Bias-Variance trade-off characterization in a classification problem
What differences with regression ?

Yann-Ael LE BORGNE
yleborgn@ulb.ac.be
Machine Learning Group
Université Libre de Bruxelles - Belgium

Abstract

An important issue in machine learning theory is the so-called bias-variance trade-off, which shows that a model with a high degree of freedom has often poor generalization capabilities. In learning problems using the quadratic loss function, the well known noise-bias-variance decomposition of the mean squared error sheds light on the nature of the model expected error. This gives insights into regression problem modelling, where the quadratic loss function is particularly appropriate. However, in classification problems, results from the precited decomposition are unapplicable as the appropriate loss function is the zero-one loss function. Attempts to decompose this function into a sum of noise-bias-variance terms have been proposed during the period 1995-2000, until a very nice general framework was proposed in 2000 by Domingos. This report is an account of the general framework proposed by Domingos, in the light of the previous solutions that had been proposed. Two major interests of this theoretical account on bias-variance decomposition are: first, that the notion of bias needs to be redefined in classification problems and, second, that given appropriate definitions of noise, bias, and variance, it is possible to unify different decompositions (among which quadratic and zero-one) in a nice general theoretical framework.

Contents

1 General definitions and notations

$\mathcal{X}$: Input space, with elements x, referred to by the random variable $\mathbf{x}$.
$\mathcal{Y}$: Output space, with elements y, referred to by the random variable $\mathbf{y}$.
$\mathcal{S}$: a stochastic process over $\mathcal{X} * \mathcal{Y}$, entirely defined by an unknown joint probability $p(x, y)$.

D_N : A set of N samples from $\mathcal{S}$.
$\hat{y} = h(x, \alpha_N)$: A predictive model with parameters α_N, identified by a learning procedure from D_N, giving prediction $\hat{y}$ for an x.
$C(y, \hat{y})$: A loss function, defining the accuracy loss of predicting $\hat{y}$ when the actual value is y. $C(y, y) = 0$.

In the following, all notations involving y and their derivatives (y^*, $\hat{y}$, ...) depend on some implicit x, omitted to simplify the writing. Moreover, depending on the context, a variable name can sometimes refer to a parameter or to a random variable. Examples of such variables are D_N (and consequently α_N and $\hat{y}$), y and x. For the sake of clarity, random variable are in bold face to distinguish them from fixed variables.

2 Introduction

The goal of supervised learning is to study and develop predictive modelling methods that can, given a set of examples D_N from $\mathcal{X} * \mathcal{Y}$, predict what is the most likely y value given some x (to find a relation linking x to y). Different modelling methods exist, for example neural networks, decision trees, polynomial regressions, etc..., and for each modelling method, different levels of complexity can be chosen (number of degrees in a polynomial regression, number of units in a neural network). A model is defined by parameters, and a learning procedure fixes values for these parameters according to the set of examples D_N. The higher the number of parameters, the greater the flexibility of the model. Thus, a naive view of machine learning would be to always choose the most complex model, so that one is sure to be able to approximate the stochastic process $\mathcal{S}$ under study. However, the search space for parameters in complex models may so large that it is computationnaly impossible to find the right parameters, even with unlimited number of examples. Moreover, the number of examples is often fixed, and often inferior to a decent ratio with respect to the dimensionality of $\mathcal{X} * \mathcal{Y}$. The following study concerns the case where the number of examples is limited. Thus, a designer is virtually always confronted to the following dilemna : on one hand, if the model is too simple, it will give a poor approximation of the phenomenon (underfitting). On the other hand, if the model is too complex, it will be able to fit exactly the examples available, without finding

a consistent way of modelling $\mathcal{S}$ (overfitting).
In order to avoid the undefitting/overfitting effects, and thus to improve modelling methods, a better theoretical understanding of the learning process is needed. The study of the generalization error through the loss function is a fruitful way to get insight into machine learning problems. A loss function is used to estimate the 'cost' of predicting $\hat{y}$ when the true value is y. By averaging the loss function over $\mathcal{Y}$, $\mathcal{X}$, and all possible D_N, one gets the generalization error expected between a model and the phenomenon to model. This error depends clearly on the different factors of the modelling frame, namely the uncertainty linking y to x in $\mathcal{S}$, the complexity of the model, the learning procedure and the number of examples available.
For the quadratic loss function, used in regression problems, the well-known noise-bias-variance decomposition can nicely put into evidence the different aspects causing the generalization error. However, for misclassification loss functions, used in classification problems, the decomposition proves to be more complex, and has even not be fully resolved into its most general form. Some different approaches have been considered, and in 2000 Domingos proposed a unified decomposition for quadratic and misclassification loss functions.
The purpose of this report is to present the main different kinds of decompositions that have been so far proposed. Section 3 is a presentation of the generalization error and the quadratic loss function decomposition in regression problems. In section 4, a general frame for classification problems will be presented. Then follows section 5 that presents Friedman's analysis of two-class classification problems. Section 6 gives an example of a misleading decomposition. Finally, section 7 presents Domingos unified frame for decomposing loss functions, and corresponding loss function decompositions are presented.

3 Regression

3.1 Problem definition

$\mathcal{X}$ and $\mathcal{Y}$ are real-valued. $p(x, y)$ is a joint probability density function.
$\hat{y} = h(x, \alpha_N)$ is a real-valued function.
The inacurracy between $\hat{y}$ and y is assessed by the quadratic loss function $C(y, \hat{y}) = (y - \hat{y})^2$. It could be argued that a better choice be the absolute loss function, so that the penalty is linear with the distance, with $C(x) = |y - \hat{y}|$. This loss function is rarely used as minimizing it proves to be difficult as it is not derivable.
In the quadratic loss function, the penalty is not linear with the distance, and this loss function is therefore more sensitive to 'outliers'. Wether or not this behaviour is a disadvantage is however arguable, and this loss function is consequently very largely used in regression problems.

At one point x, the error depends on the model $\hat{y} = h(x, \alpha_N)$, which bases its prediction on a specific D_N, and on the actual y, which may vary from one test to another. D_N (and therefore $\hat{y}$) and y can vary, and we will therefore consider in this case the associated random variable $\mathbf{D_N}$ and $\mathbf{y}$. The general mean error between the model and $\mathcal{S}$ is given by the mean squared error $MSE(x)$, i.e. the average of $(y - \hat{y})^2$ over all possible y and D_N :

$$MSE(x) = E_{\mathbf{y},\mathbf{D_N}}[(\mathbf{y} - \hat{\mathbf{y}})^2]$$

The generalized error between the model and the stochastic process $\mathcal{S}$ is the expected value of $MSE(x)$ over $\mathcal{X}$:

$$MISE = E_{\mathbf{x}}[MSE(\mathbf{x})]$$

This error is referred to as the 'mean integrated squared error' for the quadratic loss function.

3.2 Decomposition of the mean squared error

This decomposition is obtained in two steps. The first one consists in injecting $E_{\mathbf{y}}[\mathbf{y}|x]$ into $MSE(x)$:

$$\begin{aligned}
MSE(x) &= E_{\mathbf{y},\mathbf{D_N}}[(\mathbf{y} - E[\mathbf{y}|x] + E[\mathbf{y}|x] - \hat{\mathbf{y}})^2] \\
&= E_{\mathbf{y},\mathbf{D_N}}[(\mathbf{y} - E[\mathbf{y}|x])^2] + E_{\mathbf{y},\mathbf{D_N}}[(E[\mathbf{y}|x] - \hat{\mathbf{y}})^2] \\
&\quad -2 * E_{\mathbf{y},\mathbf{D_N}}[(\mathbf{y} - E[\mathbf{y}|x])(E[\mathbf{y}|x] - \hat{\mathbf{y}})] \\
&= E_{\mathbf{y}}[(\mathbf{y} - E[\mathbf{y}|x])^2] + E_{\mathbf{y},\mathbf{D_N}}[(E[\mathbf{y}|x] - \hat{\mathbf{y}})^2] \\
&\quad -2 * E_{\mathbf{y}}[(\mathbf{y} - E[\mathbf{y}|x])] * E\mathbf{D_N}[(E[\mathbf{y}|x] - \hat{\mathbf{y}})] \\
&= E_{\mathbf{y}}[(\mathbf{y} - E[\mathbf{y}|x])^2] + E_{\mathbf{D_N}}[(E[\mathbf{y}|x] - \hat{\mathbf{y}})^2] \qquad (1)
\end{aligned}$$

The first term $E_{\mathbf{y}}[(\mathbf{y} - E_{\mathbf{y}}[\mathbf{y}|x])^2]$ depends only on $\mathcal{S}$ and characterizes the error coming from the approximation of $\mathcal{S}$ at x to the expected value $E_{\mathbf{y}}[\mathbf{y}|x]$. Why doing so ?
If the relation between y and x is sought, it means that one supposes a sort of deterministic relation $y^* = f(x)$ between these variables . This ideal deterministic relation is, however, often noisy because of noise when getting empirical measures, or because one lacks some variables x to more precisely isolate the relation between x and y. The measure y read is therefore the sum of y^* and some noise component $\mathbf{w}$ with $E_{\mathbf{w}}[\mathbf{w}] = 0$, and therefore the component $E_{\mathbf{y}}[(\mathbf{y} - E_{\mathbf{y}}[\mathbf{y}|x])^2]$ is the variance of this noise, $\sigma^2_{\mathbf{w}}$ (This only holds in the presence of gaussian noise, which is often a reasonable assumption).
In this scheme, the first part of this decomposition makes it possible to separate the MSE in an irreducible error component that only depends on the

noise of $\mathcal{S}$, and a reducible error component $E_{\mathbf{D_N}}[(E_{\mathbf{y}}[\mathbf{y}|x] - \hat{\mathbf{y}})^2]$ that only depends on the adequation of the model to the function $y^* = f(x)$.

