Chapter 3

Bias and variance in supervised learning

In this chapter, the important concepts of bias and variance are introduced. After an intuitive introduction to the bias/variance compromise, we discuss in Section 3.2 the bias/variance decompositions of the mean square error (in the context of regression problems) and of the classification error (in the context of classification problems). Section 3.3 provides an empirical study through various experiments which aim at providing some insight about how the parameters of the learning problem and of the learning algorithm influence bias and variance. Section 3.4 discusses the differences between two notions of variance: error variance and parameter variance. The last section of this chapter provides some analytical characterizations of bias and variance obtained in the context of statistical learning theory, in particular quantitative error bounds obtained in the context of neural networks.

3.1 Why more general is not necessarily better

As introduced in the previous chapter, the problem of supervised learning appears to be very straightforward and could be reduced to a simple optimization problem. Indeed, at first sight, it is "sufficient" to find a model in the family $\mathcal{H}$ which realizes the minimal error on the learning set. Further, if no good enough model is found in this family, it should be sufficient to extend the family or to exchange it for a more powerful one in terms of model flexibility. Thus, to avoid too many trials and errors before getting acceptable accuracy, it would be quite natural to directly use the largest available hypothesis space (e.g. one having the so-called universal approximation property). Unfortunately, in practice, good results on the learning set do not necessarily imply good generalization performance on the universe $\mathcal{U}$ (which is the actual goal of supervised learning), especially if the "size" of the hypothesis space is large in comparison to the sample size.

To explain intuitively why larger hypothesis spaces do not necessarily lead to better models, we use the simple unidimensional regression problem described in the top of Figure 3.1. In this synthetic problem, learning outputs are generated according to

$$Y = f_B(A) + \varepsilon,$$

where ε is distributed according to $N(0, \sigma)$. With squared error loss, the best possible model is thus f_B and its average (and also local) squared error is σ^2. Let us consider two extreme situations of a bad model structure choice.[1]

[1]Notice that in this example both A and Y are *continuous* real valued variables. For the sake of

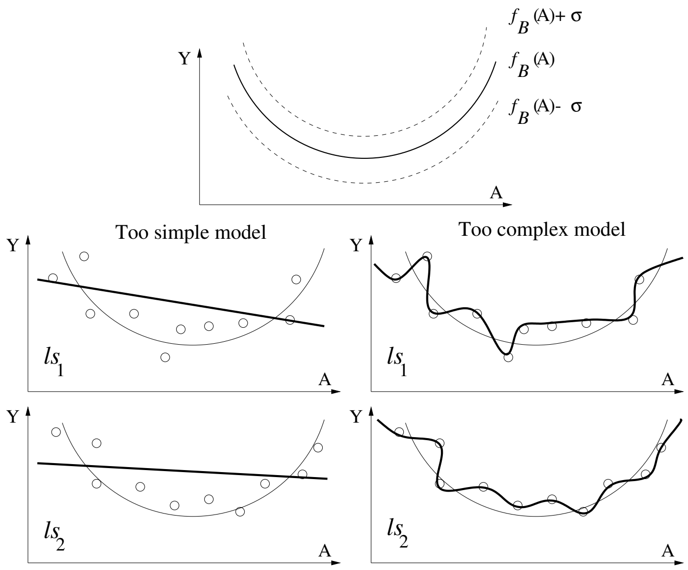


Figure 3.1: Top, our illustrative problem. Left, a linear model fitted to two learning samples. Right, a neural network with two hidden layers of 10 neurons fitted to the same samples

- A too simple model structure: using a linear model $Y = w \cdot A + b$ and minimizing squared error on the learning set, we obtain the estimations given in the left part of Figure 3.1 for two different learning set choices. These models are not very good, neither on their learning sets, nor in generalization. Whatever the learning set, there will always remain an error due to the fact that the model is too simple with respect to the complexity of f_B.

- A too complex model: by using a very complex model like a neural network with two hidden layers of ten neurons each, we get the functions at the right part of Figure 3.1 for the same learning sets. This time, models receive an almost perfect score on the learning set. However, their generalization errors are still not very good because of two phenomena. First, the learning algorithm is able to match perfectly the learning set and hence also the noise term. We say in this case that the learning algorithm "overfits" the data. Second, even if there is no noise, there will still remain some errors due to the high complexity of the model. Indeed, the learning algorithm has many different models at its disposal and if the learning set size is relatively small, several of them will realize a perfect match of the learning set. As at most one of them is a perfect image of the Bayes model, any other choice by the learning algorithm will result in suboptimality.

The main source of error is very different in both cases. In the first case, the error is essentially independent of the particular learning set and must be attributed to the lack of complexity of the model. This source of error is called **bias**. In the second case, on the other hand, the error may be attributed to the variability of the model from one learning set to another (which is due on the one hand to overfitting and on the other hand to the sparse nature of the learning set with respect to the complexity of the model). This source of error is called **variance**. Note that in the first case there is also a dependence

uniformity of notation, we nevertheless proceed by using upper case letters to denote the probability distribution of the variables.

of the model on the learning set and thus some variability of the predictions. However the resulting variance is negligible with respect to bias. In general, bias and variance both depend on the complexity of the model but in opposite direction and thus there must exist an optimal tradeoff between these two sources of error. As a matter of fact (see subsequent discussions), this optimal tradeoff depends also on the smoothness of the Bayes model and on the sample size. An important consequence of this is that automatic learning as *empirical risk minimization in a fixed hypothesis space*, as formulated in the previous chapter, is not suitable for most practical problems. Indeed, because of variance, we should always take care of not increasing too much the complexity of the model structure with respect to the complexity of the problem and the size of the learning sample.

In the next section, we give a formal additive decomposition of the mean (over all learning set choices) squared error into two terms which represent the bias and the variance effect. Some propositions of similar decompositions in the context of 0-1 loss-functions are also discussed. They show some fundamental differences between the two types of problems although bias and variance concepts are always useful.

3.2 Bias and variance decompositions

A learning algorithm produces a model f_{LS} from a learning sample LS of size N[2]. For a fixed sample size N, LS is a random variable. Hence, the model f_{LS} and its prediction $f_{LS}(\underline{a})$ at $\underline{a}$ are also random. In what follows, we will denote by $P_{LS}(y|\underline{a})$ the probability distribution of this output, i.e. the probability that a model (produced by a given learning algorithm, from samples of size N corresponding to a given learning problem) outputs y at the point $\underline{a}$:

$$P_{LS}(y|\underline{a}) \stackrel{\triangle}{=} P(f_{LS}(\underline{a}) = y) = \sum_{ls \in (\underline{\mathcal{A}},\mathcal{Y})^N | f_{ls}(\underline{a})=y} P(LS = ls). \tag{3.1}$$

The error of these models $Err(f_{LS})$ is again a random variable and we are interested in studying the expected value of this error (over the set of all learning sets of a given size N) $E_{LS}\{Err(f_{LS})\}$. The expectations may be inverted to give:

$$\begin{aligned} E_{LS}\{Err(f_{LS})\} &= E_{LS}\{E_{\underline{A},Y}\{L(Y, f_{LS}(\underline{A}))\}\} && (3.2)\\ &= E_{\underline{A}}\{E_{LS}\{E_{Y|\underline{A}}\{L(Y, f_{LS}(\underline{A}))\}\}\} && (3.3)\\ &= E_{\underline{A}}\{E_{LS}\{Err(f_{LS}(\underline{A})\}\}, && (3.4)\end{aligned}$$

where $Err(f_{LS}(\underline{A}))$ is the local error at point $\underline{A}$. Note here the symmetry (interchangeability) between the two random variables Y and $f_{LS}(\underline{A})$ in (3.3). The first one is distributed at $\underline{a}$ according to $P(y|\underline{a})$ ($= P(Y = y|\underline{A} = \underline{a})$) while the second one is distributed according to $P_{LS}(y|\underline{a})$. These distributions are illustrated in Figure 3.2, left in the case of a regression problem (density), right in the case of a three-class classification problem (discrete probability). The bias/variance decompositions will be expressed in terms of the parameters of these two distributions.

3.2.1 Regression error

In the case of a regression problem and if we use the square error criterion as a loss measure, we know that the best possible model is the function $f_B(\underline{a})$ which associates to each

[2]For simplicity reasons, the dependence on the learning set size N and on the learning algorithm will be forgotten in the notations throughout this chapter.

