EE2211 : Introduction to Machine Learning (Fall 2020)
Overfitting and the Bias-Variance Tradeoff

Vincent Y. F. Tan

In this brief document, we provide a interesting example of the perils of overfitting. We will also provide analytical example that delineates the bias-variance tradeoff. This whole discussion is optional reading.

1 An Example of Overfitting

As you have learned, in machine learning, we want to learn a model that fits the *pattern* of the data and not the data itself. Learning an overly complex model that "hugs" the data too tightly may result in overfitting. In this section, we describe an example in which overfitting has catastrophic consequences. This example, which should be of interest to the structural engineers in the class, is taken from Brian Stacey's report [Sta16], with additional analysis from Nate Silver [Sil12], whose book [Sil12] I strongly recommend.

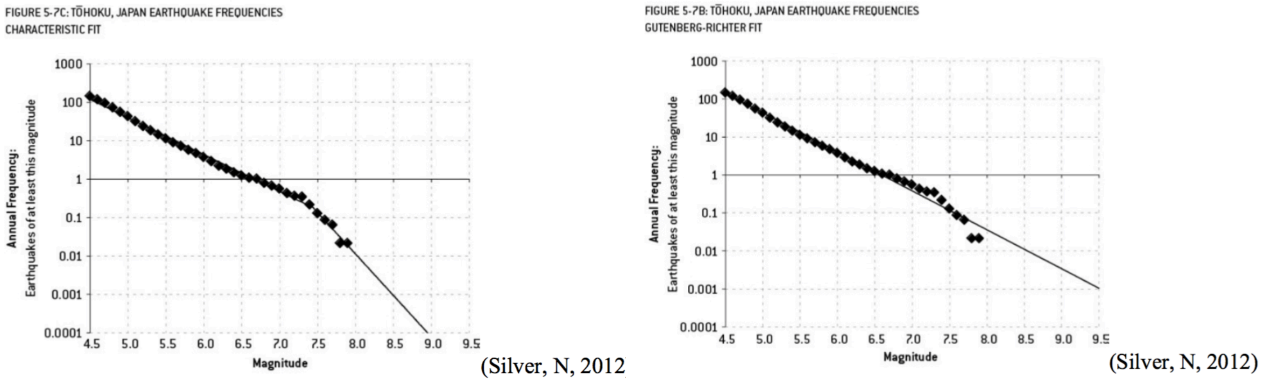


FIGURE 5-7C: TŌHOKU, JAPAN EARTHQUAKE FREQUENCIES
CHARACTERISTIC FIT

(Silver, N, 2012

FIGURE 5-7B: TŌHOKU, JAPAN EARTHQUAKE FREQUENCIES
GUTENBERG-RICHTER FIT

(Silver, N, 2012)

(a) Overfitting

(b) Linear Fit (Gutenberg-Richter)

Figure 1: Plots of the predictions based on historical data

As you probably have heard of, in March 2011, there was a major earthquake in Japan that affected the Fukushima nuclear power plant. In fact, the design of the plant was based on historical earthquake data over more than 400 years. It was designed to withstand an earthquake of magnitude 8.6 (on the Richter scale). However, the earthquake in March 2011 had a magnitude of 9.0. Could this have been prevented with better engineering? Perhaps if the engineers had taken NUS EE2211, they would have been in a better position to prevent the Fukushima disaster by using a more conservative and sturdy design for the plant.

The historical earthquake data is shown as scatter plots in Fig. 1. As can be seen, small earthquakes occur frequently while massive earthquakes occur rarely. The engineers saw the data and because of the kink around magnitude 7.3, they fitted a *polynomial* model (of order > 1), resulting in Fig. 1(a). Note that the regression curve "hugs" the points very closely. There is, however, another method known as the Gutenberg-Richter model which uses simple *linear* regression to predict frequency versus magnitude. It looks like Fig. 1(b). The models in Fig. 1(a) and Fig. 1(b) respectively say that an earthquake of magnitude 9.0

will occur on average once every 10^4 years and 500 years. Since the data is collected over 400 years, an earthquake of magnitude 9.0 is rather likely based on the second model. Hence, the overfitting error that the structural engineers made resulted in them designing a power plant that was not sufficiently strong to withstand a 9.0-earthquake, one that may occur once every 500 (as opposed to 10^4) years. This shows the importance of fitting the pattern, and not the actual data.