The second term $E_{\mathbf{D_N}}[(y^* - \hat{\mathbf{y}})^2]$ can be further decomposed. In this form, it is the expected squared distance beween $\hat{\mathbf{y}}$ and the 'optimal' y^*. $\hat{\mathbf{y}}$ depends on the model chosen, the learning procedure, and a specific D_N. More insight into the MSE can be gained by considering the density $p(\hat{y}|x)$, irrespective of the optimal value y^*. If we suppose that this density is of a Gaussian sort, we can inject $\hat{y}^* = E_{\mathbf{D_N}}[\hat{\mathbf{y}}]$, the expected $\hat{y}$ over all possible training sets D_N, into the model error $E_{\mathbf{D_N}}[(y^* - \hat{\mathbf{y}})^2]$:

$$\begin{aligned} E_{\mathbf{D_N}}[(y^* - \hat{\mathbf{y}})^2] &= E_{\mathbf{D_N}}[(y^* - \hat{y}^* + \hat{y}^* - \hat{\mathbf{y}})^2] \\ &= E_{\mathbf{D_N}}[(y^* - \hat{y}^*)^2] + E_{\mathbf{D_N}}[\hat{y}^* - \hat{\mathbf{y}})^2] \\ &\qquad -2 * E_{\mathbf{D_N}}[(y^* - \hat{\mathbf{y}}) * (\hat{y}^* - \hat{\mathbf{y}})] \\ &= (y^* - \hat{y}^*)^2 + E_{\mathbf{D_N}}[\hat{y}^* - \hat{\mathbf{y}})^2] \end{aligned} \tag{2}$$

The first term $(y^* - \hat{y}^*)^2$ relates to the expected accuracy we can get from the chosen model $h(x, \alpha_{\mathbf{N}})$ for a given x, independent of a training set $\mathbf{D_N}$ and the noise $\mathbf{w}$. It therefore qualifies the adequation of the chosen model with respect to the phenomenon to model. This term is called the 'bias' of the model, sometimes referred to as "squared bias" as it is raised to the power two.
The second term $E_{\mathbf{D_N}}[\hat{y}^* - \hat{\mathbf{y}})^2]$ quantifies how much the predicted value of $h(x, \alpha_{\mathbf{N}})$ will vary around the expected prediction value $\hat{y}^*$ for different training set $\mathbf{D_N}$. It therefore quantifies the sensitivity of the model prediction given a training set. As this term consists of taking the order two moment of the random variable $\hat{\mathbf{y}}$, this term is referred to as the "variance" of the model.
Taking 1 and 2, we have :

$$MSE(x) = \sigma^2_{\mathbf{w}} + (y^* - \hat{y}^*)^2 + E_{\mathbf{D_N}}[\hat{y}^* - \hat{\mathbf{y}})^2]$$

3.3 Illustration of the decomposition

3.3.1 Uncertainty related to $\mathcal{S}$

Let us consider a phenomenon where the relation linking y to x is of a quadratic form $y = x^2 + w$, where w is the noise (with mean 0 and standard deviation $\sigma_x = 1$), and x takes values on the interval $[0; 5]$ (see figure 3.3.1).

At one x, y follows a probability density, which is here a gaussian with mean $y^* = E_{\mathbf{w}}[(x^2 + \mathbf{w})|x] = x^2$ and standard deviation σ_x. This density is illustrated on figure 3.3.1, for $x = 3$.

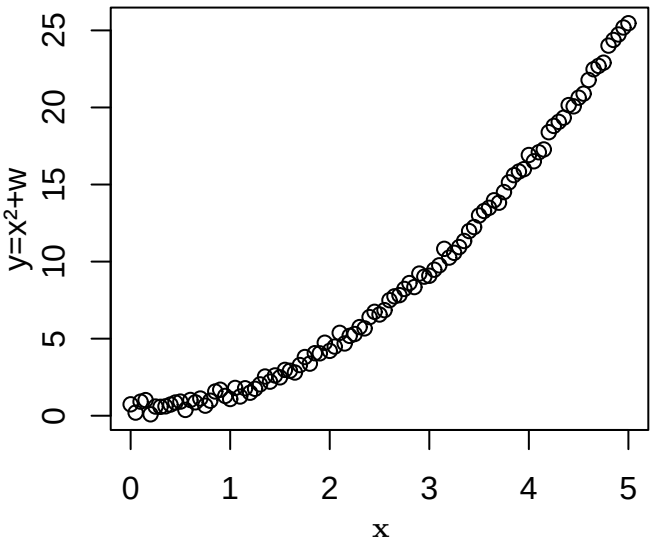

Figure 1: Samples of the non-deterministic relation $y = x^2 + w$

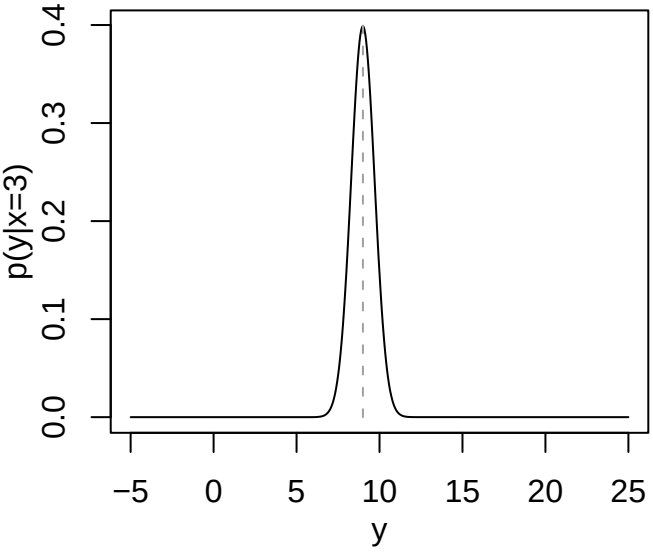

Figure 2: The probability density $p(y|x{=}3)$

3.3.2 Uncertainty related to the model

Suppose that the model chosen to approximate this relation is a linear model of the form $\hat{y} = \alpha_1 * x + \alpha_0$, and that a set D_N of $N = 100$ samples is available.

A linear regression using the least square technique allows the identification of the coefficients α_1 and α_0 from the training set D_N. On figure 3.3.2 are illustrated four different realization of this procedure for different training sets D_N.

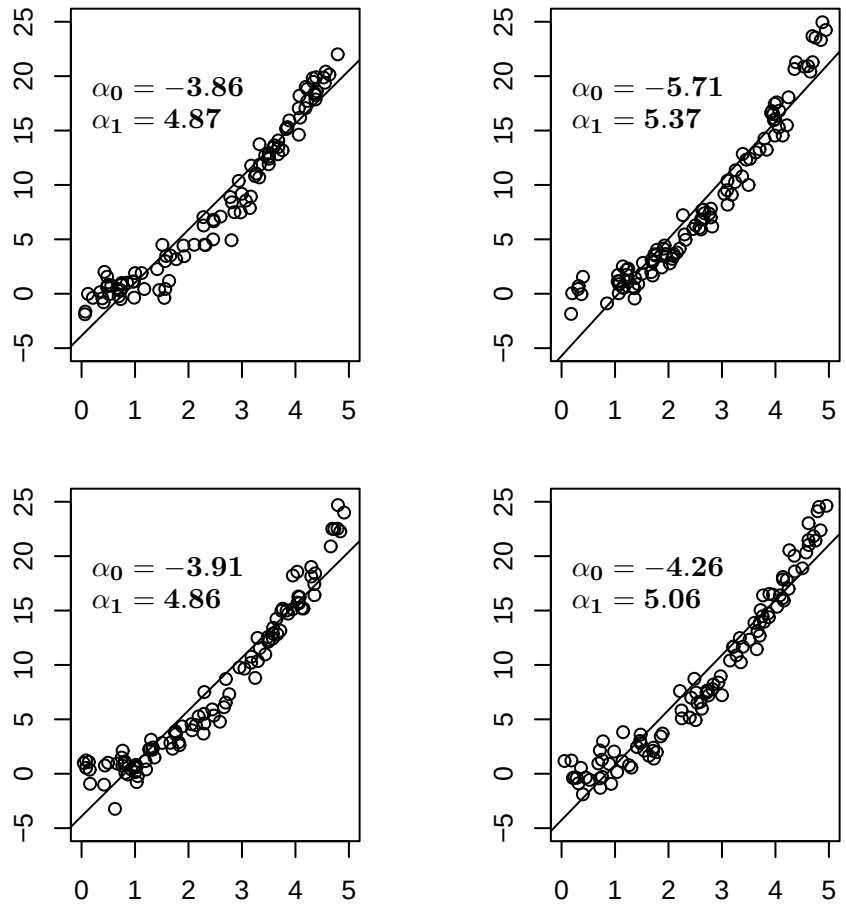

Figure 3: Four slightly different models due to variations in the training set

At one point x, depending on the training set D_N, the predicted value $\hat{y} = h(x, \alpha_N)$ will be different. On figure 3.3.2 is illustrated the probability density of $\hat{y}$ for $x = 3$ (obtained after five thousand realizations of the above procedure). The density has a gaussian shape, with mean value $E_{\mathbf{y}}[\mathbf{y}|x{=}3] = 10.81$.

