National University of Singapore
School of Computing
CS3244: Machine Learning
Solution to W05-Tutorial - 04

Bias, Variance, and Overfitting

Colab Notebook Solutions:[1] 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 week 06 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 SVM 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.*

[1] The notebooks are shared as additional exercises. They will be neither tested in exams nor discussed during the tutorial. However, the solutions will be provided.

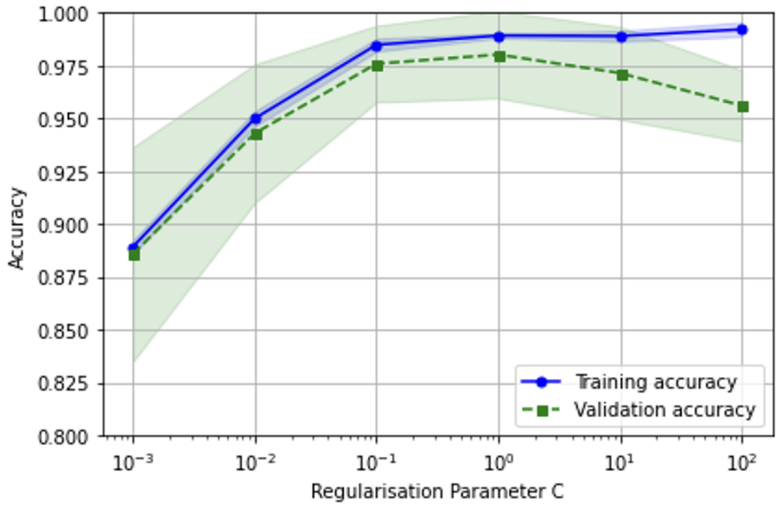

Figure 1: Validation Curve: (Source: Python Machine Learning by Sebastian Raschka)

(b) What do the blue and green shaded areas represent? How are they calculated?
Each blue region 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 region 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 neighbours 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 $f(x_0)$.

(c) In the training set, each sample follows the assumption of $y_i = f(x_i) + \epsilon_i$. Choose all the right choices about this assumption:

- A. ϵ_i comes from a Gaussian distribution
- B. all ϵ_i have the same mean
- C. all y_i have the same mean
- D. all y_i have the same variance

A, B, D. Reason: This question will help you understand the data generation process. Considering the data generation process of $y_i = f(x_i) + \epsilon_i$, $f(x_i)$ is a function of x_i, which is not a random variable. However, ϵ_i is a random variable which follows the

Gaussian distribution $\epsilon \sim N(0, \sigma^2)$, and it is randomly sampled for each i and added to $f(x_i)$. Therefore, we can get

$$E[y_i] = E[f(x_i) + \epsilon_i] = E[f(x_i)] + E[\epsilon_i] = f(x_i)$$

$$Var[y_i] = Var[f(x_i) + \epsilon_i] = Var[\epsilon_i] = \sigma^2$$

Note: $f(x_i)$ is a constant here, not a random variable. The property of $Var(X + a) = Var(X)$, where X is a random variable and a is a constant, can be used here.

4. [**[2]] **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 $\theta^\top x + b = 0$. $\vec{\theta}$ 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 $\vec{\theta}$

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

$\theta^\top x_1 + b = -1$ 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 $\frac{1}{2}||\theta||^2 = \frac{1}{2}\theta^\top\theta$ The objective of SVM is then to minimize the following

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

[2] This is offered as an optional exercise to have a better theoretical understanding.

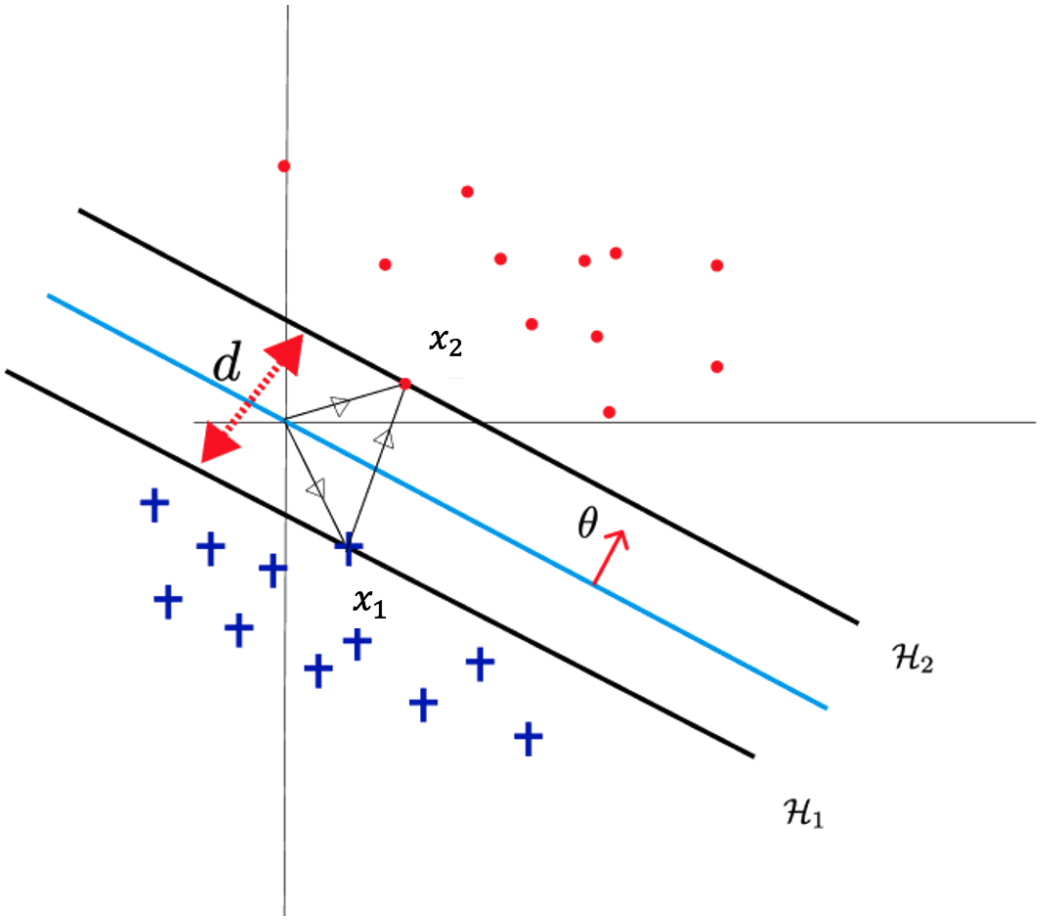

Figure 2: Support Vector Machines