2 Bias-Variance Tradeoff for Linear Models with Ridge Regression

Now for some theory. Consider the linear model

$$y = w_0^* + w_1^* x_1 + \ldots + w_d^* x_d + e \tag{1}$$

where e is zero-mean noise with variance σ^2 (not necessarily Gaussian. We can also write (1) more compactly as

$$y = f(\mathbf{x}) + e \tag{2}$$

where $f(\mathbf{x}) = \tilde{\mathbf{x}}\mathbf{w}^*$ and the *bias-augmented* sample is $\tilde{\mathbf{x}} = \begin{bmatrix} 1 & \mathbf{x} \end{bmatrix} \in \mathbb{R}^{1\times(d+1)}$ and $\mathbf{w}^* = (w_0^*, w_1^*, \ldots, w_d^*)^\top \in \mathbb{R}^{d+1}$ is the unknown weight or coefficient vector. We observe samples from a dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^m$ where $\mathbf{x}_i \in \mathbb{R}^{1\times d}$ and $y_i \in \mathbb{R}$ for $1 \le i \le m$ are respectively the feature (row) vectors and targets. Stacking these observations into matrix form, we obtain

$$\mathbf{y} = \mathbf{X}\mathbf{w}^* + \mathbf{e} \tag{3}$$

where $\mathbf{X} \in \mathbb{R}^{m\times(d+1)}$ is the design matrix with a vector of ones in the first column, $\mathbf{y} = (y_1, y_2, \ldots, y_m)^\top \in \mathbb{R}^m$ is the length-m vector of targets and $\mathbf{e} = (e_1, e_2, \ldots, e_m)^\top \in \mathbb{R}^m$ is the vector of i.i.d. noises. Note that $\mathbb{E}[\mathbf{e}] = \mathbf{0}$ and $\mathrm{Cov}(\mathbf{e}) = \sigma^2\mathbf{I}$.

Suppose we do least squares regression with $\mathbf{X}$ assumed to be full column rank (usually the case when $m > d+1$). Then you know that the least squares estimate of the unknown $\mathbf{w}^*$ is

$$\hat{\mathbf{w}}_{\text{ls}} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y} \in \mathbb{R}^{d+1}. \tag{4}$$

Given a *new* (or *test*) sample $\mathbf{x} \in \mathbb{R}^d$, we can obtain its prediction as the following inner product

$$\hat{f}_{\mathcal{D}}(\mathbf{x}) = \tilde{\mathbf{x}}\hat{\mathbf{w}}_{\text{ls}} = \tilde{\mathbf{x}}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}. \tag{5}$$

Let us evaluate the bias and variance of $\hat{f}_{\mathcal{D}}(\mathbf{x})$ in (5); note that $\hat{f}_{\mathcal{D}}(\mathbf{x})$ is only random through the noise $\mathbf{e}$ (not the dataset $(\mathbf{X}, \mathbf{y})$, the true coefficients $\mathbf{w}^*$, or the test sample $\mathbf{x}$). Also recall that the bias is $\mathrm{Bias}(\hat{f}_{\mathcal{D}}(\mathbf{x})) = \mathbb{E}[\hat{f}_{\mathcal{D}}(\mathbf{x})] - f(\mathbf{x})$. The term $\mathbb{E}[\hat{f}_{\mathcal{D}}(\mathbf{x})]$, which is an expectation (or colloquially an average) over all training datasets $\mathcal{D}$, was denoted as $\hat{f}_{\text{avg}}(\mathbf{x})$ in the lecture notes. We have

$$\begin{aligned}
\mathrm{Bias}(\hat{f}_{\mathcal{D}}(\mathbf{x})) &= \mathbb{E}\big[\hat{f}_{\mathcal{D}}(\mathbf{x}) - f(\mathbf{x})\big] && (6)\\
&\overset{(5)}{=} \mathbb{E}\big[\tilde{\mathbf{x}}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y} - f(\mathbf{x})\big] && (7)\\
&\overset{(3)}{=} \mathbb{E}\big[\tilde{\mathbf{x}}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top(\mathbf{X}\mathbf{w}^* + \mathbf{e}) - f(\mathbf{x})\big] && (8)\\
&\overset{(2)}{=} \mathbb{E}\big[\tilde{\mathbf{x}}(\mathbf{X}^\top\mathbf{X})^{-1}(\mathbf{X}^\top\mathbf{X})\mathbf{w}^* + \tilde{\mathbf{x}}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{e} - \tilde{\mathbf{x}}\mathbf{w}^*\big] && (9)\\
&= \tilde{\mathbf{x}}\mathbf{w}^* + \tilde{\mathbf{x}}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbb{E}[\mathbf{e}] - \tilde{\mathbf{x}}\mathbf{w}^* = 0, && (10)
\end{aligned}$$

