DS 100/200: Principles and Techniques of Data Science **Date: October 30, 2019**

Discussion # 10

Name: Raguvir Kunani

Bias-Variance Trade-off

1. Assume that we have a function $h(x)$ and some noise generation process that produces ϵ such that $\mathbb{E}[\epsilon] = 0$ and $\text{var}(\epsilon) = \sigma^2$. Every time we query mother nature for Y at a given a x, she gives us $Y = h(x) + \epsilon$. A new ϵ is generated each time, independent of the last. We randomly sample some data $(x_i, y_i)_{i=1}^n$ and use it to fit a model $f_{\hat{\beta}}(x)$ according to some procedure (e.g. OLS, Ridge, LASSO). In class, we showed that

$$\underbrace{\mathbb{E}\left[(Y - f_{\hat{\beta}}(x))^2\right]}_{\text{model risk}} = \underbrace{\sigma^2}_{\text{obs. variance}} + \underbrace{(h(x) - \mathbb{E}\left[f_{\hat{\beta}}(x)\right])^2}_{\text{model bias}^2} + \underbrace{\mathbb{E}\left[(\mathbb{E}\left[f_{\hat{\beta}}(x)\right] - f_{\hat{\beta}}(x))^2\right]}_{\text{model variance}}.$$

   (a) Label each of the terms above. Word bank: observation variance, model variance, observation bias^2, model bias^2, model risk, empirical mean square error.

   (b) What is random in the equation above? Where does the randomness come from?

   Y is random because there is noise ($Y = h(x) + \epsilon$)

   $f_{\hat{\beta}}(x)$ is random bc we assume training data is a random sample of the population $\Rightarrow \hat{\beta}$ is random

   (c) True or false and explain. $\mathbb{E}\left[\epsilon f_{\hat{\beta}}(x)\right] = 0$

   $\mathbb{E}[\epsilon f_{\hat{\beta}}(x)] = \mathbb{E}[\epsilon]\mathbb{E}[f_{\hat{\beta}}(x)] = 0$ bc $\mathbb{E}[\epsilon] = 0$

   $\mathbb{E}[\epsilon f_{\hat{\beta}}(x)] = \mathbb{E}[\epsilon]\mathbb{E}[f_{\hat{\beta}}(x)]$ because ϵ and $f_{\hat{\beta}}(x)$ are independent

   (d) Suppose you lived in a world where you could collect as many data sets you would like. Given a fixed algorithm to fit a model f_β to your data e.g. linear regression, describe a procedure to get good estimates of $\mathbb{E}\left[f_{\hat{\beta}}(x)\right]$ (technical point: you may assume this expectation exists).

   - collect a large # of datasets $D_1, D_2, \ldots D_N$
   - compute $f_\beta(x)_i$ for each dataset D_i
   - $\mathbb{E}\left[f_{\hat{\beta}}(x)\right] = \frac{1}{n}\sum_{i=1}^{n} f_\beta(x)_i$

   (e) If you could collect as many data sets as you would like, how does that affect the quality of your model $f_\beta(x)$?

   It doesn't; collecting more data will not make our model any better

Ridge and LASSO Regression

2. Earlier, we posed the linear regression problem as follows: Find the $\vec{\beta}$ value that minimizes the average squared loss. In other words, our goal is to find $\hat{\vec{\beta}}$ that satisfies the equation below:

$$\hat{\vec{\beta}} = \underset{\vec{\beta}}{\operatorname{argmin}}\, L(\vec{\beta}) = \underset{\vec{\beta}}{\operatorname{argmin}}\, \frac{1}{n}||\vec{y} - \mathbb{X}\vec{\beta}||_2^2$$

Here, $\mathbb{X}$ is a $n \times d$ matrix, $\vec{\beta}$ is a $d \times 1$ vector and $\vec{y}$ is a $n \times 1$ vector. As we saw in lecture and in last week's discussion, the optimal $\hat{\vec{\beta}}$ is given by the closed form expression $\hat{\vec{\beta}} = (\mathbb{X}^t\mathbb{X})^{-1}\mathbb{X}^t\vec{y}$.

