Consider the regression setting, where we would like to design a learning algorithm that given a dataset $\mathcal{D}$ consisting of N i.i.d. samples: $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$ ($\boldsymbol{x}_i \in \mathbb{R}^d, y \in \mathbb{R}$), produces a predictor function (or hypothesis) $h : \mathbb{R}^d \to \mathbb{R}$. In this section, we analyze the generalization error of a learning algorithm via the so-called *Bias-Variance Decomposition* (or *Tradeoff*).

To make things concrete for our analysis, assume we sample from the joint distribution $\boldsymbol{x}, y \sim P(\boldsymbol{X}, Y)$ [*], and therefore $\mathcal{D} \sim P^N(\boldsymbol{X}, Y)$ (i.e., independetly sample N times). Given a dataset $\mathcal{D}$, our learning algorithm produces a function $h_\mathcal{D} : \mathbb{R}^d \to \mathbb{R}$ (note that this is dependent on $\mathcal{D}$). And the quantity we are interested in, the expected test error is:

$$\mathbb{E}_{\mathcal{D}\sim P^N,\,(\boldsymbol{x},y)\sim P}\left[(h_\mathcal{D}(\boldsymbol{x}) - y)^2\right]. \tag{1}$$

Before we begin the analysis, let us first define a few useful notations:

- *(Expected Label)* Denote the conditional expectation of y given $\boldsymbol{x}$ as

$$\bar{y}(\boldsymbol{x}) = \mathbb{E}_{Y|X=x}[Y|\boldsymbol{X} = \boldsymbol{x}] = \int_y yP(y|\boldsymbol{X} = \boldsymbol{x})dy.$$

- *(Expected Prediction)* Fixing our learning algorithm, denote the expected prediction given $\boldsymbol{x}$ over the dataset $\mathcal{D}$ distribution as

$$\bar{h}(\boldsymbol{x}) = \mathbb{E}_{D\sim P^N}[h_\mathcal{D}(\boldsymbol{x})] = \int_\mathcal{D} h_\mathcal{D}(\boldsymbol{x})P(\mathcal{D})d\mathcal{D}. \tag{2}$$

We are now ready to derive the *Bias-Variance Decomposition* of (1):

$$\begin{aligned}
&\mathbb{E}_{\mathcal{D}\sim P^N,\,(\boldsymbol{x},y)\sim P}\left[(h_\mathcal{D}(\boldsymbol{x}) - y)^2\right] \\
&= \mathbb{E}_{\mathcal{D}\sim P^N,\,(\boldsymbol{x},y)\sim P}\left[\left((h_\mathcal{D}(\boldsymbol{x}) - \bar{h}(\boldsymbol{x})) + (\bar{h}(\boldsymbol{x}) - y)\right)^2\right] \\
&= \mathbb{E}_{\mathcal{D}\sim P^N,\,(\boldsymbol{x},y)\sim P}\left[\left(h_\mathcal{D}(\boldsymbol{x}) - \bar{h}(\boldsymbol{x})\right)^2\right] \\
&\quad + 2\mathbb{E}_{\mathcal{D}\sim P^N,\,(\boldsymbol{x},y)\sim P}\left[(h_\mathcal{D}(\boldsymbol{x}) - \bar{h}(\boldsymbol{x}))(\bar{h}(\boldsymbol{x}) - y)\right] \\
&\quad + \mathbb{E}_{(\boldsymbol{x},y)\sim P}\left[\left(\bar{h}(\boldsymbol{x}) - y\right)^2\right].
\end{aligned}$$

