NOTES ON THE BIAS-VARIANCE DECOMPOSITION

IORDAN GANEV

1. Introduction

In supervised statistical learning, one seeks to understand the relationship between a *predictor* variable, often denoted x, and a *response* variable, often denoted y. The predictor generally belongs to a vector space of some (finite) dimension, which we denote as $\mathbb{R}^{n_x}$. The response either also belongs to a vector space or belongs to a finite set of classes. In the former case, we are dealing with a *regression* problem, while in the latter case we are dealing with a *classification* problem.

In either case, a common procedure to understand the relationship between the predictor and the response is as follows:

(1) Collect observations $(x_1, y_1), \ldots, (x_n, y_n)$. This is the so-called *training data* One assumes that these samples are independent. One also assumes they are identically distributed, so that the sampling method is sufficiently random and does not prefer some observations over others.

(2) Based on the collected data, derive a model that assigns a response y to every predictor x. For the purposes of these notes, we assume a fixed mechanism for producing models from training data, such as linear regression, logistic regression, etc.

To introduce notation, let $D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ denote the collected data, and let

$$m_D : \{\text{predictors}\} \to \{\text{responses}\}$$

be the *model* produced by the training data D, which assigns responses to predictors based on the data D. So, given a predictor $x \in \mathbb{R}^{n_x}$, we have its response $m_D(x)$. Our focus will be:

Question: How can we evaluate the quality of the model m_D?

1.1. Training error.

One approach to answering the above question is to examine the performance of the model on the training data. The general formula for this is:

$$\text{Training error :} \qquad \text{Err}^{(D)} = \frac{1}{n} \sum_i \text{Cost}(y_i, m_D(x_i))$$

where Cost is a cost function that encodes the penalty when the sampled response y_i is sufficiently different from the estimated response $m_D(x_i)$ for the sampled predictor. For regression, one commonly takes the cost function to be the Euclidean distance, so that $\text{Cost}(y_i, m_D(x_i)) = |y_i - m_D(x_i)|^2$. Then the error is the *mean square error*. For classification, the cost function may be taken to be the 0/1 function, defined as: $\text{Cost}(y_i, m_D(x_i)) = 0$ if $y_i = m_D(x_i)$ and 1 otherwise. Then the error is the *misclassification rate*. In what follows, for concreteness, we these two common choices.

1.2. **Test error.** However, the training error has the limitation that it is based solely on the observations that were used to produce the model m_D. A better assessment of the quality of a model would be to evaluate its performance on new sample, different from those used to produce the model. In fact, it would be best to understand the performance of the model on the space of all possible new samples.

To make some headway on this task, we examine what it means to sample a pair (x, y). We imagine that there is a probability density function on such pairs:

$$p : \{(\text{predictor}, \text{response})\} \to [0, 1]$$

which controls the probability of sampling a given pair. Similarly, the training data is drawn from the space of n-tuples of predictor-response pairs: $\{(\text{predictor}, \text{response})\}^n$. The assumptions of independence and identical distribution imply that the training data is sampled according to the n-fold density of p:

$$p^n : \{(\text{predictor}, \text{response})\}^n \to [0, 1], \qquad p^n(D) = \prod_{i=1}^{n} p(x_i, y_i)$$

where $D = \{(x_i, y_i)\}_{i=1}^n$. We can now formulate the *test error* of the model m_D at x as:

$$\text{Err}_x = \begin{cases} E_{D,y}(|m_D(x) - y|^2 \mid X = x) & \text{for regression} \\ \\ \Pr(m_D(x) \neq y \mid X = x) & \text{for classification} \end{cases}$$

In other words, for regression, the test error at x is the expected distance between the model value $m_D(x)$ and the response y, while for classifications, the test error at x is the probability of a mismatch between the model's classification $m_D(x)$ and the response y. These quantities are computed considering all possible choices of training data D and responses y. The former are sampled according to the conditional density of p given x, while the latter are sampled according to the density p^n. Note that the test error is a theoretical quantity that can only be computed with knowledge of the full probability density function p.

1.3. **Noise, Variance, and Bias.** In later sections, we relate the test error to the noise, the variance, and the bias. For now, we give an intuitive summary of each of these quantities. As with the test error, these are theoretical quantities that can only be computed precisely through the density function.

The *noise* at x is a measure of the variability of the response around the expected response at x under the density p. For regression, the expected response is the expected value of y given x, while for classification, the expected response is the class with the highest conditional probability given x. With high noise, there is less of a deterministic relationship between the prediction and the response. With low noise, the relationship between the predictor and the response is stronger, and the expected label of x is likely to occur.

