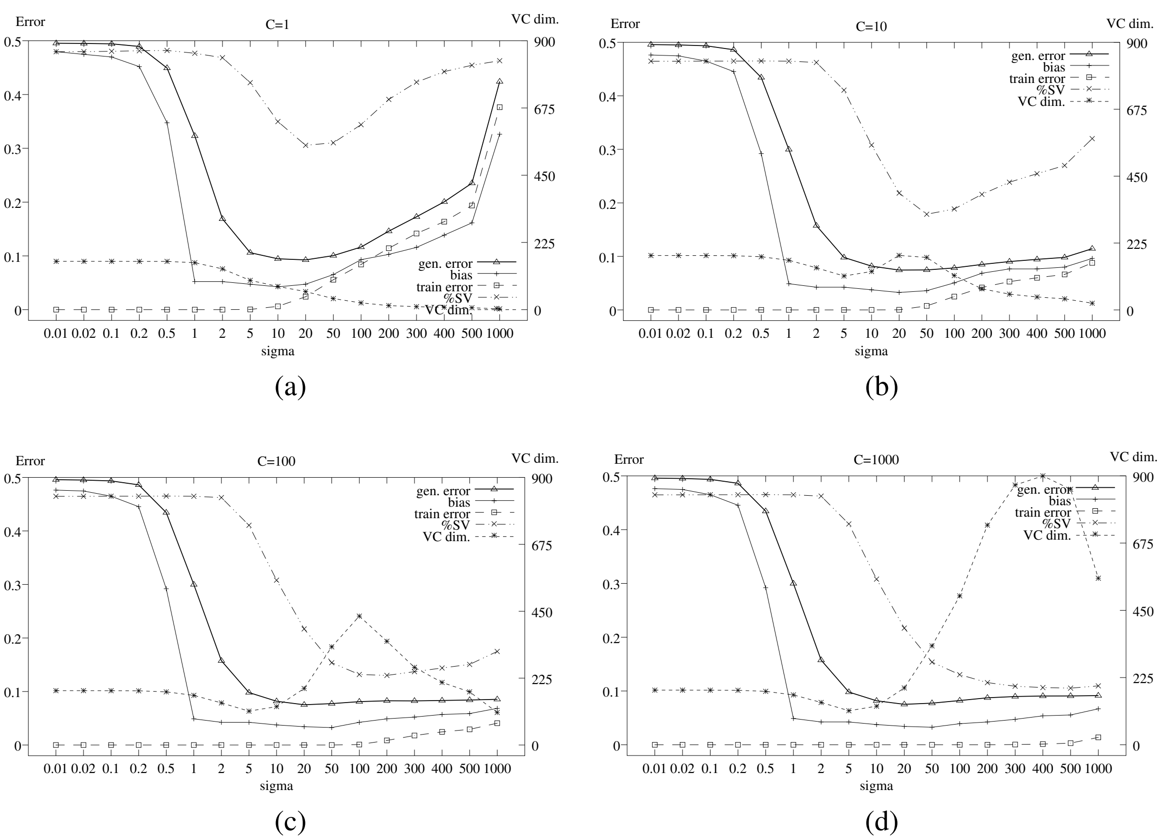

Figure 12: Letter-Two data set. Error, bias, training error, support vector rate, and estimated VC dimension in SVM RBF, while varying the σ parameter and for some fixed values of C: (a) $C = 1$, (b) $C = 10$, (c) $C = 100$, and $C = 1000$.

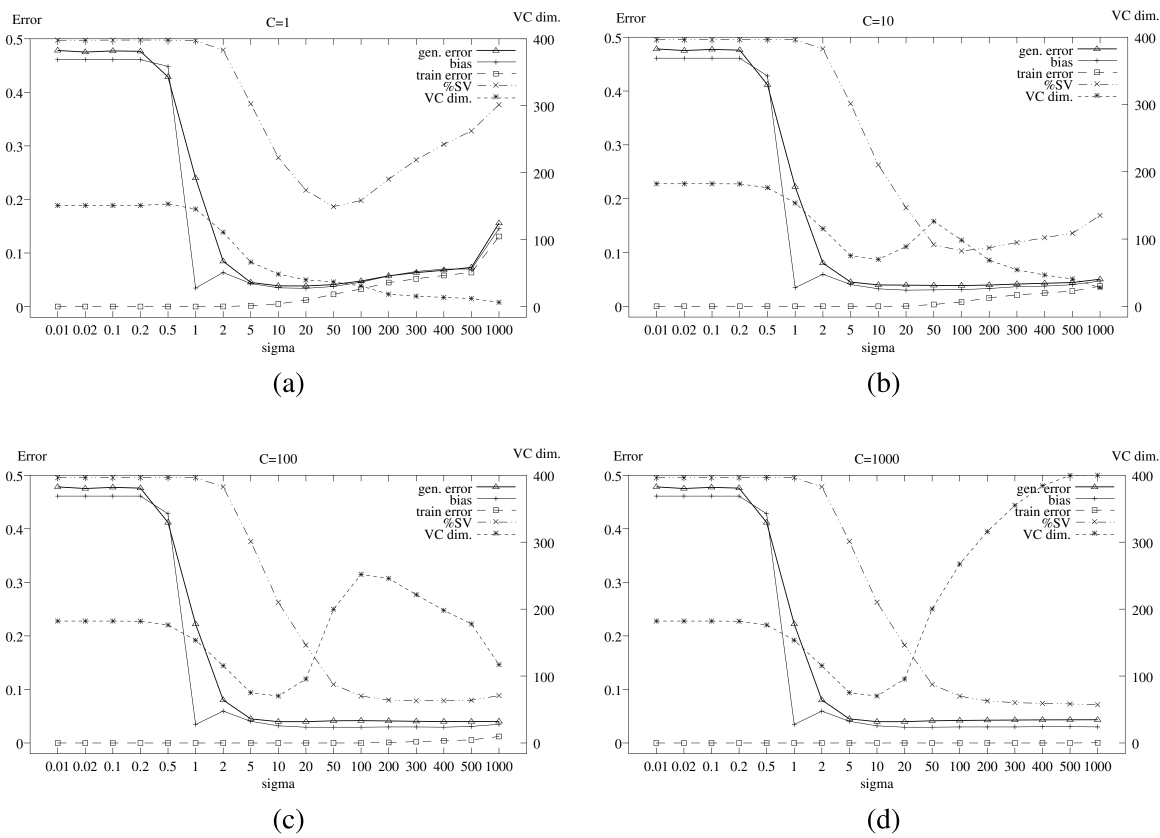

Figure 13: Grey-Landsat data set. Error, bias, training error, support vector rate, and estimated VC dimension in SVM RBF, while varying the σ parameter and for some fixed values of C: (a) $C = 1$, (b) $C = 10$, (c) $C = 100$, and $C = 1000$.

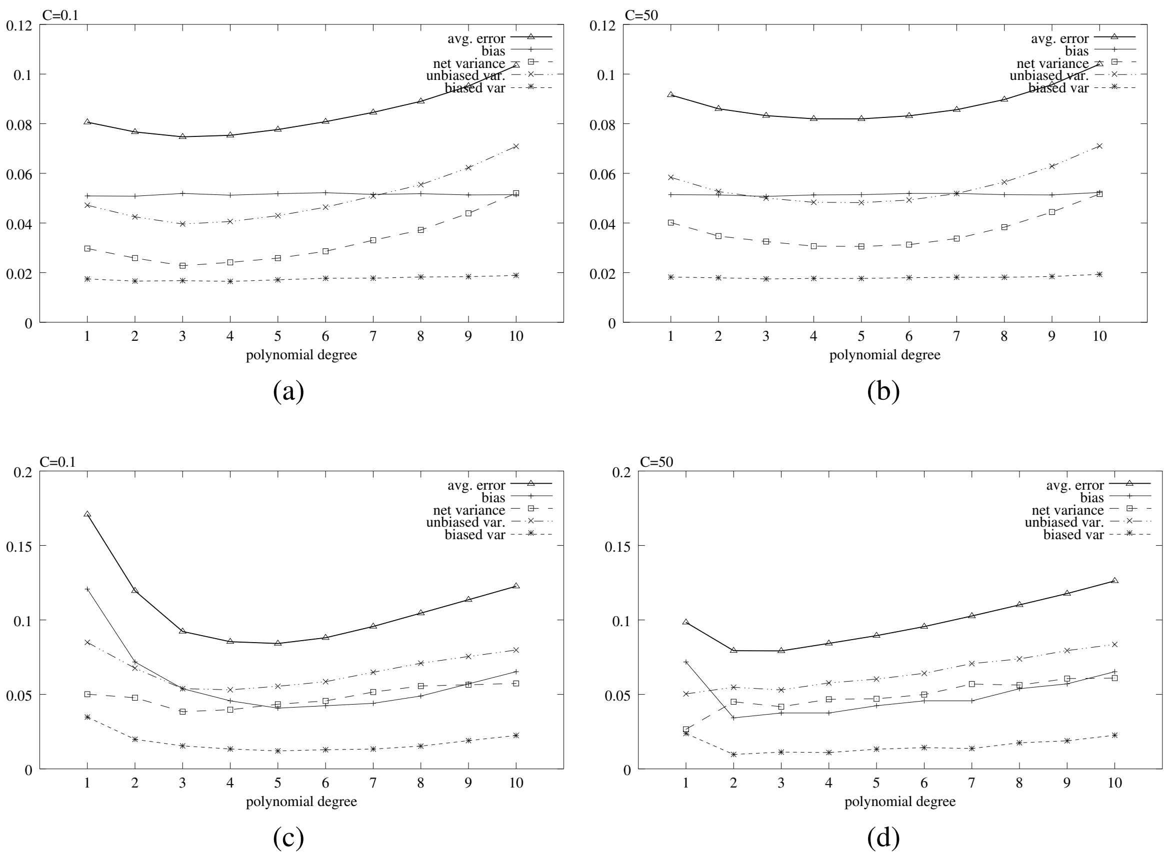

Figure 14: Bias-variance decomposition of the error in bias, net variance, unbiased and biased variance in polynomial SVM, while varying the degree and for some fixed values of C: (a) Waveform, $C = 0.1$, (b) Waveform, $C = 50$, (c) Letter-Two, $C = 0.1$, (d) Letter-Two, $C = 50$.