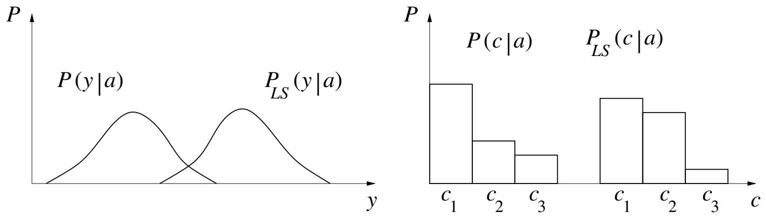


Figure 3.2: Distributions of outputs and model predictions at a point $\underline{a}$, left in regression, right in classification

input $\underline{a}$ the expectation of Y according to the conditional probability distribution $P(Y|\underline{a})$. Introducing this model in $E_{LS}\{Err(f_{LS}(\underline{a}))\}$, we get:

$$\begin{aligned}
E_{LS}\{Err(f_{LS}(\underline{a}))\} &= E_{LS}\{E_{Y|\underline{a}}\{(Y - f_{LS}(\underline{a}))^2\} && (3.5)\\
&= E_{LS}\{E_{Y|\underline{a}}\{(Y - f_B(\underline{a}) + f_B(\underline{a}) - f_{LS}(\underline{a}))^2\} && (3.6)\\
&= E_{LS}\{E_{Y|\underline{a}}\{(Y - f_B(\underline{a}))^2\} + E_{LS}\{E_{Y|\underline{a}}\{(f_B(\underline{a}) - f_{LS}(\underline{a}))^2\} + \\
&\quad E_{LS}\{E_{Y|\underline{a}}\{2 \cdot (Y - f_B(\underline{a})) \cdot (f_B(\underline{a}) - f_{LS}(\underline{a}))\} && (3.7)\\
&= E_{Y|\underline{a}}\{(Y - f_B(\underline{a}))^2\} + E_{LS}\{(f_B(\underline{a}) - f_{LS}(\underline{a}))^2\}, && (3.8)
\end{aligned}$$

since:

$$E_{Y|\underline{a}}\{(Y - f_B(\underline{a}))\} = E_{Y|\underline{a}}\{Y\} - f_B(\underline{a}) = 0 \tag{3.9}$$

by definition of the Bayes model. Symmetrically to the Bayes model, we can define the average model, $\overline{f}_{LS}$, which outputs the average prediction among all learning sets:

$$\overline{f}_{LS}(\underline{a}) = E_{LS}\{f_{LS}(\underline{a})\}. \tag{3.10}$$

Introducing this model in the second term of (3.8), we obtain:

$$\begin{aligned}
E_{LS}\{(f_B(\underline{a}) - f_{LS}(\underline{a}))^2\} &= E_{LS}\{(f_B(\underline{a}) - \overline{f}_{LS}(\underline{a}) + \overline{f}_{LS}(\underline{a}) - f_{LS}(\underline{a}))^2\} && (3.11)\\
&= E_{LS}\{(f_B(\underline{a}) - \overline{f}_{LS}(\underline{a}))^2\} + E_{LS}\{(f_{LS}(\underline{a}) - \overline{f}_{LS}(\underline{a}))^2\} + \\
&\quad E_{LS}\{2 \cdot (f_B(\underline{a}) - \overline{f}_{LS}(\underline{a})) \cdot (\overline{f}_{LS}(\underline{a}) - f_{LS}(\underline{a}))\} && (3.12)\\
&= (f_B(\underline{a}) - \overline{f}_{LS}(\underline{a}))^2 + E_{LS}\{(f_{LS}(\underline{a}) - \overline{f}_{LS}(\underline{a}))^2\} && (3.13)
\end{aligned}$$

since:

$$E_{LS}\{\overline{f}_{LS}(\underline{a}) - f_{LS}(\underline{a})\} = \overline{f}_{LS}(\underline{a}) - E_{LS}\{f_{LS}(\underline{a})\} = 0 \tag{3.14}$$

by definition of $\overline{f}_{LS}(\underline{a})$. In summary, we have the following well-known decomposition of the square error at point $\underline{a}$:

$$E_{LS}\{Err(f_{LS}(\underline{a}))\} = \sigma_R^2(\underline{a}) + \mathrm{bias}_R^2(\underline{a}) + \mathrm{var}_R(\underline{a}), \tag{3.15}$$

by defining:

$$\begin{aligned}
\sigma_R^2(\underline{a}) &= E_{Y|\underline{a}}\{(Y - f_B(\underline{a}))^2\}, && (3.16)\\
\mathrm{bias}_R^2(\underline{a}) &= (f_B(\underline{a}) - \overline{f}_{LS}(\underline{a}))^2, && (3.17)\\
\mathrm{var}_R(\underline{a}) &= E_{LS}\{(f_{LS}(\underline{a}) - \overline{f}_{LS}(\underline{a}))^2\}. && (3.18)
\end{aligned}$$

This error decomposition is well-known in estimation theory and has been introduced in the automatic learning community by Geman et al. [GBD92].

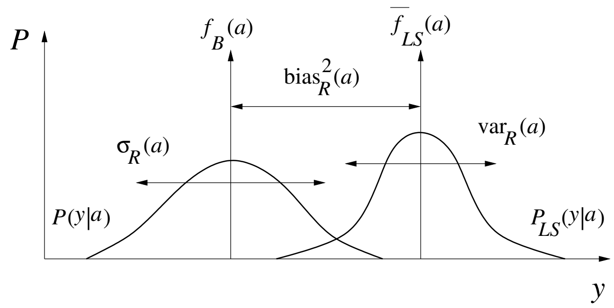


Figure 3.3: Residual error, regression bias and variance

The residual square error, $\sigma_R^2(\underline{a})$, is the error obtained by the Bayes model and it provides a theoretical lower bound which is independent of the learning algorithm. Thus the sub-optimality of a particular learning algorithm (its error with respect to the Bayes model, i.e. the second term of (3.8)) is composed of two terms:

- The (squared) bias, $\text{bias}_R^2(\underline{a})$, measuring the discrepancy between the Bayes model and the average model.
- The variance, $\text{var}_R(\underline{a})$, measuring the variability of the predictions with respect to the learning set randomness.

These three terms are illustrated in Figure 3.3. Residual error and variance are measures of the spread of the two densities and the bias is simply the distance between their means.

To explain why these two terms are effectively the consequence of the two phenomena discussed in the introduction of this chapter, we will come back to our simple regression problem. The average model is depicted in the top of Figure 3.4 for the two cases of bad model choice. Residual error, bias and variance for each position a are also drawn in the center part of the same figure. An illustration of the distributions of true and learned outputs are reproduced at the bottom of the figure. The residual error is entirely specified by the problem and loss criterion and hence independent of the algorithm and learning set used. When the model is too simple, we see on Figure 3.4 that the average model is far from the Bayes model almost everywhere and thus bias is high. The learning algorithm is not powerful enough to represent the Bayes model. On the other hand, the variance is low as the model does not match very strongly the learning set and thus the prediction at each point does not vary too much from one learning set to another. Bias is thus the dominant term of error. When the model is too complex, the distribution of predictions matches very strongly the distribution of outputs at each point. The average prediction is thus close to the Bayes model and the bias is low. However, because of the noise and the small learning set size, predictions are highly variable at each point. In this case, variance is the dominant term of error.

3.2.2 Classification error

The local mean error in classification is the probability of misclassification at point $\underline{a}$:

$$\begin{aligned} E_{LS}\{Err(f_{LS}(\underline{a}))\} &= E_{LS}\{E_{C|\underline{a}}\{1(C \neq f_{LS}(\underline{a}))\}\} && (3.19)\\ &= P(C \neq f_{LS}(\underline{a})|\underline{a}) && (3.20)\\ &= 1 - \sum_c P(c|\underline{a}) \cdot P_{LS}(c|\underline{a}). && (3.21) \end{aligned}$$

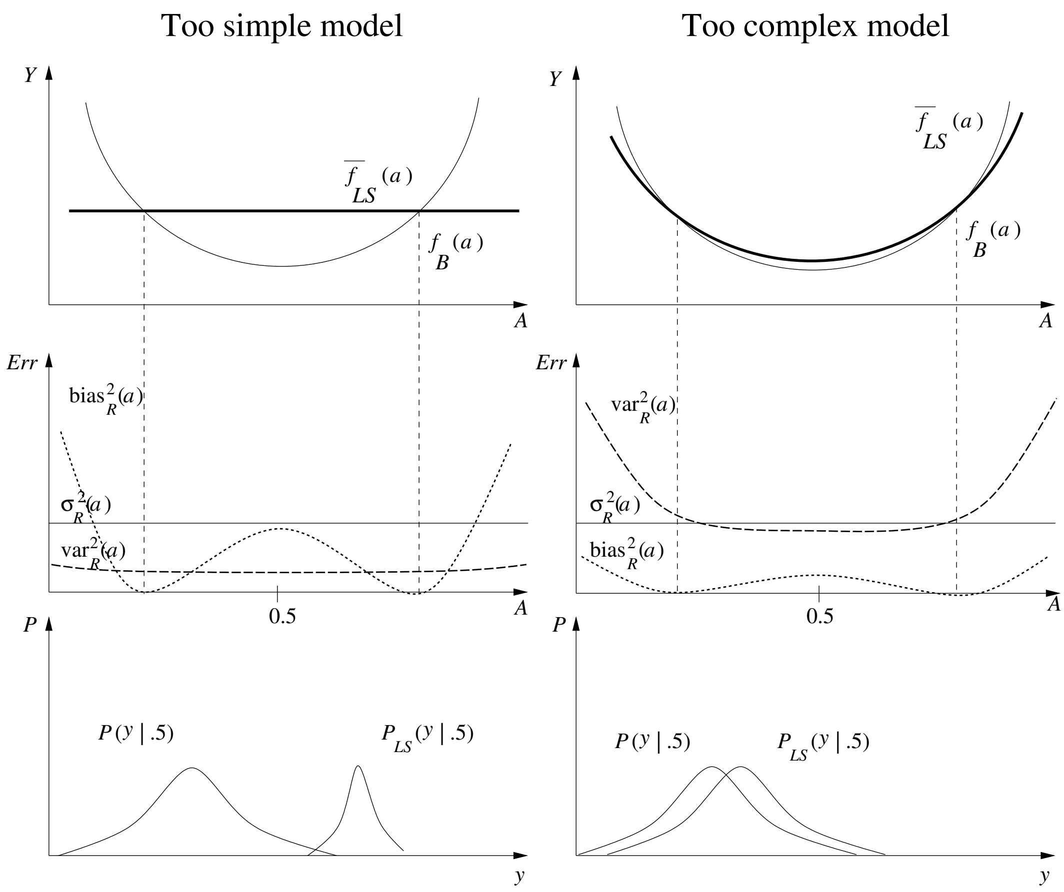


Figure 3.4: top: average model; center: residual error, bias, and, variance; bottom: distributions of outputs and predictions at the point $A = 0.5$