so the least squares estimator is *unbiased.* We emphasize that the only quantity that is random in the above calculation is the noise $\mathbf{e}$. For the variance, we first note from (5) that $\hat{f}_{\mathcal{D}}(\mathbf{x}) = \tilde{\mathbf{x}}\mathbf{w}^* + \tilde{\mathbf{x}}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{e}$ and since $\tilde{\mathbf{x}}\mathbf{w}^*$ is deterministic, the variance of $\hat{f}_{\mathcal{D}}(\mathbf{x})$ is that of $\tilde{\mathbf{x}}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{e}$. Let $\mathbf{c} = \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\tilde{\mathbf{x}}^\top$. Then

$$\begin{aligned}
\mathrm{Var}(\hat{f}_{\mathcal{D}}(\mathbf{x})) &= \mathrm{Var}\big(\hat{f}_{\mathcal{D}}(\mathbf{x})\big) && (11)\\
&\overset{(5)}{=} \mathrm{Var}\big(\mathbf{c}^\top\mathbf{e}\big) && (12)
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{i=1}^{m} c_i^2 \mathrm{Var}(e_i) && (13)\\
&= \|\mathbf{c}\|^2 \sigma^2 && (14)\\
&= \left\|\mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\tilde{\mathbf{x}}^\top\right\|^2 \sigma^2 && (15)\\
&= \tilde{\mathbf{x}}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\tilde{\mathbf{x}}^\top\sigma^2 && (16)\\
&= \tilde{\mathbf{x}}(\mathbf{X}^\top\mathbf{X})^{-1}\tilde{\mathbf{x}}^\top\sigma^2. && (17)
\end{aligned}
$$

Note that (13) holds because the e_i's (noises added on to the training samples) are assumed to be independent. This result is intuitive because $\mathrm{Var}(\hat{f}_{\mathcal{D}}(\mathbf{x}))$ is proportional to σ^2 and as the number of training samples m increases, the design matrix $\mathbf{X}^\top\mathbf{X} = \sum_{i=1}^m \mathbf{x}_i^\top\mathbf{x}_i \in \mathbb{R}^{(d+1)\times(d+1)}$ (sum of rank-one outer products) also increases linearly and so $(\mathbf{X}^\top\mathbf{X})^{-1}$ decreases as $1/m$.[1] ovRecall that

$$
\begin{aligned}
\mathrm{MSE}(\hat{f}_{\mathcal{D}}(\mathbf{x})) &= \mathbb{E}\big[(f(\mathbf{x}) + e - \hat{f}_{\mathcal{D}}(\mathbf{x}))^2\big] && (18)\\
&= \big(\mathrm{Bias}(\hat{f}_{\mathcal{D}}(\mathbf{x}))\big)^2 + \mathrm{Var}(\hat{f}_{\mathcal{D}}(\mathbf{x})) + \text{Irreducible Noise}. && (19)
\end{aligned}
$$

The three terms can be explained as follows:

- The *bias* quantifies the error caused by simplifying assumptions in the model. For example, when we use a linear function to approximate a model which is inherently quadratic, we will suffer some bias.
- The *variance* quantifies how much the estimated solution $\hat{f}_{\mathcal{D}}(\mathbf{x})$ fluctuates around its mean.
- The *irreducible error* quantifies the measurement noise that is inherent in the new test sample.

