National University of Singapore
School of Computing
CS3244: Machine Learning
Solution to Tutorial 4

Bias, Variance, and Overfitting

Colab Notebook Solutions: Bias, Variance, and Overfitting

1. **Overfitting**: Briefly answer the following questions:

   (a) When (the number of data points / noise / target complexity) increases, is overfitting less likely to occur?
   The number of data points/Noise/Target complexity: Yes/No/No.

   (b) Assume $\mathcal{H}$ is fixed and we increase the complexity of f. Will deterministic noise in general (but not necessarily in all cases) go up or down? How about stochastic noise? Is there a higher or lower tendency to overfit?
   Deterministic noise will go up in general because it gets harder for $\mathcal{H}$ to approximate f. Stochastic noise does not depend on the complexity of $\mathcal{H}$ and f. There is a higher tendency to overfit.

   (c) Assume f is fixed and we increase the complexity of $\mathcal{H}$. Will deterministic noise in general (but not necessarily in all cases) go up or down? How about stochastic noise? Is there a higher or lower tendency to overfit?
   Deterministic noise will go down in general because $\mathcal{H}$ gets higher chance to approximate f. Stochastic noise does not depend on the complexity of $\mathcal{H}$ and f. There is a higher tendency to overfit.

2. **Validation**: We will be using a strategy called Cross-Validation(CV) to measure the performance(This will be discussed in detail in the upcoming lecture). k-fold CV can be described as follows. The training dataset is divided into k groups. Then, there would be k number of training iterations and validation performance measures. In each iteration, $k-1$ number of groups of data is used to train the model and the remaining group is used to measure the validation performance. In every iteration, the validation group is different. Finally, the final performance is the average over k validation performance measures of each iteration.

   You are deciding on a regularization parameter $\mathcal{H}$ for your linear regression model. You perform 10-fold CV on your training data for the following values of $\mathcal{C}$ and get the following graph (Figure 1):

   (a) What does each blue and green point represent? How are they calculated?
   *Each blue point represents the **average** training accuracy for a value of $\mathcal{C}$. It is calculated by getting the average accuracy of all 10 training folds. Similarly, each green point represents the **average** validation accuracy for a value of $\mathcal{C}$. It is calculated by getting the average accuracy of all 10 validation folds.*

   (b) What do the blue and green shaded areas represent? How are they calculated?
   *Each blue point represents the **variance** of the training accuracy for a value of $\mathcal{C}$. It is calculated by getting the variance of accuracy of all 10 training folds. Similarly, each*

*green point represents the **variance** of the validation accuracy for a value of $\mathcal{C}$. It is calculated by getting the variance of accuracy for all 10 validation folds.*

(c) What should you select as your $\mathcal{C}$ value?
The validation accuracy is the highest when $C = 1$.

3. **Bias and Variance**: Assume $y = f(x) + \epsilon$ where $\mathrm{E}(\epsilon) = 0$ and $\mathrm{Var}(\epsilon) = \sigma^2$. Using squared-error loss, the expected prediction error of a regression fit $\hat{f}(x)$ at an input point $x = x_0$ can be written as the sum of Irreducible Error, Bias^2 and Variance. For the k-nearest regression fit, the error can be expressed as:

$$\begin{aligned} Err(x_0) &= \mathrm{E}[(y - \hat{f}_k(x_0))^2 | x_0)] \\ &= \sigma^2 + (f(x_0) - \frac{1}{k}\sum_{i=1}^{k} f(x_i))^2 + \frac{\sigma^2}{k}. \end{aligned}$$

where $x_i (i = 1, ..., k)$ are the k nearest data points. We assume that these neighbors are fixed, for simplicity.

(a) Derive the above equation by calculating bias and variance.
Irreducible Error: σ^2

Variance:

$$\begin{aligned} Var(\hat{f}_k(x_0)) &= Var(\frac{1}{k}\sum_{i=1}^{k} y(x_i)) \\ &= \frac{1}{k^2} Var(\sum_{i=1}^{k} f(x_i) + \epsilon) \\ &= \frac{1}{k^2}\sum_{i=1}^{k} Var(\epsilon) \\ &= \frac{1}{k^2}\sum_{i=1}^{k} \sigma^2 \\ &= \frac{\sigma^2}{k} \end{aligned} \tag{1}$$