The *variance* is a measure of the sensitivity of the model to the training data. With high variance, the model will *overfit* the training data and there will be a wide gap between the test error and the training error. With low variance, the model will be robust to changes in the training data. The variance is related to the flexibility of the model class from which m_D is chosen. More flexible model classes have many models to choose from; since the space of possible models is greater, the variance is generally greater. Conversely, less flexible model classes have fewer models to choose from; a training data set is more likely to produced a model close to the expected model.

The *bias* is a measure of how the model could possibly be expected do to model a real-life situation. It encodes the inherent limitations of our model class, and (as with the variance) is closely related to the flexibility of the model class. However, (unlike the variance) the bias decreases with higher flexibility. Indeed, with a more flexible model class, there is a greater possibility of finding a model in that class that fits the true relationship between the predictor and response, to the extent that there is such a true relationship. Conversely, with less flexible model classes, one may expect that there will be no possible model within the class that effectively captures the relationship between the predictor and response.

1.4. **Bias-variance trade-off.** The relationship between the bias, variance, and flexibility is known as the *bias-variance trade-off*. To summarize:

- As the flexibility of the model class increases, the variance increases.
- As the flexibility of the model class increases, the bias decreases.

Generally speaking, while flexibility is low and increasing, the bias decreases more quickly than the variance increases and the test error goes down. This is the *underfitting* regime. With high enough flexibility, the variance increases more quickly than the bias decreases, and the test error goes up again. This is the *overfitting* regime. Hence, the test error is a U-shaped curve as a function of the flexibility of the model class. In the ideal situation, one seeks a level of flexibility at the trough of the test error curve, thus minimizing the test error.

2. Bias-Variance decomposition for regression

In this section, we work in the setting of regression with the mean square error. We show that the test error can be decomposed into three parts: the noise, the variance, and the bias.

2.1. **Joint density and noise.** As before, let $\mathbb{R}^{n_x}$ be the space of predictors. Since we are working in the setting of regression, the response variable also belongs to a vector space, call it $\mathbb{R}^{n_y}$. The joint probability density function is on the product of these two vector spaces:

$$p : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \to [0, 1]$$

The *expected label* is the function:

$$\overline{y} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_y}, \qquad x \mapsto E(Y|X = x) = \frac{\int_y y p(x, y) dy}{\int_{y'} p(x, y') dy'}$$

The *noise* is the variability of a response from its expected label:

$$\text{Noise}(p) = E_{x,y}\left[(y - \overline{y}(x))^2\right] = \int_x \int_y (y - \overline{y}(x))^2 p(x, y) dy dx$$

Hence, with high noise, there is less of a deterministic relationship between the prediction and the response. With low noise, the relationship between the predictor and the response is stronger. One may also consider the noise at x, which is defined as $\text{Noise}_x = E_y(|y - \overline{y}(x)|^2 \mid X = x)$.

2.2. **The model class.** As above, suppose we have a procedure for producing a function $\mathbb{R}^{n_x} \times \mathbb{R}^{n_y}$ from a (finite) collection of training data $\{(x_i, y_i)\}$. Formally, we have a *model selection* function:

$$\mathcal{M} : \bigcup_{n=1}^{\infty} (\mathbb{R}^{n_x} \times \mathbb{R}^{n_y})^n \to \text{Fun}(\mathbb{R}^{n_x}, \mathbb{R}^{n_y})$$
$$D \mapsto m_D$$

which (deterministically) returns a function $m_D : \mathbb{R}^{n_x} \to \mathbb{R}^{n_y}$ based on training data D. Here $\bigcup_{n=1}^{\infty} (\mathbb{R}^{n_x} \times \mathbb{R}^{n_y})^n$ is the set of finite tuples $\{(x_i, y_i)\}$ of any size[1]. The *model class* of $\mathcal{M}$ is the set of all functions that can be modeled through this procedure. In other words, the model class is the image of the function $\mathcal{M}$.

2.3. **Expected classifier, bias, and variance.** Fix n and recall that the density p gives rise to a corresponding density on the n-sample *training data* is $(\mathbb{R}^{n_x} \times \mathbb{R}^{n_y})^n$ defined by $p^n(D) = \prod_{i=1}^n p(x_i, y_i)$ for $D = \{(x_i, y_i)\}_{i=1}^n$. The expected classifier produced by $\mathcal{M}$ is given by taking the point-wise expectation of m_D over all possible choices of training data:

$$\overline{m} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_y}, \qquad x \mapsto E_D[m_D(x)] = \int_D m_D(x) p^n(D) dD$$

[1]One may decide to quotient each $\mathbb{R}^{n_x} \times \mathbb{R}^{n_y}$ by the action of the symmetric group S_n.