For the least squares predictor in (5), putting the bias in (10) and variance in (17) together, we see that

$$
\mathrm{MSE}(\hat{f}_{\mathcal{D}}(\mathbf{x})) = \sigma^2\big(\tilde{\mathbf{x}}(\mathbf{X}^\top\mathbf{X})^{-1}\tilde{\mathbf{x}}^\top + 1\big). \tag{20}
$$

Now, we go a bit further and calculate explicit expressions for the bias and variance when we use ridge regression with regularization parameter $\lambda > 0$. In this case, we can jettison the assumption that $\mathbf{X}$ has full column rank and we have

$$
\hat{\mathbf{w}}_\lambda = (\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^\top\mathbf{y} \in \mathbb{R}^{d+1}. \tag{21}
$$

The prediction of the target of a new sample $\mathbf{x}$ (another row vector), denoted as $\hat{f}_{\mathcal{D},\lambda}(\mathbf{x})$, is

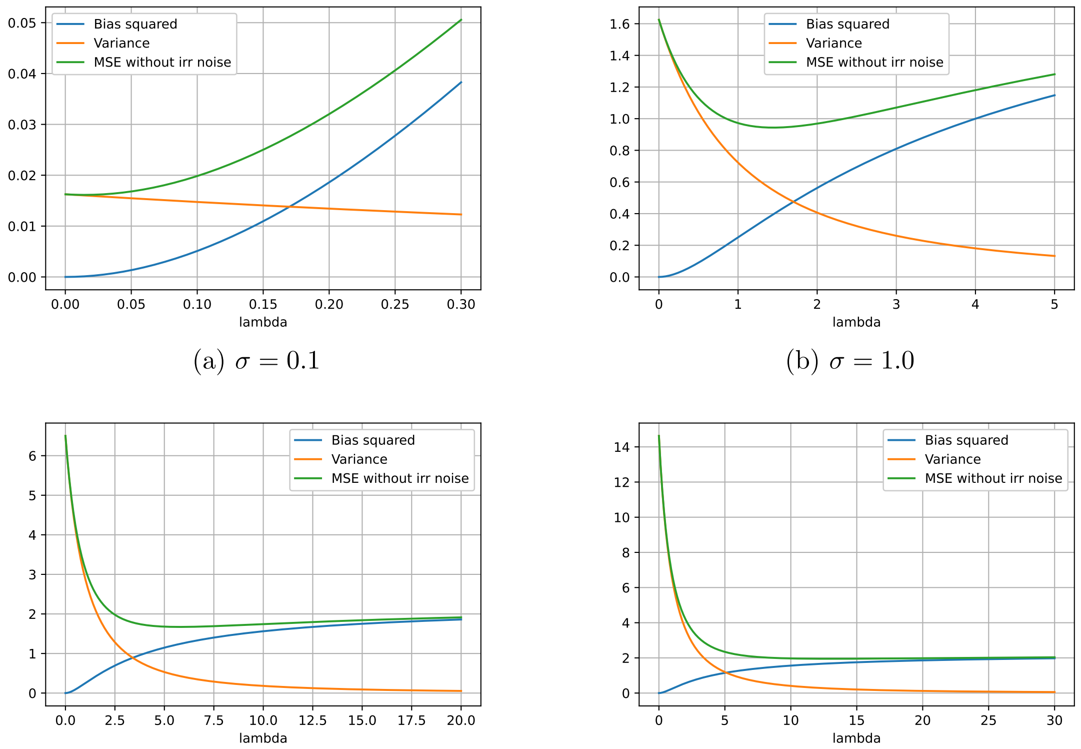
$$
\hat{f}_{\mathcal{D},\lambda}(\mathbf{x}) = \tilde{\mathbf{x}}\hat{\mathbf{w}}_\lambda = \tilde{\mathbf{x}}(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^\top\mathbf{y}, \tag{22}
$$

where $\tilde{\mathbf{x}} \in \mathbb{R}^{1\times(d+1)}$ is the bias-augmented version of $\mathbf{x}$. Note that for $\lambda > 0$, the prediction is smaller than the unregularized case because of the additional term $+\lambda\mathbf{I}$ in the inverse. Thus, this is sometimes called *weight shrinkage*. Using similar but more tedious calculations (see Appendix A.1), we obtain the bias and variance of the ridge regularized prediction in (22) as follows:

$$
\begin{aligned}
\mathrm{Bias}(\hat{f}_{\mathcal{D},\lambda}(\mathbf{x})) &= -\lambda\tilde{\mathbf{x}}(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{w}^* \quad \text{and} && (23)\\
\mathrm{Var}(\hat{f}_{\mathcal{D},\lambda}(\mathbf{x})) &= \tilde{\mathbf{x}}\big((\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1} - \lambda(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-2}\big)\tilde{\mathbf{x}}^\top\sigma^2. && (24)
\end{aligned}
$$