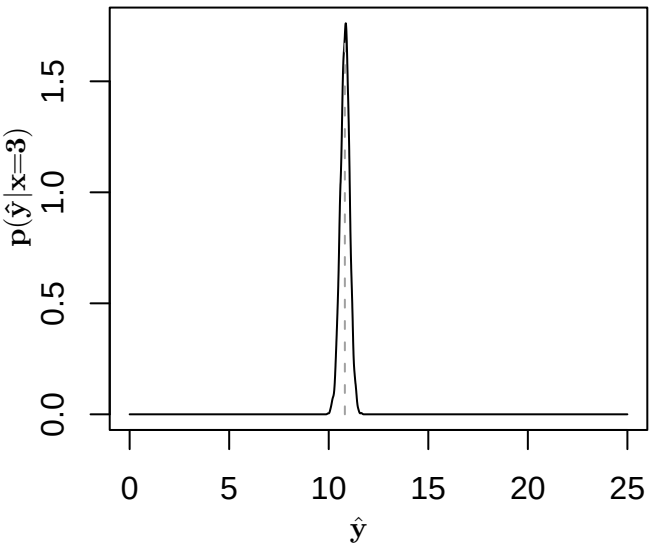

Figure 4: The probability density $p(\hat{y}|x{=}3)$

3.3.3 Combination of uncertainties

Figure 3.3.3, where densities $p(y|x{=}3)$ and $p(\hat{y}|x{=}3)$ have been grouped on the same graph, gives a graphical illustration of the noise, bias and variance quantities at one point x. One can see that the MSE defined in equation 3.1 is equivalent to the sum of the variance $\sigma_{\mathbf{y}}$ of $p(y|x)$ (the red curve), the variance $\sigma_{\hat{\mathbf{y}}}$ of $p(\hat{y}|x)$ (the blue curve), and the squared distance $d_{\mathbf{y}\hat{\mathbf{y}}}$ between the expected values of $p(y|x)$ and $p(\hat{y}|x)$ (the vertical dashed lines). Note that the sum of these three components is purely additive. The $MISE$ defined in 3.1 is the sum of the expected values of these three components over $\mathcal{X}$ (average noise, bias, and variance terms over $\mathcal{X}$).

3.4 Bias-Variance tradeoff

This decomposition put also into evidence the attention a designer should give at not choosing too complex models. It shows that in regression problems using the quadratic loss function, the error is *purely additive in its noise, bias and variance components*. In order to reduce the model error, the designer can aim at reducing either the bias or the variance, as the noise component is irreducible.

The dilemna exposed in the introduction can now be formulated in terms of bias and variance. As the model increases in complexity, its bias is likely to diminish. However, as the number of training examples is kept fixed, the parametric identification of the model may strongly vary from one D_N to

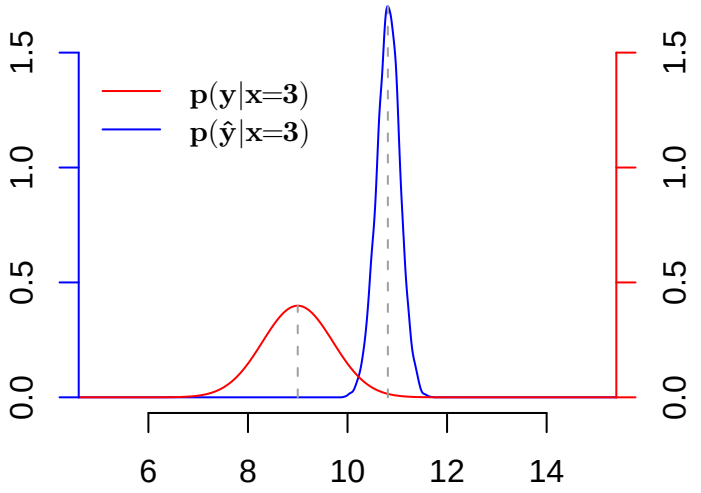

Figure 5: The probability densities $p(y|x{=}3)$ and $p(\hat{y}|x{=}3)$

another. This will increase the variance term. At one stage, the decrease in bias will be inferior to the increase in variance, warning that the model should not be too complex.
Conversely, to decrease the variance term, the designer has to simplify its model so that it is less sensitive to a specific training set. This simplification will lead to a higher bias.
This decomposition makes it possible to unveil the actors of the generalization error in regression problems with quadratic loss function.It however only applies to regression problems, and can not be transposed in classification problems, partly because the quadratic loss function is not apropriate, and partly because the notion of variance and bias has to be redefined, as will be exposed in the following.

4 Multiclass classification problems

4.1 Outline of the differences with the regression case

In a K-class classification problem, the output space $\mathcal{Y}$ is the set of classes $\mathcal{Y} = \{c_0, c_1, ..., c_{k-1}\}$. The stochastic process $\mathcal{S}$ is entirely defined by a joint probability distribution $P(y, x)$, and let $\mathbf{Y}$ and $\mathbf{x}$ be the corresponding random variables. The output $\hat{y}$ of the model $h(x, \alpha_N)$ is an element of $\mathcal{Y}$. As $\hat{y}$ depends on a particular D_N, let $\hat{\mathbf{Y}}$ and $\mathbf{D_N}$ be the corresponding random variables.

The approximation $y^* = E_{\mathbf{Y}}[\mathbf{Y}|x]$ and the loss function $C(y, \hat{y}) = (y - \hat{y})^2$ proposed is the regression do not make sense anymore, as classes c_i of $\mathcal{Y}$ may not be numeric. The fact that they can be numeric does not provide a solution as the distance between two different classes has generally no link to their corresponding numerical values. The distance between classes is therefore specific to the classification problem considered.

4.2 Misclassification loss functions

In a classification problem, an ad-hoc definition of distance has to be designed. The simplest case is to consider that the distance between a class and itself is zero, and the distance between a class and another is 1. This leads to the zero-one loss function:

$$C(y, \hat{y}) = 1(y \neq \hat{y})$$

where 1(.) is an indicator function of the truth of its argument, i.e.:

$$1(\eta) = \begin{cases} 1 & \text{if } \eta \text{ is true} \\ 0 & \text{otherwise} \end{cases}$$

This loss function expresses the fact that the value predicted by the model must be the actual value. No approximation is possible, and predicting a wrong class has the same maximum loss 1 whatever the class.
However, in some problems, it might be necessary to distinguish distances between two classes more precisely. Moreover, the distance between two classes may not be symmetric. For example, consider a in medical diagnosis classification problem with two cases, 'healthy' and 'cancerous'. The cost is higher to predict a cancerous cell as healthy than to predict a healthy cell as cancerous.
These specific distances between classes and this necessity of assymetry is expressed by a **loss matrix** C, whose components c_{ij} code for the 'cost' of predicting class j if actual class was i. Of course, we have $c_{ii} = 0$.
The misclassification loss function can be simply expressed with the loss matrix :

$$C(y, \hat{y}) = c_{y\hat{y}}$$

4.3 Optimal Bayes classifier

The notion of 'average' or 'expected' value y^* seen in the regression case has to be redefined. Let us first define the misclassification risk. Given a probability distribution $P(\mathbf{Y}|x)$, the misclassification risk $r(y')$ is the 'cost'

of predicting y':

$$r(y') = E_{\mathbf{Y}}[C(\mathbf{Y}, y')] = \sum_{y=c_0}^{y=c_{k-1}} P(\mathbf{Y} = y) * C(y, y')$$

The 'optimal' class y^* to choose at x is the one that minimizes the misclassification risk.

$$y^* = argmin'_y r(y')$$

For example, in the two-class case, if $C(c_0, c_1) = 0.1$, $C(c_1, c_0) = 0.9$, $P(\mathbf{Y} = c_0) = 0.7$ and $P(\mathbf{Y} = c_1) = 0.3$, then $r(c_0) = 0.3 * 0.9 = 0.27$ and $r(c_1) = 0.7 * 0.1 = 0.07$. The class to choose is therefore the class c_1.
If the misclassification loss function is the zero-one loss function, these expressions are simplified in:

$$r(y') = \sum_{y=c_0}^{y=c_{k-1}} P(\mathbf{Y} = y) * C(y, y') = \sum_{y \neq y'} P(\mathbf{Y} = y) = 1 - P(\mathbf{Y} = y')$$

and

$$y^* = argmin_{y'} r(y') = argmax'_y P(\mathbf{Y} = y')$$

The function y^* is called the optimal Bayes classifier. It is the classifier that minimizes the misclassification risk given a probability distribution $P(\mathbf{Y}|x)$.