The variance of the classifier produced by $\mathcal{M}$ is the expected squared difference between a classifier and the expected classifier, taken over all possible inputs:

$$\mathrm{Var}(\mathcal{M}, p) = E_{D,x}\left[(m_D(x) - \overline{m}(x))^2\right]$$

The bias-squared of the classifier is the expected squared difference between the expected classifier and the expected label, taken over all possible inputs:

$$\mathrm{Bias}^2(\mathcal{M}, p) = E_x\left[(\overline{m}(x) - \overline{y}(x))^2\right]$$

2.4. **Expected test error.** The expected test error is the mean square error of a classifier m_D over all possible test pairs (x, y) and all possible training data:

$$\mathrm{Err}(\mathcal{M}, p) = E_{D,x,y}\left[(m_D(x) - y)^2\right]$$

Proposition 2.1. *The expected test error decomposes as the sum of the variance, bias-squared, and noise:*

$$\mathrm{Err}(\mathcal{M}, p) = \mathrm{Noise}(p) + \mathrm{Var}(\mathcal{M}, p) + \mathrm{Bias}^2(\mathcal{M}, p)$$

Proof. We write:

$$\begin{aligned}(m_D(x) - y)^2 &= (m_D(x) - \overline{m}(x) + \overline{m}(x) - \overline{y}(x) + \overline{y}(x) - y)^2 \\ &= (m_D(x) - \overline{m}(x))^2 + (\overline{m}(x) - \overline{y}(x))^2 + (\overline{y}(x) - y)^2 + \text{cross terms}\end{aligned}$$

Thus, it suffices to show that the expected value of each of the cross terms is zero. For example,

$$\begin{aligned}E_{D,x,y}[(m_D(x) - \overline{m}(x))(\overline{m}(x) - \overline{y}(x))] &= E_{x,y}[E_D[m_D(x) - \overline{m}(x)](\overline{m}(x) - \overline{y})] \\ &= E_{x,y}[(E_D[m_D(x)] - \overline{m}(x))(\overline{m}(x) - \overline{y}(x))] \\ &= E_{x,y}[(\overline{m}(x) - \overline{m}(x))(\overline{m}(x) - \overline{y}(x))] = 0\end{aligned}$$

The computation that $E_{D,x,y}[(m_D(x) - \overline{m}(x))(\overline{y}(x) - y)] = 0$ follows the exact same logic. Finally,

$$\begin{aligned}E_{D,x,y}[(m_D(x) - \overline{m}(x))(\overline{y}(x) - y)] &= E_x[(m_D(x) - \overline{m}(x))E_y[(\overline{y}(x) - y) \mid X = x]] \\ &= E_x[(m_D(x) - \overline{m}(x))(\overline{y}(x) - E_y[y \mid X = x]) = 0\end{aligned}$$

This completes the proof. □

2.5. **Summary.** We have three functions of x: $\overline{y}$, $\overline{m}$, and m_D. Working relative to a point x, the expected square difference between $\overline{y}$ and y is the noise, the expected square difference between $\overline{y}$ and $\overline{m}$ is the bias-squared, and the expected square difference between $\overline{m}$ and m_D is the variance. Hence, the variance is the sensitivity of the approximating function to the training data, while the bias is the error from the inherent limitation of the model class.

3. Bias-Variance decomposition for classification

We now turn our attention to the classification setting. In this case, we can compute explicit expressions for the test error, noise, variance, and bias, but the decomposition of the regression setting does not hold.

3.1. **Joint density.** The predictor is still assumed to be continuous and belonging to $\mathbb{R}^{n_x}$, while the response belongs to one of finitely many classes. More formally, let:

$$\Omega = \Omega_Y = \{1, \ldots, C\}$$

be the possible responses. We have a density function on the space of predictor-response pairs:

$$p : \mathbb{R}^{n_x} \times \Omega \to [0, 1]$$

We denote by $p_y(x)$ the conditional probability for each class $y \in \Omega$ given a value of the predictor $x \in \mathbb{R}^{n_x}$, that is: conditional probabilities as $p_y(x)$:

$$p_y : \mathbb{R}^{n_x} \to [0, 1], \qquad p_y(x) := p(Y = y \mid X = x) = \frac{p(x, y)}{\sum_{y' \in \Omega} p(x, y')}$$

Given $x \in \mathbb{R}^{n_x}$, the conditional probabilities define a probability measure on Ω where the measure of $\{y\}$ is $p_y(x)$. The *expected label* of a given predictor value $x \in \mathbb{R}^{n_x}$ is defined as the class with the highest conditional probability. As a function:

$$\overline{y} : \mathbb{R}^{n_x} \to \Omega, \qquad \overline{y}(x) = \operatorname*{argmax}_{y \in \Omega} p_y(x)$$

Note that, in this setting of classification, argmax_y plays the role that the expected value E_y played in the regression setting. Indeed, since the classes are not linear, the usual mathematical notion of expected value generally does not make sense here.