$Var(f(x_i)) = 0$ because we assume the neighbors x_is are fixed
$Bias^2$:

$$
\begin{aligned}
\left(f(x_0) - E[\hat{f}_k(x_0)]\right)^2 &= \left(f(x_0) - E[\frac{1}{k}\sum_{i=1}^{k} y(x_i)]\right)^2 \\
&= \left(f(x_0) - \frac{1}{k}\sum_{i=1}^{k} y(x_i)\right)^2 \\
&= \left(f(x_0) - \frac{1}{k}\sum_{i=1}^{k} f(x_i) + \epsilon\right)^2 \\
&= \left(f(x_0) - \frac{1}{k}\sum_{i=1}^{k} f(x_i)\right)^2 \quad \textit{since, } E[\epsilon] = 0 \textit{ and } E[c] = c
\end{aligned} \tag{2}
$$

Assuming fixed neighbors x_i $E[\frac{1}{k}\sum_{i=1}^{k} y(x_i)] = \frac{1}{k}\sum_{i=1}^{k} y(x_i)$

(b) Describe how you would chose an optimal value of K using the above equations.

i. As we can see from the expression, smaller values of k keeping everything else constant will increase the variance.

ii. As the value of k increases, bias will increase. Why? As the number of data points increase, we will be considering points further away from x_0 and we would move further away from $f(x_0)$.

4. **[Optional] Support Vector Machines** The figure below shows two hyper-planes $\mathcal{H}_1$ and $\mathcal{H}_2$. The red points have $y_i = 1$ and the blue points have $y_i = -1$. The blue vector is a hyperplane passing through the origin and has the from $\theta^T x + b = 0$. $\vec{\theta}$ is normal to the plane. Show that the distance between the two hyperplanes is $d = \frac{2}{||\theta||}$ where $||\cdot||$ denotes the argument's norm.

Proof:

Let x_1 be a point on $\mathcal{H}_1$ and let x_2 be a point on $\mathcal{H}_2$. The closest point on $\mathcal{H}_2$ from $\mathcal{H}_1$ is

$$
\begin{aligned}
x_1 + d\frac{\theta}{||\theta||} &= x_2 \\
x_2 &= x_1 + d\vec{u}
\end{aligned} \tag{3}
$$

Where u is a unit vector along $\vec{\theta}$

$$\begin{aligned}
\theta^T x_2 + b &= 1 \\
\theta^T \{x_1 + du\} + b &= 1 \\
\theta^T du + \theta^T x_1 + b &= 1 \\
\theta^T du - 1 &= 1 \\
\theta^T du &= 2 \\
d &= \frac{2}{u\theta^T} \\
d &= \frac{2}{\frac{\theta}{||\theta||}\theta^T} \\
d &= \frac{2}{||\theta||}
\end{aligned} \tag{4}$$

$\theta^T x_1 + b = -1$ since x_1 lies on $\mathcal{H}_1$ and has to satisfy the equation.

What happens in SVM ?

We want to find θ that maximizes this distance in SVM. Maximizing this distance is same as minimizing $\frac{1}{2}||\theta||^2 = \frac{1}{2}\theta^T\theta$

The objective of SVM is then to minimize the following

$$\begin{aligned}
&\min_{\theta,b} \frac{1}{2}\theta^T\theta \\
&\textit{subjected to } y_i(\theta^T x_i + b) \geq 1
\end{aligned} \tag{5}$$