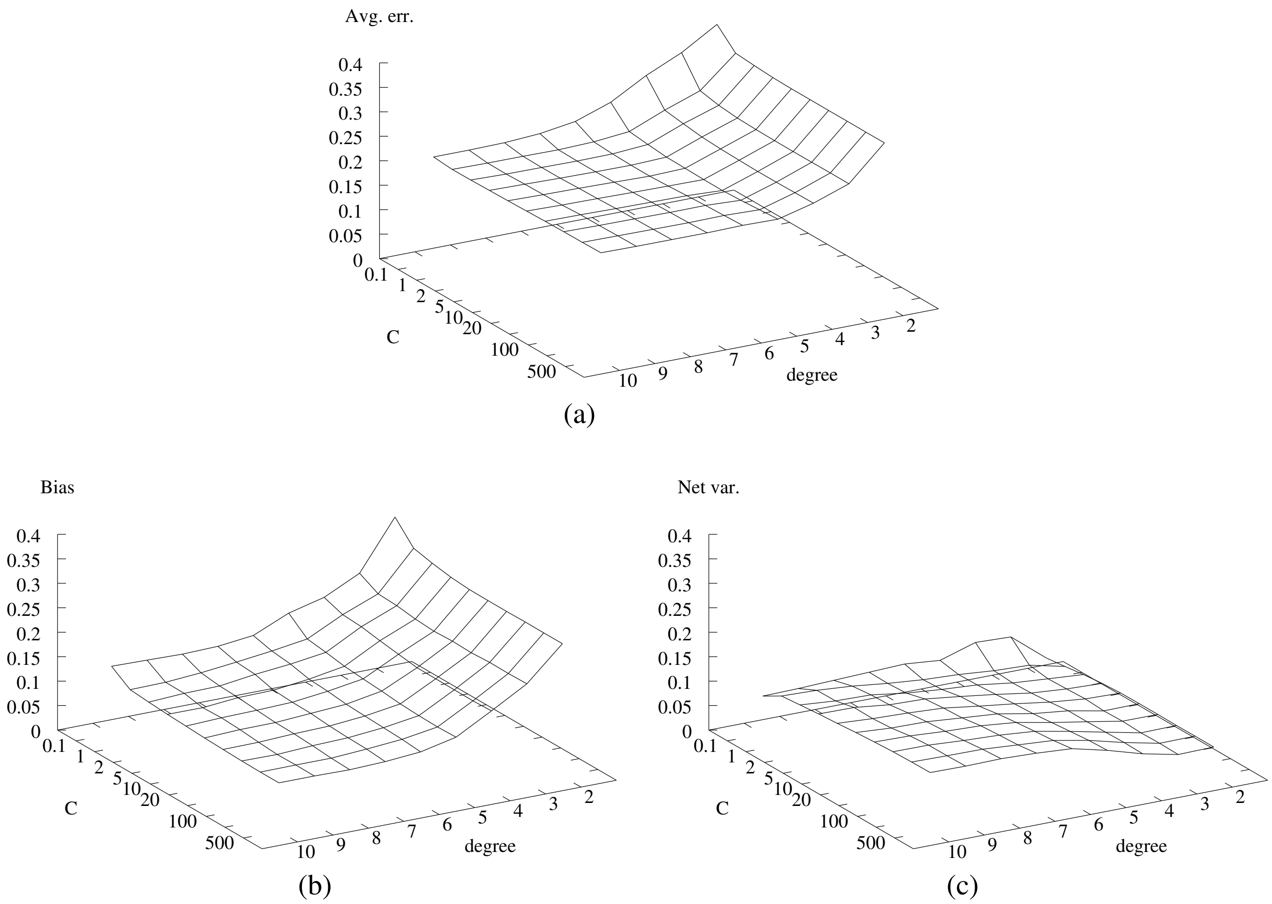

Figure 15: P2 data set. Error (a) and its decomposition in bias (b) and net variance (c), varying both C and the polynomial degree.

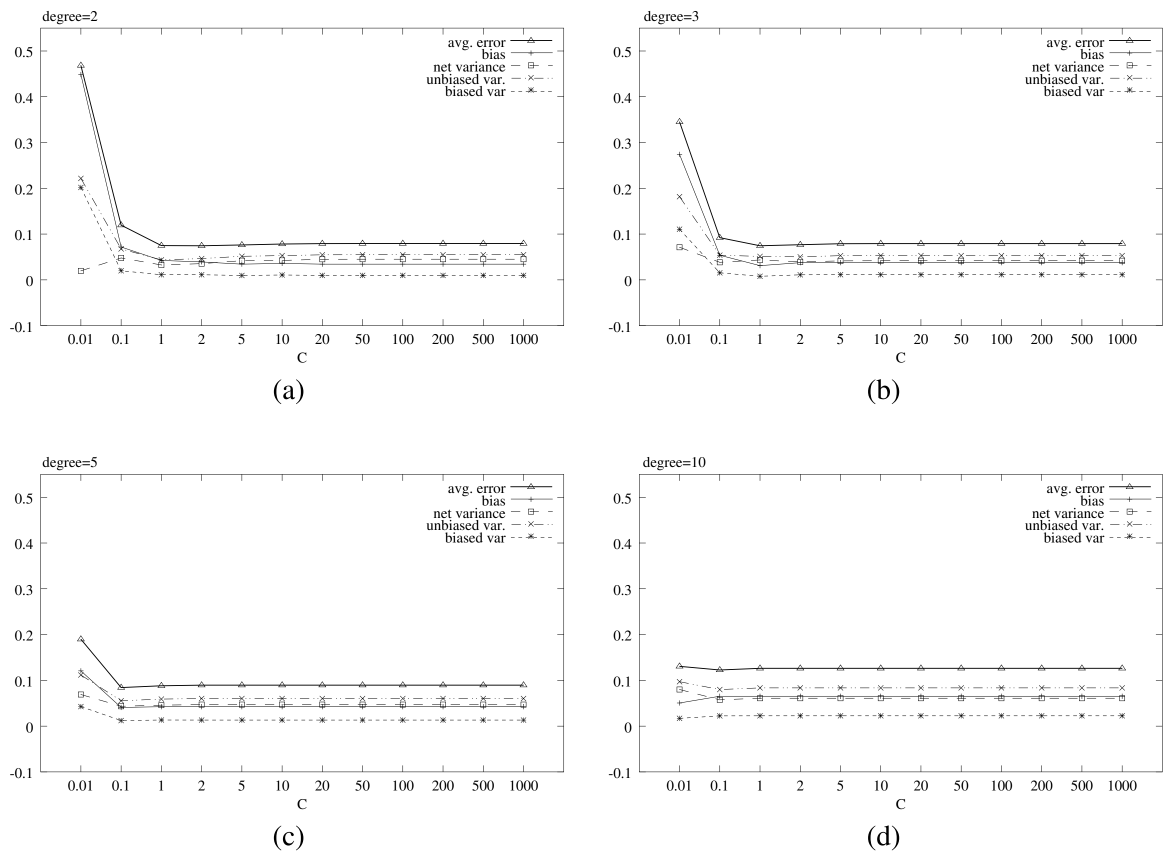

Figure 16: Letter-Two data set. Bias-variance decomposition of error in bias, net variance, unbiased and biased variance in polynomial SVM, while varying C and for some polynomial degrees: (a) $degree = 2$, (b) $degree = 3$, (c) $degree = 5$, (d) $degree = 10$

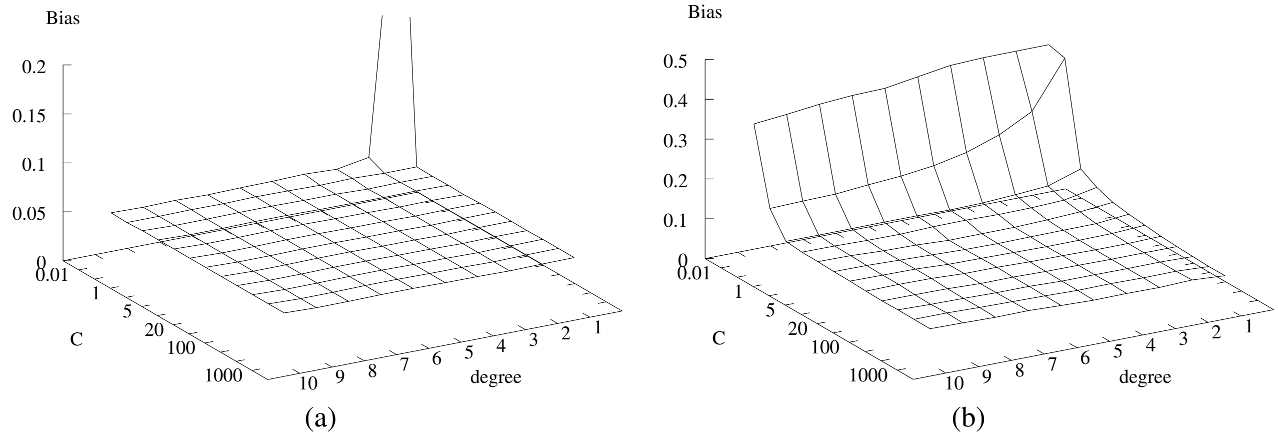

Figure 17: Bias in polynomial SVMs with (a) Waveform and (b) Spam data sets, varying both C and polynomial degree.