4.4 The mean misclassification error

To assess the generalization error of the model at a point x, we have to average the loss function at x over all possible realizations of $\mathbf{D_N}$ and $\mathbf{Y}$. Following the same logic as in the quadratic loss function, let us define the mean misclassification error. For a given x, the mean misclassification error is the expected value of the misclassification loss function over all realizations of $\mathbf{Y}$ and $\mathbf{D_N}$, that is :

$$\begin{aligned} MME(x) &= E_{\mathbf{Y},\mathbf{D_N}}[C(\mathbf{Y}, \hat{\mathbf{Y}})] \qquad (3) \\ &= E_{\mathbf{Y},\mathbf{D_N}}\left[\sum_{c,c'=c_0}^{c_{k-1}} C(c, c') * 1(\mathbf{Y} = c) * 1(\hat{\mathbf{Y}} = c')\right] \\ &= \sum_{c,c'=c_0}^{c_{k-1}} C(c, c') * E_{\mathbf{Y}}[1(\mathbf{Y} = c)] * E_{\mathbf{D_N}}[1(\hat{\mathbf{Y}} = c')] \\ &= \sum_{c,c'=c_0}^{c_{k-1}} C(c, c') * P(\mathbf{Y} = c) * P(\hat{\mathbf{Y}} = c') \qquad (4) \end{aligned}$$

The expression of the MME is rather complicated, and finding a decomposition in terms of bias, variance and noise proves to be very difficult

because of inequal interactions between classes. This decomposition will be shown, restrained to the two-class problem, in section 7.6.

For zero-one loss functions, this expression simplifies to :

$$\begin{aligned} MME(x) &= \sum_{c,c'=c_0}^{c_{k-1}} C(c,c') * P(\mathbf{Y}=c) * P(\hat{\mathbf{Y}}=c') \\ &= 1 - \sum_{c,c'=c_0}^{c_{k-1}} 1(c,c') * P(\mathbf{Y}=c) * P(\hat{\mathbf{Y}}=c') \\ &= 1 - \sum_{c=c_0}^{c_{k-1}} P(\mathbf{Y}=c) * P(\hat{\mathbf{Y}}=c) \end{aligned} \quad (5)$$

which can be compactely written as:

$$MME(x) = P(\mathbf{Y} \neq \hat{\mathbf{Y}})$$

We will now see in the next following sections an overview of the main different methods proposed to decompose the MME.

5 Friedman's analysis of the two-class problem

This study is presented in [5]. In two-class problems, the output space can be defined as $\mathcal{Y} = \{0, 1\}$, 0 standing for the first class and 1 for the second. Mapping problem classes to numerical ones makes it possible for the model $h(x, \alpha_{\mathbf{N}})$ to be real-valued.

5.1 Decision boundary and optimal Bayes classifier

In the two-class case, the optimal Bayes classifier can be defined by means of the decision boundary. First, let us define the following f function :

$$f_1(x) = P(\mathbf{Y} = 1|x)$$

$f_1(x)$ is the probability for $\mathcal{S}$ to be of class 1 at x. This function entirely determines $\mathcal{S}$ as one has $P(\mathbf{Y} = 0|x) = 1 - f_1(x)$.
As seen in section 4.3, the optimal Bayes classifier is the function y^* that minimizes the misclassification risk. In the two-class case:

$$\begin{aligned} y^* = 1 &\Leftrightarrow r(c_0) > r(c_1) \\ &\Leftrightarrow c_{10} * P(\mathbf{Y}=1) > c_{01} * P(\mathbf{Y}=0) \\ &\Leftrightarrow c_{10} * P(\mathbf{Y}=1) > c_{01} * (1 - P(\mathbf{Y}=1)) \\ &\Leftrightarrow P(\mathbf{Y}=1) > \frac{c_{01}}{c_{01}+c_{10}} \\ &\Leftrightarrow 1(P(\mathbf{Y}=1) > l) \end{aligned}$$

with $l = \frac{c_{01}}{c_{01}+c_{10}}$. l is called the decision boundary. For classes with equal costs, we have $l = \frac{1}{2}$.
By using 5.1, the optimal Bayes classifier can be expressed as:

$$y^* = 1(f_1(x) > l)$$

5.2 Nature of the model

As pointed out at the beginning of this section, the model output can be real-valued. To decide what class an instance x belongs to, the real-valued output is compared to the decision boundary in much the same way that the optimal Bayes classifier was obtained:

$$\hat{y} = 1(h(x, \alpha_N) > l)$$

The model's output needs not even be bounded by $[0, 1]$.

5.3 First part of the decomposition - The wrongly right effect

All what is developed in the following is valid for any l value. However, let us fix it to $\frac{1}{2}$ to simplify the writing. Friedman develops the MME in:

$$\begin{aligned} P(\mathbf{Y} \neq \hat{\mathbf{Y}}) &= P(\mathbf{Y} = y^*) * P(\hat{\mathbf{Y}} \neq y^*) + P(\mathbf{Y} \neq y^*) * P(\hat{\mathbf{Y}} = y^*) \\ &= P(\mathbf{Y} = y^*) * P(\hat{\mathbf{Y}} \neq y^*) + P(\mathbf{Y} \neq y^*) * (1 - P(\hat{\mathbf{Y}} \neq y^*)) \\ &= P(\hat{\mathbf{Y}} \neq y^*) * (P(\mathbf{Y} = y^*) - P(\mathbf{Y} \neq y^*)) + P(\mathbf{Y} \neq y^*) \\ &= P(\hat{\mathbf{Y}} \neq y^*) * (1 - 2 * P(\mathbf{Y} \neq y^*)) + P(\mathbf{Y} \neq y^*) \qquad (6) \end{aligned}$$

This decomposition is only possible as a two-class problem is considered here (validity of $P(\mathbf{Y} \neq \hat{\mathbf{Y}})$ *and* $P(\mathbf{Y} = y^*) \Rightarrow P(\hat{\mathbf{Y}} \neq y^*)$) . The last term in (6) is the irreducible error deriving from the uncertainty of $\mathcal{S}$. It can therefore be considered as the *noise* component. The first term of (6) is a product of the error deriving from the the model according to the optimal class y^*, and an expression depending of the *noise*. This shows that the decomposition in not additive as was the case in the mean squared error. The noise component is bounded by 0 and 0.5, by definition of y^*. The multiplicative factor $1 - 2 * P(\mathbf{Y} \neq y^*)$ will therefore always lessen the error coming from the model. In the end, the greater the noise, the greater the error, but it is not completely additive as with the MSE. This attenuation can be understood by the fact that, as there are only two classes, if the model's prediction is wrong with respect with the optimal class y^*, then it may be right if y happens to be different of y^*. This can be called the wrongly right effect, and *is a classification specific characteristic as it statistically never happens in regression.*

5.4 Second part of the decomposition

The model error $P(\hat{\mathbf{Y}} \neq y^*)$ can be analyzed in more detail, by considering the distribution $p(\mathbf{h})$ of outputs of the model $h(x, \alpha_{\mathbf{N}})$ which depends on $\mathbf{D_N}$. The probability $P(\hat{\mathbf{Y}} \neq y^*)$ that the prediction differs from y^* can then be written as:

$$P(\hat{\mathbf{Y}} \neq y^*) = 1(f_1 < l) * \int_{\frac{1}{2}}^{\infty} p(\mathbf{h}) d\mathbf{h} + 1(f_1 > l) * \int_{-\infty}^{\frac{1}{2}} p(\mathbf{h}) d\mathbf{h}$$

Friedman then argues that the distribution of $\mathbf{h}$ can be approximated by a gaussian, as each h is the final outcome of a complex averaging process. Thus, he proposes:

$$p(\mathbf{h}) = \frac{1}{\sqrt{2 * \pi * var(\mathbf{h})}} * exp\left(-\frac{1}{2}\frac{(\mathbf{h} - E_{\mathbf{h}}[\mathbf{h}])^2}{var(\mathbf{h})}\right)$$

Friedman then defines the notion of 'boundary bias' (at x) by:

$$b = sign(1/2 - f_1) * (E_{\mathbf{h}}[\mathbf{h}] - \frac{1}{2})$$

The model is therefore correct on an example x if the boundary bias is negative, and incorrect if the boundary bias is positive. The quantity of the boundary bias is not important, but its sign is decisive. All what is required of $E_{\mathbf{h}}[\mathbf{h}]$ is to be on the 'same side' as f_1. If this can be achieved, then reducing the variance will reduce the probability for h to be wrong. In this sense, variance tends to dominate bias in classification. This notion of boundary bias leads to the use of 'oversmoothing' models (Naive Bayes, K-NN), not appropriate in regression because of their high bias, but which can nonetheless perform well in classification. An oversmoothing model is a model that tends to shrink $\mathbf{h}$ to the mean output value $E_{\mathbf{Y},\mathbf{x}}[\mathbf{Y}]$, that is:

$$h(x, \alpha_N) = (1 - \eta(x)) * E_{\mathbf{Y}}[\mathbf{Y}] + \eta(x) * E_{\mathbf{Y},\mathbf{x}}[\mathbf{Y}]$$

where $0 \leq \eta(x) \leq 1$ is the smoothing coefficient, generally depending on x. As long as the decision boundary equals $E_{\mathbf{Y},\mathbf{x}}[\mathbf{Y}]$, then boundary bias in negative, and classification is correct.