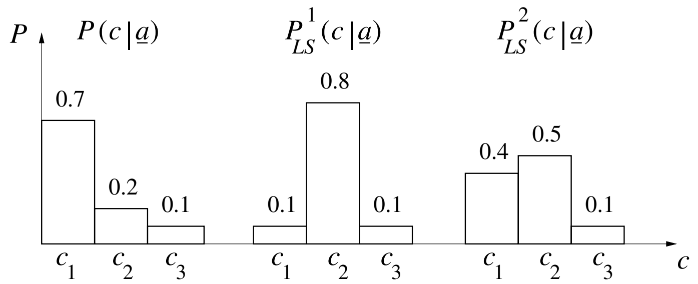


Figure 3.5: left, a classification problem; center and right, two different learning algorithms

The Bayes model in classification associates to each input $\underline{a}$ the most likely class according to the conditional class probability distribution, i.e. $f_B(\underline{a}) = \arg\max_c P(c|\underline{a})$. The residual or irreducible error at point $\underline{a}$, noted[3] $\sigma_C(\underline{a})$, becomes:

$$\sigma_C(\underline{a}) = 1 - P(f_B(\underline{a})|\underline{a}). \tag{3.22}$$

To keep symmetry with the Bayes model and by analogy with the regression problem, the equivalent in classification of the average model of regression is the *majority vote classifier* defined by:

$$f_{LS}^{MAJ}(\underline{a}) = \arg\max_c P_{LS}(c|\underline{a}), \tag{3.23}$$

which outputs at each point the class receiving the majority of votes among the distribution of classifiers induced from the distribution of learning sets (of size N). The bias in regression is the error of the average model with respect to the Bayes model. This definition yields here:

$$\text{bias}_C(\underline{a}) = 1(f_B(\underline{a}) \neq f_{LS}^{MAJ}(\underline{a})). \tag{3.24}$$

So, biased points are those for which the majority vote classifier disagrees with the Bayes classifier. On the other hand, variance can be defined as:

$$\text{var}_C(\underline{a}) = 1 - P_{LS}(f_{LS}^{MAJ}(\underline{a})|\underline{a}). \tag{3.25}$$

Indeed, this definition is certainly a measure of the variability of the predictions at a point $\underline{a}$:

- when $\text{var}_C(\underline{a}) = 0$, every model outputs the same class whatever the learning set from which it is induced;
- $\text{var}_C(\underline{a})$ is maximal when the probability of the class given by the majority vote classifier is equal to $\frac{1}{m}$ (with m the number of classes), which indeed corresponds to the most uncertain distribution of classifiers.

Unfortunately, defined in this way by analogy with the regression case, residual error, bias and variance do not sum up to give the local classification error. In other words:

$$E_{LS}\{Err(f_{LS}(\underline{a}))\} \neq \sigma_C(\underline{a}) + \text{bias}_C(\underline{a}) + \text{var}_C(\underline{a}). \tag{3.26}$$

Indeed, let us illustrate on a simple example how increased variance may decrease the average classification error in some situations. Figure 3.5 represents (at a certain point $\underline{a}$ of the input space) three probability distributions: the leftmost histogram represents the

[3]In our error decompositions for classification, we made the arbitrary choice of not using the "square" notations. Hence, we will use σ_C instead of σ_C^2 and bias_C instead of bias_C^2.

true conditional class probability distribution $P(c|\underline{a})$, showing also that the Bayes classifier would decide class c_1 at this point; the two other histograms represent the sampling distributions of the class decided by models produced by two different learning algorithms.

From Figure 3.5 we observe that both algorithms produce models which most probably will decide class c_2 (respectively with probability 0.8 and 0.5). Thus the two methods are biased ($\text{bias}^1_C(\underline{a}) = \text{bias}^2_C(\underline{a}) = 1$) since their majority vote classifiers would choose class c_2 instead of the most likely class (c_1). On the other hand, the variances of the two methods are obtained in the following way:

$$\begin{aligned} \text{var}^1_C(\underline{a}) &= 1 - 0.8 = 0.2 \\ \text{var}^2_C(\underline{a}) &= 1 - 0.5 = 0.5, \end{aligned}$$

and their average classification error rates are found to be

$$\begin{aligned} E_{LS}\{Err(f^1_{LS}(\underline{a}))\} &= 1 - 0.7 \cdot 0.1 - 0.2 \cdot 0.8 - 0.1 \cdot 0.1 = 0.76 \\ E_{LS}\{Err(f^2_{LS}(\underline{a}))\} &= 1 - 0.7 \cdot 0.4 - 0.2 \cdot 0.5 - 0.1 \cdot 0.1 = 0.61. \end{aligned}$$

Thus among these two methods with identical bias, it is the one having the largest variance which has the smallest average error rate.

It is easy to see that this happens here because of the existence of a bias. Indeed, in classification an algorithm which has small variance and high bias is an algorithm[4] which systematically (i.e. whatever the learning sample) produces a wrong[5] answer, whereas an algorithm which has a high bias but also a high variance is only wrong for a majority of learning samples, but not necessarily systematically. In other words, in classification, much variance can be beneficial because it can lead the system closer to the Bayes classification.

As a result of this counter-intuitive interaction between bias and variance terms in classification, a number of alternative measures of classification bias and variance have been proposed in the machine learning literature, none of which is really satisfactory (a detailed discussion of the most representative decompositions found in the literature is given in Appendix A). In order to study experimentally classification algorithms, we have selected among these latter a small subset of measures which seemed the most sensible to us. These measures are discussed below. One of the goals of our experiments in subsequent chapters is to show that such measures are still useful to analyze classification learning algorithms, although not plainly satisfactory from a theoretical point of view.

Decomposition proposed by Tibshirani

This decomposition was proposed first by Tibshirani [Tib96] and later on by James and Hastie [JH96]. They define the bias at point $\underline{a}$ as the difference between the (local) error probability of the majority vote classifier and that of the Bayes classifier:

$$\text{bias}_T(\underline{a}) = (1 - P(f^{MAJ}_{LS}(\underline{a})|\underline{a})) - (1 - P(f_B(\underline{a})|\underline{a})). \quad (3.27)$$

This quantity is necessarily non-negative and combined with the residual error gives the classification error of the majority vote classifier:

$$\begin{aligned} \sigma_C(\underline{a}) + \text{bias}_T(\underline{a}) &= 1 - P(f_B(\underline{a})|\underline{a}) + P(f_B(\underline{a})|\underline{a}) - P(f^{MAJ}_{LS}(\underline{a})|\underline{a}) & (3.28) \\ &= 1 - P(f^{MAJ}_{LS}(\underline{a})|\underline{a}) & (3.29) \\ &= Err(f^{MAJ}_{LS}(\underline{a})). & (3.30) \end{aligned}$$

[4] in our example, this corresponds to algorithm 1.

[5] a class different from the one chosen by the Bayes classifier.

In order to obtain an additive decomposition, Tibshirani then logically defines "variance" as the difference between the expected classification error of the models produced by the algorithm and the error of the majority vote classifier:

$$\mathrm{var}_T(\underline{a}) = E_{LS}\{Err(f_{LS}(\underline{a}))\} - Err(f_{LS}^{MAJ}(\underline{a})). \tag{3.31}$$

Because this quantity can be negative, Tibshirani names it *aggregation effect* instead of variance.

Thus the bias term in this decomposition corresponds to the additional error with respect to the residual error which would remain if we could drive variance to zero without affecting the majority vote classifier. Variance can thus be interpreted as the modification of error resulting from a complete reduction of the prediction variability. As it has been illustrated above that an increase of prediction variability may increase the error in classification, it is not surprising to note that var_T is not strictly positive. For example, for the second classifier of Figure 3.5, $E_{LS}\{Err(f_{LS}(\underline{a}))\}$ is 0.61 and $Err(f_{LS}^{MAJ}(\underline{a}))$ is 0.8, yielding an aggregation effect of -0.19. If the variance term is negative, a reduction of the prediction variability which does not affect the bias is likely to increase the error rate. On the contrary, if the variance term is positive, a reduction of the prediction variability results in a decrease of error. Notice that in many situations, the majority of points in the input space will be unbiased because most of them are located far from the optimal decision boundary, and for such points it is generally quite easy to identify the most probable class. Thus, only the points lying in regions where classes overlap may possibly be biased and hence present a negative variance. Thus, if we focus on the global expected error rate of these methods (averaged over the complete input space distribution) it is normally the case that the variance term is positive and that its reduction leads to accuracy improvement[6].