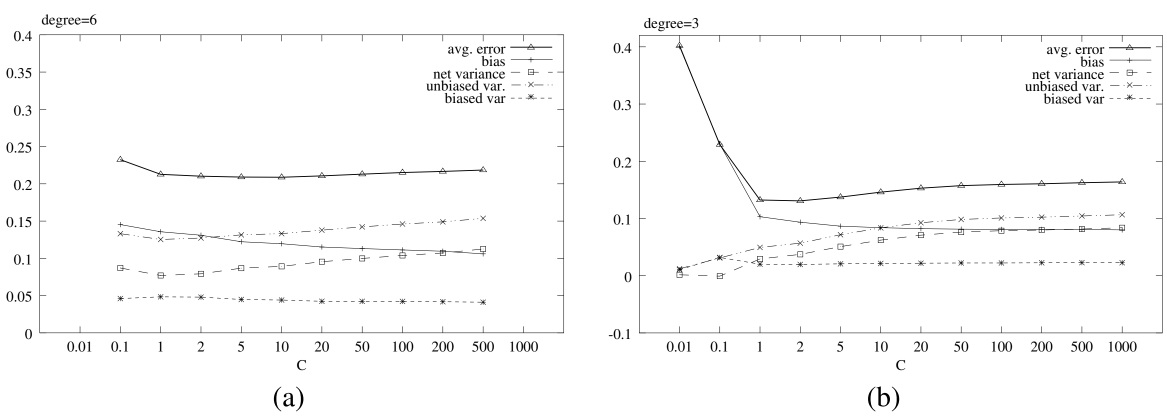

Figure 18: Bias-variance decomposition of the error in bias, net variance, unbiased and biased variance in polynomial SVM, varying C: (a) P2 data set with $degree = 6$, (b) Spam data set with $degree = 3$.

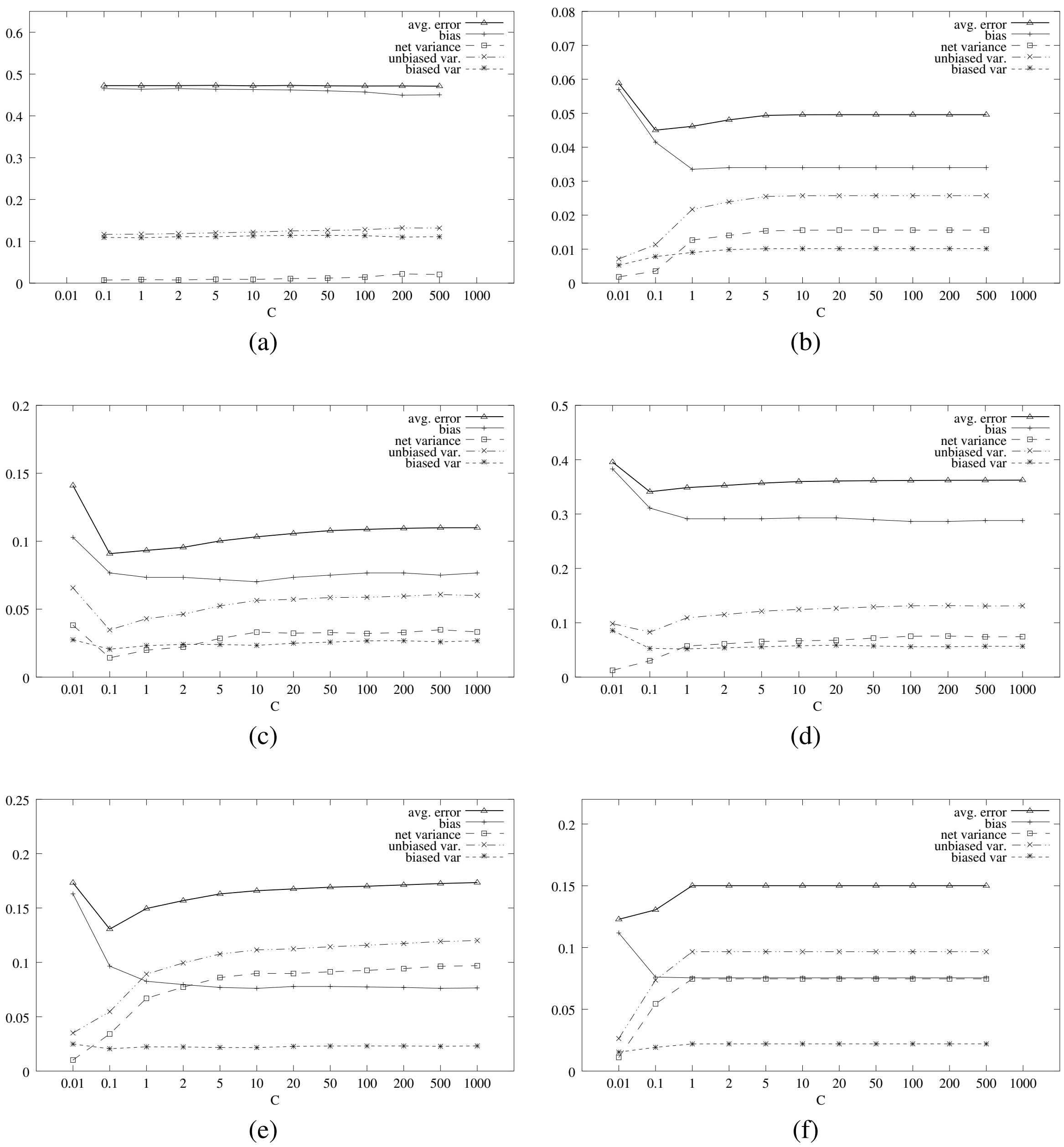

Figure 19: Bias-variance decomposition of error in bias, net variance, unbiased and biased variance in dot product SVM, varying *C*: (a) P2, (b) Grey-Landsat, (c) Letter-Two, (d) Letter-Two with added noise, (e) Spam, (f) Musk.

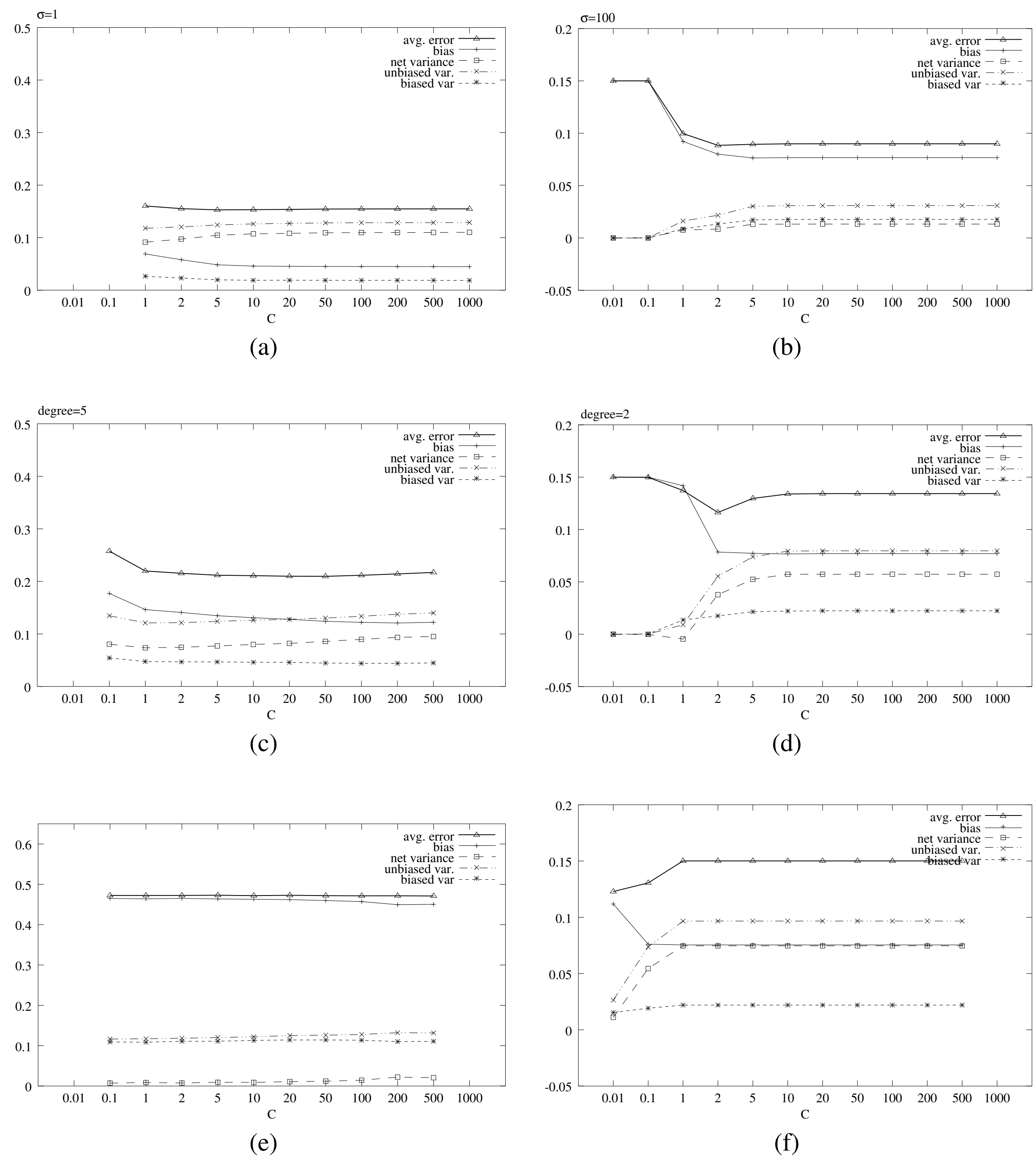

Figure 20: Bias-variance decomposition of the error in bias, net variance, unbiased and biased variance with respect to *C*, considering different kernels. (a) P2, Gaussian; (b) Musk, Gaussian (c) P2, polynomial; (d) Musk, polynomial; (e) P2, dot product; (f) Musk, dot product.