3.2. **Model.** Similarly to the regression setting, suppose we have an algorithm that takes as input a (finite) collection of training data $\{(x_i, y_i) \in \mathbb{R}^{n_x} \times \Omega\}$ and outputs an estimate for the conditional probabilities. Formally, we have a function:

$$\bigcup_{n=1}^{\infty} (\mathbb{R}^{n_x} \times \Omega)^n \to \operatorname{Fun}\left(\mathbb{R}^{n_x}, \operatorname{PrMeasure}(\Omega)\right)$$

$$D \mapsto [x \mapsto [\{y\} \mapsto \hat{p}_y^{(D)}(x)]]$$

that takes in training data and outputs a probability measure on Ω for every predictor $x \in \mathbb{R}^{n_x}$. Going one step further, this procedure allows us to assign a class to each predictor by taking the label with the highest conditional probability estimate. In notation, define:

$$m_D : \mathbb{R}^{n_x} \to \Omega, \qquad x \mapsto \operatorname*{argmax}_{y \in \Omega} \hat{p}_y^{(D)}(x)$$

Thus, we have a *model selection* function

$$\mathcal{M} : \bigcup_{n=1}^{\infty} (\mathbb{R}^{n_x} \times \Omega)^n \to \mathrm{Fun}(\mathbb{R}^{n_x}, \Omega)$$
$$D \mapsto m_D$$

3.3. **The space of training data.** Fix n to be the number of training samples. As above, the density p induces a density on the space of n-sample *training data*, namely, $p^n : (\mathbb{R}^{n_x} \times \Omega)^n \to [0,1]$, with $p^n(D) = \prod_{i=1}^n p(x_i, y_i)$ for for $D = \{(x_i, y_i)\}_{i=1}^n$. Using this density function, for $y \in \Omega$, we identify the following subset of the space of training data:

$$S_y(x) = \{D \in (\mathbb{R}^{n_x} \times \Omega)^n \mid m_D(x) = y\}$$

In other words, $S_y(x)$ is the set of training data that produce a classifier assigning the label y to x. For fixed $x \in \mathbb{R}^{n_x}$, the subsets $\{S_y(x)\}_{y\in\Omega}$ form a partition of the space of training data. Now set $q_y(x)$ to be the probability that the training data D lies in $S_y(x)$ given the predictor x and the class y. That is, we have:

$$q_y : \mathbb{R}^{n_x} \to [0,1], \qquad q_y(x) = \Pr(D \in S_y(x) \mid X = x, Y = y)$$

We note that both $S_y(x)$ and $q_y(x)$ depend on the model class $\mathcal{M}$ and the density p; we suppress this additional notatin. Finally, we define the expected classifier as the function assigning a predictor x to the class with the largest probability of being chosen by the training data:

$$\overline{m} : \mathbb{R}^{n_x} \to \Omega, \qquad \overline{m}(x) = \operatorname*{argmax}_{y\in\Omega} q_y(x) = \operatorname*{argmax}_{y\in\Omega} \Pr(m_D(x) = y)$$

3.4. **Computation of probabilities.** Fix a predictor value $x \in \mathbb{R}^{n_x}$. Define the error, noise, variance, and squared bias at x as:

$$\begin{aligned}
\mathrm{Err}_x(\mathcal{M}, p) &= \Pr(y \neq m_D(x) \mid X = x) \\
\mathrm{Noise}_x(\mathcal{M}, p) &= \Pr(y \neq \overline{y}(x) \mid X = x) \\
\mathrm{Var}_x(\mathcal{M}, p) &= \Pr(\overline{m}(x) \neq m_D(x) \mid X = x) \\
\mathrm{Bias}_x^2(\mathcal{M}, p) &= \Pr(\overline{m}(x) \neq \overline{y}(x) \mid X = x)
\end{aligned}$$

