Problems

1. LFD Problem 4.4 MATLAB implementation

(a)

$$
\begin{aligned}
E_{a,x}[f^2] =& E_{a,x}[(a_0L_0(x)+a_1L_1(x)+...+a_{Qf}L_{Qf}(x))^2]\\
&\because \int_{-1}^{1} L_k(x)L_l(x)dx = \begin{cases} 0 & l \neq k \\ \frac{2}{2k+1} & l = k \end{cases} \text{ and } E_x(f(x)) = \int_a^b P(x)f(x)dx\\
=& E_{a,x}[f^2] = E_a[\int_{-1}^{1}(a_0L_0(x)+a_1L_1(x)+...+a_{Qf}L_{Qf}(x))^2]\\
=& E_a[\sum_{q=0}^{Qf}\frac{a_q^2}{2q+1}]\\
=& E[a_0^2]+\frac{1}{3}E[a_1^2]+...+\frac{1}{2Q_f+1}E[a_{Qf}^2] \qquad (1)\\
&\because \sigma^2 = 1, \text{ then } E[a_i^2] = \sigma^2 - (E[a_i]^2)^2 = 1 - 0 = 1, \text{ for } i = 1,2,...,Qf\\
&\therefore E[a_0^2]+\frac{1}{3}E[a_1^2]+...+\frac{1}{2Q_f+1}E[a_{Qf}^2]\\
=& 1+\frac{1}{3}+\frac{1}{5}+...+\frac{1}{2Q_f+1} = \sum_{q=0}^{Q_f}\frac{1}{2q+1}\\
=& E_{a,x}[f^2] = \sum_{q=0}^{Q_f}\frac{1}{2q+1}
\end{aligned}
$$

Let the result to 1, the term can be normalized when each a_i, for $i = 1,2,...,Q_f$, is divided by the normalizer $\sqrt{\sum_{q=0}^{Q_f}\frac{1}{2q+1}}$, that is, $\tilde{a} = \frac{a_i}{\sqrt{\sum_{q=0}^{Q_f}\frac{1}{2q+1}}}$

The reason to normalize f is that normalizing $E_{a,x}[f^2] = 1$ let the noise level σ^2 automatically calibrated to the signal level.

(b)

To obtain g_2 and $g_1$0, we transform the original data $x \in X$ with a second order transformation and 10th order transformation. Then, we can find the best linear fit for the data in Z_2 and Z_{10}.

The implementation is as follows,

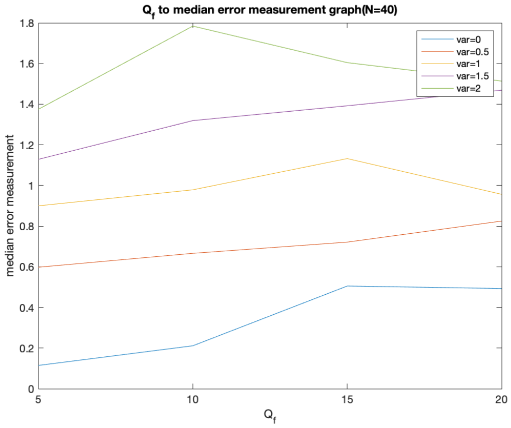
```
z_train_2 = L(x_train,2)
z_train_10 = L(x_train,10)
w2 = glmfit(z_train_2, y_train, 'normal', 'constant', 'off')
w10 = glmfit(z_train_10, y_train, 'normal', 'constant', 'off')
g2 = computeLegPoly(x_test, 2) * w2
g10 = computeLegPoly(x_text, 10) * w10
```

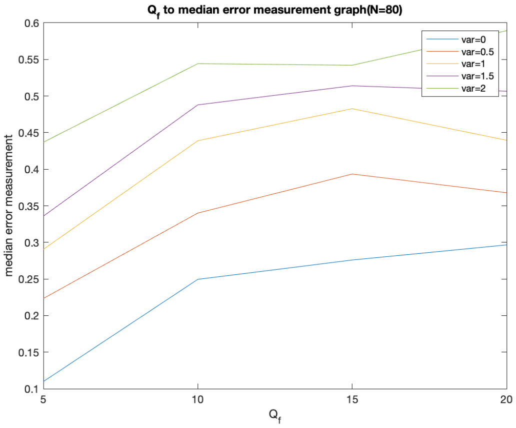
(c)

