Machine Learning, Spring 2018
Homework 4

Due on 23:59 May 3, 2018
Send to *cs*282_01@163.*com*
with subject "Chinese name+student number+HW4"

1 Hoeffding Inequality: Solution

(1) $Pr = C_{10}^{1} \times 0.1^{9} \times 0.9 + 0.1^{10} = 9.1 \times 10^{-9}$

(2) $Pr[|v - \mu| \geq 0.8] \leq 2e^{-2\epsilon^2 \times m} = 2e^{-2\times 0.8^2 \times 10} = 2 \times e^{-12.8}$

2 Bias-variance decomposition: Solution

To simplified the expression, we omit the * here, use h to denote $h(x^*)$, $\bar{h}$ to denote $\overline{h(x^*)}$, y to denote y^*, f to denote $f(x^*)$.

The proof is as follows:
$$\begin{aligned} & E[(y-h)^2] = E[(y-\bar{h}+\bar{h}-h)^2] \\ &= E[(y-\bar{h})^2 + (\bar{h}-h)^2 + 2(h-\bar{h})(y-\bar{h})] \\ &= E[(y-\bar{h})^2] + E[(h-\bar{h})^2] + 2E[(h-\bar{h})\cdot(y-\bar{h})] \end{aligned}$$

What's more, $E[(h-\bar{h})\cdot(y-\bar{h})] = E[h\cdot y - h\cdot\bar{h} - \bar{h}\cdot y + \bar{h}\cdot\bar{h}]$. Since $\bar{h}$ is a fix value, and h is independent to y, we have: $E[h\cdot y - \bar{h}\cdot y] = E[h]\cdot E[y] - \bar{h}\cdot E[y] = 0$, $E[\bar{h}\cdot\bar{h} - h\cdot\bar{h}] = 0$. Then it equals to:
$$\begin{aligned} & E[(y-h)^2] = E[(y-\bar{h})^2] + E[(h-\bar{h})^2] \\ &= E[(\bar{h}-f+f-y)^2] + E[(h-\bar{h})^2] \\ &= E[(\bar{h}-f)^2] + E[(f-y)^2] + 2E[(\bar{h}-f)\cdot(f-y)] + E[(h-\bar{h})^2] \end{aligned}$$

Generally, the mean of noise is zero, i.e., $E[(y-f)] = 0$. Then $2E[(\bar{h}-f)\cdot(f-y)] = 0$ ($\bar{h}$ and f are both fix value, independent of the distribution $\mathcal{D}$). Then we have:

$$E[(y-h)^2] = E[(\bar{h}-f)^2] + E[(f-y)^2] + E[(h-\bar{h})^2] = \textbf{bias}^2 + \sigma^2 + \textbf{variance}$$

3 Cross Validation And L2 Regularization: solution

(1) Note that: since we have 1595 samples in training data, I change k to 11, which allows a division with no remainder. $1595 = 11 \times 145$. The Fig 1 shows the 11-folder splited training data result. The MSE here we use: $MSE = 0.5sum([(y_{predict}[i] - y_{test}[i])^2 for\ i\ in\ range(m)])/m$

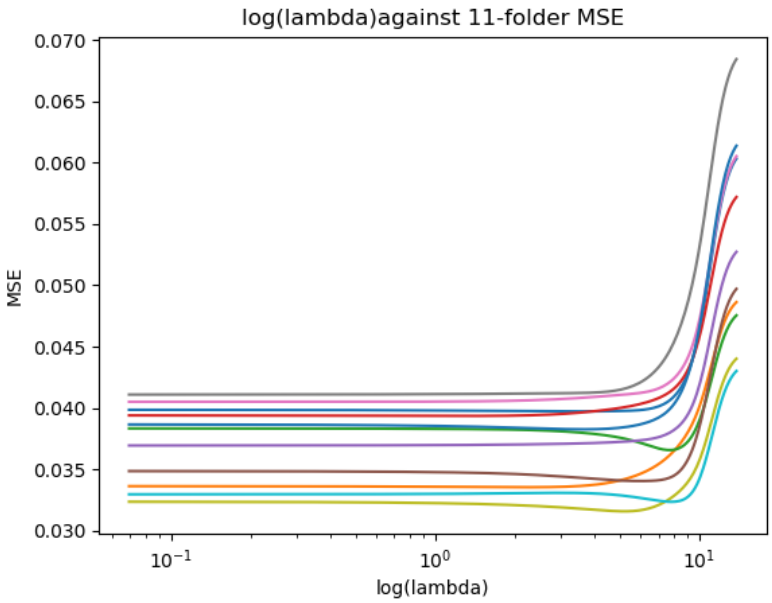

Figure 1: log(lambda) against 11-folder MSE in training data

And the average result is shown in Fig 2.

