Tanmay Gupta
+9289

MATH-680

Homework-3

(1)

Regression Shrinkage & Selection via the LASSO.

-> The paper focuses on the best subset selection technique using LASSO. ~~regression strategy~~ It also emphasizes on the fact that lasso minimizes the residual sum of squares in subject to the sum of the absolute value of the coefficient being less than a constant.

-> The findings suggest that LASSO enjoys some favourable properties of both subset selection & ridge regression.

-> Also the lasso idea is quite general & can be applied to various statis models as extensions of gen. reg. models or tree-based models.

Scanned by CamScanner

(2) [2.7]

(a)

For linear regression, the coeff. $\hat{\beta}$ depends on X.

$$\hat{\beta} = (X^T X)^{-1} X^T y.$$

Here X_{ij} = $\begin{bmatrix} x_1^T \\ x_2^T \\ x_3^T \\ \vdots \\ x_N^T \end{bmatrix}$

for the model fitting at x_0 →

$$\hat{f}(x_0) = x_0^T \hat{\beta} = x_0^T (X^T X)^{-1} X^T y$$

$\left[(X^T X)^{-1} X^T\right]_j$ is the j^{th} row of $(X^T X)^{-1} X^T$

The weights are →

$$l_i(x_0, X) = x_{01}\left[(X^T X)^{-1} X^T\right]_1 + x_{02}\left[(X^T X)^{-1} X^T\right]_2 \ldots$$

$$+ x_{0p}\left[(X^T X)^{-1} X^T\right]_p$$

$$\Rightarrow \sum_{i=1}^{p} \left(x^T D (x^T x)^{-1} x^T \right)_i$$

So the weights depend on the entire set on y_i.

for K-nearest neighbour eg. function ->

$$\hat{f}(x_0) = \frac{1}{k} \sum_{x_i \in N_k(x_0)} y_i$$

weights here are ->

$$\hat{f}(x_0) = l_i(x_0, D) = 1[x_i \in N_k(x_0)]$$

we define D as

$$D = \{ d(x_i, x_0) ; x_i \in T \}$$

Also

$$N_k(x_0) = \{ x_i : d(x_i, x_0) \leq d_k \}$$

where d_k is the k^{th} smallest distance

So for D we have to search the training set & weights depend on set X

(b)

~~Taeufdaap~~

X is fixed and Y varies

x_0 & $f(x_0)$ are fixed.

So, $E_{y|x}(f(x_0) - \hat{f}(x_0))^2$

$= f(x_0)^2 - 2 f(x_0) . E_{y|x}(\hat{f}(x_0)) + E_{y|x}((\hat{f}(x_0)^2)$

$= (f(x_0) - E_{y|x}(\hat{f}(x_0)))^2 + (E_{y|x}(\hat{f}(x_0)^2)) - ~~(E_{y|x}(\hat{f}(x_0))^2~~$
$E_{y|x}(\hat{f}(x_0)))^2$

$= (bias)^2 + var(\hat{f}(x_0))$

(3) [Ex-2.8 for solution]

For linear regression ->
Training error = 0.576
Testing error = 4.121

for KNN →

K = 1	Training error = 0	Testing error = 2.473
K = 3	" " = 0.504	" " = 3.022
K = 5	" " = 0.576	" " = 3.022
K = 7	" " = 0.648	" " = 3.022
K = 15	" " = 0.936	" " = 3.846

We can see that KNN has less testing error as compared to linear regression for all the values of K.

Relatively training error is less for K=1 & K=3 and equal for K=5. Linear reg. error is more for K=7 or K=15 than linear reg.

So as the no of K increases the training error increases.

(4) K-NN Classifier.

(4)

(a)

K	Training error	Testing error
1	0	33.4
4	27	32.2
7	27	26.2
10	26	26.4
15	28	25.8
16	30	25.2
30	27.5	21.8
48	28.5	21.9
60	28.5	21.2
70	28.5	21.8
100	28.5	22.3
150	28.5	22.6
200	46	50.1

Plotting K code

(b)

K	Training error	Testing error
1	0	24.5
4	18.5	26.2
7	21	25
10	23	22.9
13	22.5	22.2
16	24.5	22.8
30	23	23.2
45	25	23.5
60	26.5	23.8
80	28	26.2
100	27	25.9
150	30.5	30
200	53	49.6

Plots in R code.

(c)

We choose the K which gives the least testing error. The plots show similar pattern with errors

a increasing training and testing error as the no. of K is [illegible]. ~~A [illegible]~~ In the E/K plot the training error is declining.

The best K for scenario-1 is K=60
" " " " " -2 " K=13

(5)

(i)

K	Training error	Testing error
1	0	2.23
3	0.42	2.39
5	0.46	2.39
7	0.58	2.55
15	1.04	3.48

(ii) LDA training error = 0.50 %

LDA testing error = 3.50 %