Bias-Variance Analysis: Theory and Practice

Anand Avati

1 Introduction

In this set of notes, we will explore the fundamental Bias-Variance tradeoff in Statistics and Machine Learning under the squared error loss. The concepts of Bias and Variance are slightly different in the contexts of Statistics vs Machine Learning, though the two are closely related in spirit. We will first start with the classical notions from Statistics, using Linear Regression with L_2-regularization as a case study. The simplicity of Linear Regression allows us to derive closed form expressions for the Bias and Variance terms and appreciate the tradeoff better. Then we will study the notion of Bias and Variance and their decomposition in the context of Machine Learning (prediction), and see the connections to the classical notions using L_2-regularized Linear Regression as an example.

Throughout this document we will use the following notation. We are given an i.i.d. data set $S = \{(x^{(i)}, y^{(i)})\}_{i=1}^{n}$ that was generated from some data generating probability distribution having some unknown (constant) parameter $\theta^* \in \mathbb{R}^d$. Here $x^{(i)} \in \mathbb{R}^d$ and $y^{(i)} \in \mathbb{R}$. For notational convenience, let $X \in \mathbb{R}^{n \times d}$ denote the design matrix, and $\vec{y} \in \mathbb{R}^n$ the vector of labels. In the case of regression X is considered given (constant).

2 Bias and Variance in Statistical Inference

We start with the classical setting of statistical inference. Our goal in statistical inference is to construct an estimator for the unknown parameter θ^* given the observed data set S.

Let us indicate our estimator as $\hat{\theta}_n$, where n is the size of the dataset used to fit the model. For example, in the case of Linear Regression, $\hat{\theta}_n = (X^T X)^{-1} X^T \vec{y}$. Note that $\hat{\theta}_n$ is a random variable even though θ^* was not. This is because $\hat{\theta}_n$ is a (deterministic) function of the noisy data set S, where noise is typically in the labels $\vec{y}$. It is worth noting that the randomness in $\hat{\theta}_n$ therefore indirectly depends on θ^*, due to this noise. The distribution of $\hat{\theta}_n$ is commonly called the *Sampling distribution*. The Bias and Variance of the estimator $\hat{\theta}_n$ are just the (centered) first and second moments of its sampling distribution.

We call $\text{Bias}(\hat{\theta}_n) \equiv \mathbb{E}[\hat{\theta}_n - \theta^*]$ the *Bias* of the estimator $\hat{\theta}_n$. The estimator $\hat{\theta}_n$ is called *Unbiased* if $\mathbb{E}[\hat{\theta}_n - \theta^*] = 0$ (i.e. $\mathbb{E}[\hat{\theta}_n] = \theta^*$) for all values of θ^*.

Similarly, we call $\text{Var}(\hat{\theta}_n) \equiv \text{Cov}[\hat{\theta}_n]$ the *Variance* of the estimator. Note that, unlike Bias, the Variance of the estimator does not directly depend on the true parameter θ^*.

The Bias and Variance of an estimator are not necessarily directly related (just as how the first and second moment of any distribution are not necessarily related). It is possible to have estimators that have high or low bias and have either high or low variance. Under the squared error, the Bias and Variance of an estimator are related as:

$$
\begin{aligned}
\text{MSE}(\hat{\theta}_n) &= \mathbb{E}\left[\|\hat{\theta}_n - \theta^*\|^2\right] \\
&= \mathbb{E}\left[\|\hat{\theta}_n - \mathbb{E}[\hat{\theta}_n] + \mathbb{E}[\hat{\theta}_n] - \theta^*\|^2\right] \\
&= \mathbb{E}\left[\|\hat{\theta}_n - \mathbb{E}[\hat{\theta}_n]\|^2 + \underbrace{\|\mathbb{E}[\hat{\theta}_n] - \theta^*\|^2}_{\text{Constant}} + 2(\underbrace{\hat{\theta}_n - \mathbb{E}[\hat{\theta}_n]}_{\text{Zero Mean}})^T(\mathbb{E}[\hat{\theta}_n] - \theta^*)\right] \\
&= \mathbb{E}\left[\|\hat{\theta}_n - \mathbb{E}[\hat{\theta}_n]\|^2\right] + \|\mathbb{E}[\hat{\theta}_n] - \theta^*\|^2 \\
&= \mathbb{E}\left[\text{tr}\left[(\hat{\theta}_n - \mathbb{E}[\hat{\theta}_n])(\hat{\theta}_n - \mathbb{E}[\hat{\theta}_n])^T\right]\right] + \|\mathbb{E}[\hat{\theta}_n] - \theta^*\|^2 \\
&= \text{tr}\left[\text{Var}(\hat{\theta}_n)\right] + \|\text{Bias}(\hat{\theta}_n)\|^2.
\end{aligned}
$$

It is quite often the case that techniques employed to reduce Variance results in an increase in Bias, and vice versa. This phenomenon is called the *Bias Variance Tradeoff*. Balancing the two evils (Bias and Variance) in an optimal way is at the heart of successful model development. Now we will do a case study of Linear Regression with L_2-regularization, where this trade-off can be easily formalized.