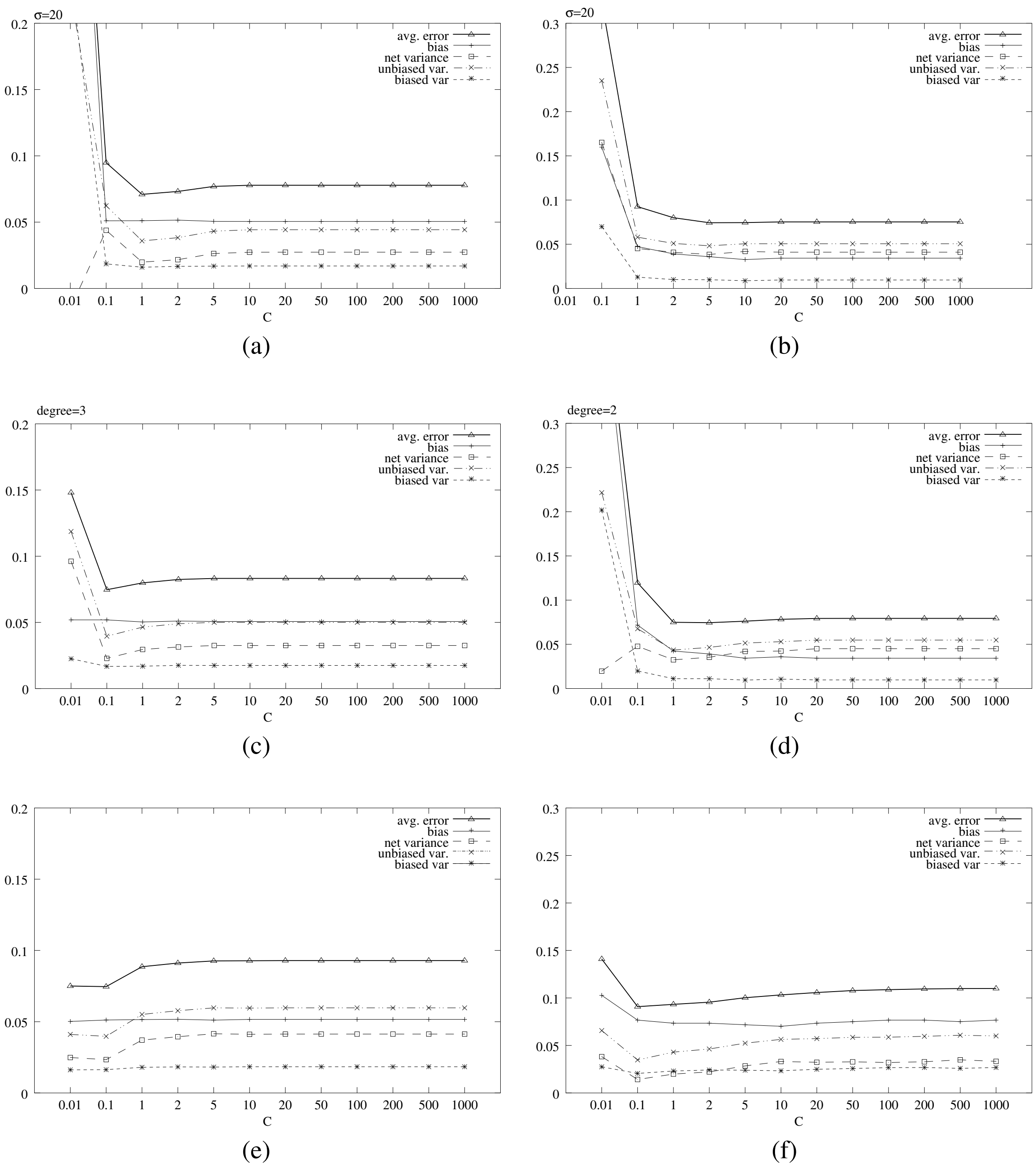

Figure 21: Bias-variance decomposition of the error in bias, net variance, unbiased and biased variance with respect to *C*, considering different kernels. (a) Waveform, Gaussian; (b) Letter-Two, Gaussian (c) Waveform, polynomial; (d) Letter-Two, polynomial; (e) Waveform, dot product; (f) Letter-Two, dot product.

Kernel type	**Avg. Error**	**Bias**	**Var. unb.**	**Var. bias.**	**Net Var.**
RBF	0.0901 ± 0.0087	0.0805 ± 0.0126	0.0237 ± 0.0039	0.0141 ± 0.0025	0.0096 ± 0.0019
Poly	0.1158 ± 0.0069	0.0782 ± 0.0083	0.0585 ± 0.0071	0.0109 ± 0.0018	0.0376 ± 0.0047
D-prod	0.1305 ± 0.0133	0.1179 ± 0.0140	0.0285 ± 0.0084	0.0159 ± 0.0045	0.0126 ± 0.0035

Table 3: Evaluation of the variation of the estimated values of bias variance decomposition with the *Musk* data set. RBF-SVM stands for SVM with Gaussian kernel; Poly-SVM for SVM with polynomial kernel and D-prod SVM for SVM with dot product kernel. Net Var. Var unb. and Var. bias. stand for net, unbiased and biased variance. For each value is represented the mean value over 100 replicated experiments and the corresponding value of the standard deviation.

In our experiments we used relatively small training sets (100 examples), while the number of input variables ranged from 2 (*P2* data set) to 166 (*Musk* data set). Hence, even if for each SVM model (that is for each combination of SVM parameters) we used 200 training sets $D_i, 1 \leq i \leq 200$ in order to train 200 different classifiers f_{D_i}, you could wonder whether the estimated quantities (average error, bias, net-variance, unbiased and biased variance) could be noisy. An extensive evaluation of the sensitivity of the estimated quantities to the sampling procedure would be very expensive. Indeed if we replicate only 10 times our experiments on all the data sets, we should train and test more than 5 millions of different SVMs. Anyway, in order to get insights about this problem, we performed 100 replicates of our experiments limited only to the *Musk* data set (that is the data set with the largest dimensionality in our experiments), for a subset of the parameters near the optimal ones. We found that the standard deviation of the estimated values is not too large. For instance, considering the best model for Gaussian, polynomial and dot product kernels we obtained the values shown in Table 5.3. It seems that the computed quantities are not too noisy, even if we need more experiments to confirm this result.

5.4 Bias-Variance Decomposition with Noisy Data

While the estimation of the noise is quite straightforward with synthetic data, it is a difficult task with "real" data James (2003). For these reasons, and in order to simplify the computation and the overall analysis, in our experiments we did not explicitly consider noise.

Anyway, noise can play a significant role in the bias-variance analysis. Indeed, according to Domingos, with the 0/1 loss the noise is linearly added to the error with a coefficient equal to $2P_D(f_D(\mathbf{x}) = y_*) - 1$ (Equation 6). Hence, if the classifier is accurate, that is if $P_D(f_D(\mathbf{x}) = y_*) \gg 0.5$, then the noise $N(\mathbf{x})$, if present, influences the expected loss. In the opposite situation also, with very bad classifiers, that is when $P_D(f_D(\mathbf{x}) = y_*) \ll 0.5$, the noise influences the overall error in the opposite sense: it reduces the expected loss. If $P_D(f_D(\mathbf{x}) = y_*) \approx 0.5$, that is if the classifier performs a sort of random guessing, then $2P_D(f_D(\mathbf{x}) = y_*) - 1 \approx 0$ and the noise has no substantial impact on the error.

Hence if we know that the noise is small we can disregard it, but what about the effect of noise when it is present but not explicitly considered in the bias-variance decomposition of the error? The analysis of the results in the data set Letter-Two without and with 20 % added noise shows that the

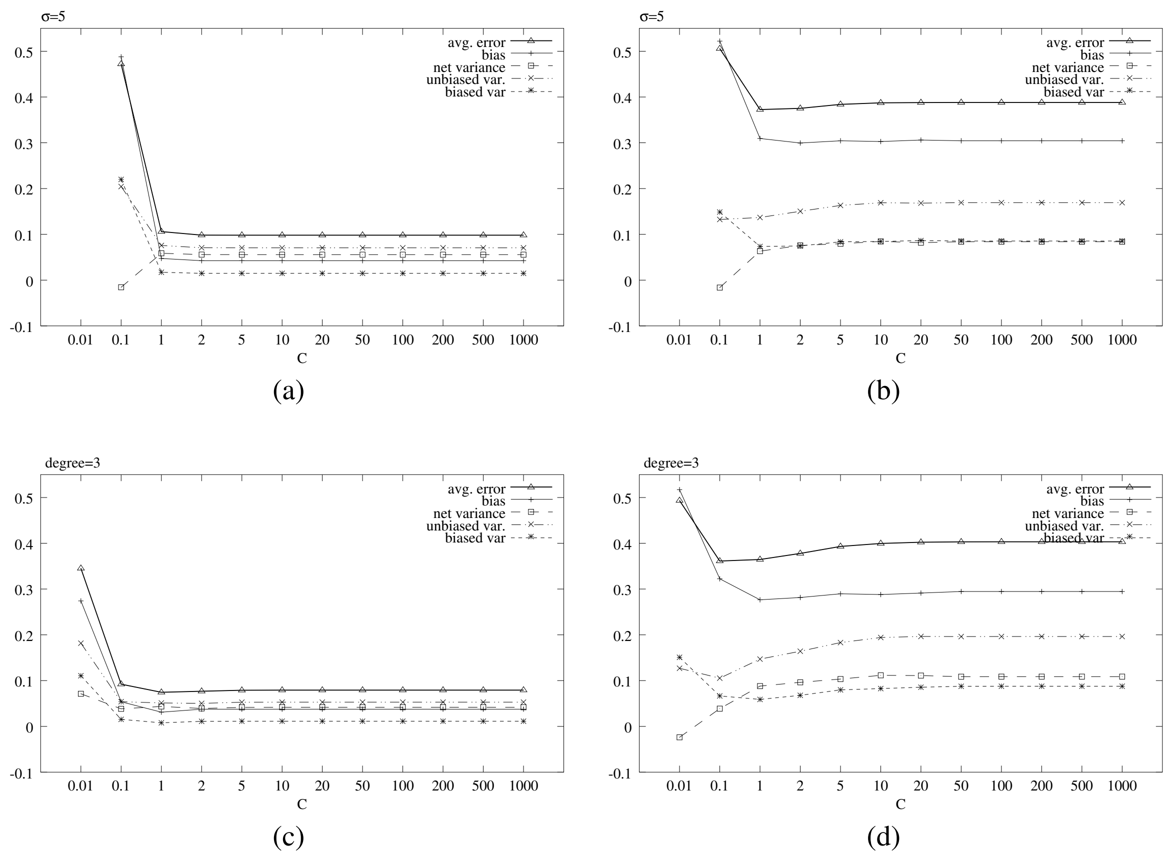

Figure 22: Effect of noise on bias and variance. The bias-variance decomposition of the error is shown while varying the C regularization parameter with polynomial and Gaussian kernels. (a) Letter-Two: Gaussian kernel, $\sigma = 5$, (b) Letter-Two with added noise: Gaussian kernel, $\sigma = 5$, (c) Letter-Two: polynomial kernel, degree = 3, (d) Letter-Two with added noise: polynomial kernel, degree = 3.