5.5 Application and illustration

For example, K-Nearest Neighbours is a kind of oversmoothing model. It approximates y^* at x by averaging the values of the k nearest training examples to x. Therefore, it tends to shrink an estimate h to the mean $E_{\mathbf{Y},\mathbf{x}}[\mathbf{Y}]$, and in clasification problem, this makes it possible to classify the instance correctly whereas in a regression problem that would not have been appropriate. For certain problem, Naive Bayes classifiers or logit regression would also exhibit this property. See [5] for a detailed account of the experiments.

As an illustration, consider the following example. $\mathcal{X} = [0; 1]$, $\mathcal{Y} = \{0; 1\}$, and $\mathcal{S}$ is defined by:

$$\begin{cases} P(\mathbf{Y} = 1) = 0.9 & \text{if } 0 \leq x < 0.5 \\ P(\mathbf{Y} = 1) = 0.1 & \text{if } 0.5 \leq x \leq 1 \end{cases}$$

The size of D_N is $N = 100$, and the model $h(x, \alpha_N)$ chosen is a simple linear regression using the least square technique. On figure 5.5 are illustrated four models for four different realizations of D_N. Classes 0 and 1 are balanced, and therefore the decision boundary is $l = 0.5$. Classification is achieved by testing $h(x, \alpha_N) > 0.5$. If true, predicted class is 1, otherwise, it is 0. Given the nature of the underlying phenomenon, note that we have here a kind of oversmoothing model.

Theoretically, we know that the best model would be one for which $h(x, \alpha_N) > 0.5$ for $x < 0.5$ and $h(x, \alpha_N) < 0.5$ for $x > 0.5$. Practically, the model obtained will depend on the D_N it was given. On figure 5.5 is represented the probability density of $h(x, \alpha_N)$ at $x = 0.45$ for all possible realization of D_N. At $x = 0.45$, we have $y^* = 1$, and as the expected value of $E_{\mathbf{D_N}}[h(0.45, \alpha_\mathbf{N})] = 0.56$ is superior to 0.5, the boundary bias is negative, which means that on average, the model is correct at that point. The model error $P(\hat{\mathbf{Y}} \neq 1)$ corresponds to the dark area. We can compute

$$P(\hat{\mathbf{Y}} \neq y^*) = P(\mathbf{h} < 0.5) = \int_{-\infty}^{\frac{1}{2}} p(\mathbf{h}) d\mathbf{h} = 0.054$$

which means that there is a 5% probability that the model be wrong at that point, with respect to the optimal value $y^* = 1$.

Computing equation (6) gives:

$$\begin{aligned} P(\hat{\mathbf{Y}} \neq y^*) * (1 - 2 * P(\mathbf{Y} \neq y^*)) + P(\mathbf{Y} \neq y^*) &= 0.054 * (1 - 2 * 0.1) + 0.1 \\ &= 0.054 - 2 * 0.54 * 0.1 + 0.1 \\ &= 0.1432 \end{aligned}$$

which is the sum of the model error and the noise, minus the probability of the 'wrongly right' effect.

5.6 Shortcomings

Friedman's analysis is limited to two class problems. All what has been presented can not be extended to k-class problems, $k > 2$. What is more, he supposes that the distribution $\mathbf{h}$ be Gaussian, and this assumption may not be validated. In that case, the expectancy $E_\mathbf{h}[\mathbf{h}]$ could actually not reflect correctly the class that is on average chosen by the model ($argmax_c\ P(\hat{\mathbf{Y}} = c|x)$).

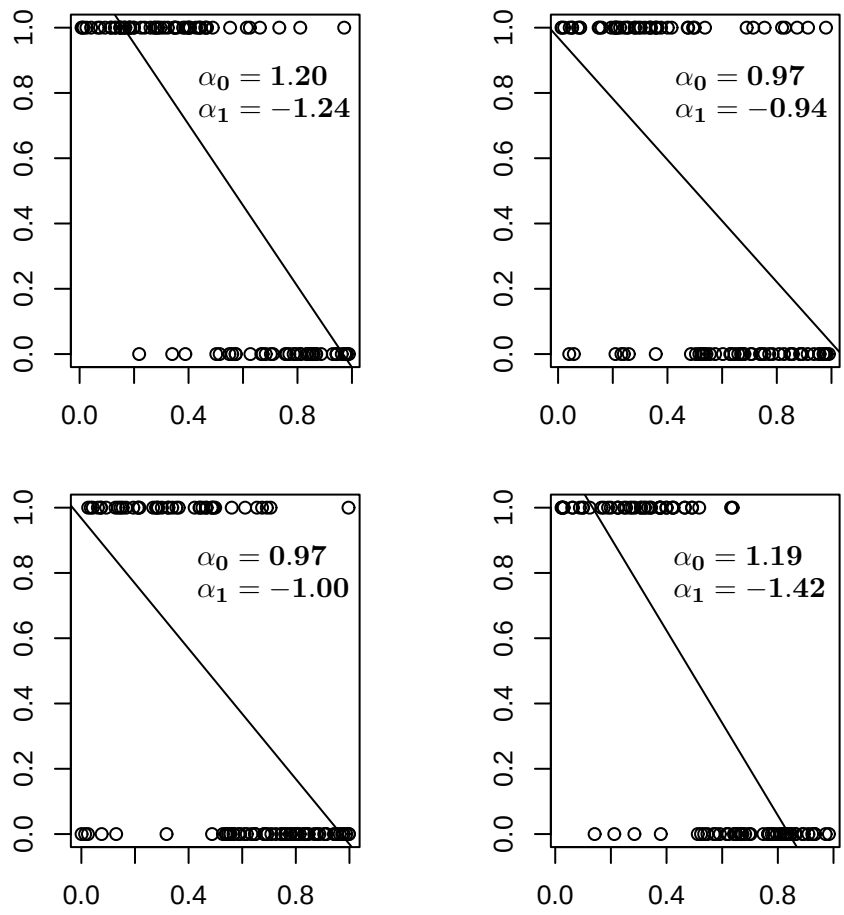

Figure 6: Four slightly different models due to variations in the training set

6 Misleading decomposition

6.1 Use of the quadratic loss function

In a two class problem, it is posible to use the quadratic loss function. The quadratic decomposition makes no real sense, as will be shown, and is presented as the qualitative conclusions will be reused later in section 6.2. Reintroducing 3.2 with replacement of regression concepts $\sigma^2_{\mathbf{w}}$ and y^* by their original values, we have :

$$\begin{aligned} MSE(x) &= E_{\mathbf{y},\mathbf{D_N}}[(\mathbf{y} - \hat{\mathbf{y}})^2] \\ &= E_{\mathbf{y}}[(\mathbf{y} - E[\mathbf{y}|x])^2] + (E[\mathbf{y}|x] - E_{\mathbf{D_N}}[\hat{\mathbf{y}}])^2 + E_{\mathbf{D_N}}[E_{\mathbf{D_N}}[\hat{\mathbf{y}}] - \hat{\mathbf{y}})^2] \end{aligned}$$

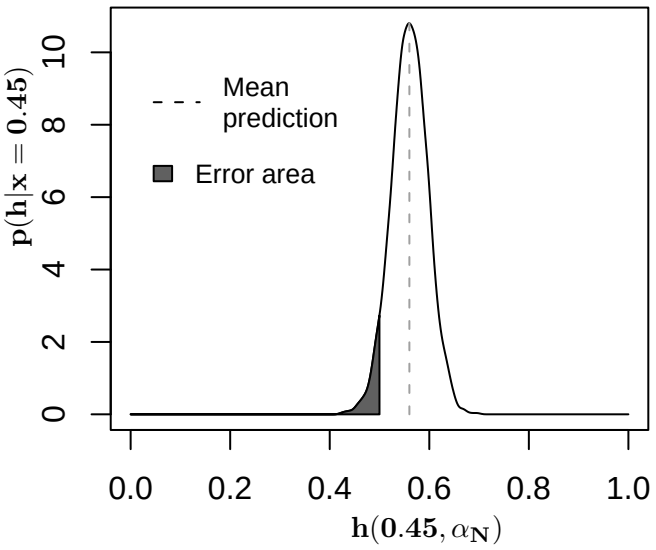

Figure 7: Probability density of $\mathbf{h}(0.45, \alpha_N)$. We have $E_{\mathbf{D_N}}[h(0.45, \alpha_{\mathbf{N}})] = 0.56$ and $P(\mathbf{h} < 0.5) = 0.054$.

One could argue that :

1. The *noise* term measures the degree of uncertainty of $\mathcal{S}$ at x. It equals zero only if $\mathcal{S}$ is deterministic at x, and is maximum when both classes have the same probability to appear at x. Moreover, it does not depend on the model nor on the training set.