2.1 Bias Variance Tradeoff in Linear Regression with L_2-regularization

Recall that in Linear Regression we make the assumption $y^{(i)} = \theta^{*T} x^{(i)} + \epsilon^{(i)}$ where each $\epsilon^{(i)} \sim \mathcal{N}(0, \tau^2)$ i.i.d.. We assume X is given, and hence constant. For notational simplicity let $\vec{\epsilon} \in \mathbb{R}^n \sim \mathcal{N}(\vec{0}, \tau^2 I)$ where $\vec{\epsilon}_i = \epsilon^{(i)}$. So, $\vec{y} = X\theta^* + \vec{\epsilon}$. Recall that Linear Regression with L_2-regularization (with regularization parameter $\lambda > 0$) minimizes the cost function

$$J(\theta) = \frac{\lambda}{2}\|\theta\|_2^2 + \frac{1}{2}\sum_{i=1}^{n}\left(y^{(i)} - \theta^T x^{(i)}\right)^2,$$

and enjoys a closed form solution

$$\begin{aligned}
\hat{\theta}_n &= \arg\min_{\theta \in \mathbb{R}^d} J(\theta) \\
&= \arg\min_{\theta \in \mathbb{R}^d} \left[\frac{\lambda}{2}\|\theta\|_2^2 + \frac{1}{2}\sum_{i=1}^{n}\left(y^{(i)} - \theta^T x^{(i)}\right)^2\right] \\
&= \arg\min_{\theta \in \mathbb{R}^d} \left[\frac{\lambda}{2}\|\theta\|_2^2 + \frac{1}{2}\|X\theta - \vec{y}\|_2^2\right] \\
&= (X^T X + \lambda I)^{-1} X^T \vec{y}.
\end{aligned}$$

Consider the eigendecomposition of the symmetric Positive Semi Definite (PSD) matrix $X^T X$:

$$X^T X = U \underbrace{\begin{bmatrix} \sigma_1^2 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma_d^2 \end{bmatrix}}_{\text{diag}\left(\sigma_1^2, \dots, \sigma_d^2\right)} U^T,$$

where $U^T U = U U^T = I$, and $\{\sigma_1^2, \sigma_2^2, \dots, \sigma_d^2\}$ are the eigenvalues, where some of the σ_i^2 could be 0. However, even when X (and hence $X^T X$) is not full rank, $(X^T X + \lambda I)$ is always symmetric and Positive Definite (PD) since we are adding $\lambda > 0$ to all the diagonal elements of $X^T X$:

$$X^T X + \lambda I = U \begin{bmatrix} \sigma_1^2 + \lambda & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma_d^2 + \lambda \end{bmatrix} U^T.$$

This implies

$$\left(X^TX+\lambda I\right)^{-1} = U\begin{bmatrix} \frac{1}{\sigma_1^2+\lambda} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \frac{1}{\sigma_d^2+\lambda} \end{bmatrix} U^T.$$

Therefore $\hat{\theta}_n$ always exists and is unique. Now we analyze the Bias and Variance of $\hat{\theta}_n$. We start with the expression for the estimator

$$\begin{aligned}
\hat{\theta}_n &= (X^TX+\lambda I)^{-1}X^T\vec{y} \\
&= (X^TX+\lambda I)^{-1}X^T(X\theta^*+\vec{\epsilon}) \\
&= \left[\left(X^TX+\lambda I\right)^{-1}X^TX\right]\theta^* + \left[\left(X^TX+\lambda I\right)^{-1}X^T\right]\vec{\epsilon}
\end{aligned}$$