main effect of noise in this specific situation consists in incrementing the bias and consequently the average error. Indeed, with Gaussian kernels (Figure 22 (a) and (b)) the bias is raised to about 0.3, with an increment of about 0.25 with respect to the data set without noise, while the net-variance is incremented only by about 0.02, as the increment of the unbiased variance is counter-balanced by the increment of the biased variance. A similar behavior is registered also with polynomial (Figure 22 (c) and (d)) and dot product kernels (Figure 19 (c) and (d)).

6. Characterization of Bias-Variance Decomposition of the Error

Despite the differences observed in different data sets, common patterns of bias and variance can be detected for each of the kernels considered in this study. Each kernel presents a specific charac-

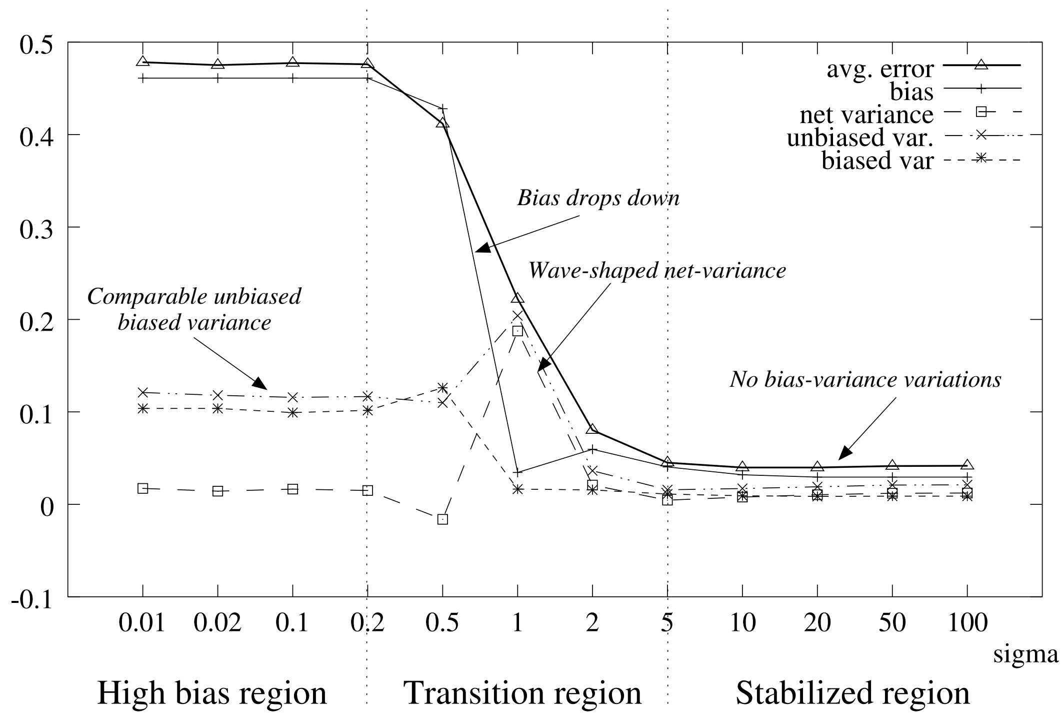

Figure 23: The 3 regions of the error in RBF-SVM with respect to σ.

terization of bias and variance with respect to its specific parameters, as explained in the following sections.

6.1 Gaussian Kernels

Error, bias, net-variance, unbiased and biased variance show a common trend in the 7 data sets we used in the experiments. Some differences, of course, arise in the different data sets, but we can distinguish three different regions in the error analysis of *RBF-SVM*, with respect to increasing values of σ (Figure 23):

1. **High bias region**. For low values of σ, error is high: it depends on high bias. Net-variance is about 0 as biased and unbiased variance are equivalent. In this region there are no remarkable fluctuations of bias and variance: both remain constant, with high values of bias and comparable values of unbiased and biased variance, leading to net-variance values near to 0. In some cases biased and unbiased variance are about equal, but different from 0, in other cases they are equal, but near to 0.

2. **Transition region**. Suddenly, for a critical value of σ, the bias decreases rapidly. This critical value depends also on C: for very low values of C, we have no learning, then for higher values the bias drops. Higher values of C cause the critical value of σ to decrease (Figure 4 (b) and 5). In this region the increase in net-variance is lower than the decrease in bias: so the average error decreases. The boundary of this region can be determined at the point where the error

stops decrementing. This region is characterized also by a particular trend of the net-variance. We can distinguish two main behaviors:

(a) **Wave-shaped net-variance**. Net-variance first increases and then decreases, producing a wave-shaped curve with respect to σ. The initial increment of the net-variance is due to the simultaneous increment of the unbiased variance and decrement of the biased variance. In the second part of the transition region, biased variance stabilizes and unbiased variance decreases, producing a parallel decrement of the net-variance. The rapid decrement of the error with σ is due to the rapid decrement of the bias, after which the bias stabilizes and the further decrement of the error with σ is determined by the net-variance reduction (Figure 4c, 5).

(b) **Semi-wave-shaped net-variance**. In other cases the net-variance curve with σ is not so clearly wave-shaped: the descending part is very reduced (Figure 5 e, f). In particular in the musk data set we have a continuous increment of the net-variance (due to the continuous growing of the unbiased variance with σ), and no wave-shaped curve is observed (at least for $C > 10$, Figure 11 d).

In both cases the increment of net-variance is slower than the increment in bias: as a result, the average error decreases.

3. **Stabilized region**. This region is characterized by small or no variations in bias and net-variance. For high values of σ both bias and net-variance stabilize and the average error is constant (Figure 4, 5). In other data sets the error increases with σ, because of the increment of the bias (Figure 11 a,b) or the unbiased variance (Figure 11 c,d).

In the first region, bias rules SVM behavior: in most cases the bias is constant and close to 0.5, showing that we have a sort of random guessing, without effective learning. It appears that the area of influence of each support vector is too small (Figure 7), and the learning machine overfits the data. This is confirmed by the fact that in this region the training error is about 0 and almost all the training points are support vectors.

In the transition region, the SVM starts to learn, adapting itself to the data characteristics. Bias rapidly goes down (at the expenses of a growing net-variance), but for higher values of σ (in the second part of the transition region), sometimes net-variance also goes down, working to lower the error (Figure 5).

Even if the third region is characterized by no variations in bias and variance, sometimes for low values of C, the error increases with σ (Figure 10 a, 12 a), as a result of the bias increment; on the whole RBF-SVMs are sensitive to low values of C: if C is too low, then bias can grow quickly. High values of C lower the bias(Figure 12 c, d).

6.2 Polynomial and Dot Product Kernels

For polynomial and dot product SVMs, we have also characterized the behavior of SVMs in terms of average error, bias, net-variance, unbiased and biased variance, even if we are not able to distinguish between different regions clearly defined.

However, common patterns of the error curves with respect to the polynomial degree, considering bias, net-variance and unbiased and biased variance can be noticed.

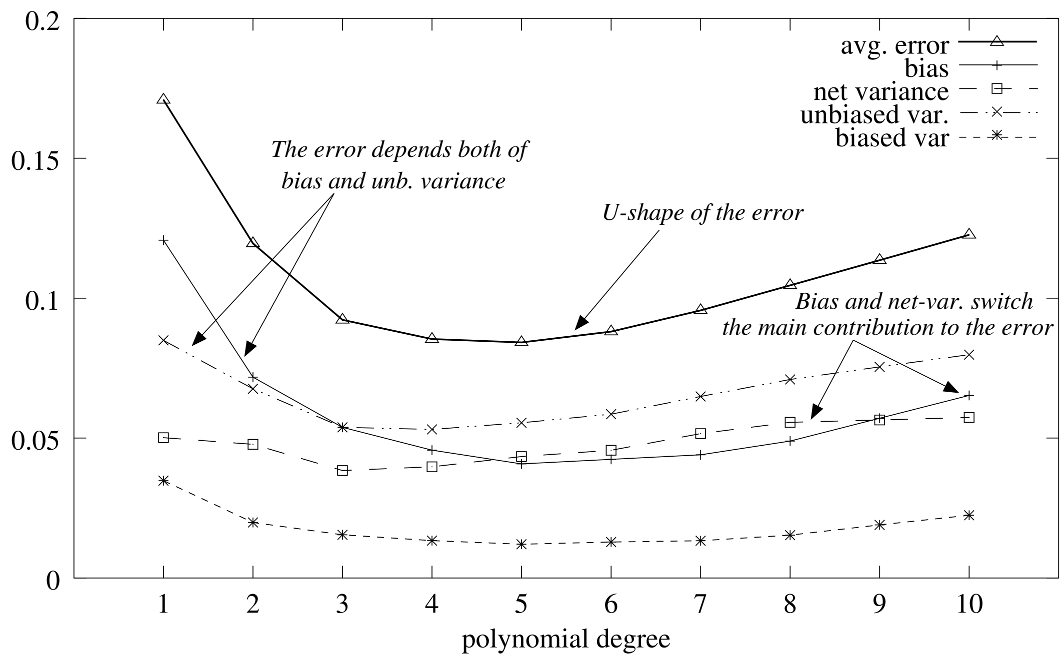

Figure 24: Behaviour of polynomial SVM with respect of the bias-variance decomposition of the error.