2. The *variance* term measures the variability of the model prediction. It equals zero only if the prediction is always the same regardless of the training set. As the prediction becomes more sensitive to the training set, this component increases. Finally, it does not depend on the actual value y.

3. The $bias^2$ measures the difference between the probability for $\mathbf{y}$ to be 1 and the probability for $\hat{\mathbf{y}}$ to be 1. Therefore, if the model can learn to exactly reproduce the stochastic phenomenon at x, the bias is 0.

Behind the seductive appearance of these properties, there are issues that need to raised. In regression, the bias at x was the difference between the expected value of $\mathbf{y}$, and the average output of the model (irrespective of the training sets). In classification, one feels that the notion of average value has to be redefined according to classification concepts. As presented in section 4, the average value should be the class that maximizes a probability distribution, or the class that minimizes the misclassification risk. A desirable

property of bias in a classication scheme is that the optimal Bayes classifier have a zero bias. This is not the case in this decomposition. The bias obtained here rather reflects the discrepancy between the uncertainty of $\mathcal{S}$ and the uncertainty of the model averaged over all training sets. It therefore contains a mix of the *noise* and *variance* components. So to major shortcomings of this bias are:

1. It does not equal zero for an optimal Bayes classifier.
2. It is not independent of the *noise* and *variance* components.

6.2 A quadratic misclassification decomposition

This solution was proposed in 1996 by Kohavi and Wolpert [9]. Let us consider the squared sum:

$$\begin{aligned}\frac{1}{2}\sum_{c=c_0}^{c_{k-1}}\Big(P(\mathbf{Y}=c)-P(\hat{\mathbf{Y}}=c)\Big)^2 &= \frac{1}{2}\left(\sum_{c=c_0}^{c_{k-1}}P(\mathbf{Y}=c)^2\right)\\ &\quad+\frac{1}{2}\left(\sum_{c=c_0}^{c_{k-1}}P(\hat{\mathbf{Y}}=c)^2\right)\\ &\quad-\sum_{c=c_0}^{c_{k-1}}P(\mathbf{Y}=c)*P(\mathbf{Y}=c) \quad (7)\end{aligned}$$

By subtracting (5) with (7), and reassembling the terms, one gets:

$$\begin{aligned}P(\mathbf{Y}\neq\hat{\mathbf{Y}}) &= \frac{1}{2}\sum_{c=c_0}^{c_{k-1}}\Big(P(\mathbf{Y}=c)-P(\hat{\mathbf{Y}}=c)\Big)^2 \quad (8)\\ &\quad+\frac{1}{2}\left(1-\left(\sum_{c=c_0}^{c_{k-1}}P(\mathbf{Y}=c)^2\right)\right) \quad (9)\\ &\quad+\frac{1}{2}\left(1-\left(\sum_{c=c_0}^{c_{k-1}}P(\hat{\mathbf{Y}}=c)^2\right)\right) \quad (10)\end{aligned}$$

Kohavi and Wolpert refered (8) as *squared bias*, (9) as *noise*, and (10) as *variance*. These definitions of $bias^2$, *variance* and *noise* share exactly the same properties that those of the quadratic decomposition presented above. Therefore, for the same reasons, this decomposition does not provide much insight into the nature of bias and variance for classification problems.

7 A unified frame for loss function decompositions

A very nice solution to the misclassification loss function decomposition, consistent with concepts of bias and variance used in the quadratic decomposition, was presented by Domingos in [3], [4]. Prior to his decompositions are definitions of some useful concept that make it possible to unify his misclassification loss function decomposition to the standard quadratic one.

7.1 Definitions

First, for a given x, Domingos introduces clearly the notion of optimal value already seen in subsections 3.2 and 4.3:
An **optimal value** y^*, characterizing a stochastic process $\mathcal{S}$ at x, depends on the loss function $C(y, \hat{y})$. It is the value that minimizes the expected value of the loss function over all possible y.

$$y^* = argmin'_y E_{\mathbf{Y}}[C(\mathbf{Y}, y')]$$

This definition is consistent with optimal values defined in subsections 3.2 and 4.3. For other loss functions, e.g. the absolute loss function defined in subsection 3.1, this optimal value would be the median.
Second, a similar definition is introduced for a model:
A **main value** $\hat{y}^*$ for a model at x is the value that minimizes this expected value of the loss function over all possible training sets D_N.

$$\hat{y}^* = argmin'_y E_{\mathbf{D_N}}[C(\hat{\mathbf{Y}}, y')]$$

This definition is consistent with the average value $\hat{y}^*$ used in subsection 3.2, and extends the concept to other kinds of loss functions.
From these definitions, consistent definitions for the *noise*, *bias*, and *variance* components of a decomposition (at a given x) are introduced:
Noise : Loss incurred by the variations of y relative to the optimal prediction:

$$N(x) = E_{\mathbf{Y}}[C(\mathbf{Y}, y^*)]$$

Bias : Loss incurred by the main prediction relative to the optimal prediction:

$$B(x) = C(y^*, \hat{y}^*)$$

Variance : Average loss incurred by the prediction relative to the main prediction:

$$V(x) = E_{\mathbf{D_N}}[C(\hat{y}^*, \hat{\mathbf{Y}})]$$

Finally, once these concepts defined, Domingos shows how to decompose quadratic and misclassification loss functions in the general mean error form :

$$ME(x) = E_{\mathbf{Y},\mathbf{D_N}}[C(\mathbf{Y}, \hat{\mathbf{Y}})] = c_1 * N(x) + B(x) + c_2 * V(x)$$

where c_1 and c_2 are multiplicative factors, potentially depending on x. Averaged over x, this gives the mean integrated error:

$$MIE = E_{\mathbf{x},\mathbf{Y},\mathbf{D_N}}[C(\mathbf{Y}, \hat{\mathbf{Y}})] = E_{\mathbf{x}}[c_1 * N(\mathbf{x})] + E_{\mathbf{x}}[B(\mathbf{x})] + E_{\mathbf{x}}[c_2 * V(\mathbf{x})]$$

7.2 Quadratic loss function decomposition

$c_1 = c_2 = 1$. The proof was already demonstrated in subsection 3.2, but the concepts of optimal and main values are now clearly stated. Moreover, given the new bias concept defined above, the presence of a square is considered in this decomposition as a consequence of the quadratic loss function. The term 'squared bias' does not hold anymore here.

7.3 Misclassification loss function decomposition

Domingos showed how to obtain the decomposition for zero-one loss in multi-class problems, which we present in subsection 7.5, and for general misclassification loss function in two-class problems, which we present in subsection 7.6. The general misclassification loss function in general multi-class problems, see formula (4), remains unproved for the moment.
It is interesting to note that with the notion of bias defined above, the bias in classification is binary. This follows from the bias concept definition, and so is consistent with a 'squared bias' for the quadratic loss function decomposition. This suggestion was first proposed by Dietterich and Kong [2], [8], who had developed a decomposition in noise-free settings, with binary bias and potentially negative variance. This negative variance had been criticized [9], as a variance component should not be negative. Domingos' decomposition is similar in the concept to the Dietterich and Kong's one, but through the above definition of variance, the variance is positive, although as we will see its coefficient c_2 can be negative, which concpetually sort the problem out. Moreover, his decomposition goes further than Dietterich and Kong's one as it includes the noise component.
The rationale of the following decompositions is made of three steps:

1. The first step considers only the expected loss over $\mathbf{y}$, and decomposes it into a weighted noise component and an error which depends only on the optimal value and y^*. The goal is therefore to find a coefficient c_0 that can verify this decomposition:

$$E_{\mathbf{Y}}[C(\mathbf{Y}, \hat{y})] \quad = \quad C(y^*, \hat{y}) + c_0 * E_{\mathbf{Y}}[C(\mathbf{Y}, y^*)] \tag{11}$$

   The coefficient c_0 can depend on $\hat{y}$, and is therefore a function of D_N and x.

2. The second step considers only the expected loss over D_N between the optimal value and different $\hat{y}$. It aims at decomposing this expected

loss in a weighted variance term and a bias. The goal is therefore to find a coefficient c_2 that can verify this decomposition:

$$E_{\mathbf{D_N}}[C(y^*, \hat{\mathbf{Y}})] = C(y^*, \hat{y}^*) + c_2 * E_{\mathbf{D_N}}[C(\hat{y}^*, \hat{\mathbf{Y}})] \tag{12}$$

3. The third step consists in unifying (11) and (12) by averaging 11 over D_N. One obtains:

$$\begin{aligned} MME(x) &= E_{\mathbf{Y},\mathbf{D_N}}[C(\mathbf{Y}, \hat{\mathbf{Y}})] \\ &= E_{\mathbf{D_N}}[c_0 * E_{\mathbf{Y}}[C(\mathbf{Y}, y^*)] + C(y^*, \hat{\mathbf{Y}})] \\ &= E_{\mathbf{D_N}}[c_0 * E_{\mathbf{Y}}[C(\mathbf{Y}, y^*)]] + E_{\mathbf{D_N}}[C(y^*, \hat{\mathbf{Y}})] \\ &= c_1 * E_{\mathbf{Y}}[C(\mathbf{Y}, y^*)] + C(y^*, \hat{y}^*) \\ &\quad + c_2 * E_{\mathbf{D_N}}[C(\hat{y}^*, \hat{\mathbf{Y}})] \end{aligned} \tag{13}$$

with $c_1 = E_{\mathbf{D_N}}[c_0]$

As will be seen, and as was already partly showed by Friedman in subsection 5.3, these decomposition are not additive, and there exists interactions between components, due to the finite number of categories in $\mathcal{Y}$.