Variance measure proposed by Kohavi and Wolpert

While var_T is a measure of the effect on error of a modification of the prediction variability, it does not really measure the stability of the prediction according to the learning set randomness. Therefore we will use in our analysis of the bias/variance profile of classification algorithms another, more adequate measure of this variability, namely the variance term introduced by Kohavi and Wolpert (see [KW96] and the appendix A):

$$\mathrm{var}_{KW}(\underline{a}) = \frac{1}{2}(1 - \sum_c P_{LS}(c|\underline{a})^2). \tag{3.32}$$

This measure is actually equivalent to an entropy measure[7] for the categorical variable $f_{LS}(\underline{a})$. It is non-negative and equal to zero if and only if $P_{LS}(f_{LS}^{MAJ}(\underline{a})|\underline{a}) = 1$, and maximal when $P_{LS}(c|\underline{a}) = \frac{1}{m}$, where m is the number of classes. However, contrary to var_T, a decrease of the value of this measure, for fixed majority vote classifier, should not be interpreted as a decrease of the average error rate.

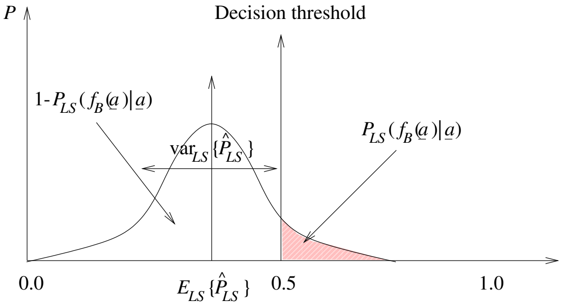
3.2.3 Variance of probability estimates

Many classification algorithms work by first computing an estimate $\hat{P}_{LS}(c|\underline{a})$ of the conditional class probability distribution at $\underline{a}$ and then deriving a classification model by:

$$f_{LS}(\underline{a}) = \arg\max_c \hat{P}_{LS}(c|\underline{a}). \tag{3.33}$$

[6] Actually, in none of our experiments, we have found a negative global variance term.

[7] Actually, it is proportional to the so-called Gini index (or quadratic entropy) of the discrete random variable $f_{LS}(\underline{a})$ (see for example [Weh96] for a discussion of entropy measures).



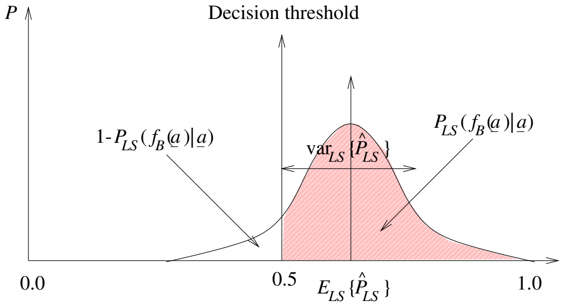


Figure 3.6: distribution of $\hat{P}_{LS}(f_B(\underline{a})|\underline{a})$, left when $E_{LS}\{\hat{P}_{LS}(f_B(\underline{a})|\underline{a})\}$ is lower than 0.5, right when $E_{LS}\{\hat{P}_{LS}(f_B(\underline{a})|\underline{a})\}$ is greater than 0.5

Obviously, if these probability estimates have a small variance and bias the corresponding classifier is stable with respect to random variations of the learning set and close to the Bayes classifier. Thus, a complementary approach to study bias and variance of a classification algorithm is to connect, in a quantitative way, this probability estimate variance, i.e.

$$\text{var}_{LS}\{\hat{P}_{LS}(c|\underline{a})\} = E_{LS}\{(\hat{P}_{LS}(c|\underline{a}) - E_{LS}\{\hat{P}_{LS}(c|\underline{a})\})^2\}, \tag{3.34}$$

to the classification error of the resulting classification rule (3.33).

We reproduce here the derivations carried out by Friedman [Fri97], in the particular case of a two-class problem. Let us denote by $f_B(\underline{a})$ the most probable class at $\underline{a}$ (i.e. the class chosen by the Bayes classifier) and by $f_{\neg B}(\underline{a})$ the other class. The mean classification error at $\underline{a}$ may then be written (with some elementary manipulations) as:

$$\begin{aligned} E_{LS}\{Err(f_{LS}(\underline{a}))\} &= 1 - P_{LS}(f_B(\underline{a})|\underline{a}) \cdot P(f_B(\underline{a})|\underline{a}) - P_{LS}(f_{\neg B}(\underline{a})|\underline{a}) \cdot P(f_{\neg B}(\underline{a})|\underline{a}) \\ &= 1 - P(f_B(\underline{a})|\underline{a}) + (1 - P_{LS}(f_B(\underline{a})|\underline{a})) \cdot (2P(f_B(\underline{a})|\underline{a}) - 1). \end{aligned} \tag{3.35}$$

Notice that in (3.35) $P_{LS}(f_B(\underline{a})|\underline{a})$ denotes the probability that a model induced from a random learning set outputs the class $f_B(\underline{a})$, and, according to (3.33), this is the case when $\hat{P}_{LS}(f_B(\underline{a})|\underline{a})$ is greater than 0.5.

In order to relate this quantity to the probability estimate variance let us have a look at the graphs of Figure 3.6 which represent two hypothetical distributions of $\hat{P}_{LS}(f_B(\underline{a})|\underline{a})$. Note that on these figures the probability $P_{LS}(f_B(\underline{a})|\underline{a})$ is represented by the area at the right of the decision threshold 0.5. Hence, $1 - P_{LS}(f_B(\underline{a})|\underline{a})$ is the area to the left of 0.5. Assuming a Gaussian distribution[8] for $\hat{P}_{LS}(f_B(\underline{a})|\underline{a})$, this area is computed by:

$$1 - P_{LS}(f_B(\underline{a})|\underline{a}) = \Phi(\frac{E_{LS}\{\hat{P}_{LS}(f_B(\underline{a})|\underline{a})\} - 0.5}{\text{var}_{LS}\{\hat{P}_{LS}(f_B(\underline{a})|\underline{a})\}}) \tag{3.36}$$

where $\Phi(z)$ is the upper tail of the standard normal distribution (see Figure 3.7).

So the local error $Err(f_{LS}(\underline{a}))$ is linked to the regression variance of $\hat{P}_{LS}(f_B(\underline{a})|\underline{a})$ by the following expression:

$$E_{LS}\{Err(f_{LS}(\underline{a}))\} = 1 - P(f_B(\underline{a})|\underline{a}) + \Phi(\frac{E_{LS}\{\hat{P}_{LS}(f_B(\underline{a})|\underline{a})\} - 0.5}{\text{var}_{LS}\{\hat{P}_{LS}(f_B(\underline{a})|\underline{a})\}}) \cdot (2.P(f_B(\underline{a})|\underline{a}) - 1). \tag{3.37}$$

Their interaction is easy to understand from Figure 3.6 and the form of $\Phi(z)$:

[8]This Gaussian assumption is certainly not satisfied in all cases (e.g. $\hat{P}_{LS}(c|\underline{a})$ estimated by a tree with only pure terminal nodes will take only two values, 0 or 1), but the main conclusions drawn upon this assumption still remain valid.

$$\Phi(z) = \frac{1}{\sqrt{2\pi}} \int_{z}^{+\infty} e^{-\frac{u^2}{2}} du.$$

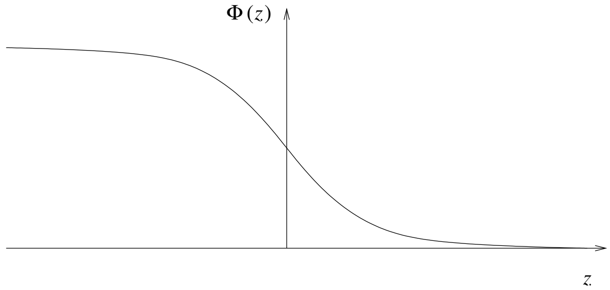


Figure 3.7: Upper tail area of the standard normal distribution

- when the average probability estimate of the majority class (i.e. $E_{LS}\{\hat{P}_{LS}(f_B(\underline{a})|\underline{a})\}$) is lower than 0.5, an increase of variance will increase $P_{LS}(f_B(\underline{a})|\underline{a})$ (at the left part of Figure 3.6) and hence yield a decrease of the classification error;

- on the other hand, when $E_{LS}\{\hat{P}_{LS}(f_B(\underline{a})|\underline{a})\}$ is greater than 0.5, an increase of variance will increase $1 - P_{LS}(f_B(\underline{a})|\underline{a})$ (at the right part Figure 3.6) and hence yield an increase of the classification error.

Actually, $E_{LS}\{\hat{P}_{LS}(f_B(\underline{a})|\underline{a})\}$ greater than 0.5 corresponds to the case $f_{LS}^{MAJ}(\underline{a}) = f_B(\underline{a})$, i.e. $\text{bias}_C(\underline{a}) = 0$ and $E_{LS}\{\hat{P}_{LS}(f_B(\underline{a})|\underline{a})\}$ lower than 0.5 to the case $f_{LS}^{MAJ}(\underline{a}) \neq f_B(\underline{a})$, i.e. $\text{bias}_C(\underline{a}) = 1$. The conclusions are thus similar to what we found in our illustrative problem above: *in classification, variance is beneficial for biased points and detrimental for unbiased ones.*

Another conclusion can be drawn from (3.37): whatever the regression bias on the approximation of $\hat{P}_{LS}(f_B(\underline{a})|\underline{a})$, the classification error can be driven to its minimum value by reducing solely the variance, under the assumption that $E_{LS}\{\hat{P}_{LS}(f_B(\underline{a})|\underline{a})\}$ is greater than 0.5. This means that perfect classification rules can be induced from very bad (rather biased, but of small variance) probability estimators.