The average loss curve shows in general a U shape with respect to the polynomial degree, and this shape may depend on both bias and unbiased variance or in some cases mostly on the unbiased variance according to the characteristics of the data set. From these general observations we can schematically distinguish two main global pictures of the behaviour of polynomial SVM with respect to the bias-variance decomposition of the error:

1. **Error curve shape bias-variance dependent**.
   In this case the shape of the error curve is dependent both on the unbiased variance and the bias. The trend of bias and net-variance can be symmetric or they can also have non coincident paraboloid shape, depending on C parameter values (Figure 14 c, d and 15). Note that bias and net variance show often opposite trends (Figure 15).

2. **Error curve shape unbiased variance dependent**.
   In this case the shape of the error curve is mainly dependent on the unbiased variance. The bias (and the biased variance) tend to be degree independent, especially for high values of C (Figure 14 a, b) .

Figure 24 schematically summarizes the main characteristics of the bias-variance decomposition of error in polynomial SVM. Note however that the error curve depends for the most part on both variance and bias: the prevalence of the unbiased variance (Figure 14 a, b) or the bias seems to depend mostly on the distribution of the data.

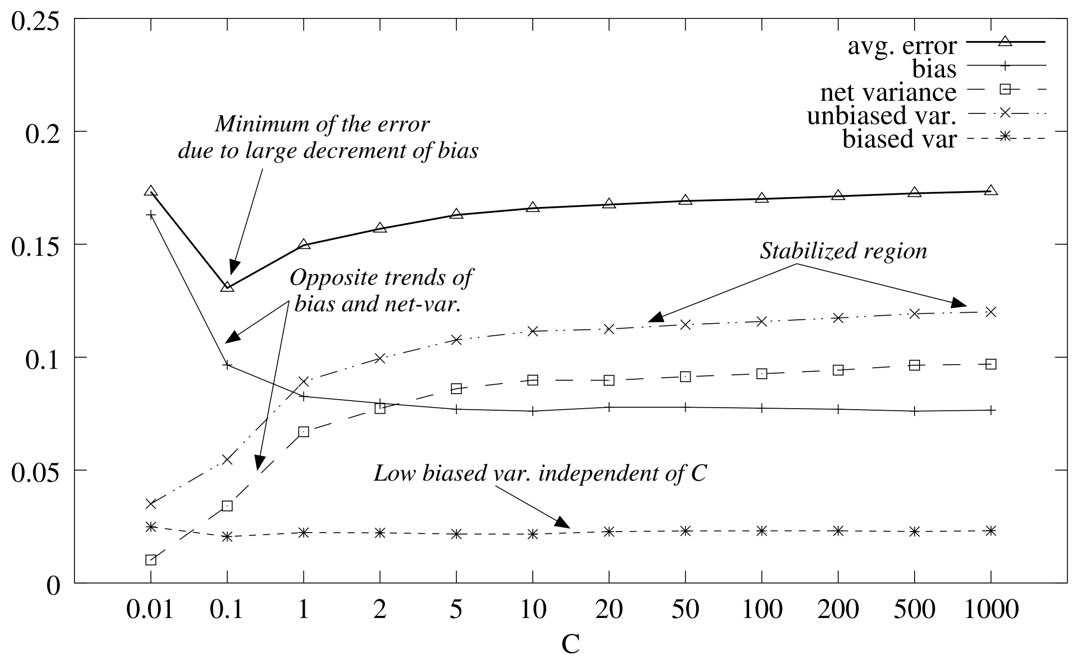

Figure 25: Behaviour of the dot product SVM with respect of the bias-variance decomposition of the error.

The increment of the values of C tends to flatten the U shape of the error curve: in particular for large C values bias becomes independent with respect to the degree (Figure 17). Moreover the C parameter plays also a regularization role (Figure 18)

Dot product SVM are characterized by opposite trends of bias and net-variance: bias decrements, while net-variance grows with respect to C; then, for higher values of C both stabilize. The combined effect of these symmetric curves produces a minimum of the error for low values of C, as the initial decrement of bias with C is larger than the initial increment of net-variance. Then the error slightly increases and stabilizes with C (Figure 19). The shape of the net-variance curve is determined mainly by the unbiased variance: it increases and then stabilizes with respect to C. On the other hand the biased variance curve is flat, remaining small for all values of C. A schematic picture of this behaviour is given in Figure 25.

7. Two Directions for Developing Ensembles of SVMs

In addition to providing insights into the behavior of SVMs, the analysis of the bias-variance decomposition of the error can identify the situations in which ensemble methods might improve SVM performance.

On several real-world problems, SVM ensembles are reported to give improvements over single SVMs (Kim et al., 2002; Valentini et al., 2003), but few works showed also negative experimental results about ensembles of SVMs (Buciu et al., 2001; Evgeniou et al., 2000). In particular Evgeniou et al. (2000) experimentally found that leave-one-out error bounds for kernel machines ensembles are tighter that the equivalent ones for single machines, but they showed that with accurate parameters tuning single SVMs and ensembles of SVMs perform similarly.

In this section we propose to exploit bias-variance analysis in order to develop ensemble methods well tuned to the bias-variance characteristics of the base learners. In particular we present two possible ways of applying bias-variance analysis to develop SVM-based ensemble methods.

7.1 Bagged Ensemble of Selected Low-Bias SVMs

From a general standpoint, considering different kernels and different parameters of the kernel, we can observe that the minimum of the error, bias and net-variance (and in particular unbiased variance) do not match. For instance, considering RBF-SVM we see that we achieve the minimum of the error, bias and net-variance for different values of σ (see, for instance, Figure 5). Similar considerations can also be applied to polynomial and dot product SVM. Often, modifying parameters of the kernel, if we gain in bias we lose in variance and vice versa, even if this is not a rule.

Under the bootstrap assumption, bagging reduces only variance. From bias-variance decomposition we know that unbiased variance reduces the error, while biased variance increases the error. Hence bagging should be applied to low-biased classifiers, because the biased variance will be small.

Summarizing, we can schematically consider the following observations:

- We know that bagging lowers net-variance (in particular unbiased variance) but not bias (Breiman, 1996b).

- SVMs are strong, low-biased learners, but this property depends on the proper selection of the kernel and its parameters.

- If we can identify low-biased base learners with a relatively high unbiased variance, bagging can lower the error.

- Bias-variance analysis can identify SVMs with low bias.

Hence a basic high-level algorithm for a general *Bagged ensemble of selected low-bias SVMs* is the following:

1. Estimate bias-variance decomposition of the error for different SVM models

2. Select the SVM model with the lowest bias

3. Perform bagging using as base learner the SVM with the estimated lowest bias.

This approach combines the low bias properties of SVMs with the low unbiased variance properties of bagging and should produce ensembles with low overall error. We named this approach *Lobag*, that stands for *Lo*w bias *bag*ging. Using SVMs as base learners, depending on the type of kernel and parameters considered, and on the way the bias is estimated for the different SVM models, different algorithmic variants can be provided: For instance, depending on the type of kernel and parameters considered, different implementations can be given:

1. Selecting the RBF-SVM with the lowest bias with respect to the C and σ parameters.

2. Selecting the polynomial-SVM with the lowest bias with respect to the C and degree parameters.

3. Selecting the dot-prod-SVM with the lowest bias with respect to the C parameter.

4. Selecting the SVM with the lowest bias with respect to the kernel.

Another issue is how to implement the estimation of the bias-variance decomposition of the error for different SVM models. We could use cross-validation in conjunction with bootstrap replicates, or out-of-bag estimates (especially if we have small training sets), or hold-out techniques in conjunction with bootstrap replicates if we have sufficiently large training sets.

A first implementation of this approach, using an out-of-bag estimate of the bias-variance decomposition of the error, has been proposed, and quite encouraging results have been achieved (Valentini and Dietterich, 2003).

Another problem is the estimate of the noise in real data sets. A straightforward approach simply consists in disregarding it, but in this way we could overestimate the bias (see Section 5.4). Some heuristics are proposed in James (2003), but the problem remains substantially unresolved.

It is worth noting that this approach can be viewed as an alternative way for tuning SVM parameters, using an ensemble instead of a single SVM. From this standpoint recent works proposed to automatically choose multiple kernel parameters (Chapelle et al., 2002; Grandvalet and Canu, 2003), setting, for instance different σ values for each input dimension in Gaussian kernels, by applying a minimax procedure to iteratively maximize the margin of the SVM and to minimize an estimate of the generalization error over the set of kernel parameters (Chapelle et al., 2002). This promising approach could be in principle extended to minimize the bias, instead of the overall error. To this purpose we need to solve non trivial problems such as providing an upper bound for the bias and the variance, or at least an easy to compute their estimator having, if possible, an analytical expression. This approach could represent a new interesting research line that could improve the performances and/or reduce the computational burden of the Lobag method.

7.2 Heterogeneous Ensembles of SVM

The analysis of bias-variance decomposition of error in SVM shows that the minimum of the overall error, bias, net-variance, unbiased and biased variance occur often in different SVM models. These different behaviors of different SVM models could be in principle exploited to produce diversity in ensembles of SVMs. Although the diversity of base learner itself does not assure the error of the ensemble will be reduced (Kuncheva et al., 2001b), the combination of accuracy and diversity in most cases does (Dietterich, 2000a). As a consequence, we could select different SVM models as base learners by evaluating their accuracy and diversity through the bias-variance decomposition of the error.