Proposition 3.1. *Given $x \in \mathbb{R}^{n_x}$, we have:*

$$\begin{aligned}
\mathrm{Err}_x(\mathcal{M}, p) &= 1 - \sum_{y\in\Omega} p_y(x) q_y(x) \\
\mathrm{Noise}_x(p) &= 1 - p_{\overline{y}(x)}(x) = 1 - \max_y p_y(x) \\
\mathrm{Var}_x(\mathcal{M}, p) &= 1 - q_{\overline{m}(x)}(x) = 1 - \max_y q_y(x) \\
\mathrm{Bias}_x^2(\mathcal{M}, p) &= \begin{cases} 1 & \text{if } \overline{y}(x) \neq \overline{m}(x) \\ 0 & \text{else} \end{cases}
\end{aligned}$$

Proof. For readability, we suppress the notation $(\mathcal{M}, p)$. For the first claim, we compute:

$$\begin{aligned}\mathrm{Err}_x &= \Pr(y \neq m_D(x) \mid X = x) \\ &= \sum_y \Pr(y \neq m_D(x) \mid X = x\ ,\ Y = y)\Pr(Y = y \mid X = x) \\ &= \sum_y (1 - \Pr(D \in S_y(x)))p_y(x) = \sum_y (1 - q_y(x))p_y(x) \\ &= 1 - \sum_y p_y(x)q_y(x)\end{aligned}$$

For the second claim:

$$\mathrm{Noise}_x = \Pr(y \neq \overline{y}(x) \mid X = x) = \sum_{y \neq \overline{y}(x)} \Pr(Y = y \mid X = x) = 1 - p_{\overline{y}(x)}(x)$$

For the third claim:

$$\mathrm{Var}_x = \Pr(m_D(x) \neq \overline{m}(x) \mid X = x) = \sum_{y \neq \overline{m}(x)} \Pr(D \in S_y(x) \mid X = x) = 1 - q_{\overline{m}(x)}(x)$$

The last claim is immediate. □

3.5. **Two class case.** We examine the case of two classes in more detail.

Proposition 3.2. *Suppose* $\Omega = \{0, 1\}$*.For any* $x \in \mathbb{R}^{n_x}$*, we have:*

$$\mathrm{Err}_x(\mathcal{M}, p) = \mathrm{Noise}_x(p) + |(p_1(x) - p_0(x))(\mathrm{Var}_x(\mathcal{M}, p) - \mathrm{Bias}_x^2(\mathcal{M}, p))| \tag{3.1}$$

Proof. Set $p(x) = p_1(x) = 1 - p_0(x)$ and $q(x) = q_1(x) = 1 - q_0(x)$. Then:

$$\begin{aligned}\mathrm{Err}_x &= p(x) + q(x) - 2p(x)q(x) \\ \mathrm{Noise}_x &= \min(p(x), 1 - p(x)) \\ \mathrm{Var}_x &= \min(q(x), 1 - q(x)) \\ \mathrm{Bias}_x^2 &= \begin{cases} 1 & \text{if } p(x) \leq 1/2 < q(x) \text{ or } q(x) \leq 1/2 < p(x) \\ 0 & \text{otherwise} \end{cases}\end{aligned}$$

There are four cases:

(1) $p(x) \leq 1/2$ and $q(x) \leq 1/2$, so that $\mathrm{Noise}_x = p(x)$, $\mathrm{Var}_x = q(x)$, and $\mathrm{Bias}_x^2 = 0$.
(2) $p(x) \leq 1/2$ and $q(x) > 1/2$, so that $\mathrm{Noise}_x = p(x)$, $\mathrm{Var}_x = 1 - q(x)$, and $\mathrm{Bias}_x^2 = 1$.
(3) $p(x) > 1/2$ and $q(x) \leq 1/2$, so that $\mathrm{Noise}_x = 1 - p(x)$, $\mathrm{Var}_x = q(x)$, and $\mathrm{Bias}_x^2 = 1$.
(4) $p(x) > 1/2$ and $q(x) > 1/2$, so that $\mathrm{Noise}_x = 1 - p(x)$, $\mathrm{Var}_x = 1 - q(x)$, and $\mathrm{Bias}_x^2 = 0$.

For each of these cases, a direct computation verifies that formula 3.1 holds. □

We examine Equation 3.1 in more detail. Note that the squared bias is either 0 or 1, depending on whether $p(x)$ and $q(x)$ lie on the same side of $1/2$ or not, respectively. If the squared bias is zero, then we have:

$$\text{Err}_x = \text{Noise}_x + |2p(x) - 1|\text{Var}_x$$

Thus, decreasing the variance decreases the error. On the other hand, suppose the squared bias is one, so that $p(x)$ and $q(x)$ lie on opposite sides of $1/2$. Then, noting that the variance is at most $1/2$, we have:

$$\text{Err}_x = \text{Noise}_x + |2p(x) - 1|(1 - \text{Var}_x)$$

In this case, increasing the variance will lead to a decrease in the error.