In experiments of subsequent chapters, we will often complete our analysis of bias/variance profile of classification algorithms by the regression decomposition of these probability estimates.

3.2.4 Bias and variance estimation (bootstrap method)

Bias and variance definitions make intensive use of the knowledge of the distribution of learning samples, $P(\underline{A}, Y)$. In the machine learning problem definition however, the only knowledge we have about this distribution is a data set of randomly drawn samples. So, in practice, true values of bias/variance terms have to be exchanged for some estimates. To this end, and assuming we have enough data, the available dataset is divided into two disjoint parts, PS (P for "pool") and TS (T for "test"), used in the following way:

- As we do not have access to an infinite source of learning sets, PS is used to approximate the learning set generation mechanism. A good candidate for this is to replace sampling from $P(\underline{A}, Y)$ by sampling with replacement from PS. This is called "bootstrap" sampling in the statistical literature [ET93] and the idea behind it is to use the empirical distribution of the finite sample as an approximation of the true distribution. The bigger is PS with respect to the learning set size, the better will be the approximation. Thus, starting from a set $PS = \{(\underline{a}_1, y_1), \ldots, (\underline{a}_{M_p}, y_{M_p})\}$, the estimation procedure for the expectation of a function $g(LS)$ is the following: Draw T learning sets of size N (with $N \leq M_p$)with replacement from PS, $\{ls_1, \ldots, ls_T\}$, compute $g(ls_i)$ for $i = 1, \ldots, T$ and average:

$$E_{LS}\{g(LS)\} \simeq \frac{1}{T}\sum_{i=1}^{T} g(ls_i). \tag{3.38}$$

For example, to estimate the average regression model, build a model f_{ls_i} from each learning set ls_i and compute:

$$\overline{f}_{LS}(\underline{a}) \simeq \frac{1}{T}\sum_{i=1}^{T} f_{ls_i}(\underline{a}). \tag{3.39}$$

Similarly, the probability $P_{LS}(c|\underline{a})$ that a model induced from a random learning set gives the class c at point $\underline{a}$ is estimated by:

$$P_{LS}(c|\underline{a}) \simeq \frac{1}{T}\sum_{i=1}^{T} 1(f_{ls_i}(\underline{a}) = c), \tag{3.40}$$

which yields the majority vote classifier:

$$f_{LS}^{MAJ}(\underline{a}) \simeq \arg\max_{c} \sum_{i=1}^{T} 1(f_{ls_i}(\underline{a}) = c). \tag{3.41}$$

- The set TS, which is independent of PS, is used as a test sample to estimate errors and bias and variance terms. Denoting by $\{(\underline{a}_1, y_1), \ldots, (\underline{a}_{M_v}, y_{M_v})\}$ this set, the expectation of a function $g(\underline{a}, y)$ is approximated by:

$$E_{\underline{A},Y}\{g(\underline{A}, Y)\} \simeq \frac{1}{M_v}\sum_{i=1}^{M_v} g(\underline{a}_i, y_i). \tag{3.42}$$

For example, the mean error of a model f is estimated by:

$$Err(f) = E_{\underline{A},Y}\{L(f(\underline{A}), Y)\} \simeq \frac{1}{M_v}\sum_{i=1}^{M_v} L(f(\underline{a}_i), y_i). \tag{3.43}$$

However, some of the previously defined terms are difficult to estimate without any knowledge of the problem other than a dataset TS. Indeed, the estimation of the Bayes model $f_B(\underline{a})$ (or more generally of the distribution $P(Y|\underline{A})$) from data is nothing but the goal of supervised learning and if we had a way to estimate it from data, we would not need to bother about estimating bias and variance to study machine learning algorithms. So, the bias and variance terms which make explicit use of these latter will be mostly impossible to estimate for real datasets (from which we do not have any knowledge of the underlying distribution). For example, the regression noise and bias terms (3.16 - 3.17) depend on the Bayes model and thus are impossible to estimate separately only from data. The solution adopted by most authors to circumvent this problem is to assume that there is no noise, i.e. $f_B(a_i) = y_i, i = 1, \ldots, M_v$, and to estimate the bias term from TS by:[9]

$$E_{\underline{A}}\{\text{bias}_R^2(\underline{A})\} = E_{\underline{A}}\{(f_B(\underline{A}) - \overline{f}_{LS}(\underline{A}))^2\} \simeq \frac{1}{M_v}\sum_{i=1}^{M_v}(y_i - \overline{f}_{LS}(\underline{a}_i))^2, \tag{3.44}$$

[9]Note that only average values (over the input space) and not local values of bias or variance terms may be estimated as a finite sample can not cover the whole input space.

using in this last expression the estimation of the average model given by (3.39). If it happens that actually there is noise, the last expression is an estimation of the error of the average model and hence, this amounts at estimating the sum of the residual error and bias terms. The fact that we can not distinguish errors which are due to noise or bias is not very dramatic, since in our experiments we will be mainly interested in studying relative variations of bias and variance more than their absolute values and the part of (3.44) which is due to residual error is constant. The regression variance may then be estimated by the following expression:

$$E_{\underline{A}}\{\mathrm{var}_R(\underline{A})\} \simeq \frac{1}{M_v}\sum_{i=1}^{M_v}\frac{1}{T}\sum_{i=1}^{T}(f_{ls_i}(\underline{a}_i) - \overline{f}_{LS}(\underline{a}_i))^2, \tag{3.45}$$

or equivalently by the difference between the mean error and the sum of noise and bias as estimated by (3.44).

In classification, the estimate of the classification bias given by Tibshirani can also be obtained by assuming no noise (i.e. if $P(c|\underline{a}) = 1(c = f_B(\underline{a}))$) by:

$$E_{\underline{A}}\{\mathrm{bias}_T(\underline{A})\} \simeq \frac{1}{M_v}\sum_{i=1}^{M_v} 1(y_i \neq f_{LS}^{MAJ}(\underline{a}_i)), \tag{3.46}$$

where the majority vote classifier is estimated according to (3.41). Again, if there is noise, expression (3.46) is actually an estimation of the mean error of the majority vote classifier which is also the additive combination of residual error σ_C and Tibshirani's bias term bias_T. According to its definition, Tibshirani's variance is estimated by the difference between the mean error and expression (3.46).

The preceding procedure may yield very unstable estimators (suffering of high variance) especially if the available amount of data is not sufficiently large. Several techniques are possible to stabilize the estimations. For example, a simple method is to use several random divisions of the datasets into PS and TS and to average the estimates found for each separation. More complex estimates may be constructed to further reduce the variance of the estimation (see for example [Tib96, Wol97, Web00]). However in our study, we will try to choose sufficiently large data set so that avoidance of such complex estimation techniques would not be too damaging.

3.3 Empirical study of bias and variance

Bias and variance both contribute to the error and we have explained in the beginning of this chapter that they are affected in an opposite direction by variations in the complexity of the model. In addition to the model complexity, it is interesting to find out and demonstrate how the parameters of the learning problem and algorithm influence bias and variance. To evaluate these influences, some experiments are carried out in this section on a regression and a classification problem. Both are derived from an artificial problem introduced in [Fri91] which has 10 input attributes $(A_1, \ldots, A_{10})$ all independently and uniformly distributed in $[0, 1]$. The regression output variable Y is obtained as follows:

$$Y = f_B(\underline{A}) + \varepsilon, \tag{3.47}$$

with

$$f_B(\underline{A}) = 10sin(\pi A_1 A_2) + 20(A_3 - 0.5)^2 + 10A_4 + 5A_5, \tag{3.48}$$

and ε is a noise term independent of $\underline{A}$ and distributed according to a Gaussian distribution of zero mean and unit variance. Thus, Y is actually independent of the last five input

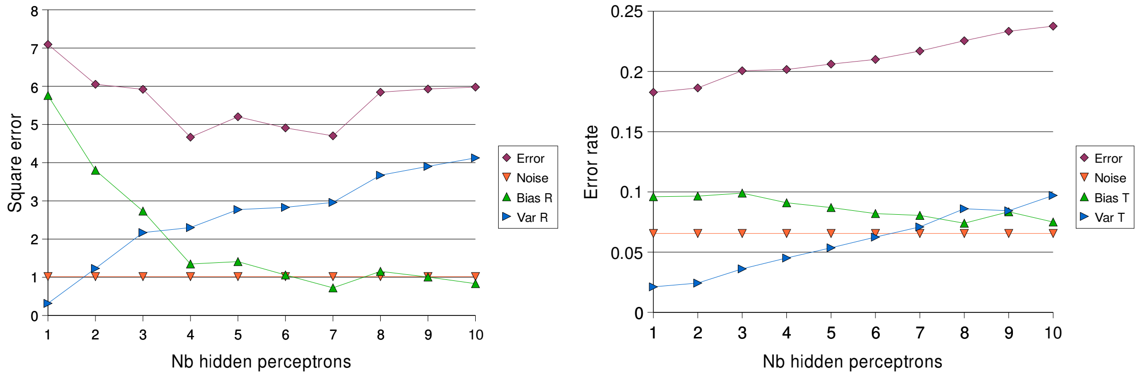


Figure 3.8: Evolution of regression (left) and classification (right) bias and variance with respect to the number of perceptrons in the hidden layer of a neural network

variables which are included to make the problem more representative of real situations where in general some variables are irrelevant. A classification problem is defined from this regression problem by defining the following binary classification output:

$$C = \begin{cases} c_1 & \text{if } Y < 14.0 \\ c_2 & \text{if } Y \geq 14.0 \end{cases} \tag{3.49}$$

The threshold on Y has been chosen in order to balance the two classes. Bias and variance decompositions are estimated using the protocol given in the previous section. 8000 cases are forming the pool sample and the test sample contains 2000 cases. When not stated otherwise, the learning sample size N is equal to 300 and 50 of samples are drawn from PS to estimate expectations according to LS.