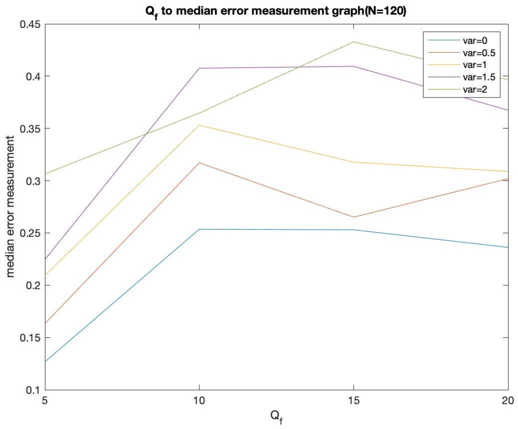
To compute analytically E_{out} for a given g_{10},

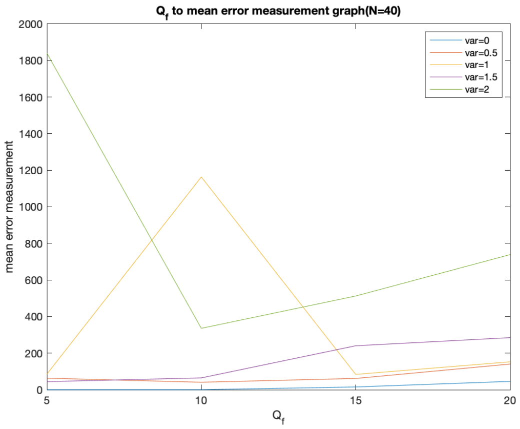
$$
\begin{aligned}
E_{out} =& E_{x,y}[(g_{10}(x) - y)^2] \\
=& E_{x,\epsilon}[(g_{10}(x) - (f(x) + \sigma\epsilon))^2] \\
=& E_{x,\epsilon}[(g_{10}(x) - f(x))^2 - 2(g_{10}(x) - f(x))\sigma\epsilon + (\sigma\epsilon)^2] \\
=& E_{x,\epsilon}[(\sum_{i=0}^{10} w_i L_i(x) - \sum_{j=0}^{Q_f} a_j L_j(x))^2 - 2(\sum_{i=0}^{10} w_i L_i(x) - \sum_{j=0}^{Q_f} a_j L_j(x))\sigma\epsilon + (\sigma\epsilon)^2] \\
\because& E_\epsilon[\epsilon] = 0 \rightarrow E_{x,\epsilon}[-2(\sum_{i=0}^{10} w_i L_i(x) - \sum_{j=0}^{Q_f} a_j L_j(x))\sigma\epsilon] = 0 \\
\because& E_\epsilon[\epsilon^2] = 1 \rightarrow E_{x,\epsilon}[(\sigma\epsilon)^2] = \sigma^2 \\
E_{out} =& E_{x,\epsilon}[(\sum_{i=0}^{10} w_i L_i(x) + \sum_{j=0}^{Q_f} a_j L_j(x))^2 + \sigma^2] \\
=& E_{x,\epsilon}[(\sum_{i=0}^{10} w_i L_i(x))^2 - 2\sum_{j=0}^{min(10,Q_f)} (w_j a_j L_j(x))^2 + \sum_{k=0}^{Q_f}(a_k L_k(x))^2 + \sigma^2] \\
=& E_\epsilon[\sum_{i=0}^{10} \frac{w_i^2}{2i+1} - 2\sum_{j=0}^{min(10,Q_f)} \frac{w_j a_j}{2j+1} + \sum_{k=0}^{Q_f} \frac{a_k^2}{2k+1} + \sigma^2] \\
=& \sum_{i=0}^{10} \frac{w_i^2}{2i+1} - - 2\sum_{j=0}^{min(10,Q_f)} \frac{w_j a_j}{2j+1} + \sum_{k=0}^{Q_f} \frac{a_k^2}{2k+1} + \sigma^2
\end{aligned}
\tag{2}
$$