Observe the middle term is 0 since

$$\begin{aligned}
&\mathbb{E}_{\mathcal{D}\sim P^N,\,(\boldsymbol{x},y)\sim P}\left[(h_\mathcal{D}(\boldsymbol{x}) - \bar{h}(\boldsymbol{x}))(\bar{h}(\boldsymbol{x}) - y)\right] \\
&= \mathbb{E}_{(\boldsymbol{x},y)\sim P}\left[\mathbb{E}_{\mathcal{D}\sim P^N}\left[(h_\mathcal{D}(\boldsymbol{x}) - \bar{h}(\boldsymbol{x}))(\bar{h}(\boldsymbol{x}) - y) \mid (\boldsymbol{X}, Y)\right]\right] \\
&\qquad\qquad \text{(Law of Total Expectation}^\dagger\text{ and independence of } \mathcal{D} \text{ and } (\boldsymbol{x}, y)) \\
&= \mathbb{E}_{(\boldsymbol{x},y)\sim P}\left[(\bar{h}(\boldsymbol{x}) - y) \cdot \mathbb{E}_{\mathcal{D}\sim P^N}\left[(h_\mathcal{D}(\boldsymbol{x}) - \bar{h}(\boldsymbol{x})) \mid (\boldsymbol{X}, Y)\right]\right] \\
&= \mathbb{E}_{(\boldsymbol{x},y)\sim P}\left[(\bar{h}(\boldsymbol{x}) - y) \cdot 0\right] \qquad \text{(since (2))} \\
&= 0.
\end{aligned}$$

Hence,

$$\mathbb{E}_{\mathcal{D}\sim P^N,\,(\boldsymbol{x},y)\sim P}\left[(h_\mathcal{D}(\boldsymbol{x}) - y)^2\right] = \underbrace{\mathbb{E}_{\mathcal{D}\sim P^N,\,(\boldsymbol{x},y)\sim P}\left[\left(h_\mathcal{D}(\boldsymbol{x}) - \bar{h}(\boldsymbol{x})\right)^2\right]}_{\text{Variance}} + \mathbb{E}_{(\boldsymbol{x},y)\sim P}\left[\left(\bar{h}(\boldsymbol{x}) - y\right)^2\right]. \tag{3}$$

[*] In an slight abuse of notation, we here used P to denote the density function.

[†] This is a special application of the *Law of Total Expectation*, which many of you probably have already used in HW0. We can easily derive this from definition: $\mathbb{E}_{X,Y}[g(X,Y)] = \int_x\int_y g(x,y)p(x,y)dydx = \int_x\int_y g(x,y)p(y|x)p(x)dydx = \int_x \left(\int_y g(x,y)p(y|x)dy\right)p(x)dx = \int_x \mathbb{E}_{Y|X=x}[g(X,Y)|X=x]p(x)dx = \mathbb{E}_X[\mathbb{E}_{Y|X}[g(X,Y)|X]]$.

Note that at this point, we already have the Variance term: $\mathbb{E}_{\mathcal{D}\sim P^N,\,(\boldsymbol{x},y)\sim P}\left[\left(h_{\mathcal{D}}(\boldsymbol{x})-\bar{h}(\boldsymbol{x})\right)^2\right]$, which expresses, over all possible dataset $\mathcal{D}$, how much does the prediction generated by our algorithm deviate from the expected prediction in expectation. The higher this quantity is, the bigger the change in prediction when we vary our dataset. In particular, when *overfitting* happens, our learning algorithm produces prediction very finetuned for every dataset, thus resulting in a high deviation from the expected prediction.

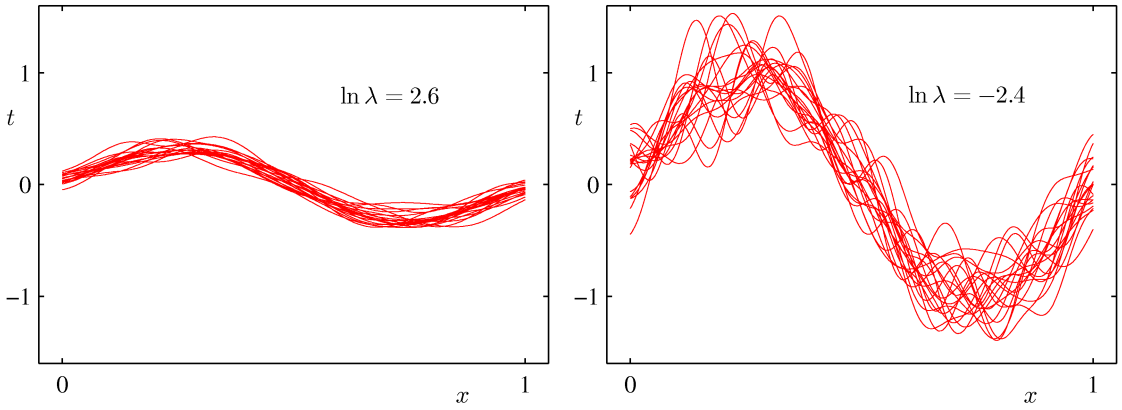

Figure 1: Plot of 20 of 100 predictor functions produced by the linear regression with regularization $\lambda \in \{\exp(2.6), \exp(-2.4)\}$ learning algorithm, trained with sample $N = 25$ over 100 datasets. Examples of low/high variance learning algorithms. Taken from [1].