To compute the Bias of this model, we take the expectation of the above and observe that (remember, X is considered constant in regression):

$$\begin{aligned}
\mathbb{E}[\hat{\theta}_n] &= \mathbb{E}\left[\left[\left(X^TX+\lambda I\right)^{-1}X^TX\right]\theta^* + \left[\left(X^TX+\lambda I\right)^{-1}X^T\right]\vec{\epsilon}\right] \\
&= \left[\left(X^TX+\lambda I\right)^{-1}X^TX\right]\theta^* + \left[\left(X^TX+\lambda I\right)^{-1}X^T\right]\mathbb{E}\left[\vec{\epsilon}\right] \\
&= \left[\left(X^TX+\lambda I\right)^{-1}X^TX\right]\theta^* + \left[\left(X^TX+\lambda I\right)^{-1}X^T\right]\vec{0} \\
&= \left[\left(X^TX+\lambda I\right)^{-1}X^TX\right]\theta^* \\
&= U\begin{bmatrix} \frac{1}{\sigma_1^2+\lambda} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \frac{1}{\sigma_d^2+\lambda} \end{bmatrix} U^TX^TX\theta^* \\
&= U\begin{bmatrix} \frac{1}{\sigma_1^2+\lambda} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \frac{1}{\sigma_d^2+\lambda} \end{bmatrix} U^TU\begin{bmatrix} \sigma_1^2 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma_d^2 \end{bmatrix} U^T\theta^* \\
&= U\begin{bmatrix} \frac{1}{\sigma_1^2+\lambda} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \frac{1}{\sigma_d^2+\lambda} \end{bmatrix}\begin{bmatrix} \sigma_1^2 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma_d^2 \end{bmatrix} U^T\theta^* \\
&= U\begin{bmatrix} \frac{\sigma_1^2}{\sigma_1^2+\lambda} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \frac{\sigma_d^2}{\sigma_d^2+\lambda} \end{bmatrix} U^T\theta^*.
\end{aligned}$$

From the above, we can make a few observations. First, when $\lambda = 0$, we see that $\mathbb{E}[\hat{\theta}_n] = \theta^*$. This implies that standard linear regression estimator (without regularization) is Unbiased. Second, the above expression is essentially a "shrunk" θ^* because all the eigenvalues of the matrix are less than one. In fact, the more regularization we add (i.e. larger λ), the smaller the eigenvalues will be, and hence the stronger the "shrinkage" towards 0. This implies that the estimator $\hat{\theta}_n$ of L_2-regularized Linear Regression is Biased (towards 0 in this case).

Though we paid the price of adding regularization in the form of having a Biased estimator, we do however gain something in return: reduced variance. In order to analyze the variance of the estimator $\hat{\theta}_n$, first recall the following property of multivariate Gaussians:

$$\text{If } \vec{\epsilon} \sim \mathcal{N}(\vec{0}, \tau^2 I), \text{ then } A\vec{\epsilon} \sim \mathcal{N}(\vec{0}, A(\tau^2 I)A^T).$$

This gives us (again, remember X is given, and hence constant in regression):

$$\begin{aligned}
\text{Cov}[\hat{\theta}_n] &= \text{Cov}\left[\left(X^TX + \lambda I\right)^{-1} X^T \vec{y}\right] \\
&= \text{Cov}\left[\left(X^TX + \lambda I\right)^{-1} X^T \left(X\theta^* + \vec{\epsilon}\right)\right] \\
&= \text{Cov}\left[\underbrace{\left(X^TX + \lambda I\right)^{-1} X^TX\theta^*}_{\text{Constant}} + \left(\underbrace{\left(X^TX + \lambda I\right)^{-1} X^T}_{\text{Constant}}\right)\vec{\epsilon}\right] \\
&= \text{Cov}\left[\left(\left(X^TX + \lambda I\right)^{-1} X^T\right)\vec{\epsilon}\right] \\
&= \left[\left(X^TX + \lambda I\right)^{-1} X^T\right] \text{Cov}[\vec{\epsilon}] \left[\left(X^TX + \lambda I\right)^{-1} X^T\right]^T \qquad \text{(using above property)} \\
&= \left[\left(X^TX + \lambda I\right)^{-1} X^T\right] \tau^2 I \left[X\left(X^TX + \lambda I\right)^{-1}\right] \\
&= \tau^2 \left(X^TX + \lambda I\right)^{-1} \left(X^TX\right) \left(X^TX + \lambda I\right)^{-1} \\
&= \tau^2\ U \begin{bmatrix} \frac{1}{\sigma_1^2+\lambda} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \frac{1}{\sigma_d^2+\lambda} \end{bmatrix} U^T\ U \begin{bmatrix} \sigma_1^2 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma_d^2 \end{bmatrix} U^T\ U \begin{bmatrix} \frac{1}{\sigma_1^2+\lambda} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \frac{1}{\sigma_d^2+\lambda} \end{bmatrix} U^T \\
&= U \begin{bmatrix} \frac{\tau^2\sigma_1^2}{(\sigma_1^2+\lambda)^2} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \frac{\tau^2\sigma_d^2}{(\sigma_d^2+\lambda)^2} \end{bmatrix} U^T.
\end{aligned}$$