Our results show that the "optimal region" (low average loss region) is quite large in RBF-SVMs (Figure 4). This means that C and σ do not need to be tuned extremely carefully. From this point of view, we can avoid time-consuming model selection by combining RBF-SVMs trained with different σ values all chosen from within the "optimal region." For instance, if we know that the error curve looks like the one depicted in Figure 23, we could try to fit a sigmoid-like curve using only few values to estimate where the stabilized region is located. Then we could train an heterogeneous ensemble of SVMs with different σ parameters (located in the low bias region) and average them according to their estimated accuracy.

A high-level algorithm for *Heterogeneous Ensembles of SVMs* could include the following steps:

1. Individuate the "optimal region" through bias-variance analysis of the error

2. Select the SVMs with parameters chosen from within the optimal region defined by bias-variance analysis.

3. Combine the selected SVMs by majority or weighted voting according to their estimated accuracy.

We could use different methods or heuristics to find the "optimal region" (see Section 5.1.3) and we have to define also the criterion used to select the SVM models inside the "optimal region" (for instance, improvement of the diversity). The combination could be performed using also other approaches, such as minimum, maximum, average and OWA aggregating operators (Kittler et al., 1998) or Behavior-Knowledge space method (Huang and C. Y., 1995), Fuzzy aggregation rules (Wang et al., 1998), Decision templates (Kuncheva et al., 2001a) or Meta-learning techniques (Prodromidis et al., 1999). Bagging and boosting (Freund and Schapire, 1996) methods can also be combined with this approach to further improve diversity and accuracy of the base learners.

7.3 Numerical Experiments with Low Bias Bagged SVMs

In order to show that these research directions could be fruitful to follow further, we performed numerical experiments on different data sets to test the Lobag ensemble method using SVMs as base learners. We compared the results with single SVMs and classical bagged SVM ensembles. We report here some preliminary results.More detailed results are reported in Valentini and Dietterich (2003).

We employed the 7 different two-class data sets described in Section 4.1, using small $\mathcal{D}$ training sets and large test $\mathcal{T}$ sets in order to obtain a reliable estimate of the generalization error: the number of examples for $\mathcal{D}$ was set to 100, while the size of $\mathcal{T}$ ranged from a few thousands for the "real" data sets to ten thousands for synthetic data sets. Then we applied the Lobag algorithm setting the number of samples bootstrapped from $\mathcal{D}$ to 100, and performing an out-of-bag estimate of the bias-variance decomposition of the error. The selected lobag, bagged and single SVMs were finally tested on the separated test set $\mathcal{T}$.

Table 7.3 shows the results of the experiments. We measured 20 outcomes for each method: 7 data sets, and 3 kernels (Gaussian, polynomial, and dot product) applied to each data set except P2 for which we did not apply the dot product kernel (because it was obviously inappropriate). For each pair of methods, we applied the McNemar test (Dietterich, 1998) to determine whether there was a significant difference in predictive accuracy on the test set.

On nearly all the data sets, both bagging and Lobag outperform the single SVMs independently of the kernel used. The null hypothesis that Lobag has the same error rate as a single SVM is rejected at or below the 0.1 significance level in 17 of the 20 cases, while the null hypothesis that bagging has the same error rate as a single SVM is rejected at or below the 0.1 level in 13 of the 20 cases. Most importantly, Lobag generally outperforms standard bagging. Lobag is statistically significantly better than bagging in 9 of the 20 cases, and significantly inferior only once.

These preliminary results show the feasibility of our approach, as shown also by similar experiments presented in Valentini and Dietterich (2003), but we need more experimental studies and

Kernel type	**E$_{\text{lobag}}$**	**E$_{\text{bag}}$**	**E$_{\text{single}}$**	**Confidence level**		
				L/B	**L/S**	**B/S**
	Data set *P2*					
Polyn.	0.1735	0.2008	0.2097	0.001	0.001	0.001
Gauss.	0.1375	0.1530	0.1703	0.001	0.001	0.001
	Data set *Waveform*					
Linear	0.0740	0.0726	0.0939	1	0.001	0.001
Polyn.	0.0693	0.0707	0.0724	1	0.1	0.1
Gauss.	0.0601	0.0652	0.0692	0.001	0.001	0.001
	Data set *Grey-Landsat*					
Linear	0.0540	0.0540	0.0650	1	0.001	0.001
Polyn.	0.0400	0.0440	0.0480	1	0.1	1
Gauss.	0.0435	0.0470	0.0475	0.1	0.1	1
	Data set *Letter-Two*					
Linear	0.0881	0.0929	0.1011	1	0.025	0.05
Polyn.	0.0701	0.0717	0.0831	1	0.05	0.1
Gauss.	0.0668	0.0717	0.0799	1	1	1
	Data set *Letter-Two with added noise*					
Linear	0.3535	0.3518	0.3747	1	1	0.1
Polyn.	0.3404	0.3715	0.3993	1	0.05	0.1
Gauss.	0.3338	0.3764	0.3829	0.05	0.025	1
	Data set *Spam*					
Linear	0.1408	0.1352	0.1760	0.05	0.001	0.001
Polyn.	0.0960	0.1034	0.1069	0.1	0.025	1
Gauss.	0.1130	0.1256	0.1282	0.005	0.001	1
	Data set *Musk*					
Linear	0.1291	0.1291	0.1458	1	0.001	0.001
Polyn.	0.1018	0.1157	0.1154	0.001	0.001	1
Gauss.	0.0985	0.1036	0.0936	0.05	1	0.05

Table 4: Results of the experiments using pairs of train $\mathcal{D}$ and test $\mathcal{T}$ sets. E_{lobag}, E_{bag} and E_{SVM} stand respectively for estimated error of lobag, bagged and single SVMs on the test set $\mathcal{T}$. The three last columns show the confidence level according to the Mc Nemar test. $\mathbf{L/B}$, $\mathbf{L/S}$ and $\mathbf{B/S}$ stand respectively for the comparison Lobag/Bagging, Lobag/Single SVM and Bagging/Single SVM. If the confidence level is equal to 1, no significant difference is registered.

applications to real problems in order to better understand when and in which conditions this approach could be fruitful.

8. Conclusion and Future Works

We applied bias-variance decomposition of the error as a tool to gain insights into SVM learning algorithm. In particular we performed an analysis of bias and variance of SVMs, considering Gaussian, polynomial, and dot product kernels. The relationships between parameters of the kernel and bias, net-variance, unbiased and biased variance have been studied through an extensive experimentation involving training, testing, and bias-variance analysis of more than half million of SVMs.

We discovered regular patterns in the behavior of the bias and variance, and we related those patterns to the parameters and kernel functions of the SVMs. The characterization of bias-variance decomposition of the error showed that in Gaussian kernels we can individuate at least three different regions with respect to the σ parameter, while in polynomial kernels the U shape of the error can be determined by the combined effects of bias and unbiased variance. The analysis revealed also that the expected trade-off between bias and variance holds systematically for dot product kernels, while other kernels showed more complex relationships.

The information supplied by bias-variance analysis suggests two promising approaches for designing ensembles of SVMs. One approach is to employ low-bias SVMs as base learners in a bagged ensemble. The other approach is to apply bias-variance analysis to construct a heterogeneous, diverse set of accurate and low-bias classifiers. We are designing and experimenting with both of these approaches.

An outgoing development of this work extends this analysis to bagged and boosted ensemble of SVMs, in order to achieve more insights about the behavior of SVM ensembles based on resampling methods.

In our experiments we did not explicitly consider the noise: analyzing the role of the noise in the decomposition of the error (Section 5.4) could help to develop ensemble methods specifically designed for noisy data.

Moreover in our experiments we did not explicitly consider the characteristics of the data. Nonetheless, such as we could expect and as our experiments suggested, different data characteristics influence bias-variance patterns in learning machines. To this purpose we plan to explicitly analyze the relationships between bias-variance decomposition of the error and data characteristics, using data complexity measures based on geometrical and topological characteristics of the data (Li and Vitanyi, 1993; Ho and Basu, 2002).

Acknowledgments

We thanks the anonymous reviewers for their comments and suggestions.

Appendix A.

In this appendix we discuss the notions of systematic and variance effect introduced by James (2003), showing that these quantities are reduced respectively to the bias and the net-variance when the $0/1$ loss is used and the noise is disregarded.

James (2003) provides definitions of bias and variance that are similar to those provided by Domingos (2000c). Indeed bias and variance definitions are based on quantities that he named the systematic part *sy* of the prediction y and the systematic part *st* of the target t. These correspond respectively to the Domingos main prediction (Equation2) and optimal prediction (Equation1). Moreover James distinguishes between bias and variance and *systematic* and *variance effects*. Bias and variance satisfy respectively the notion of the difference between the systematic parts of y and t, and the variability of the estimate y. Systematic effect *SE* represents the change in error of predicting t when using *sy* instead of *st* and the variance effect VE the change in prediction error when using y

instead of sy in order to predict t. Using Domingos' notation (y_m for sy, and y_* for st) the variance effect is

$$VE(y,t) = E_{y,t}[L(y,t)] - E_t[L(t,y_m)],$$

while the systematic effect corresponds to

$$SE(y,t) = E_t[L(t,y_m)] - E_t[L(t,y_*)].$$

In other words the systematic effect represents the change in prediction error caused by bias, while the variance effect the change in prediction error caused by variance.

While for the squared loss the two sets of bias-variance definitions match, for general loss functions the identity does not hold. In particular for the 0/1 loss James proposes the following definitions for noise, variance and bias with 0/1 loss:

$$\begin{aligned} N(\mathbf{x}) &= P(t \neq y_*), \\ V(\mathbf{x}) &= P(y \neq y_m), \\ B(\mathbf{x}) &= I(y_* \neq y_m), \end{aligned} \tag{10}$$

where $I(z)$ is 1 if z is true and 0 otherwise.