To prevent overfitting, we saw that we can instead minimize the sum of the average squared loss plus a regularization function $\lambda\mathcal{S}(\vec{\beta})$. If use the function $\mathcal{S}(\vec{\beta}) = ||\vec{\beta}||_2^2$, we have "ridge regression". If we use the function $\mathcal{S}(\vec{\beta}) = ||\vec{\beta}||_1$, we have "LASSO regression". For example, if we choose S($\vec{\beta}$) = $||\vec{\beta}||_2^2$, our goal is to find $\hat{\vec{\beta}}$ that satisfies the equation below:

$$\hat{\beta} = \underset{\vec{\beta}}{\operatorname{argmin}}\, L(\vec{\beta}) = \underset{\vec{\beta}}{\operatorname{argmin}}\, \frac{1}{n}||\vec{y} - \mathbb{X}\vec{\beta}||_2^2 + \lambda||\vec{\beta}||_2^2 = \underset{\vec{\beta}}{\operatorname{argmin}}\, \frac{1}{n}\sum_{i=1}^{n}(y_i - \mathbb{X}_{i,\cdot}^T\vec{\beta})^2 + \lambda\sum_{j=1}^{d}\beta_j^2$$

Recall that λ is a hyperparameter that determines the impact of the regularization term. Though we did not discuss this in lecture, we can also find a closed form solution to ridge regression: $\hat{\vec{\beta}} = (\mathbb{X}^T\mathbb{X} + \lambda\mathbf{I})^{-1}\mathbb{X}^T\vec{y}$. It turns out that $\mathbb{X}^T\mathbb{X} + \lambda\mathbf{I}$ is guaranteed to be invertible (unlike $\mathbb{X}^T\mathbb{X}$ which might not be invertible).

(a) As model complexity increases, what happens to the bias and variance of the model?

Complexity increases ⇒ bias decreases (we can "overfit" the data)
variance increases (overfitting ⇒ not generalizable)

(b) In terms of bias and variance, how does a regularized model compare to ordinary least squares regression?

regularized model has → higher bias (not minimizing the L_2 loss directly)
→ lower variance (encourages smaller weights)

(c) In ridge regression, what happens if we set $\lambda = 0$? What happens as λ approaches ∞?

$\lambda = 0$ ⇒ loss function becomes $\frac{1}{n}||y - X\beta||_2^2$ (same as least squares)

$\lambda = \infty$ ⇒ placing infinite penalty on model weights ⇒ $\hat{\beta} = \vec{0}$

(d) How does model complexity compare between ridge regression and ordinary least squares regression? How does this change for large and small values of λ?

- model complexity decreases with increasing regularization
- increasing λ increases regularization, vice versa

(e) If we have a large number of features (10,000+) and we suspect that only a handful of features are useful, which type of regression (Lasso vs Ridge) would be more helpful in interpreting useful features?

LASSO encourages model weights to become 0 (just memorize this)

(f) What are the benefits of using ridge regression?

- decrease model variance (most important benefit)
- closed-form solution: $\hat{\beta} = (X^T X + \lambda I)^{-1} X^T y$
  - don't need to worry about rank of X like before

Cross Validation

3. After running 5-fold cross validation, we get the following mean squared errors for each fold and value of λ:

Fold Num	$\lambda = 0.1$	$\lambda = 0.2$	$\lambda = 0.3$	$\lambda = 0.4$	Row Avg
1	80.2	70.2	91.2	91.8	83.4
2	76.8	66.8	88.8	98.8	82.8
3	81.5	71.5	86.5	88.5	82.0
4	79.4	68.4	92.3	92.4	83.1
5	77.3	67.3	93.4	94.3	83.0
Col Avg	79.0	68.8	90.4	93.2	

How do we use the information above to choose our model? Do we pick a specific fold? a specific lambda? or a specific fold-lambda pair? Explain.

we use cross-validation to pick a specific λ, which we will then use to train our model on the entire training set

In this case, we choose $\lambda = 0.2$ bc it has lowest MSE across all folds (Col. Avg.)

4. You build a model with two regularization hyperparameters λ and γ. You have 4 good candidate values for λ and 3 possible values for γ, and you are wondering which λ, γ pair will be the best choice. If you were to perform five-fold cross-validation, how many validation errors would you need to calculate?

(4 choices for λ) x (3 choices for γ) x (5 folds) = 60 validation errors

5. In the typical setup of k-fold cross validation, we use a different parameter value on each fold, compute the mean squared error of each fold and choose the parameter whose fold has the lowest loss.

○ A. True

● B. False

we use the same parameter value across folds so we can compute the average error across folds for that choice of parameter value