surajrampure.com

DS 100: Principles and Techniques of Data Science **Date: October 10, 2018**

Discussion #7

Name:

$\text{var}(\square) = E\left[(E(\square) - \square)^2\right]$

$h(x)$: true values

$Y = h(x) + \epsilon$: observe

$f_{\hat{\theta}}(x)$: model

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 $f_{\hat{\theta}}(x)$ according to some procedure (e.g. OLS, Ridge, LASSO). In class, we showed that

Bias-Variance Decomp.

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

   (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.

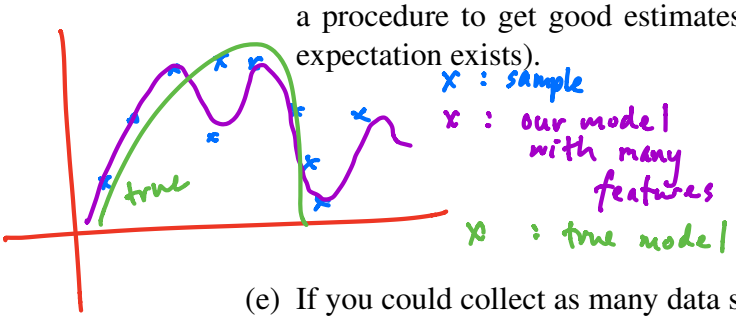
   (b) What is random in the equation above? Where does the randomness come from?

Y is random as it depends on ϵ

$f_{\hat{\theta}}(x)$ depends on Y, ∴ it is also random

   (c) ~~True or false and explain: $\mathbb{E}[\epsilon f_{\theta}(x)] = 0$~~

   (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_θ to your data e.g. linear regression, describe a procedure to get good estimates of $\mathbb{E}[f_{\hat{\theta}}(x)]$ (technical point: you may assume this expectation exists).



b) Y is random as it involves random noise

$f_{\hat{\theta}}(x)$ depends on Y,

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

- gives good estimate of your desired model
- doesn't tell you quality of the model

model(x) = 12

2. We find the optimal $\boldsymbol{\theta}$ that minimizes squared loss with L_1 regularization:

$$\hat{\theta} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \frac{1}{n}||\mathbf{y} - \boldsymbol{\Phi}\boldsymbol{\theta}||_2^2 + \lambda||\boldsymbol{\theta}||_1 = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \frac{1}{n}\sum_{i=1}^{n}(y_i - \boldsymbol{\Phi}_{\mathbf{i},\cdot}{}^{\mathbf{T}}\boldsymbol{\theta})^{\mathbf{2}} + \lambda\sum_{\mathbf{j=1}}^{\mathbf{d}}|\theta_{\mathbf{j}}|.$$

You receive all your (nice, clean, numerical) data in a single `DataFrame` called CompleteData. The first column contains the responses and the remaining d columns hold the features. You want to use 60% of your data in the training set and implement part of 5-fold cross-validation with the following pseudocode:

```
Phi_train, Y_train, Phi_test, Y_test =\
    make_train_test_split(CompleteData, 0.60)
lambdas = make_lambdas(from=0.1, to=0.4, by=0.1)

n = count_rows(...)
fold_size = n / k
idx = range(n)
randomly shuffle the ordering of idx
folds = [idx[i * fold_size : (i+1) * fold_size] for i in range(k)]

for i, fold in enumerate(folds):
    for j, lam in enumerate(lambdas):
      mse[i, j] = calculate_mse_lasso(Phi__, Y__, fold, lam)
```

(a) What should the `...` be in `count_rows` above? Your choices are `CompleteData`, `Phi_train`, and `Phi_test`.

(b) What should the blanks be in `calculate_mse_lasso` above? Your choices are `train` and `test`.

(c) Describe an algorithm for `calculate_mse_lasso`.

take λ with least error!

(d) 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	

should all be close

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.

Ridge Regression

$\|y-\Phi\theta\|_2^2 + 10^{100}\|\theta\|_2^2$ $\lambda\theta = c$

3. Ridge regression is a variant of least squares that involves regularization. The problem is stated as follows:

OLS

$$\hat{\theta} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\, L(\boldsymbol{\theta}) = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\, \frac{1}{n}||\mathbf{y} - \boldsymbol{\Phi}\boldsymbol{\theta}||_2^2 + \lambda||\boldsymbol{\theta}||_2^2 = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\, \frac{1}{n}\sum_{i=1}^{n}(y_i - \boldsymbol{\Phi}_{\mathbf{i},\cdot}{}^T\boldsymbol{\theta})^2 + \lambda\sum_{j=1}^{d}\theta_j^2$$

Here, λ is a hyperparameter that determines the impact of the regularization term. $\boldsymbol{\Phi}$ is a $n \times d$ matrix, $\boldsymbol{\theta}$ is a $d \times 1$ vector and $\mathbf{y}$ is a $n \times 1$ vector. The optimal choice is $\hat{\boldsymbol{\theta}} = (\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \lambda\mathbf{I})^{-1}\boldsymbol{\Phi}^T\mathbf{y}$.