Note that since $\mathrm{Bias}(\hat{f}_{\mathcal{D},\lambda}(\mathbf{x})) \neq 0$ in general, the regularized least squares solution is *biased*. We will see, however, that its variance is smaller than that of the unregularized solution. Indeed, by comparing (17) and (24), we see that for any $\sigma^2 > 0$, $\mathrm{Var}(\hat{f}_{\mathcal{D},\lambda}(\mathbf{x})) \le \mathrm{Var}(\hat{f}_{\mathcal{D}}(\mathbf{x}))$ with equality if and only if $\lambda = 0$.

[1] More precisely, if $\mathbf{x}_i$ are sampled i.i.d. from a distribution $p(\mathbf{x})$, then as m grows, $\frac{1}{m}\mathbf{X}^\top\mathbf{X} \to \mathbf{C}$ where $\mathbf{C} = \mathbb{E}[\mathbf{x}_i^\top\mathbf{x}_i]$ is the covariance matrix of $\mathbf{x}_i$ for any i. Thus, $(\mathbf{X}^\top\mathbf{X})^{-1}$ decays at a rate of $1/m$.



(a) $\sigma = 0.1$

(b) $\sigma = 1.0$

(c) $\sigma = 2.0$

(d) $\sigma = 3.0$

Figure 2: Plots of $\text{Bias}(\hat{f}_{\mathcal{D},\lambda}(\mathbf{x}))^2$ and $\text{Var}(\hat{f}_{\mathcal{D},\lambda}(\mathbf{x}))$ and their sum for various noise levels σ

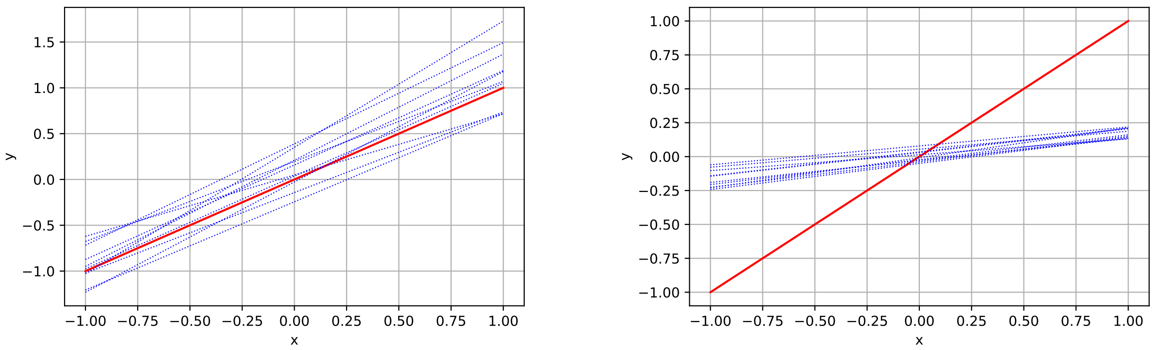
3 An Example of the Bias-Variance Tradeoff

We provide an example in which we can compute the bias and variance in (23) and (24) in closed form. Say $d = 1$ and we observe the two (scalar) training samples $\mathbf{x}_1 = -1$ and $\mathbf{x}_2 = 1$. Let $\mathbf{w}^* = [0, 1]^\top$ so the true model is nothing but the simple linear model $y = x + e$ (cf. Eqn. (2)). Then

$$\mathbf{X} = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}, \quad \mathbf{X}^\top \mathbf{X} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}, \quad (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I})^{-1} = \begin{bmatrix} 1/(2+\lambda) & 0 \\ 0 & 1/(2+\lambda) \end{bmatrix}. \tag{25}$$

Suppose our test sample is $\mathbf{x} = 1.5$, i.e., we are trying to predict the target of a point outside the range of values within our dataset. We plot $\text{Bias}(\hat{f}_{\mathcal{D},\lambda}(\mathbf{x}))^2$, $\text{Var}(\hat{f}_{\mathcal{D},\lambda}(\mathbf{x}))$ and their sum for different noise variances σ^2 in Fig. 2. A few remarks are in order:

- The bias is always zero when $\lambda = 0$ or when the least squares estimator in (4) is used; this is in line with (10). However, the squared bias grows as λ increases.
- The variance is initially large when $\lambda = 0$. However, it tends to zero as λ increases. This is in line with (24). In essence, with more regularization, the solution stabilizes and the variance of the prediction on a new test sample is reduced. It holds that $\text{Var}(\hat{f}_{\mathcal{D},\lambda}(\mathbf{x})) \to 0^+$ as $\lambda \to 0^+$.
- The sum of the squared bias and variance, which is the mean-squared error minus the irreducible error (cf. Eqn. (19)), has a "sweet spot" at around $\lambda \approx 1.44$ for $\sigma = 1.0$.
- The effect of regularization is more produced for large noise, i.e., when σ is increased. Indeed, observe that the reduction of the sum of the squared bias and the variance from the case in which $\lambda = 0$ is



(a) $\sigma = 0.3, \lambda = 0.1$

(b) $\sigma = 0.3, \lambda = 10$

Figure 3: Plots the predicted lines given 10 noise realizations

more pronounced when the noise is larger. When the noise is large, there is severe overfitting and we are essentially fitting the curve to the noise. Hence, it is imperative to regularize the solution. For small σ, say $\sigma = 0.1$, the benefit of regularization is not obvious.

The code to generate Fig. 2 is provided in Appendix A.2. You can play with it to generate other bias-variance plots.

In Fig. 3, we generate 10 datasets $\mathcal{D}$ as follows. Fixing $\mathbf{x}_1 = -1$ and $\mathbf{x}_2 = 1$, we generate 10 corresponding $\mathbf{y}$'s according to the true model in (2). Using the datasets, we learned the regression lines based on $\hat{\mathbf{w}}_\lambda$ in (21). Here, we fix the noise to be $\sigma = 0.3$ and consider two values of $\lambda \in \{0.1, 10\}$. For the small $\lambda = 0.1$, we notice that the regression lines have low bias on average; their averaged value is close to the ground truth red line $y = x$. However, they have large variability among themselves. For the large regularization parameter $\lambda = 10$, we see that the lines have high bias; they do not seem to approximate the ground truth very well indeed. However, among the 10 lines, they are close to one another and hence when λ is large, the variance of the solution is small. Finally, we see that for large λ, the intercept and slope of the regression lines are both small, which is intuitive as if λ is large $\hat{\mathbf{w}}_\lambda$ defined in (21) is small.

A.1 Proofs of (23) and (24)

For the bias, we have

$$\begin{aligned}
\mathrm{Bias}(\hat{f}_{\mathcal{D},\lambda}(\mathbf{x})) &= \mathbb{E}\big[\hat{f}_{\mathcal{D},\lambda}(\mathbf{x}) - f(\mathbf{x})\big] && (26)\\
&\overset{(22)}{=} \mathbb{E}\big[\tilde{\mathbf{x}}(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^\top\mathbf{y} - \tilde{\mathbf{x}}\mathbf{w}^*\big] && (27)\\
&\overset{(3)}{=} \mathbb{E}\big[\tilde{\mathbf{x}}(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^\top(\mathbf{X}\mathbf{w}^* + \mathbf{e}) - \tilde{\mathbf{x}}\mathbf{w}^*\big] && (28)\\
&= \mathbb{E}\big[\tilde{\mathbf{x}}(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1}(\mathbf{X}^\top\mathbf{X}\mathbf{w}^* + \mathbf{X}^\top\mathbf{e}) - \tilde{\mathbf{x}}\mathbf{w}^*\big] && (29)\\
&= \mathbb{E}\big[\tilde{\mathbf{x}}(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1}((\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})\mathbf{w}^* - \lambda\mathbf{w}^* + \mathbf{X}^\top\mathbf{e}) - \tilde{\mathbf{x}}\mathbf{w}^*\big] && (30)\\
&= \tilde{\mathbf{x}}\mathbf{w}^* - \lambda\tilde{\mathbf{x}}(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{w}^* + \mathbb{E}\big[\tilde{\mathbf{x}}(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^\top\mathbf{e}\big] - \tilde{\mathbf{x}}\mathbf{w}^* && (31)\\
&= -\lambda\tilde{\mathbf{x}}(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{w}^* && (32)
\end{aligned}$$

as desired. For the variance, we first note that the prediction can be simplified as follows:

$$\begin{aligned}
\hat{f}_{\mathcal{D},\lambda}(\mathbf{x}) &= \tilde{\mathbf{x}}(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^\top(\mathbf{X}\mathbf{w}^* + \mathbf{e}) && (33)\\
&= \tilde{\mathbf{x}}(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^\top\mathbf{X}\mathbf{w}^* + \tilde{\mathbf{x}}(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^\top\mathbf{e}. && (34)
\end{aligned}$$