From the above expression we observe that as we add more regularization (i.e. larger λ), the smaller the spectrum (i.e. all the eigenvalues) of the covariance of the estimator $\hat{\theta}_n$, and hence smaller tr $\Big[\text{Var}(\hat{\theta}_n)\Big]$.

Thus we clearly see the Bias Variance trade-off as a function of λ. The larger the value of λ, the higher the Bias (undesirable) but also smaller the Variance (desirable) of $\hat{\theta}_n$, and vice versa. There exists a sweet spot for λ that minimizes the sum of the two evils, and finding that sweet spot is better explained in the context of prediction, which is the next section.

3 Bias and Variance in Prediction

In a prediction (Supervised Machine Learning) setting, our goals are different from statistical inference. Instead of constructing an estimator $\hat{\theta}_n$ for the unknown parameter, we wish to learn a function f that can predict y given x well (with respect to some loss function). As before, we are given a training set $S = \{(x^{(i)}, y^{(i)})\}_{i=1}^n$. We make the assumption that $y = f(x) + \epsilon$, where ϵ satisfies $\mathbb{E}[\epsilon] = 0$ and $\mathbb{V}[\epsilon] = \tau^2$ (ϵ is not necessarily Gaussian). Further, we *define* (*not assume*) the "true" f as

$$f(x') \equiv \mathbb{E}[y|x = x'].$$

Our task now is to construct a hypothesis $\hat{f}_n$ given a fixed size training set S that mimics f well on all future unseen examples. In other words, $\hat{f}_n$ needs to have good *generalization error*. We will only consider the case where the generalization error is the expected squared error loss on an unseen example.

Suppose $\hat{f}_n$ is obtained with some (unspecified) training process over S. As before, note that $\hat{f}_n$ is random, and the randomness comes due to the $\epsilon^{(i)}$'s embedded in the training set examples. Consider a new unseen example pair (y_*, x_*) and the corresponding generalization error, where the expectation is over the randomness in ϵ embedded in the test example, and in $\hat{f}_n$:

$$\begin{aligned}
\text{MSE}(\hat{f}_n) &= \mathbb{E}\left[\Big(y_* - \hat{f}_n(x_*)\Big)^2\right] \\
&= \mathbb{E}\left[\Big(\epsilon + f(x_*) - \hat{f}_n(x_*)\Big)^2\right]
\end{aligned}$$

$$
\begin{aligned}
&= \mathbb{E}\left[\epsilon^2\right] + \mathbb{E}\left[\left(f(x_*) - \hat{f}_n(x_*)\right)^2\right] + \mathbb{E}\left[2\epsilon(f(x_*) - \hat{f}_n(x_*))\right] \\
&= \mathbb{E}\left[\epsilon^2\right] + \mathbb{E}\left[\left(f(x_*) - \hat{f}_n(x_*)\right)^2\right] + \underbrace{\mathbb{E}\left[\epsilon\right]}_{=0}\mathbb{E}\left[2(f(x_*) - \hat{f}_n(x_*))\right] \qquad \text{(i.i.d. } \epsilon\text{)} \\
&= \mathbb{E}\left[\epsilon^2\right] + \mathbb{E}\left[\left(f(x_*) - \hat{f}_n(x_*)\right)^2\right] \\
&= \mathbb{E}\left[\epsilon^2\right] + \mathbb{E}\left[f(x_*) - \hat{f}_n(x_*)\right]^2 + \mathbb{V}\left[f(x_*) - \hat{f}_n(x_*)\right] \qquad (\mathbb{E}[X^2] = \mathbb{V}[X] + \mathbb{E}[X]^2) \\
&= \underbrace{\tau^2}_{\text{Irreducible error}} + \underbrace{\mathbb{E}\left[\hat{f}_n(x_*) - f(x_*)\right]^2}_{\text{Bias}^2} + \underbrace{\mathbb{V}\left[\hat{f}_n(x_*)\right]}_{\text{Variance}} \qquad (\mathbb{V}[a - X] = \mathbb{V}[X])
\end{aligned}
$$

The above decomposition suggests similarities with the statistical inference setting. The Bias and Variance are, as before, just the (centered) first and second moments of $\hat{f}_n$ (skipping x_* in the notation). Bias of $\hat{f}_n$ at input x_* is defined as $\mathbb{E}[\hat{f}_n - f]$, and Variance is $\mathbb{V}[\hat{f}_n]$. Such a clean decomposition into Bias and Variance terms exists only for the squared error loss. Proposals have been made for more general losses, though none are widely accepted.