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

λ↑, complexity decreases
- bias increases - variance decreases (and vice versa)

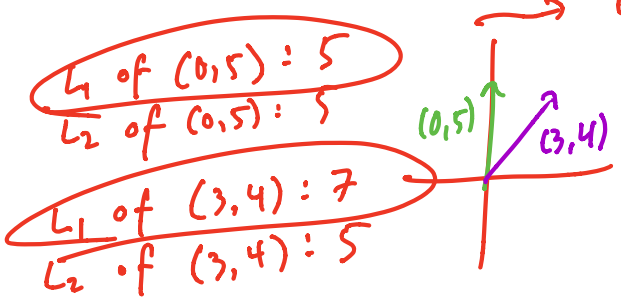
(b) In terms of bias and variance, how does the a regularized regression estimator compare to ordinary least squares regression?

OLS has the most complex model
∴ RR has higher bias, lower var

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

λ=0: this is just OLS λ→∞ : θ→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?



LASSO: L_1
RIDGE: L_2

$x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$

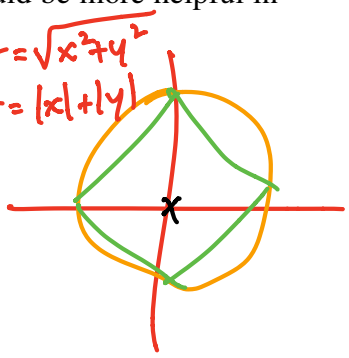
$L_1(x) = |x_1| + |x_2| + \dots + |x_n|$

$L_2(x) = (x_1^2 + x_2^2 + \dots + x_n^2)^{1/2}$

(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: $L(\theta) = \|y - \Phi\theta\|_2^2 + \lambda\|\theta\|_1$

$L_2: r = \sqrt{x^2+y^2}$
$L_1: r = |x| + |y|$



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

- prevents overfitting
- ensures a solution

(g) On last week's discussion, we discussed possible situations where the matrix $\Phi^T\Phi$ was not invertible, such as the presence of linearly dependent columns or an insufficient number of observations. In this question, we will demonstrate that the L_2 regularization penalty always ensures that the matrix $(\mathbf{\Phi}^T\mathbf{\Phi} + \lambda\mathbf{I})^{-1}$ is invertible, resulting in a unique solution.

$a^Ta = \|a\|_2^2$

i. A symmetric matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ is **positive semi-definite** if for every non-zero vector $\mathbf{v} \in \mathbb{R}^n$, we have $\mathbf{v}^T\mathbf{A}\mathbf{v} \geq 0$. Given a matrix $\mathbf{\Phi} \in \mathbb{R}^{n\times d}$ (think our feature matrix), show that $\mathbf{\Phi}^T\mathbf{\Phi}$ is positive semi-definite.

$v^T\Phi^T\Phi v = (\Phi v)^T\Phi v = \|\Phi v\|_2^2 \geq 0$

ii. A symmetric matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ is **positive definite** if for every non-zero vector $\mathbf{v} \in \mathbb{R}^n$, we have $\mathbf{v}^T\mathbf{A}\mathbf{v} > 0$. Notice that the inequality is now strict. Given a matrix $\mathbf{\Phi} \in \mathbb{R}^{n\times d}$ (think our feature matrix) and $\lambda > 0$ (our regularization hyperparameter), show that $\mathbf{\Phi}^T\mathbf{\Phi} + \lambda\mathbf{I}$ is positive definite.

$v^T(\Phi^T\Phi + \lambda I)v = v^T\Phi^T\Phi v + \lambda v^Tv = \|\Phi v\|_2^2 + \lambda\|v\|_2^2 > 0$

iii. Prove that a positive-definite matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ is always invertible by showing that its null space only contains the zero vector i.e.

$$N(\mathbf{A}) = \{\mathbf{0}\}$$

Proof by Contradiction

Assume there exists some $v \in N(A)$, $v \neq 0$
$\rightarrow Av = 0$

$v^TAv = v^T0 = 0$

but, we assume A is PD, i.e. $v^TAv > 0$

contradiction! $\therefore N(A) = \{0\}$