STAT 5630 - Statistical Machine Learning

due friday september 20, 2019

Max Ryoo (hr2ee)

Professor: Xiwei (Denny) Tang

Homework1

problem 1 *Linear Regression*

Given that:

$$Y = f(x) + \epsilon,$$
$$E[\epsilon] = 0$$
$$Var[\epsilon] = \sigma^2$$
$$Err(x^*) = E[(Y^* - \hat{f}(x^*))^2]$$

Show that:

$$Err(x^*) = Bias(\hat{f})^2 + Var(\hat{f}) + \sigma^2$$

Proof.

$Err(x^*) = E[(Y^* - \hat{f}(x^*))^2]$

$= E[Y^{*2} - 2Y^*\hat{f}(x^*) + \hat{f}(x^*)^2]$ — Because we know that $Y^* = f(x^*) + \epsilon$

$= E[\{f(x^*) + \epsilon\}^2 - 2(\{f(x^*) + \epsilon\}\hat{f}(x^*)) + \hat{f}(x^*)^2]$

$= E[f(x^*)^2] + 2E[f(x^*)\epsilon] + E[\epsilon^2] - 2E[f(x^*)\hat{f}(x^*)] - 2E[\epsilon\hat{f}(x^*)] + E[\hat{f}(x^*)^2]$ — If we rearrange ...

$= E[f(x^*)^2] - 2E[f(x^*)\hat{f}(x^*)] + E[\hat{f}(x^*)^2] + 2E[f(x^*)\epsilon] + E[\epsilon^2] - 2E[\epsilon\hat{f}(x^*)]$ — by factoring we can summarize

$= E[(f(x^*) - \hat{f}(x^*))^2] + 2E[f(x^*)\epsilon] + E[\epsilon^2] - 2E[\hat{f}(x^*)\epsilon]$ — We then use the fact that ϵ and $f(x)$ are independent...

$= E[(f(x^*) - \hat{f}(x^*))^2] + 2E[f(x^*)]E[\epsilon] + E[\epsilon^2] - 2E[\hat{f}(x^*)]E[\epsilon]$ — since $E[\epsilon] = 0$...

$= E[(f(x^*) - \hat{f}(x^*))^2] + 0 + E[\epsilon^2] - 0$ — Using the definition of variance $Var(x) = E[x^2] - E[x]^2$

$= E[(f(x^*) - \hat{f}(x^*))^2] + Var(\epsilon) + E[\epsilon]^2$

$= E[(f(x^*) - \hat{f}(x^*))^2] + \sigma^2 + 0$ — Using definition of variance where $x = f(x^*) - \hat{f}(x^*)$...

$= Var(f(x^*) - \hat{f}(x^*)) + E[f(x^*) - \hat{f}(x^*)]^2 + \sigma^2$ — Bias is defined as the expected difference between estimator and true value so...

$= Var(f(x^*) - \hat{f}(x^*)) + Bias(\hat{f})^2 + \sigma^2$ — $f(x^x)$ is a constant number so ...

$$= Var(\hat{f}) + Bias(\hat{f})^2 + \sigma^2 \qquad \text{Rearranging...}$$
$$= Bias(\hat{f})^2 + Var(\hat{f}) + \sigma^2$$

□

PROBLEM 2 *Degree of Freedom*

1. For 1NN show that df = n

   Proof.

$$\hat{Y}(x) = \frac{1}{k} \sum_{x_i \in N_k(x)} y_i \qquad \text{Def' of KNN}$$
$$= y_i \qquad \text{Since when } k = 1 \text{ every point will be its own group}$$
$$df = \frac{1}{\sigma^2} \sum_{i=1}^{n} Cov(\hat{Y}_i, Y_i) \qquad \text{Def' of DF}$$
$$= \frac{1}{\sigma^2} \sum_{i=1}^{n} Cov(Y_i, Y_i) \qquad \hat{Y}_i \text{ is } Y_i \text{ as shown above}$$
$$= \frac{1}{\sigma^2} [n * \sigma^2] \qquad Cov(X, X) = Var(X) = \sigma^2$$
$$df = n$$