3.1 Prediction with L_2-regularized Linear Regression

Now let us tie all this back to the case of L_2-regularized Linear Regression. Let us assume $f(x) = \theta^{*T}x$ where θ^* is the true unknown parameter. Now $\hat{f}_n(x) = \hat{\theta}_n^T x$ where $\hat{\theta}_n$ is the L_2-regularized Linear Regression estimator. We can see the relation between the Bias and Variance terms of prediction and of inference as follows:

$$
\begin{aligned}
\text{Bias}(\hat{f}_n) &= \mathbb{E}[\hat{f}_n(x) - f(x)] \\
&= \mathbb{E}[\hat{\theta}_n^T x - \theta^{*T} x] \\
&= \mathbb{E}[\hat{\theta}_n - \theta^*]^T x \\
&= \text{Bias}(\hat{\theta}_n)^T x. \\
\text{(and similarly)} \quad \text{Var}(\hat{f}_n) &= x^T \left[\text{Var}(\hat{\theta}_n)\right] x.
\end{aligned}
$$

The irreducible error appears only in the prediction setting, as it is an artifact of the noise in the *test example* (there is no such test example in the inference setting). In other words, the noise in the training data contributes to the

Variance term, and the noise in the test example manifests itself as the irreducible error term.

In order to minimize the generalization error, we need to reduce one or more of the decomposed components. There is nothing we can do to reduce irreducible error, since it is just noise in the data (i.e., the same x could have different y values in different examples). Thus we are left with balancing the Bias and Variance terms. In the case of L_2-regularized Linear Regression, we could consider adjusting the λ value. In the inference setting, we saw that increasing λ reduces the Variance but increases the Bias. This tradeoff directly translates into the prediction setting as well, based on the above relations. Back then it was not clear what might be a good sweet spot for setting the λ value. However in the prediction setting, there is an obvious answer: choose λ to be the value that minimizes the squared error (generalization error) in cross-validation.

4 Bias and Variance in practice

To wrap things up, we can relate the Bias Variance decomposition to the commonly used terms *overfitting* and *underfitting* in the following informal way:

- *Overfitting* relates to having a *High Variance* model or estimator. To fight overfitting, we need to focus on reducing the Variance of the estimator, such as: increase regularization, obtain larger data set, decrease number of features, use a smaller model, etc.
- *Underfitting* relates to having a *High Bias* model or estimator. To fight underfitting, we need to focus on reducing the Bias in the estimator, such as: decrease regularization, use more features, use a larger model, etc.

The first step in improving generalization error is to characterize which component in the decomposition has the highest contribution, and go after that component. Unfortunately there is no theoretically sound yet tractable way of calculating the breakdown. However there are certain heuristics that are extremely useful. Loosely speaking:

- Training error can be treated as the amount of Bias in the model or estimator. If the model is unable to fit the training data itself well, then it is likely that the model has High Bias. This is the underfitting regime.

- Gap between cross-validation error and Training error can be treated as the Variance of the model or the estimator. If the Training error is low but the Cross Validation error is high, it is very likely that model has High Variance. This is the overfitting regime.

We should ***always*** analyze the model performance by looking at the training error and cross-validation error *simultaneously*. This is the only tractable (albeit heuristic) way to obtain an estimate of the Bias and Variance components. Only then should we take steps that are targeted towards addressing either Bias or Variance purposefully.

Steps taken to fight overfitting (i.e. fight High Variance) generally do not necessarily help fight underfitting (i.e. High Bias). For example, it is futile to spend time and resources in obtaining more data (technique to fight High Variance) when the training error itself is high (symptom of High Bias).

Similarly steps taken to fight underfitting (i.e. fight High Bias) generally do not necessarily help fight overfitting (i.e. High Variance). For example, it is futile to switch to a larger neural network (technique to fight High Bias) when the gap between cross-validation error and training error is high (symptom of High Variance).

Many times steps taken to fight one (either High Bias or High Variance) can end up worsening the other. This is essentially how the Bias Variance trade-off is encountered in practice.