7.4 Zero-One loss function decomposition for two classes

7.4.1 Decomposition

Let us first show the proof with only two classes:

First part from formula (11):

$$E_{\mathbf{Y}}[C(\mathbf{Y}, \hat{y})] = C(y^*, \hat{y}) + c_0 * E_{\mathbf{Y}}[C(\mathbf{Y}, y^*)]$$

is proved with $c_0 = 1$ if $\hat{y} = y^*$ and $c_0 = -1$ if $\hat{y} \neq y^*$. It is trivially true if $\hat{y} = y^*$. If $\hat{y} \neq y^*$, we have, given that there are only two classes:

$$\begin{aligned} E_{\mathbf{Y}}[C(\mathbf{Y}, \hat{y})] &= P(\mathbf{Y} \neq \hat{y}) \\ &= 1 - P(\mathbf{Y} = \hat{y}) && (14) \\ &= 1 - P(\mathbf{Y} \neq y^*) && (15) \\ &= C(y^*, \hat{y}) - E_{\mathbf{Y}}[C(\mathbf{Y}, y^*)] \end{aligned}$$

Second part from formula (12):

$$E_{\mathbf{D_N}}[C((y^*, \hat{\mathbf{Y}})] = C(y^*, \hat{y}^*) + c_2 * E_{\mathbf{D_N}}[C(\hat{y}^*, \hat{\mathbf{Y}})]$$

is proved with $c_2 = 1$ if $\hat{y}^* = y^*$ and $c_2 = -1$ if $\hat{y}^* \neq y^*$. The proof is very similar to the first part. If $\hat{y}^* = y^*$, it is trivially true. If $\hat{y}^* \neq y^*$, we have :

$$\begin{aligned} E_{\mathbf{D_N}}[C(y^*, \hat{\mathbf{Y}})] &= P(\hat{\mathbf{Y}} \neq y^*) \\ &= 1 - P(\hat{\mathbf{Y}} = y^*) && (16) \\ &= 1 - P(\hat{\mathbf{Y}} \neq \hat{y}^*) && (17) \\ &= C(y^*, \hat{y}^*) - E_{\mathbf{D_N}}[C(\hat{\mathbf{Y}}, \hat{y}^*)] \end{aligned}$$

We can now obtain $MME(x)$ with formula (13):

$$MME(x) = c_1 * E_{\mathbf{Y}}[C(\mathbf{Y}, y^*)] + C(y^*, \hat{y}^*) + c_2 * E_{\mathbf{D_N}}[C(\hat{y}^*, \hat{\mathbf{Y}})]$$

with $c_1 = E_{\mathbf{D_N}}[c_0] = P(\hat{\mathbf{Y}} = y^*) - P(\hat{\mathbf{Y}} \neq y^*)$, and $c_2 = 1$ if $B(x) = 0$ or $c_2 = -1$ if $B(x) = 1$.

We have also:

$$MIE = E_{\mathbf{x}}[(P(\hat{\mathbf{Y}} = y^*) - P(\hat{\mathbf{Y}} \neq y^*)) * N(\mathbf{x})] + E_{\mathbf{x}}[B(\mathbf{x})] + E_{\mathbf{x}}[(1 - 2 * B(\mathbf{x})) * V(\mathbf{x})]$$

7.4.2 Discussion on coefficients

1. Coefficient c_1 shows that the error coming from the noise component is partly offset by the unstability of the model over D_N. It is the same wrongly-right effect that Friedman had shown (see subsection 5.3). The same qualitative conclusions hold : A classifier will of course perform better is there is no noise in the stochastic process under study and if it is itself not too unstable with respect to a specific training set D_N. However, in case of noise and variance, then the error of these two components are not additive, but partly offset each other as there are only two classes. The two extreme cases are: if $P(\hat{\mathbf{Y}} = y^*) = 1$, then the classifier is optimal, $P(\hat{\mathbf{Y}} \neq y^*) = 0$ and the noise component is the origin of the error. If $P(\hat{\mathbf{Y}} = y^*) = 0$, then $P(\hat{\mathbf{Y}} \neq y^*) = 1$ and the bias is one. The noise component will be subtracted to the bias, as actually in this case the classifier is only right when y is not the optimal value.

2. The bias of the model over $\mathcal{X}$, $E_{\mathbf{x}}[B(\mathbf{x})]$, can be understood in the context of classification as the *probability* that the model main value differ from the stochastic process optimal value. This is what is conceptually desirable for bias in classification models.

3. The variance of the model over $\mathcal{X}$, $E_{\mathbf{x}}[(2B(\mathbf{x}) - 1) * P(\hat{\mathbf{Y}} \neq \hat{y}^*)]$, is not independent of the bias as was the case for the quadratic loss function variance. It is important to consider the variance term in two ways. It absolute value is indeed the variance of the model, whereas it weighted value is what Domingos qualified as the 'net' variance. The net variance is the variance contribution in the decomposition, which can be lower than the absolute variance is the model is biased on some examples. *Distinction between these two types of variance is conceptually fundamental in this decomposition.*

7.5 Zero-One loss function decomposition for k-classes, $k > 2$

7.5.1 Decomposition

The proof for general multi-class problem is very similar to the one presented above, and will only complicate somewhat coefficients c_0, c_1 and c_2. In the first proof, we could go from (14) to (15) as there were only two classes. If we consider more classes, we can relate $P(\mathbf{Y} = \hat{y})$ and $P(\mathbf{Y} \neq y^*)$:

$$\begin{aligned} P(\mathbf{Y} = \hat{y}) &= P(\mathbf{Y} = \hat{y}|\mathbf{Y} = y^*) * P(\mathbf{Y} = y^*) \\ &\quad + P(\mathbf{Y} = \hat{y}|\mathbf{Y} \neq y^*) * P(\mathbf{Y} \neq y^*) \\ &= P(\mathbf{Y} = \hat{y}|\mathbf{Y} \neq y^*) * P(\mathbf{Y} \neq y^*) \end{aligned}$$

as we suppose $\hat{y} \neq y^*$. Thus, the first part of the decomposition becomes:

$$\begin{aligned} E_{\mathbf{Y}}[C(\mathbf{Y}, \hat{y})] &= P(\mathbf{Y} \neq \hat{y}) \\ &= 1 - P(\mathbf{Y} = \hat{y}) \\ &= 1 - P(\mathbf{Y} = \hat{y}|\mathbf{Y} \neq y^*) * P(\mathbf{Y} \neq y^*) \\ &= C(y^*, \hat{y}) - P(\mathbf{Y} = \hat{y}|\mathbf{Y} \neq y^*) * E_{\mathbf{Y}}[C(\mathbf{Y}, y^*)] \quad (18) \end{aligned}$$

and we have $c_0 = -P(\mathbf{Y} = \hat{y}|\mathbf{Y} \neq y^*)$.

The same reasoning holds for the second part of the decomposition. In the proof for 2 classes, we could go from (16) to (17) as there were only two classes. With more classes, we can relate $P(\hat{\mathbf{Y}} = y^*)$ and $P(\hat{\mathbf{Y}} \neq \hat{y}^*)$ by:

$$\begin{aligned} P(\hat{\mathbf{Y}} = y^*) &= P(\hat{\mathbf{Y}} = y^*|\hat{\mathbf{Y}} = \hat{y}^*) * P(\hat{\mathbf{Y}} = \hat{y}^*) \\ &\quad + P(\hat{\mathbf{Y}} = y^*/\hat{\mathbf{Y}} \neq \hat{y}^*) * P(\hat{\mathbf{Y}} \neq \hat{y}^*) \\ &= P(\hat{\mathbf{Y}} = y^*/\hat{\mathbf{Y}} \neq \hat{y}^*) * P(\hat{\mathbf{Y}} \neq \hat{y}^*) \end{aligned}$$

As we suppose $y^* \neq \hat{y}^*$. Thus, the second part of the decomposition becomes:

$$\begin{aligned} E_{\mathbf{D_N}}[C(y^*, \hat{y})] &= P(\hat{\mathbf{Y}} \neq y^*) \\ &= 1 - P(\hat{\mathbf{Y}} = y^*) \\ &= 1 - P(\hat{\mathbf{Y}} = y^*|\hat{\mathbf{Y}} \neq \hat{y}^*) * P(\hat{\mathbf{Y}} \neq \hat{y}^*) \\ &= C(y^*, \hat{y}^*) - P(\hat{\mathbf{Y}} = y^*|\hat{\mathbf{Y}} \neq \hat{y}^*) * E_{\mathbf{D_N}}[C(\hat{\mathbf{Y}}, \hat{y}^*)] (19) \end{aligned}$$

and we have $c_2 = -P(\hat{\mathbf{Y}} = y^*|\hat{\mathbf{Y}} \neq \hat{y}^*)$

We can now obtain $MME(x)$ with (13):

$$MME(x) = c_1 * E_{\mathbf{Y}}[C(\mathbf{Y}, y^*)] + C(y^*, \hat{y}^*) + c_2 * E_{\mathbf{D_N}}[C(\hat{y}^*, \hat{\mathbf{Y}})]$$

with $c_1 = E_{\mathbf{D_N}}[c_0] = P(\hat{\mathbf{Y}} = y^*) - P(\hat{\mathbf{Y}} \neq y^*) * P(\mathbf{Y} = \hat{y}|\mathbf{Y} \neq y^*)$, and $c_2 = 1$ if $B(x) = 0$ or $c_2 = -P(\hat{\mathbf{Y}} = y^*|\hat{\mathbf{Y}} \neq \hat{y}^*)$ if $B(x) = 1$.