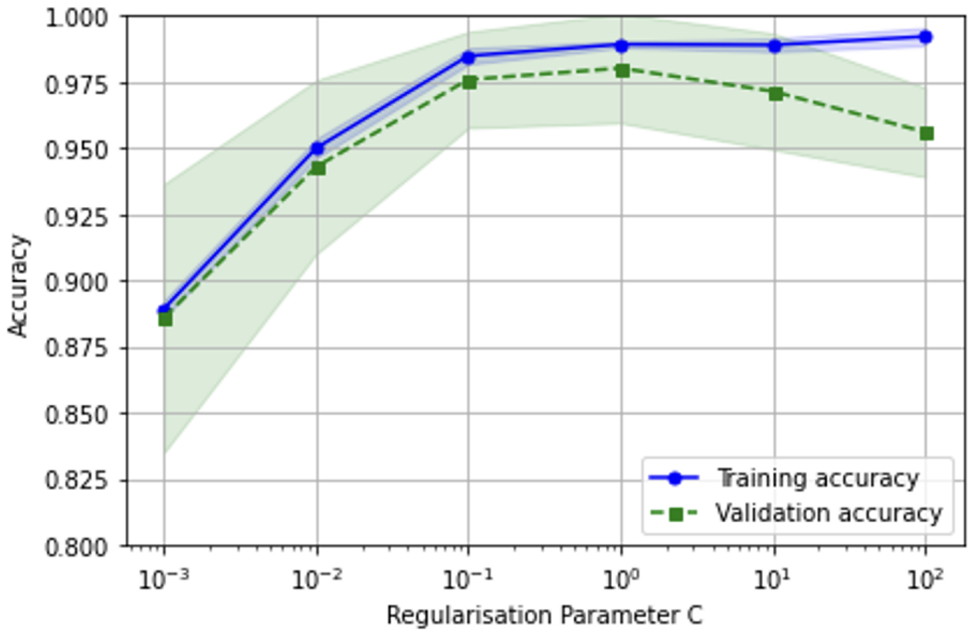

Figure 1: Validation Curve: (Source: Python Machine Learning by Sebastian Raschka)

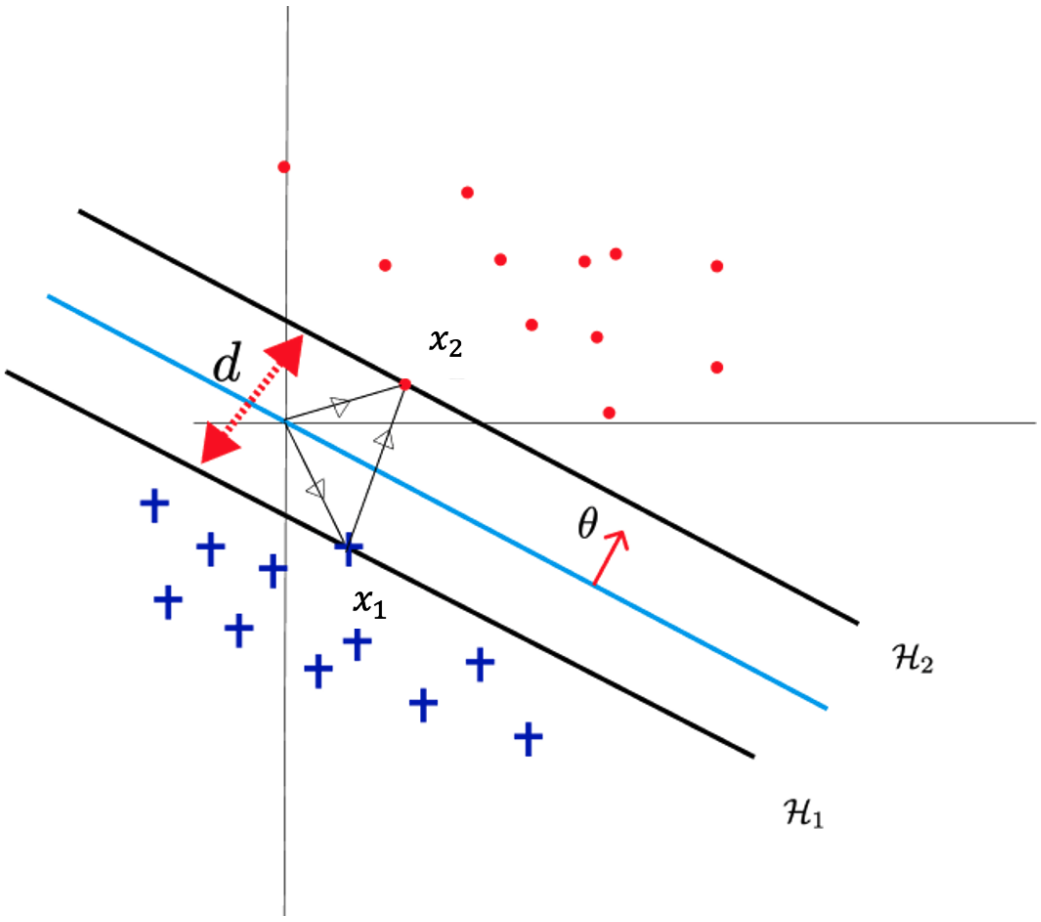

Figure 2: Support Vector Machines