   □

2. If $\hat{y}_i = \bar{y} = \frac{1}{n} \sum_{i=1}^{n} \hat{y}_i$ e.g., nNN, show that df=1

   Proof.

$$df = \frac{1}{\sigma^2} \sum_{i=1}^{n} Cov(\hat{Y}_i, Y_i) \qquad \text{Def' of DF}$$
$$= \frac{1}{\sigma^2} \sum_{i=1}^{n} Cov(\bar{Y}, Y_i) \qquad \text{Given Substitution}$$
$$= \frac{1}{\sigma^2} \sum_{i=1}^{n} Cov(\frac{1}{n} \sum_{j=1}^{n} Y_j, Y_i) \qquad \text{Def' of } \bar{Y}$$
$$= \frac{1}{\sigma^2} \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{n} Cov(Y_j, Y_i) \qquad \text{Properties of Covariance/rearranging}$$

   Since we assume Y_j and Y_i are independent...

   When $i \neq j$... $Cov(Y_j, Y_i) = 0$

   When $i = j$... $Cov(Y_j, Y_i) = \sigma^2$

   This case of $i = j$ happense n times. Thus the expression will be...

$$= \frac{1}{\sigma^2} * \frac{1}{n} * \sigma^2 * n$$
$$df = 1$$

   □

3. For Linear Regression $Y_{nx1} = X_{nxp}B_{px1} + \epsilon_{nx1}$ show that df = p

Proof.

$Y_{nx1} = X_{nxp}B_{px1} + \epsilon_{nx1}$	Let's rewrite def' of Y to $\hat{Y}$
$\hat{Y} = X\hat{\beta}$	Def of $\hat{\beta}$
$= X(X^TX)^{-1}X^TY$	Let's say $H = X(X^TX)^{-1}X^T$...
$= HY$	
$df = \frac{1}{\sigma^2}Trace(Cov(\hat{Y}, Y))$	Def of DF in matrix form
$= \frac{1}{\sigma^2}Trace(Cov(HY, Y))$	Substitution of $\hat{Y}$..
$= \frac{1}{\sigma^2}Trace(H(Cov(Y, Y))$	$Cov(Ab, b) = Cov(b)$ if A is a constant matrix
$= \frac{\sigma^2}{\sigma^2}Trace(H)$	Since $Cov(Y, Y) = \sigma^2$
$= Trace(X(X^TX)^{-1}X^T)$	Original Definition...
$= Trace(X^TX(X^TX)^{-1})$	$Trace(BA) = Trace(AB)$
$df = p$	$X^TX(X^TX)^{-1}$ is Identity Matrix so Trace of p by p Identity matrix is p