7.5.2 Discussion on coefficients

1. Coefficient c_1 now shows that the error coming from the noise component will depend more strongly on $P(\hat{\mathbf{Y}} = y^*)$, i.e. on the ability of the classifier to predict the right class. The coefficient c_1 will only decrease in cases where both the $\mathbf{Y}$ value and the predicted value are equal, but differ from the optimal class y^*. The higher the number of classes, the less this probability. Therefore, as the number of classes increases, the advantage of the wrongly-right effect vanishes quickly.

2. As the number of classes increases, it is also more likely for the classifier to be biased. Whereas in the two-class case the variance had a negative coefficient when the classifier was biased on an example, in the multi-class case the variance will diminish the error due to bias only in the case where, by chance, the classifier wrongly predict the right class (this can work only in the presence of noise). Therefore, the higher the number of classes, the less likely the probability for variance to decrease the bias error.

7.5.3 Consequences for the classifier design

The two precedent observations show that in classification problems with many classes, careful attention should be given at reducing the bias. Alternatively, given that these decompositions proved that the lower the number of classes, the higher the tolerance for errors, it is sensible to break down a classification problem to several two-class problems. See [6] for experimental evidence on improving classifier performance by using pairwise classifiers.

7.6 General misclassification loss function decomposition for two-class problems

7.6.1 Decomposition

In this case, the loss function is defined by a loss matrix, as presented in section 4.4. In a similar manner than the two preceding decomposition, Domingos shows that:

$$E_{\mathbf{Y},\mathbf{D_N}}[C(\mathbf{Y},\hat{\mathbf{Y}})] = c_1 * E_{\mathbf{Y}}[C(\mathbf{Y},y^*)] + C(y^*,\hat{y}^*) + c_2 * E_{\mathbf{D_N}}[C(\hat{y}^*,\hat{\mathbf{Y}})]$$

with $c_1 = P(\mathbf{Y} = \hat{y}) - \frac{C(y^*,\hat{y})}{C(\hat{y},y^*)} * P(\mathbf{Y} \neq y^*)$ and $c_2 = 1$ if $\hat{y}^* = y^*$ or $c_2 = -\frac{C(y^*,\hat{y}^*)}{C(\hat{y}^*,y^*)}$ if $\hat{y}^* \neq y^*$.

7.6.2 Discussion on coefficients

This loss function is not anymore bounded by 0 and 1 as were the previous ones. The different costs apply to misclassification can raise or decrease

strongly the three noise-bias-variance components.
Example : Suppose two classes 0 and 1, and $C(0,1) = 1$ and $C(1,0) = 10$, i.e. it is ten times more costly two predict class 0 when the actual class is 1 than the contrary.

1. case 1 : $y^* = \hat{y}^* = 0$. $N(x) = C(1,0) * P(\mathbf{Y} = 1)$ will be a function of the high $C(1,0)$ coefficient, and $V(x) = C(0,1) * P(\hat{\mathbf{Y}} = 1)$ will be a function of the low coefficient $C(0,1)$. Therefore the noise will be the main actor in the mean error.

2. Case 2 : $y^* = \hat{y}^* = 1$. $N(x) = C(0,1) * P(\mathbf{Y} = 0)$ will be a function of the low $C(0,1)$ coefficient, and $V(x) = C(1,0) * P(\hat{\mathbf{Y}} = 0)$ will be a function of the high coefficient $C(1,0)$. Therefore the variance will tend to dominate the mean error. Notice too that the noise coefficient might become negative as $\frac{C(y^*,\hat{y})}{C(\hat{y},y^*)}$ will be high. It can even decrease the error coming from variance.

3. Case 3 : $y^* = 0$ and $\hat{y}^* = 1$, so bias is low. $N(x) = C(1,0) * P(\mathbf{Y} = 1)$ will be a function of the high $C(1,0)$ coefficient, and $V(x) = C(1,0) * P(\hat{\mathbf{Y}} = 0)$ will be a function of the high coefficient $C(1,0)$. The $V(x)$ coefficient $\frac{C(y^*,\hat{y}^*)}{C(\hat{y}^*,y^*)}$ will be low and negative, so the global variance component will be proportional to the low bias cost. The noise will therefore tends to dominate.

4. Case 4 : $y^* = 1$ and $\hat{y}^* = 0$, so bias is high. $N(x) = C(0,1) * P(\mathbf{Y} = 0)$ will be a function of the low $C(0,1)$ coefficient, and $V(x) = C(0,1) * P(\hat{\mathbf{Y}} = 1)$ will be a function of the low coefficient $C(0,1)$. The $V(x)$ coefficient $\frac{C(y^*,\hat{y}^*)}{C(\hat{y}^*,y^*)}$ will be high and negative, so the global variance component will be proportional to the high bias cost. The term $\frac{C(y^*,\hat{y})}{C(\hat{y},y^*)}$ in the noise coefficient is likely to be negative if there is some variance. In this case, the bias will tend to dominate.

As before, the three components strongly interact with one another, but in a rather more complicated way. On unbiased cases, if optimal class is 1, variance is likely to dominate (expensive bad prediction), and if optimal class is 0, noise is likely to dominate (expensive cost of ill-determinated class at x). On biased examples, if optimal class is 0, noise is likely to dominate (expensive cost of ill-determinated class at x). If optimal class is 1, bias is likely to dominate. These are general trends that rather apply in cases where noise and variance have about equal importance.

8 Conclusion

The main purpose of this report was to present the differences between classification and regression problems, the consequences it had on the choice of

the loss function, and finally to discuss the conceptual desiderata one expects from a generalization error decomposition into noise, bias and variance components.
This report has covered the following points:

1. Generelization error decomposition of regression and classification models have different characteristics. These differences are due to the nature of the output space, which is infinite and ordered in regression problems, and finite and nominal in classification problems. Consequently, chances for phenomenon noise and model variance to interact are statistically zero in regression, whereas in classification this probability increases as the number of classes decreases. Therefore, whereas in regression the error coming from the noise, bias and variance components is purely additive, there can be interactions in a classification problem, which will sometimes decrease the absolute value of some of these components.

2. The concepts of boundary bias and oversmoothing classifiers defined by Friedman in section 5 help to explain why simple models like K-Nearest Neighbours or Naive Bayes classifiers can sometimes perform better than more sophisticated models. It is however emphasized that this is in no case a general rule.

3. The concepts of noise, bias, and variance have been for long accepted in regression problems. However, they remained until recently ill-defined in classification problems. With the concepts defined by Domingos, it becomes possible to define noise, bias, and variance in a consistent way for quadratic and misclassification loss functions.

4. These decompositions allow better understanding of certain types of machine learning algorithms. For example, bagging [1] or ensemble techniques [7], can be easily explained through the variance term of the decomposition. Breaking down classification problems with many classes to a set of 2-class classification problems should also provide better performances given the interaction between noise and variance components in classification problems.

9 acknowledgments

This report was supported by the **COMP^{2}SYS** project, sponsored by the Human Resources and Mobility program of the European Community (MEST-CT-2004-505079).

References

[1] L. Breiman. Bagging predictors. Technical report, Department of Statistics, University of California, 1994.

[2] T. Dietterich. Machine learning bias, statistical bias, and statistical variance of decision tree algorithms, 1995.

[3] P. Domingos and G. Hulten. A unified bias-variance decomposition and its applications. In *Proceedings of the 17th International Conference on Machine Learning*, pages 231–238, 2000.

[4] P. Domingos and G. Hulten. A unified bias-variance decomposition for zero-one and squared loss. In *Proceedings of the 17th International Conference on Artificial Intelligence*, pages 564–569, 2000.

[5] J. H. Friedman. On bias, variance, 0/1 loss, and the curse of dimensionality. In *Data Mining and Knwledge Discovery*, pages 55–77, 1996.

[6] J. Furnkranz. Round robin classification. *Journal of Machine Learning*, 2002.

[7] J. Han H. Wang; W. Fan; P. Yu. Mining concept-drifting data streams using ensemble classifiers. In *SIGKDD*, 2003.

[8] E. Kong and T. Dietterich. Error-correcting output coding corrects bias and variance. In *Proceedings of the 12th International Conference on Machine Learning*, 1995.

[9] D. H. Wolpert R. Kohavi. Bias plus variance decomposition for zero-one loss functions. In *Proceedings of the 13th Internation Conference on Machine Learning*, pages 275–283, 1996.