3.3.1 Hypothesis space size

Bias mainly depends on the model complexity (the size of the hypothesis space) but also on the amount of optimization carried out by the machine learning method (search procedure): the larger the number of candidate models implicitly visited during the search, the smaller bias. On the other hand, the more search the learning algorithm carries out, the more dependent is the model on the learning sample and hence the higher will be variance.

Single hidden layer perceptrons

Figure 3.8 corresponds to a sequence of simulations carried out on the above described regression and classification problems with a neural network package using the Levenberg-Marquardt optimization method [Muñ96]. Notice that this algorithm starts from initial weights which values are chosen according to a random number generator, thus introducing some randomness in the algorithm in addition to the variance related to the learning sample randomness itself. The maximum number of iterations and stopping criteria where chosen in these simulations so as to ensure good convergence to the local minimum of the empirical risk.

The horizontal axis in these figures corresponds to the number of neurons in the hidden layer, i.e. the parameter which controls here the size of the hypothesis space. Notice that the number of adjustable parameters of these neural networks varies between about 10 (a single hidden neuron) and 100 (ten hidden neurons).

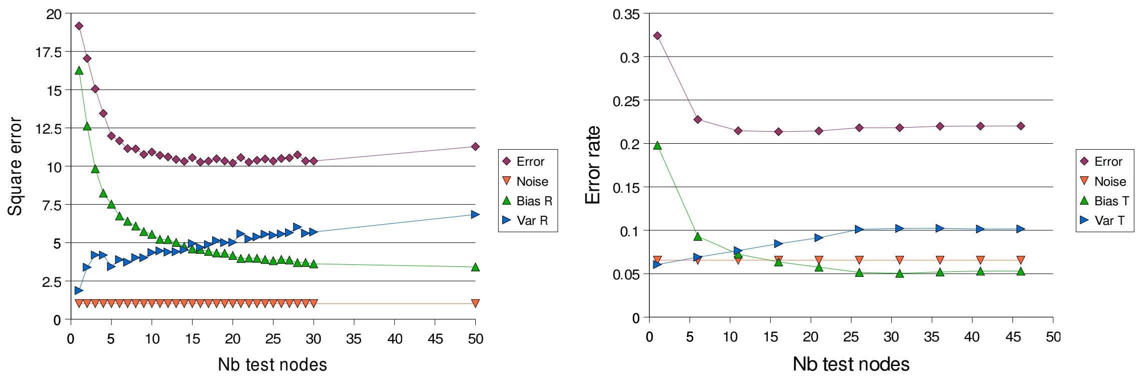


Figure 3.9: Evolution of regression (left) and classification (right) bias and variance with respect to the number of tests in a decision/regression tree

The left graph shows the results corresponding to the regression problem and the right one corresponds to classification (in the latter case, the neural network has two neurons in the output layer, trained on the indicator variables of the two classes; the corresponding classification rule consist of choosing the class corresponding to the largest output layer activation). The yellow curves correspond to the residual error which, obviously, does not depend on the number of neurons in the hidden layer. The three other curves show the variation of bias, variance, and average error (the latter curve is obtained as the sum of the three other ones).

In the regression case, the curves clearly show the expected evolution of bias, which decreases with increasing complexity, of variance, which increases, and, of the mean square error which first decreases and then increases. The curves thus are consistent with the theory of the bias/variance compromise, which in this particular problem corresponds to 4 hidden neurons.

Considering the curves corresponding to the classification case, where Tibshirani's bias and variance terms are used, similar behavior can be observed, with the only difference that in this case the optimal complexity corresponds to one single hidden neuron (which actually corresponds to a linear decision boundary).

Decision/regression trees

Figure 3.9 shows similar curves, where the complexity parameter corresponds to the number of terminal nodes of the tree[10]. The same protocol and learning samples than in the preceding simulation have been used to yield these graphics. Again, as predicted by the theory, variance increases monotonically with complexity while bias decreases, yield in terms of average error and optimal complexity, which is slightly large for regression than for classification.

Comparing the average error rates of Figs. 3.9 and 3.8 we notice, in passing, that the neural networks significantly outperform decision and regression trees on this particular problem.

3.3.2 The learning problem: Bayes model and noise

Bias and variance are also related to the complexity of the Bayes model $f_B(\underline{a})$. At fixed complexity of the model, the bias increases with the complexity of f_B, because the absolute

[10]The trees have been built according to the best first procedure which is described in Chapter 5.

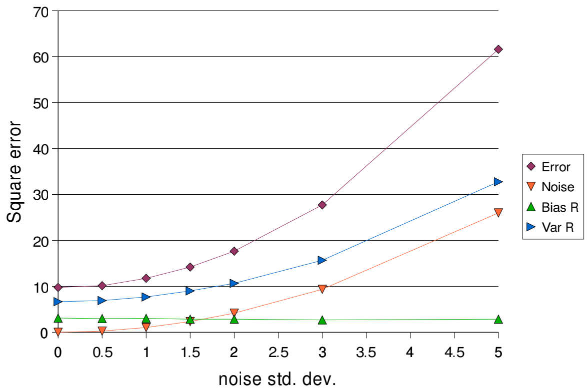


Figure 3.10: Evolution of bias and variance of regression trees with respect to the standard deviation of the noise

value of the bias depends on the way the complexity of the model matches the complexity of the function. The variance on the other hand remains mainly unaffected by the complexity of the Bayes model[11]. The best tradeoff between bias and variance with respect to the model complexity is thus - unfortunately, but not surprisingly - problem dependent. Note that the selection of the optimal value of complexity, in a problem dependent way, is discussed in the next chapter in the context of several learning algorithms.

Another problem dependent parameter that influences the tradeoff is the noise. In regression, bias which depends essentially only on the Bayes model and the average model should be unaffected by random additive noise. Variance however increases with the noise. This is confirmed by the results shown in Figure 3.10, corresponding to simulations with our synthetic problem where the noise level has been artificially increased progressively. The curves show the bias/variance profile for increasing value of the standard deviation of the noise ε in equation (3.47). The residual error is of course increasing. And indeed, while bias remains almost constant, variance is monotonically increasing. Note that, the fact that variance decreases with decreasing noise level must not be interpreted as "no noise = no variance"[12]. Besides, the variance of regression trees with zero noise remains quite important in Figure 3.10.

3.3.3 Learning sample size

Whatever the learning algorithm, we expect the variance to decrease when the learning sample size increases, since the more samples are gathered, the more stable should be the induced model.[13]

On the other hand, the evolution of bias with respect to the sample size is algorithm dependent. When the complexity of the model is fixed, bias is typically independent of the learning sample size. The left part of Figure 3.11 shows that this is indeed the case with linear regression for example. However, the right part of the same figure obtained with regression trees of fixed number of terminal nodes (7 here) shows that in this case, bias is a slightly increasing function of the learning sample size although the complexity is

[11] This statement may appear here as unjustified, but it is essentially supported by various theoretical considerations, in particular those provided in Section 3.5.

[12] Which is a statement found in various forms in the literature on automatic learning, e.g. in the form "Overfitting is due to noise...".

[13] Note that the "whatever" in this sentence, should be understood as "whatever reasonable", because it is of course possible to imagine learning algorithms which variance would increase with sample size.