□

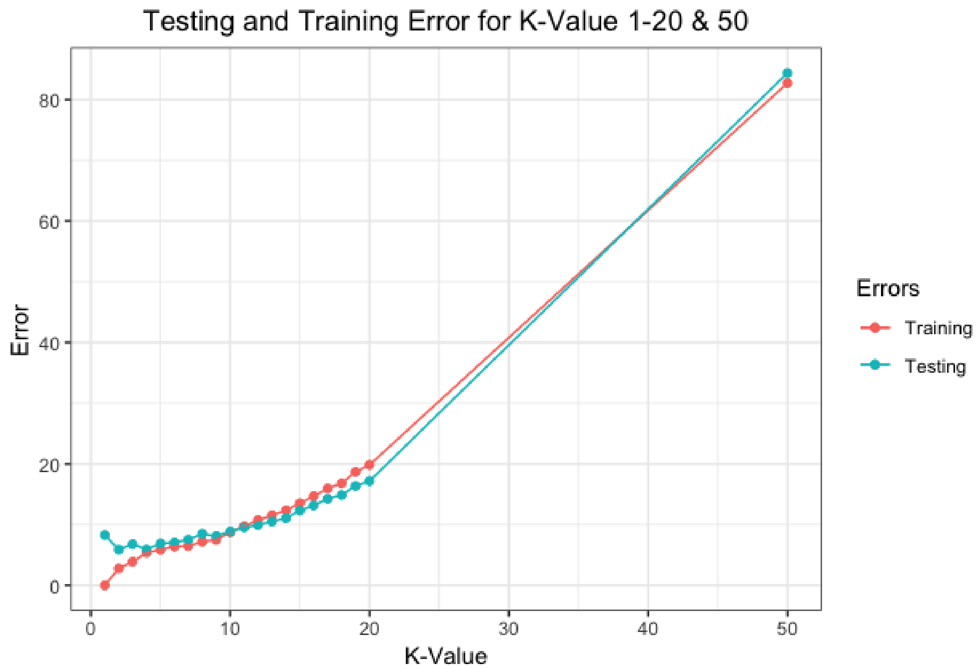
PROBLEM 3 *Coding*

1. KNN $k = 1 - 20, 50$

K-Value	Training Error	Testing Error
1	0	8.26798956869082
2	2.75032853281866	5.87808874456799
3	3.87920153460641	6.76340373145879
4	5.37699387388468	5.89500759673447
5	5.86287264356344	6.8791861922426
6	6.37722334515822	7.04528583725008
7	6.46712050610632	7.5281469700618
8	7.1783506328033	8.48220854262109
9	7.50362009283095	8.10092075246868
10	8.74333081753121	8.8558638698073
11	9.7261423169281	9.51482319082283
12	10.7316031393558	9.92003380174122
13	11.5442207261346	10.5116422513043
14	12.3392168312112	11.0471601959592
15	13.5021723407372	12.2833236813108
16	14.6759068825225	13.1642926397814
17	15.9456296118624	14.2636489576971
18	16.8121702481737	14.8591266026838
19	18.6724312855749	16.3472782229595
20	19.8339988847786	17.1650222033091
50	82.7541538986692	83.537833441402

The best K value was 2, which yielded a Training Error of 2.75032853281866 and a Testing Error of 5.87808874456799.

The following shows the training and testing error for different values of K...



2. Linear Regression
   When using Linear Regression Function the **Training Error** was 18.93879 and the **Testing Error** was 20.86782. This was evidently worse than the KNN model with a K value of 2.

   Based on the observed data we can infer that the KNN model did better by first looking at the respective training and testing errors, which KNN did better by much with a K-value of 2. Also, we can intuitively see that the KNN model will work well due tot he points of the generated data not being a linear relationship.

3. Cross Validation
   The table containing the Training and Validation error is included in the next page...

   For performing Cross Validation on the dataset with 40 training samples and 10 testing samples (all from the traning dataset) the best K value was 6. The testing error was fairly close between $K < 10$ and after 10 started increasing in Testing Errors. This K value differed from when Cross Validation wasn't performed. Previously the best K Value had been 2. Generally the Testing Errors while doing cross validation increased by a slight amount compared to the respective errors resulted from not doing cross validation.

   Also from the graph of Training & Testing Error for different values of K, the training error continually increased while the testing error was similar to one another for $K < 10$ and followed the continually increasing pattern after 10.

K-Value	Training Error	Testing Error
1	0	10.7635768456058
2	1.10874732630578	9.05760359106116
3	2.63363172968399	8.96355223125826
4	3.78676509539604	9.12264624464492
5	4.56053382129996	8.98941755091128
6	5.16661579647145	8.80561076996737
7	5.76883006485741	8.81221053881018
8	6.40027853160722	9.01720593674725
9	7.08088375651029	9.47268787488331
10	7.81197663170943	10.1030077747225
11	8.57268731829235	10.7919763306006
12	9.35907771973858	11.4687294542375
13	10.1925304957847	12.177054441347
14	11.0750245890924	13.0017756234942
15	12.0027519044555	13.9150396383121
16	12.9643203929651	14.859055181052
17	13.9653486071805	15.8456306140482
18	15.0060593770221	16.8907128827717
19	16.0942219236379	17.9609160990587
20	17.2059721724784	19.0711521649568
40	48.3769115081294	49.6139549479778

The following graph was plotted to show the training and testing error for different values of K using cross validation...

The reason that Max K value was set to 40 is due to the fact that the training dataset using cross validation will only have 40 points. Based on the definition of KNN K can only be set to the number of records in the dataset since having a $K = n$ means that each point will be a group with itself. Having a $K > n$ does not apply and does not make logical sense to KNN.