(2) From the result above, we choose the best performance model: $\log(\lambda) \approx 5.2068507128$. Then we use the whole training data to build model, and test on testing data, the result is shown in Fig 3. The loss of testing data is: 0.036646418879

(3) The relationship between λ and number of small coefficients is shown in Fig 4. Here we set the threhold of "small" as: 1e-03. Actually, too positive or too negative coefficients both indicate a strong relationship to the crime-rate, so what really matters(the real small, we consider) should be the coefficients with an very small absolute value, so here we also draw another figure with abs-threhold in Fig 5.

From the graph we can see that, when λ is very large, the number of small coefficients grow exponential with λ's growth, and the loss is increasing also

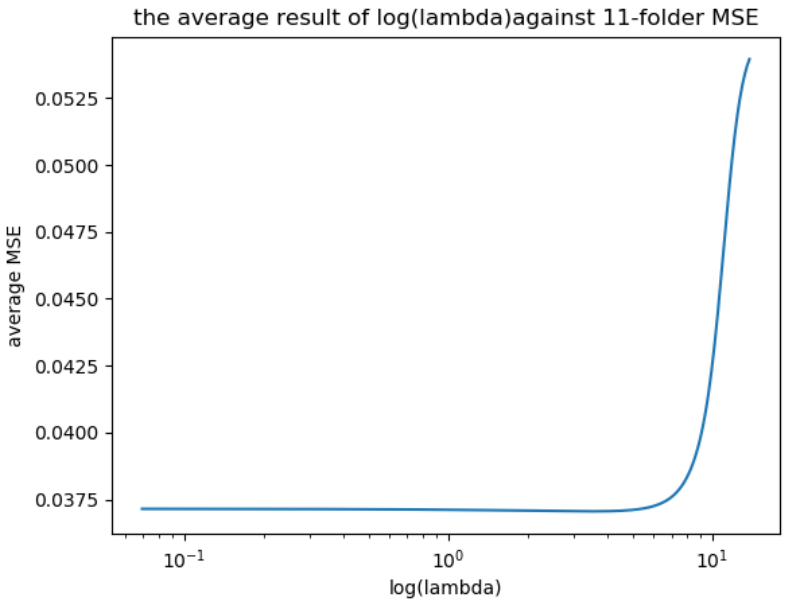

Figure 2: log(lambda) against 11-folder MSE in training data

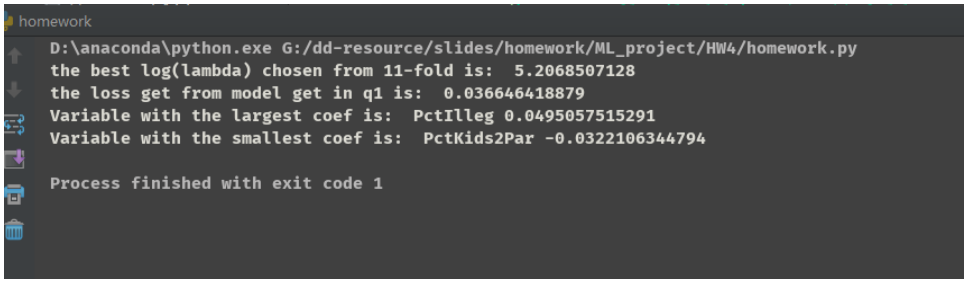

Figure 3: running result

exponentially. Actually, it reveals a phenomenon that, when λ is getting larger, the limitation on the polynomial order is getting larger result in under-fitting. Now considering an extreme example (when $\log(\lambda) \gg 10$ in Fig 5, the number of abs-small coef is nearly to 95), then almost only $\theta_0 \neq 0$, which means the model is a y-fixed line(parallel to the ground), obviously it is a bad and useless model.