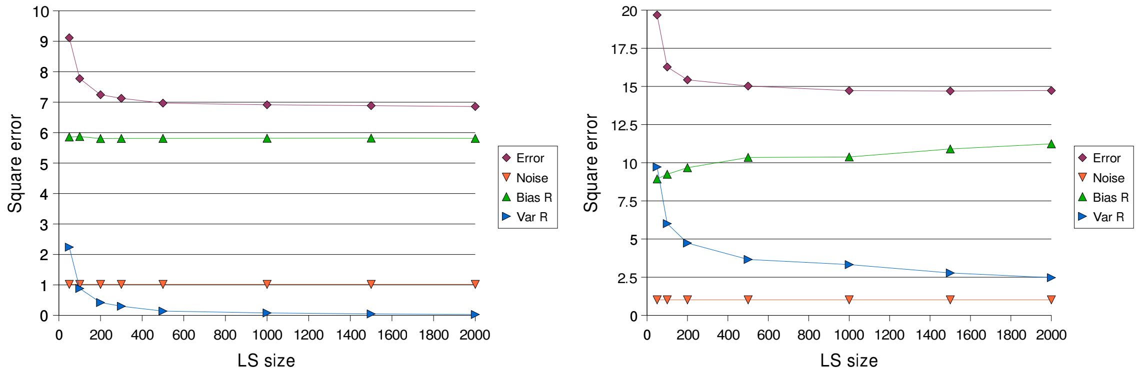


Figure 3.11: Influence of the learning set size at fixed complexity. Left, with linear regression, right, with regression trees of 7 terminal nodes

fixed. This counterintuitive result can nevertheless be explained. In the case of regression trees, the average model $\overline{f}_{LS}$ is impossible to represent with a tree of fixed complexity (the hypothesis space of fixed complexity trees is not closed with respect to the aggregation operator used). Indeed, the average model corresponding to a finite number of trees of small complexity can be represented only by a tree of much larger complexity. And, in the limit, the average of an infinite number of trees can actually represent a smooth function, which is not at all representable by a finite tree, which can only represent piecewise constant functions. Thus, when the learning sample size is small, the variance is high and trees induced from different learning sets are very diverse. On the other hand, when the learning sample size increases, trees becomes closer and closer to each other and the average model is less and less flexible, hence bias increases (slightly in Figure 3.11). Notice however that, because variance decreases much more significantly than bias increases this figure suggests that for increasing learning sample size the optimal complexity will also increase, which is indeed the case (see chapter 5).

In several algorithms however, the model complexity or the search algorithm depends indirectly[14] on the size of the learning set and hence the bias should decrease in these cases with the learning sample size. For example, the size of fully grown decision/regression trees (i.e. which are build so as to minimize the empirical risk) is automatically adapted to the learning sample size and hence their bias is actually a decreasing function of the learning set size. Figure 3.12 shows us that under these conditions bias and variance indeed decrease as the sample size goes up, left with regression trees, right with decision trees.

3.3.4 Learning algorithm

The bias/variance tradeoff is also different from one learning algorithm to another. Some algorithms intrinsically present high variance but low bias and other algorithms present high bias but low variance. For example, linear models and Naive Bayes method because of their strong hypothesis often suffer from a high bias. On the other hand, because of their small number of parameters, their variance is small and, on some problems, they may therefore be competitive with more complex algorithms, even if their underlying hypothesis is clearly violated. While the bias of the nearest neighbor method (1-NN), neural networks and regression/decision trees generally are smaller, the increase in flexibility of their model is paid by an increase of the variance.

[14] Actually, those algorithms which aim at optimizing the bias/variance tradeoff by adjusting the model complexity to the problem specifics.

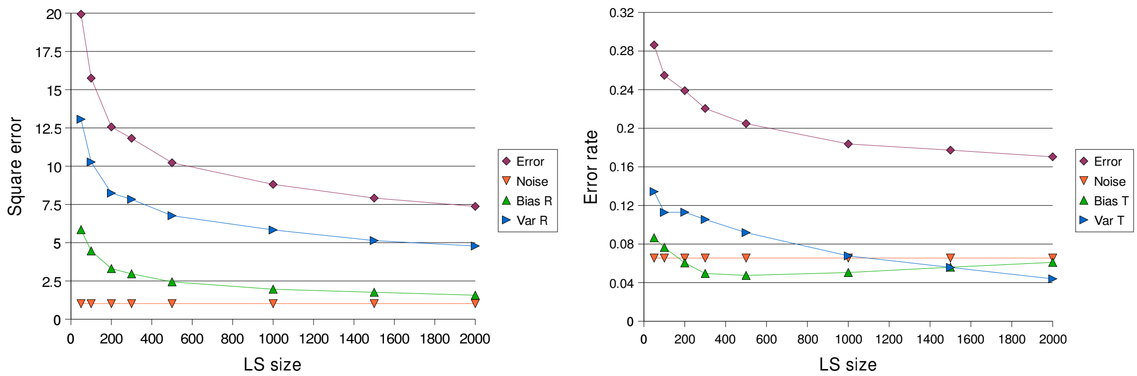


Figure 3.12: Influence of the learning set size at adjustable complexity. Left, with regression trees, right, with decision trees

Table 3.1: Bias/variance decompositions of several algorithms in regression

Method	Square error	σ_R^2	bias_R^2	var_R
Linear regression	7.0	1.0	5.8	0.2
k-NN (k=1)	15.4	1.0	4.0	10.4
k-NN (k=10)	8.5	1.0	6.2	1.3
MLP (10)	2.0	1.0	0.2	0.8
MLP (10-10)	4.6	1.0	0.4	3.2
Regression tree	10.2	1.0	2.5	6.7

The variance of decision trees will be studied in detail later but it is partly due to the discretization of numerical attributes and to the hierarchical nature of their induction.

The variance of 1-NN rule (which bias is negligible, for large sample sizes) is easier to understand. Since, the decision at a point of the input space depends on the closest sample in the learning set, a small perturbation of this sample may lead to a large change in this prediction. In the case of complex neural networks, there are generally several (possibly an infinite number of) possible parameterizations of the networks which allow to realize very small values of the empirical risk on the learning sample. So, according to the initial choice of parameters and the particular learning sample, the optimization procedure will be trapped into different local optima and hence the induced model will also be highly variable.

To illustrate this discussion, we have carried out a number of simulations on the regression problem in order to compare different variants of four basic learning algorithms. Table 3.1 provides results obtained with learning samples of size 500, the numbers being obtained by the same protocol as before (50 random learning samples of size 500 used to estimate expected values with respect to the LS variable, and a validation sample composed of 2000 objects, in order to compute averages over the input space). The table confirms the following general trends:

- **Linear regression:** variance is negligible, which is mainly due here to the fact the number of parameters of this method (11) is very small with respect to the number of objects in the learning sample. The error of the linear regression is mainly due to its bias, which is a consequence of the non-linearity of the Bayes model corresponding to this regression problem.

- **k-Nearest-Neighbor:** in our experiments we have used the Euclidean distance in the space of 10 attributes scaled to make their variance equal to 1. Note that among the 10 input attributes their are actually 5 which are irrelevant, and this is known to be detrimental to the performance of k-NN. The table shows a very high variance when $k = 1$ and also a rather high bias. Increasing the value of k to 10 allows to reduce variance significantly, but at the price of increasing the bias by 50%. All in all the k-NN method is less accurate than linear regression in spite of the fact that it may in principle handle the non-linearity. Note that the rather high bias displayed here by the k-NN method is due to the fact that the average distance to the nearest neighbors is too high compared to the speed of variation of the Bayes model. This distance is a result of the combined effect of high dimensionality in the input space and small sample size.

- **Multilayer perceptrons:** this method provides overall the best results in this problem, having both negligible bias and small variance. The two simulations correspond respectively to one single hidden layer with ten neurons (denoted (10) in the table) and two hidden layers with ten neurons each (noted (10-10) in the table). The variance of the more complex structure is significantly more important.

- **Regression trees:** in this table the trees are fully grown. Their bias is small but not negligible, and variance is so high that in the end the method is less accurate than linear regression and the 10-NN method. Note that the bias is due here to the fact that the smooth function f_B is difficult to approximate with a simple staircase model, whose complexity is moreover limited by the sample size (to about 20 terminal nodes, on average).

3.4 Two other kinds of variances

The variance expressed in our decompositions and studied in the preceding experiments:

$$E_{\underline{A}}\{\mathrm{var}_{LS}\{f_{LS}(\underline{A})\}\} \tag{3.50}$$

is the variance of the predictions given by the model, averaged over the whole input space. It translates the effect of the variability of the model on the mean square error or on the classification error. Nevertheless, the variability of the prediction comes with (and is correlated to) two other types of variances: the variance of the error realized by a model and the variance of the parameters of this model.

3.4.1 Error variance

The variance of the error of a model is expressed by:

$$\mathrm{var}_{LS}\{Err(f_{LS})\}. \tag{3.51}$$

If this variance is high, the accuracy of a model produced from a learning set is very variable from one learning set to another. At the time of writing this thesis, and to our best knowledge, no detailed study of the link between this variance and the variance of the predictions has been carried out. In what follows we will merely provide a very short discussion of it.

Intuitively, if error variance is high at a point $\underline{a}$, then the variance of the predictions at this point must also be high. In other words, small prediction variance implies small error variance. On the other hand, the opposite is not necessarily true: a high prediction variance does not necessarily translate into a high error variance. This is due to the fact that errors at different points in the input space can be correlated in different ways, and it is possible that the increase of error in one region is cancelled by a decrease in another region. Thus errors may be rather stable in spite of a significant prediction variance.

In this dissertation, we will focus on the reduction of prediction variance, which will as a by-product also reduce error variance, should it be high.

3.4.2 Parameter variance

If a model is parameterized by a parameter α (e.g. a weight in a neural network or a discretization threshold in a decision tree), the learned value of this parameter α_{LS} depends on the learning set and hence is also a random variable. We can thus compute its variance with respect to the learning set randomness:

$$\mathrm{var}_{LS}\{\alpha_{LS}\}. \tag{3.52}$$

If this variance is high, the value of the parameter is highly unstable from one learning set to another. When the model is intrinsically interpretable, its practical interpretability is limited if the parameter variance is high and so this variance should be reduced as well. Again, there is not a bi-directional quantitative relationship between parameter variance and prediction variance. Indeed, high prediction variance should come with a high parameter variance but the opposite is not true: even if the prediction variance is not high, the parameter variance may be important. It depends on the way these parameters intervene to make up the final prediction.

