Name: Honda Li

Short Answer and "True or False" Conceptual Questions

A.0

(A.0.a)

The bias and variance is similar to the concepts of precision and accuracy in Experimental Physics. Under the context of machine learning, Bias refers to part of the learning errors caused by a model being too simple, in a way that it just cant represent the joint probability density function with its simplicity.

The variance refers to the variance of the random variable $\hat{f}(x)$, which depends on the samples we observed.

Bias-variance trade off relates 2 types of learning errors (Bias, Variance) with the model complexity.

(A.0.b)

Usually, higher the model complexity, higher the variance, lower the model complexity, higher the bias.

(A.0.c)

False. The bias decreases. Because the bias is: $\mathbb{E}\left[\left(f(x) - \mathbb{E}\left[\hat{f}(x)\right]\right)^2\right]$. The number of sample seems to be irrelevant to the amount of bias we have for the model.

(A.0.d)

True when we fix the model complexity. This is absolutely true when we consider a linear regression. If we take infinitely many data, the best line-fit will converge.

This is even more true when we just consider taking the average as a way of making prediction (Which is not a bad way to predict the output given $X = x$), then, this is literally the Central limit theorem, as we have more and more samples, the variance of the sample average gets smaller.

(A.0.e)

Yes, and this is exactly the idea behind regularization, where it tries to change the objective of the optimization problem a bit, allowing the models to use less features and prevent it from over-fitting.

(A.0.f)

We should use the test set to tune the hyper-parameters for the models. This is true because the test set is indicative of true performance of the model.
Consider hypertunning the polynomial regression using training set only, where p is the degree of polynomial. Then, for any given sample that is normalized, there is a $p = n$ that where n is the total number of samples and th polynomial will fit perfectly. This is obviously over-fitting. Hence we should hypertune using the test data set.

(A.0.g)

False. It's an underestimate. This is a conclusion from the lecture.

Maximum Likelihood Estimator(MLE)

(A.1)

(A.1.a)

Objective: Find the expression for the maximum=likelihood estimate for the parameter λ for the poisson distribution, interns of the goal count. Assume idd rvs.

Here we will assume that observations obtained takes the form $x_1, x_2, \cdots x_N$, and then we derive the best estimator for λ in this much general context.

I will shut up and just show you the math:

$$\prod_{n=1}^{N} \text{Poi}(x_n|\lambda) \tag{A.1.a.1}$$

$$\sum_{n=1}^{N} \log\left(\text{Poi}(x_n|\lambda)\right)$$

$$\sum_{n=1}^{N} \left(-\lambda + x_n \ln(\lambda) + \log(x_n!)\right)$$

Notice that, only some of the terms are relevant to the parameter λ, therefore, the optimization problem we are solving is:

$$\lambda^+ = \underset{\lambda}{\text{argmax}} \left\{ -N\lambda + \ln(\lambda) \sum_{i=1}^{N} (x_n) \right\} \tag{A.1.a.2}$$

To solve it, we just take the derivative, set it to zero and then solve for λ, because this function is a function that has a single local maximum.

$$\partial_\lambda \left[-N\lambda + \ln(\lambda) \sum_{i=1}^{N} (x_n) \right] = 0 \tag{A.1.a.3}$$

$$-N + \frac{\sum_{n=1}^{N} x_n}{\lambda^+} = 0$$

$$\implies \lambda^+ = \frac{\sum_{n=1}^{N} x_n}{N}$$

Therefor, for this particular problem, the best estimator will be the average of all the observation, which is just:

$$\frac{2+4+6+1}{5} = 2.6$$

And that is the answer for the question.

(A.1.b)

The derivation of the best estimator in the general context is shown in A.1.a.

The numerical value for six observations is:

$$\frac{2+4+6+1+3}{7} = 2.6666666666\cdots$$

(A.1.c)

The numerical results for 5 observations has been shown in A.1.a and A.1.b respectively.