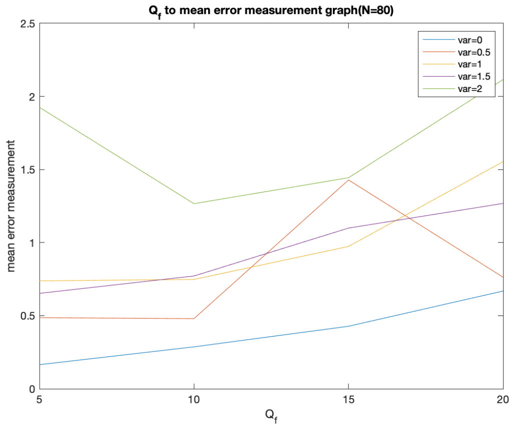
(d)

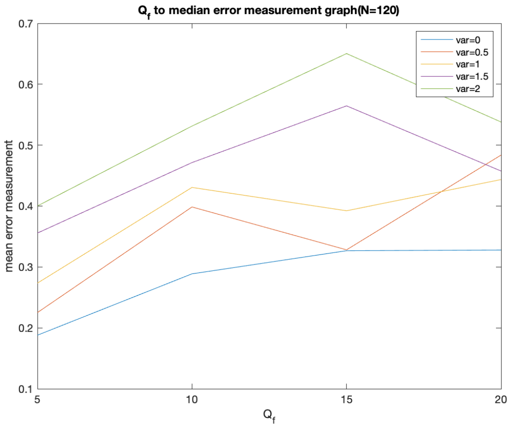


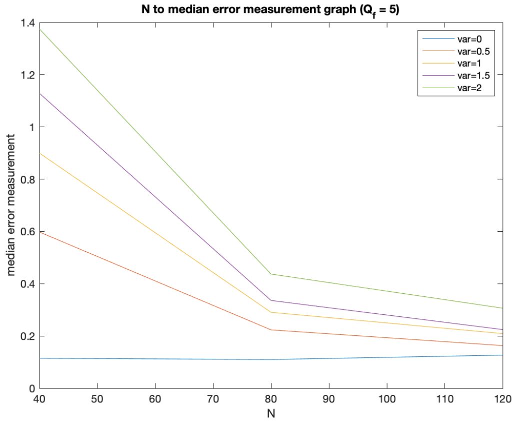


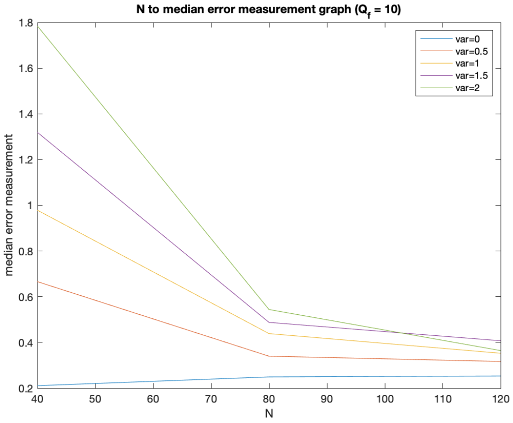

Q_f to median error measurement graph(N=120)
median error measurement
Q_f
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
5
10
15
20
var=0
var=0.5
var=1
var=1.5
var=2

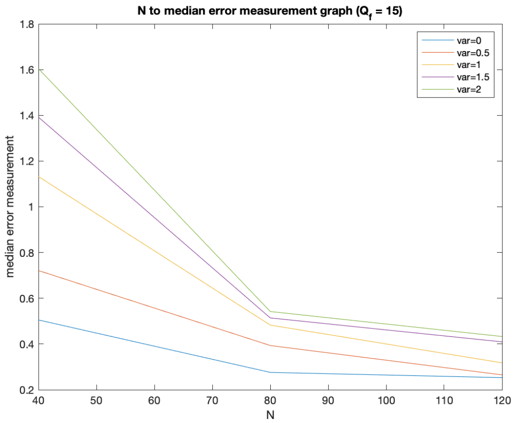

Q_f to mean error measurement graph(N=40)
mean error measurement
Q_f
0
200
400
600
800
1000
1200
1400
1600
1800
2000
5
10
15
20
var=0
var=0.5
var=1
var=1.5
var=2