We can apply the same trick again for the second term in (3) and complete the *Bias-Variance Decomposition*:

$$
\begin{aligned}
&\mathbb{E}_{(\boldsymbol{x},y)\sim P}\left[\left(\bar{h}(\boldsymbol{x})-y\right)^2\right] \\
&= \mathbb{E}_{(\boldsymbol{x},y)\sim P}\left[\left((\bar{h}(\boldsymbol{x})-\bar{y}(\boldsymbol{x}))+(\bar{y}(\boldsymbol{x})-y)\right)^2\right] \\
&= \mathbb{E}_{(\boldsymbol{x},y)\sim P}\left[\left(\bar{h}(\boldsymbol{x})-\bar{y}(\boldsymbol{x})\right)^2\right] + \mathbb{E}_{(\boldsymbol{x},y)\sim P}\left[(\bar{y}(\boldsymbol{x})-y)^2\right] \\
&\quad + 2\mathbb{E}_{(\boldsymbol{x},y)\sim P}\left[(\bar{h}(\boldsymbol{x})-\bar{y}(\boldsymbol{x}))(\bar{y}(\boldsymbol{x})-y)\right] \qquad (4)\\
&= \underbrace{\mathbb{E}_{(\boldsymbol{x},y)\sim P}\left[\left(\bar{h}(\boldsymbol{x})-\bar{y}(\boldsymbol{x})\right)^2\right]}_{\text{Bias}^2} + \underbrace{\mathbb{E}_{(\boldsymbol{x},y)\sim P}\left[(\bar{y}(\boldsymbol{x})-y)^2\right]}_{\text{Noise}}.
\end{aligned}
$$

We can show the last term in (4) is 0 in a similar way as we did before:

$$
\begin{aligned}
&\mathbb{E}_{(\boldsymbol{x},y)\sim P}\left[(\bar{h}(\boldsymbol{x})-\bar{y}(\boldsymbol{x}))(\bar{y}(\boldsymbol{x})-y)\right] \\
&= \mathbb{E}_{\boldsymbol{x}\sim\boldsymbol{X}}\left[\mathbb{E}_{y\sim Y|\boldsymbol{X}=\boldsymbol{x}}\left[(\bar{h}(\boldsymbol{x})-\bar{y}(\boldsymbol{x}))(\bar{y}(\boldsymbol{x})-y)\mid \boldsymbol{X}=\boldsymbol{x}\right]\right] && \text{(Law of Total Expectation)}\\
&= \mathbb{E}_{\boldsymbol{x}\sim\boldsymbol{X}}\left[(\bar{h}(\boldsymbol{x})-\bar{y}(\boldsymbol{x}))\mathbb{E}_{y\sim Y|\boldsymbol{X}=\boldsymbol{x}}\left[(\bar{y}(\boldsymbol{x})-y)\mid \boldsymbol{X}=\boldsymbol{x}\right]\right] \\
&= \mathbb{E}_{\boldsymbol{x}\sim\boldsymbol{X}}\left[(\bar{h}(\boldsymbol{x})-\bar{y}(\boldsymbol{x}))\left(\bar{y}(\boldsymbol{x})-\mathbb{E}_{y\sim Y|\boldsymbol{X}=\boldsymbol{x}}\left[y\mid \boldsymbol{X}=\boldsymbol{x}\right]\right)\right] \\
&= \mathbb{E}_{\boldsymbol{x}\sim\boldsymbol{X}}\left[(\bar{h}(\boldsymbol{x})-\bar{y}(\boldsymbol{x}))\left(\bar{y}(\boldsymbol{x})-\bar{y}(\boldsymbol{x})\right)\right] && \text{(Definition of } \bar{y}(\boldsymbol{x}))\\
&= \mathbb{E}_{\boldsymbol{x}\sim\boldsymbol{X}}\left[(\bar{h}(\boldsymbol{x})-\bar{y}(\boldsymbol{x}))\cdot 0\right] \\
&= 0.
\end{aligned}
$$