(A.2)

Objective: Find the MLE for the uniform distribution on $[o, \theta]$, where θ is the value we want to estimate.

Suppose that an observations has been made: $x_1, x_2, \cdots x_N$, and we assume that they are idd, and we want to find the likelihood of such an observation is generated using the Uniform distribution. And this will be given by the expression:

$$\prod_{n=1}^{N} \underbrace{\mathbb{P}\left(X = x_n\right)}_{\frac{1}{\theta}\mathbf{1}\{0 \leq x_n \leq \theta\}} \tag{A.2.1}$$

Observe that, if any of the observation is beyond the range $[0, \theta]$, we will have zero likelihood, so let's assume that $\theta \leq \max_{1 \leq i \leq N}(x_i)$, then we will have this expression for the likelihood:

$$\prod_{n=1}^{N} \frac{1}{\theta} = \frac{1}{\theta^N} \tag{A.2.2}$$

And taking the log on that we have:

$$\log\left(\frac{1}{\theta^N}\right) = -N \log(\theta) \tag{A.2.3}$$

Observe that that function is monotonically decreasing as the value of θ get larger and larger, therefore, to maximize the likelihood, we need the value of θ to be as small as possible, and the smallest possible such θ that is not giving us zero likelihood is: $\max_{1 \leq i \leq N}(x_i)$, therefore, best estimate is given by:

$$\theta^+ = \max_{1 \leq i \leq N}(x_i)$$

Over-fitting

A.3

1. $S = \{(x_i, y_i)\}_{i=1}^{N}$ drawn from idd with underlying joint distribution $\mathcal{D}$.

2. The training set is break into $S_{\text{train}}, S_{\text{test}}$, and $S = S_{\text{Train}} \cup S_{\text{Test}}$, notice that overlapping is possible.

3. And the true least square error, given a predictive model f is: $\epsilon(f) = \mathbb{E}\left[(f(x)d - y)^2\right]$.

A.3.a

We want to show that that, the expected bias of the model f over the training set and the test set is the same as expected value of the bias over the distribution $\mathcal{D}$, hence it's unbiased.

Start by considering the expected value of $\epsilon_{\text{train}}(f)$:

$$\mathbb{E}_{\text{train}}\left[\frac{1}{N_{\text{train}}} \sum_{(x,y) \in S_{\text{train}}} (f(x) - y)^2\right] \tag{A.3.a.1}$$

$$= \frac{1}{N_{\text{train}}} \sum_{(x,y) \in S_{\text{train}}} \underbrace{\mathbb{E}_{\text{train}}\left[(f(x) - y)^2\right]}_{\epsilon(f)}$$

$$\hat{\epsilon}(f) = \hat{\epsilon}_{\text{train}}(f)$$

For each term $f(x) - y$, we assume that it's idd, and therefore, its expected value is going to be the same as if it's drawn from the distribution $\mathcal{D}$, because each sample of the train set is drawn in this way, therefore,

we can conclude that $\mathbb{E}\left[(f(x) - y)^2\right]$ is going to give $\epsilon(f)$.

The proof for $\hat{\epsilon}_{\text{test}}(f)$ is going to be exact same because they are both drawn from the same idd joint distribution: $\mathcal{D}$. And hence we know that the bias estimator for any given model is going to be unbiased.

A.3.b

$\mathbb{E}_{\text{train}}\left[\hat{\epsilon}_{\text{train}}(\hat{f})\right] \neq \mathbb{E}\left[\epsilon(\hat{f})\right]$ Consider Linear regression where 2 the sample size is so small ($d - 1$ samples where d is the number of features) that we have a perfect fit model $\hat{f}$ then the bias of the model is going to be zero under the training set, however, the bias over the real distribution is not, because $\mathcal{D}$ could be non-linear and there could be noises, which both will introduce some biased.