We will see in subsequent chapters that the variance of the parameters of decision trees (discretization thresholds and attribute choice at interior nodes of the tree) turns out to be very high, which makes their interpretation questionable. Thus, we will study in detail this variance in Chapter 5 and investigate several techniques liable to reduce it in order to improve interpretability.

3.5 Analytical results related to bias and variance

Bias/variance decompositions split the error into two terms whose interpretation gives some feeling about the way error evolves with respect to various parameters. The empirical study of the previous section gives some clues about the qualitative behavior of bias and variance with respect to these parameters but what we can quantitatively expect is still an open question. We should add that, although the main trends that have been illustrated hold in a large number of situations, there are also a number of exceptions[15].

In some simple cases, it is possible to express analytically bias or variance.

3.5.1 Linear models

Let us assume a regression problem where $Y = f_B(\underline{A}) + \varepsilon$ with $E_\varepsilon\{\varepsilon\} = 0$ and $\mathrm{var}_\varepsilon\{\varepsilon\} = \sigma^2$.

Let us further assume that $f_B(\underline{A})$ is a linear function of the attributes and that we use a least squares multiple linear regression method to approximate this function from a learning sample. Then one can show (see [HTF01]) that the following two conditions are satisfied:

$$E_{\underline{A}}\{\mathrm{bias}_R^2(\underline{A})\} = 0, \tag{3.53}$$

and

$$E_{\underline{A}}\{\mathrm{var}_R(\underline{A})\} = \frac{n}{N}\sigma^2, \tag{3.54}$$

[15] e.g. it is possible to find situations where the increase of the number of parameters of a model leads actually to a decrease of variance, rather than an increase.

where n is the number of attributes and N the sample size.

Thus variance is proportional to the dimensionality of the input space and inversely proportional to the sample size. More generally, even if the Bayes model is not linear, the variance of linear models is increasing with the number of attributes (and hence of weights) and a decreasing function of the learning sample size (as confirmed by Figure 3.11 p. 47).

Notice also the fact that under "no noise" conditions variance vanishes, which led some people to extrapolate that variance is *due* to noise in general (which is not true). Indeed, even for linear regression, variance may be observed under no-noise conditions if the Bayes model is not linear.

3.5.2 K-NN

If we further assume input attribute values of the learning cases are fixed and only their output values are changing from one learning set to another, the variance of the k-NN may be approximated by (see also [HTF01]):

$$E_{\underline{A}}\{\mathrm{var}_R(\underline{A})\} = \frac{\sigma^2}{k}, \tag{3.55}$$

showing that variance is a decreasing function of the number of neighbors. Again, variance is increasing with the noise level.

3.5.3 Single hidden layer perceptrons

Derivation of analytical formulas for bias and variance is possible only in very simple cases, such as those just discussed. For example, in more general situations, like in the case of neural networks or decision trees, no such formulas have been (and possibly could be) derived.

Nevertheless, sometimes it is possible to give at least analytical bounds on error. Barron [Bar94], for example, gives bounds on the mean square error for single hidden layer neural networks. Assuming there is no noise ($\sigma_R^2 = 0$), these bounds are:

$$E_{LS}\{Err(f_{LS})\} \leq O(\frac{C_{f_B}^2}{L}) + O(\frac{nL}{N}\log N), \tag{3.56}$$

where L is the number of units in the hidden layer and C_{f_B} is a measure of the complexity of the Bayes model based on its Fourier representation. This decomposition is similar to a bias/variance decomposition. Actually, the first contribution is a bound on the error of the best neural network of this complexity, denoted by f_L^*, with respect to the Bayes model:

$$E_{\underline{A}}\{(f_B(\underline{A}) - f_L^*(\underline{A}))^2\}. \tag{3.57}$$

This latter error is called *approximation error* by Barron.

The second term in (3.56) is actually a bound on the average error of a learned neural network with respect to f_L^*:

$$E_{LS,\underline{A}}\{(f_{LS}(\underline{A}) - f_L^*(\underline{A}))^2\}. \tag{3.58}$$

This term is called *estimation error* by Barron and the bound is obtained under the assumption that the neural network optimization method is indeed able to find the set of parameters minimizing the empirical risk.

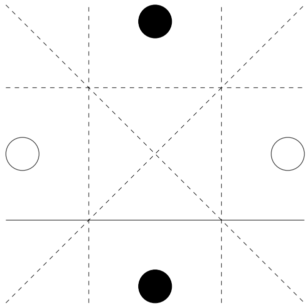


Figure 3.13: a configuration of four points that can not be shattered by a single line

Although this decomposition does not correspond exactly to the usual decomposition, these two terms may be assimilated respectively to a bias and a variance term. An interpretation of these bounds confirms our intuitive and empirical analyses. Indeed, the bound on bias is an increasing function of the Bayes model complexity and decreases with the complexity of the model, L. On the other hand, the bound on variance increases with the model complexity and the number of input attributes, while it decreases with the learning sample size. So, these bounds give an analytical justification of the form of the curves obtained in the previous section.

3.5.4 Statistical learning theory

When the complexity of the model increases, the learning algorithm is able to realize lower and lower errors on the learning sample (empirical risk) even if the Bayes model is very complex, and simultaneously approximation error and bias are decreasing. On the other hand, the estimation error of the learned model and the variance of the method will increase. Thus, the empirical risk is correlated with bias and approximation error, and the difference between the generalization error $Err(f_{ls})$ and the empirical risk $\hat{Err}(f_{ls}, ls)$ is a quantity related to variance.

While several authors have given bounds on this difference, the statistical learning theory of Vapnik [Vap95] certainly gives rise to the most general bounds. In classification, these bounds are based on a general measure of the representation capacity of a set of functions which is called the VC-dimension. The VC-dimension of a hypothesis space $\mathcal{H}$ of binary-valued functions is the largest number of points such that there exists a configuration of this number of points that can be *shattered* by members of $\mathcal{H}$. A set of points is said to be shattered by $\mathcal{H}$ if whatever the binary class assigned to each point, there exists a function in $\mathcal{H}$ that perfectly reproduces this classification. For example, in a bi-dimensional space, the VC-dimension of linear discriminant functions is three since it is impossible to find four points in the two-dimensional space which can be shattered by the set of linear discriminants (see Figure 3.13, for an example of 4 points which classification can not be obtained by a linear discriminant), while it is possible to shatter three points in general position. Notice that if the VC-dimension is infinite, then there exist samples of any size for which the empirical risk can be driven to zero within the hypothesis space whatever the chosen labels. The main contribution of Vapnik in automatic learning theory was to show that the empirical risk minimization principle in automatic learning is justifiable only if the VC-dimension of the hypothesis space is finite.[16]

[16]Infinite VC-dimension means that the hypothesis space is so large that it becomes too "easy" to find apparently excellent hypotheses, for most learning problems and any sample size, so that the empirical risk minimization principle looses its discrimination power between hypotheses.

The other main contribution of Vapnik was to provide errors bounds in the case of finite samples, based on the VC-dimension. For example, he proves the following result for binary classification problems with misclassification error: for any function $h \in \mathcal{H}$, we have with probability at least $1 - \alpha$:

$$Err(h) \leq \hat{Err}(h, ls) + \varepsilon, \tag{3.59}$$

with

$$\varepsilon = 4\frac{d}{N}[\ln(\frac{2N}{d}) + 1] - \frac{1}{N}\ln(\frac{\alpha}{4}), \tag{3.60}$$

where d is the VC-dimension of the set of functions $\mathcal{H}$. This bound shows again that the difference between generalization error and empirical risk increases with the model complexity (by means of the VC-dimension) and decreases with the learning set size. Vapnik provides similar bounds for the regression case (see [Vap95]).

Because of their generality (the results are distribution independent), these bounds are often too loose to be of practical use from the quantitative point of view. Another drawback of this approach is the difficulty to compute the exact VC-dimension of many hypothesis spaces used in automatic learning. However, qualitatively, these theoretical results are very useful to understand and design learning algorithms. A practical application of these bounds will be described in the next chapter.

3.6 Discussion

In automatic learning, bias and variance both contribute to prediction error, whatever the learning problem, algorithm, and sample size. The error decomposition allows us to better understand the way an automatic learning algorithm will respond to changing conditions. It allows us also to compare different methods in terms of their weaknesses. This understanding can then be exploited in order to select methods in practice, to study their performances in research, and to guide us in order to find appropriate ways to improve automatic learning methods.

The empirical simulations provided in Section 3.3 aimed at providing a quantitative comparison of methods and of various effects. As far as this thesis is concerned, one of the main outcomes is the very high variance of decision and regression tree induction, which significantly decreases the attractiveness of this method in spite of its other intrinsic qualities (interpretability, flexibility, computational efficiency). It suggests that, in order to improve these methods it is necessary to find a way to reduce its variance, ideally without jeopardizing its other nice features. We believe that these observations justify our focus on variance in decision tree induction in this thesis, and our research for general methods able to reduce this variance in an effective way. This is the topic of the second part of this thesis, while in the third part we will concentrate on practical applications in the context of time series classification, where variance is even more important than in non-temporal problems. Before we switch to these considerations, the next chapter is dedicated to classical variance reduction techniques in supervised learning.