So, one way we might select lambda is how well it performs on the test data as this is where we can assess our models accuracy in a real world setting. Though many other factors come into play, increasing lambda will increase the bias to reduce variance which may be something we try to avoid.

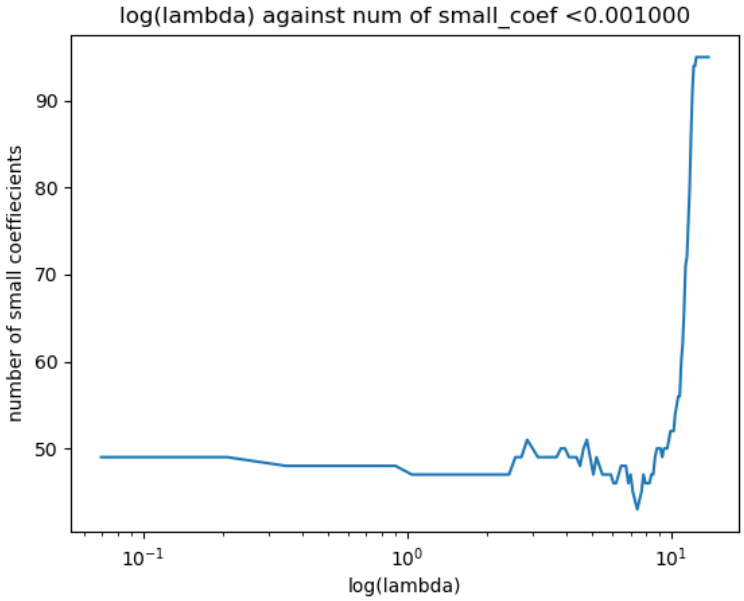

Figure 4: log(lambda) against num of small coef

Another idea all together would be to consider choosing lambda based on principles of cross validation.

(4) From the result above, choose the best performance model: $\log(\lambda) \approx 5.2068507128$, and get:

the most positive variable coefficient: **"PctIlleg"** = 0.0495057515291

the most negative variable coefficient: **"PctKids2Par"** = - 0.0322106344794

which is shown in Fig 3. Which means: The houses with higher 'percentage of kids born to never married' leads to a higher crime rate; And the house with higher 'percentage of kids in family housing with two parents' leads to lower crime rate.

Since the most positive coefficient is the number of children born to unmarried parents and the most negative coefficient is the percentage of kids who have two parents, we can say that parents are often positive roles in childrens lives. It also seems interpretable that having a high percentage of parents to children would benefit the community and decrease crime (as seen by the negative coefficient).

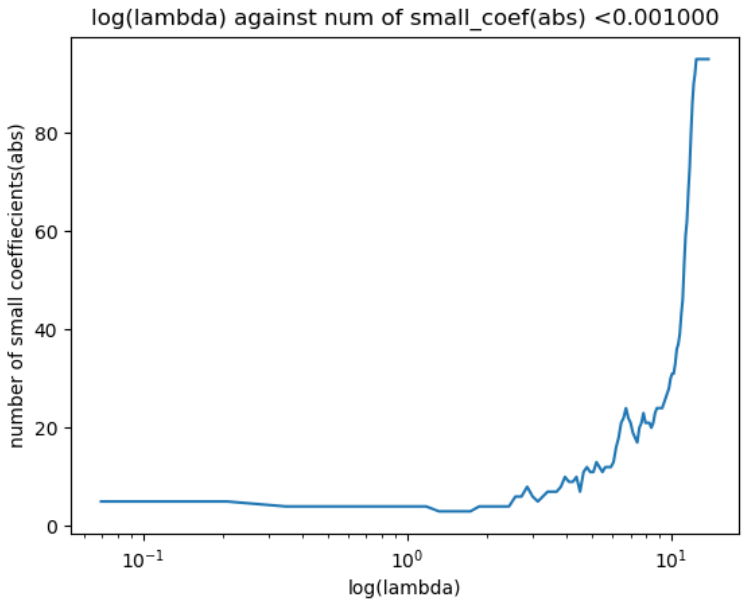

Figure 5: log(lambda) against num of small coef(abs)