A.3.c

Some explanations are needed for this question. Consider the following expression:

$$\underbrace{\mathbb{E}_{\text{train}}\left[\hat{\epsilon}_{\text{train}}(\hat{f}_{\text{train}})\right]}_{\epsilon(\hat{f}_{\text{train}})} \leq \mathbb{E}_{\text{train, test}}\left[\hat{\epsilon}_{\text{test}}(\hat{f}_{\text{train}})\right] \tag{A.3.b.1}$$

This is saying that, the expected value of the error on trained estimator evaluated using the test samples by enumerating over all the test samples is less than The expected value of the error when we train the model with train set and get the error with the test set. The estimator of the model is obtained via the following procedures:

Let $\mathcal{F} = \{f_1, f_2 \cdots\}$ be a collection of functions and $\hat{f}$ will minimizes the error given the training set. Now let's consider the error the expected value of the error on the test set of the estimator obtained from the train set:

$$\mathbb{E}_{\text{train, test}}\left[\hat{\epsilon}_{\text{test}}(\hat{f}_{\text{train}})\right] = \sum_{f \in \mathcal{F}} \mathbb{E}_{\text{train, test}}\left[\hat{\epsilon}_{\text{test}}(f)\mathbf{1}\{\hat{f}_{\text{train}} = f\}\right] \tag{A.3.b.2}$$

For all the model f in the model family $\{$, if f equals to the model estimator on the train set, then we get the expected value of the test error on f. Notice that the expected value operator goes in because each model f from the family is independently trained and the train, test samples drawn from the populations are independent events.

By the independence of the train set and test set we can make the claim that (here we are fixing on one of the models f from $\mathcal{F}$):

$$\mathbb{E}_{\text{train, test}}\left[\hat{\epsilon}_{\text{test}}(f)\mathbf{1}\{\hat{f}_{\text{train}} = f\}\right] = \underbrace{\mathbb{E}_{\text{test}}\left[\hat{\epsilon}_{\text{test}}(f)\right]}_{\epsilon(f)} \mathbb{E}_{\text{train}}\left[\mathbf{1}\{\hat{f}_{\text{train}} = f\}\right] \tag{A.3.b.3}$$

Take not that, in order to compute the sum expected value, we need the probability that f is indeed the estimated model obtained by the train set. Intuitively, depending the randomness of the samples from the populations, the function $\hat{f}_{\text{train}}$ might be different from time to time, but they should be closed to the true model because it's not biased. And in this case, we can make the claim that:

$$\mathbb{E}_{\text{train}}\left[\mathbf{1}\{\hat{f}_{\text{train}} = f\}\right] = \mathbb{P}\left(\hat{f}_{\text{train}} = f\right) \tag{A.3.b.4}$$

So then go back to A.3.b.2:

$$\mathbb{E}_{\text{train, test}}\left[\hat{\epsilon}_{\text{test}}(\hat{f}_{\text{train}})\right] = \sum_{f \in \mathcal{F}} \mathbb{E}_{\text{train, test}}\left[\hat{\epsilon}_{\text{test}}(f)\mathbf{1}\{\hat{f}_{\text{train}} = f\}\right] \tag{A.3.b.4}$$

$$= \sum_{f \in \mathcal{F}} \epsilon(f)\mathbb{P}\left(\hat{f}_{\text{train}} = f\right)$$

$$= \mathbb{E}_{\mathcal{F}}\left[\epsilon(f)\right]$$

4

And notice that, because $\hat{f}_train$ has the best errors among all the functions in $\mathcal{F}$, therefore it's less than the above quantity:

$$\underbrace{\mathbb{E}_{\text{train}}\left[\hat{\epsilon}_{\text{train}}(\hat{f}_{\text{train}})\right]}_{\epsilon(\hat{f}_{\text{train}})} \leq \mathbb{E}_{\mathcal{F}}\left[\epsilon(f)\right] \tag{A.3.b.5}$$

Polynomial Regression

Ridge Regression on MNIST