The squared bias term $\mathbb{E}_{(\boldsymbol{x},y)\sim P}\left[\left(\bar{h}(\boldsymbol{x}) - \bar{y}(\boldsymbol{x})\right)^2\right]$ is the expected squared difference between the expected prediction $\bar{h}(\boldsymbol{x})$ (over all datasets) and the expected label $\bar{y}(\boldsymbol{x})$ (over all datasets). In other words, this term represents the extent to which the average prediction over all data sets differs from the ground truth function. The higher the Bias^2, the further our learning algorithm like to *"bias"* away from the ground truth function (therefore inherently unable to approximate the function).

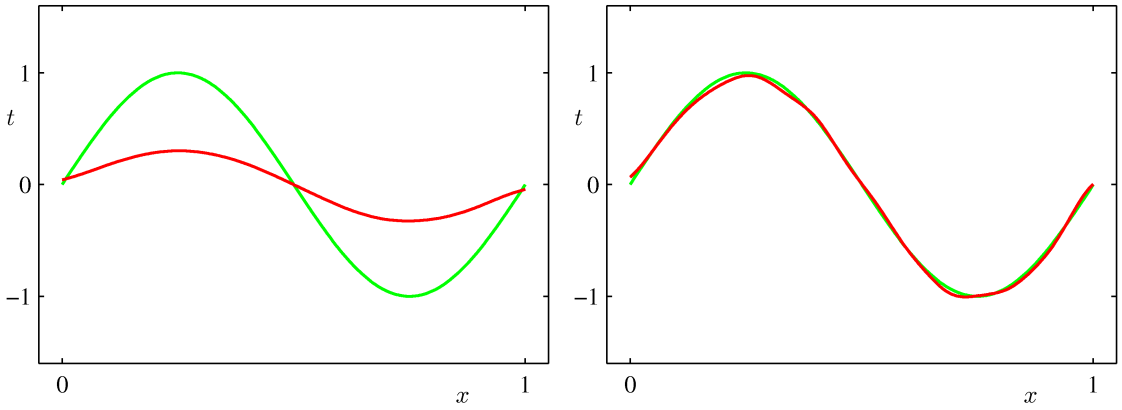

Figure 2: Plot of *average* of 100 predictor functions produced with the same setting as in Figure 1. Examples of high (left)/low (right) bias learning algorithms. Taken from [1].

Finally, the Noise term $\mathbb{E}_{(\boldsymbol{x},y)\sim P}\left[(\bar{y}(\boldsymbol{x}) - y)^2\right]$ is the expected (over $\boldsymbol{x}$) variance of sampled label y. It is intrinsic to the data and sampling process and cannot be erased by any learning algorithm (notice that this term is not involved with $\mathcal{D}$ in any way). We have now completed our derivation of the *Bias-Variance Decomposition*, summarized by the following equation:

$$\mathbb{E}_{\mathcal{D}\sim P^N,\,(\boldsymbol{x},y)\sim P}\left[(h_{\mathcal{D}}(\boldsymbol{x}) - y)^2\right] = \underbrace{\mathbb{E}_{\mathcal{D}\sim P^N,\,(\boldsymbol{x},y)\sim P}\left[\left(h_{\mathcal{D}}(\boldsymbol{x}) - \bar{h}(\boldsymbol{x})\right)^2\right]}_{\text{Variance}} + \underbrace{\mathbb{E}_{(\boldsymbol{x},y)\sim P}\left[\left(\bar{h}(\boldsymbol{x}) - \bar{y}(\boldsymbol{x})\right)^2\right]}_{\text{Bias}^2} + \underbrace{\mathbb{E}_{(\boldsymbol{x},y)\sim P}\left[(\bar{y}(\boldsymbol{x}) - y)^2\right]}_{\text{Noise}}.$$

References

[1] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006. ISBN: 0387310738.