The variance effect for the 0/1 loss can be expressed as

$$\begin{aligned} VE(y,t) = E_{y,t}[L(y,t) - L(t,y_m)] &= P_{y,t}(y \neq t) - P_t(t \neq y_m) = \\ = 1 - P_{y,t}(y = t) - (1 - P_t(t = y_m)) &= P_t(t = y_m) - P_{y,t}(y = t), \end{aligned} \tag{11}$$

while the systematic effect is

$$\begin{aligned} SE(y,t) = E_t[L(t,y_m)] - E_t[L(t,y_*)] &= P_t(t \neq y_m) - P_t(t \neq y_*) = \\ = 1 - P_t(t = y_m) - (1 - P_t(t = y_*)) &= P_t(t = y_*) - P_t(t = y_m). \end{aligned} \tag{12}$$

If we let $N(\mathbf{x}) = 0$, considering Equation 7, 10 and Equation 11 the variance effect becomes

$$\begin{aligned} VE(y,t) = P_t(t = y_m) - P_{y,t}(y = t) &= P(y_* = y_m) - P_y(y = y_*) = \\ = 1 - P(y_* \neq y_m) - (1 - P_y(y \neq y_*)) &= 1 - B(\mathbf{x}) - (1 - EL(\mathcal{L},\mathbf{x})) = \\ EL(\mathcal{L},\mathbf{x}) - B(\mathbf{x}) = V_n(\mathbf{x}), & \end{aligned} \tag{13}$$

while from Equation 10 and Equation 12 the systematic effect becomes

$$\begin{aligned} SE(y,t) = P_t(t = y_*) - P_t(t = y_m) &= 1 - P_t(t \neq y_*) - (1 - P_t(t \neq y_m)) = \\ P(y* \neq y_m) = I(y* \neq y_m) = B(\mathbf{x}). & \end{aligned} \tag{14}$$

Hence if $N(\mathbf{x}) = 0$, it follows that the variance effect is equal to the net-variance (Equation 13), and the systematic effect is equal to the bias (Equation 14).

References

E. L. Allwein, R. E. Schapire, and Y. Singer. Reducing multiclass to binary: a unifying approach for margin classifiers. *Journal of Machine Learning Research*, 1:113–141, 2000.

E. Bauer and R. Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting and variants. *Machine Learning*, 36(1/2):525–536, 1999.

O. Bousquet and A. Elisseeff. Stability and Generalization. *Journal of Machine Learning Research*, 2:499–526, 2002.

L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996a.

L. Breiman. Bias, variance and arcing classifiers. Technical Report TR 460, Statistics Department, University of California, Berkeley, CA, 1996b.

L. Breiman. Random Forests. *Machine Learning*, 45(1):5–32, 2001.

I. Buciu, C. Kotropoulos, and I. Pitas. Combining Support Vector Machines for Accurate Face Detection. In *Proc. of ICIP'01*, volume 1, pages 1054–1057, 2001.

O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee. Choosing multiple parameters for support vector machines. *Machine Learning*, 46(1):131–159, 2002.

S. Cohen and N. Intrator. Automatic Model Selection in a Hybrid Perceptron/Radial Network. In *Multiple Classifier Systems. Second International Workshop, MCS 2001, Cambridge, UK*, volume 2096 of *Lecture Notes in Computer Science*, pages 349–358. Springer-Verlag, 2001.

T. G. Dietterich. Approximate statistical test for comparing supervised classification learning algorithms. *Neural Computation*, (7):1895–1924, 1998.

T. G. Dietterich. Ensemble methods in machine learning. In J. Kittler and F. Roli, editors, *Multiple Classifier Systems. First International Workshop, MCS 2000, Cagliari, Italy*, volume 1857 of *Lecture Notes in Computer Science*, pages 1–15. Springer-Verlag, 2000a.

T. G. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting and randomization. *Machine Learning*, 40(2):139–158, 2000b.

T. G. Dietterich and G. Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, (2):263–286, 1995.

P. Domingos. A unified bias-variance decomposition. Technical report, Department of Computer Science and Engineering, University of Washington, Seattle, WA, 2000a.

P. Domingos. A Unified Bias-Variance Decomposition and its Applications. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 231–238, Stanford, CA, 2000b. Morgan Kaufmann.

P. Domingos. A Unified Bias-Variance Decomposition for Zero-One and Squared Loss. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence*, pages 564–569, Austin, TX, 2000c. AAAI Press.

T. Evgeniou, L. Perez-Breva, M. Pontil, and T. Poggio. Bounds on the Generalization Performance of Kernel Machine Ensembles. In P. Langley, editor, *Proc. of the Seventeenth International Conference on Machine Learning (ICML 2000)*, pages 271–278. Morgan Kaufmann, 2000.

Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In *Proceedings of the 13th International Conference on Machine Learning*, pages 148–156. Morgan Kauffman, 1996.

J. H. Friedman. On bias, variance, 0/1 loss and the curse of dimensionality. *Data Mining and Knowledge Discovery*, 1:55–77, 1997.

S. Geman, E. Bienenstock, and R. Doursat. Neural networks and the bias-variance dilemma. *Neural Computation*, 4(1):1–58, 1992.

Y. Grandvalet and S. Canu. Adaptive Scaling for Feature Selection in SVMs. In S. Becker, S. Thrun, and K. Obermayer, editors, *NIPS 2002 Conference Proceedings, Advances in Neural Information Processing Systems*, volume 15, Cambridge, MA, 2003. MIT Press.

T. Heskes. Bias/Variance Decompostion for Likelihood-Based Estimators. *Neural Computation*, 10:1425–1433, 1998.

T. K. Ho and M. Basu. Complexity measures of supervised classification problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):289–300, 2002.

Y. S. Huang and Suen. C. Y. Combination of multiple experts for the recognition of unconstrained handwritten numerals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17: 90–94, 1995.

G. James. Variance and bias for general loss function. *Machine Learning*, (2):115–135, 2003.

T. Joachims. Making large scale SVM learning practical. In Smola A. Scholkopf B., Burges C., editor, *Advances in Kernel Methods - Support Vector Learning*, pages 169–184. MIT Press, Cambridge, MA, 1999.

H. C. Kim, S. Pang, H. M. Je, D. Kim, and S. Y. Bang. Pattern Classification Using Support Vector Machine Ensemble. In *Proceedings of the International Conference on Pattern Recognition, 2002*, volume 2, pages 20160–20163. IEEE, 2002.

J. Kittler, M. Hatef, R. P. W. Duin, and Matas J. On combining classifiers. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(3):226–239, 1998.

E. M. Kleinberg. A Mathematically Rigorous Foundation for Supervised Learning. In J. Kittler and F. Roli, editors, *Multiple Classifier Systems. First International Workshop, MCS 2000, Cagliari, Italy*, volume 1857 of *Lecture Notes in Computer Science*, pages 67–76. Springer-Verlag, 2000.

R. Kohavi and D. H. Wolpert. Bias plus variance decomposition for zero-one loss functions. In *Proc. of the Thirteenth International Conference on Machine Learning, The Seventeenth International Conference on Machine Learning*, pages 275–283, Bari, Italy, 1996. Morgan Kaufmann.

E. Kong and T. G. Dietterich. Error-correcting output coding correct bias and variance. In *The XII International Conference on Machine Learning*, pages 313–321, San Francisco, CA, 1995. Morgan Kauffman.

L. I. Kuncheva, J. C. Bezdek, and R. P. W. Duin. Decision templates for multiple classifier fusion: an experimental comparison. *Pattern Recognition*, 34(2):299–314, 2001a.

L. I. Kuncheva, F. Roli, G. L. Marcialis, and C. A. Shipp. Complexity of Data Subsets Generated by the Random Subspace Method: An Experimental Investigation. In J. Kittler and F. Roli, editors, *Multiple Classifier Systems. Second International Workshop, MCS 2001, Cambridge, UK*, volume 2096 of *Lecture Notes in Computer Science*, pages 349–358. Springer-Verlag, 2001b.

L. I. Kuncheva and C. J. Whitaker. Measures of diversity in classifier ensembles. *Machine Learning*, 51:181–207, 2003.

M. Li and P Vitanyi. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer-Verlag, Berlin, 1993.

L. Mason, P. Bartlett, and J. Baxter. Improved generalization through explicit optimization of margins. *Machine Learning*, 2000.

C. J. Merz and P. M. Murphy. UCI repository of machine learning databases, 1998. www.ics.uci.edu/mlearn/MLRepository.html.

A. Prodromidis, P. Chan, and S. Stolfo. Meta-Learning in Distributed Data Mining Systems: Issues and Approaches. In H. Kargupta and P. Chan, editors, *Advances in Distributed Data Mining*, pages 81–113. AAAI Press, 1999.

R. E. Schapire. A brief introduction to boosting. In Thomas Dean, editor, *16th International Joint Conference on Artificial Intelligence*, pages 1401–1406. Morgan Kauffman, 1999.

R. E. Schapire, Y. Freund, P. Bartlett, and W. Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *The Annals of Statistics*, 26(5):1651–1686, 1998.

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

R. Tibshirani. Bias, variance and prediction error for classification rules. Technical report, Department of Preventive Medicine and Biostatistics and Department od Statistics, University of Toronto, Toronto, Canada, 1996.

G. Valentini and T. G. Dietterich. Low Bias Bagged Support Vector Machines. In T. Fawcett and N. Mishra, editors, *Machine Learning, Proceedings of the Twentieth International Conference (ICML 2003)*, pages 752–759, Washington D. C., USA, 2003. AAAI Press.

G. Valentini and F. Masulli. NEURObjects: an object-oriented library for neural network development. *Neurocomputing*, 48(1–4):623–646, 2002.

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

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

D. Wang, J. M. Keller, C. A. Carson, K. K. McAdoo-Edwards, and C. W. Bailey. Use of fuzzy logic inspired features to improve bacterial recognition through classifier fusion. *IEEE Transactions on Systems, Man and Cybernetics*, 28B(4):583–591, 1998.