The first part is non-random. So the variance of $\hat{f}_{\mathcal{D},\lambda}(\mathbf{x})$ is precisely that of the noise term $\tilde{\mathbf{x}}(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^\top\mathbf{e}$. Let $\mathbf{c} = \mathbf{X}(\mathbf{X}^\top\mathbf{X}+\lambda\mathbf{I})^{-1}\tilde{\mathbf{x}}^\top$. Then by the same steps as those for the variance of the unregularized case (see steps leading to (14)),

$$\begin{aligned}
\mathrm{Var}(\hat{f}_{\mathcal{D},\lambda}(\mathbf{x})) &= \|\mathbf{c}\|^2\sigma^2 && (35)\\
&= \tilde{\mathbf{x}}(\mathbf{X}^\top\mathbf{X}+\lambda\mathbf{I})^{-1}\mathbf{X}^\top\mathbf{X}(\mathbf{X}^\top\mathbf{X}+\lambda\mathbf{I})^{-1}\tilde{\mathbf{x}}^\top\sigma^2 && (36)\\
&= \tilde{\mathbf{x}}(\mathbf{X}^\top\mathbf{X}+\lambda\mathbf{I})^{-1}\left[(\mathbf{X}^\top\mathbf{X}+\lambda\mathbf{I})-\lambda\mathbf{I}\right](\mathbf{X}^\top\mathbf{X}+\lambda\mathbf{I})^{-1}\tilde{\mathbf{x}}^\top\sigma^2 && (37)\\
&= \tilde{\mathbf{x}}(\mathbf{X}^\top\mathbf{X}+\lambda\mathbf{I})^{-1}\big(\mathbf{I}-\lambda(\mathbf{X}^\top\mathbf{X}+\lambda\mathbf{I})^{-1}\big)\tilde{\mathbf{x}}^\top\sigma^2 && (38)\\
&= \tilde{\mathbf{x}}\big((\mathbf{X}^\top\mathbf{X}+\lambda\mathbf{I})^{-1}-\lambda(\mathbf{X}^\top\mathbf{X}+\lambda\mathbf{I})^{-2}\big)\tilde{\mathbf{x}}^\top\sigma^2 && (39)
\end{aligned}$$

as desired.

A.2 Code to Generate the Bias-Variance Plots

```
import numpy as np
import matplotlib.pyplot as plt

X = np.array([[1, -1], [1, 1]])
XTX = X.T @ X
N = 1000
lam = np.linspace(0.0001,5,N)
x_test = 1.5
x = np.array([1, x_test])
w_s = np.array([0, 1]).T
sigma = 1
bias = np.zeros(N)
var = np.zeros(N)

for i in range(0, len(lam)):
    reg = lam[i]*np.identity(2)
    bias[i] = -reg @ x @ np.linalg.inv(XTX + reg) @ w_s
    var[i] = x @ ( np.linalg.inv(XTX + reg) - lam[i] * np.linalg.inv(XTX + reg)
                  @ np.linalg.inv(XTX + reg))@x.T*sigma**2

MSE = np.power(bias,2)+var
plt.plot(lam, np.power(bias,2), label = 'Bias squared')
plt.plot(lam, var, label = 'Variance')
plt.plot(lam, MSE, label = 'MSE without irr noise')
plt.legend()
plt.grid()
plt.xlabel('lambda')
plt.savefig('bias_variance.eps', format='eps')
print(lam[np.argmin(MSE)])
```

References

[Sil12] N. Silver. *The Signal and the Noise: The Art and Science of Prediction.* Penguin Books Ltd, 2012.

[Sta16] B. Stacey. Fukushima: The failure of predictive models. *MPRA Paper No. 69383*, 2016. Online at https://mpra.ub.uni-muenchen.de/69383/.