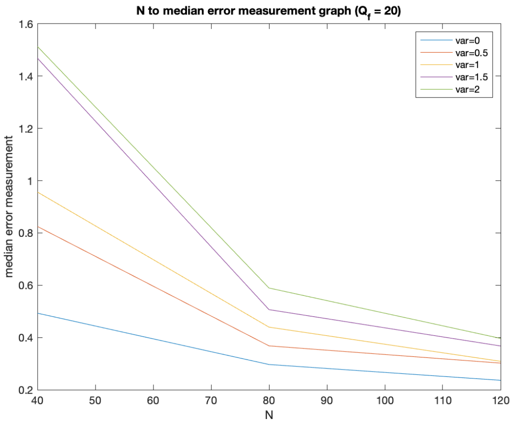

Q_f to mean error measurement graph(N=80)
mean error measurement
Q_f
0
0.5
1
1.5
2
2.5
5
10
15
20
var=0
var=0.5
var=1
var=1.5
var=2


Q_f to median error measurement graph(N=120)
mean error measurement
Q_f
0.1
0.2
0.3
0.4
0.5
0.6
0.7
5
10
15
20
var=0
var=0.5
var=1
var=1.5
var=2


N to median error measurement graph (Q_f = 5)
median error measurement
N
0
0.2
0.4
0.6
0.8
1
1.2
1.4
40
50
60
70
80
90
100
110
120
var=0
var=0.5
var=1
var=1.5
var=2


N to median error measurement graph (Q_f = 10)
median error measurement
N
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
40
50
60
70
80
90
100
110
120
var=0
var=0.5
var=1
var=1.5
var=2





The above are the results based on different size of training set, variance, and Q_f, and using median and mean to measure the out-of sample error.

The explanations are as follows,

First, serious overfitting happens when using mean error measurement, model complexity is large, number of training data is small, and nose coefficient is large.

Especially, error measurements grows positively larger when model complexity and N goes larger.

Given sufficiently large N, say N = 80 or 120, the difference complex model and simple model grows larger when model complexity, Q_f, goes up.

Without stochastic noise, σ, error measurement is larger than those in the same N and Q_f with stochastic noise.

Summing up,

overfitting goes up when stochastic noise, σ, goes up, especially when N is small.

overfitting goes up when model complexity, Q_f, goes up.

overfitting goes down when number of training data, N, goes up.

Comments:

The result using mean error measurement is very volatile as it counts very extreme case into average, for example, extreme positive or negative ϵ or σ.

Using median error measurement can give us more consistent result due to picking median value.

Table 1: Results Table with Different Q_f, N, Variance, and Method of Measurement

Mean error measurement				Median error measurement			
$var = 0$	$N = 40$	$N = 80$	$N = 120$	$var = 0$	$N = 40$	$N = 80$	$N = 120$
$Q_f = 5$	0.1564	0.1649	0.1880	$Q_f = 5$	0.1147	0.1099	0.1267
$Q_f = 10$	0.2369	0.2868	0.2887	$Q_f = 10$	0.2112	0.2495	0.2536
$Q_f = 15$	15.1976	0.4277	0.3267	$Q_f = 15$	0.5052	0.2759	0.2530
$Q_f = 20$	45.6467	0.6688	0.3278	$Q_f = 20$	0.4930	0.2966	0.2362
$var = 0.5$	$N = 40$	$N = 80$	$N = 120$	$var = 0.5$	$N = 40$	$N = 80$	$N = 120$
$Q_f = 5$	63.0766	0.4862	0.2251	$Q_f = 5$	0.5978	0.2234	0.1632
$Q_f = 10$	40.9573	0.4799	0.3985	$Q_f = 10$	0.6661	0.3401	0.3171
$Q_f = 15$	61.6690	1.4284	0.3281	$Q_f = 15$	0.7213	0.3934	0.2651
$Q_f = 20$	140.4305	0.7632	0.4843	$Q_f = 20$	0.8249	0.3679	0.3019
$var = 1$	$N = 40$	$N = 80$	$N = 120$	$var = 1$	$N = 40$	$N = 80$	$N = 120$
$Q_f = 5$	85.6	0.7384	0.2735	$Q_f = 5$	0.8998	0.2905	0.2092
$Q_f = 10$	1163	0.7479	0.4306	$Q_f = 10$	0.9787	0.4389	0.3531
$Q_f = 15$	83.87	0.9739	0.3923	$Q_f = 15$	1.1324	0.4827	0.3177
$Q_f = 20$	152.6	1.554	0.4435	$Q_f = 20$	0.9565	0.4396	0.3088
$var = 1.5$	$N = 40$	$N = 80$	$N = 120$	$var = 1.5$	$N = 40$	$N = 80$	$N = 120$
$Q_f = 5$	44.32	0.6528	0.3557	$Q_f = 5$	1.1287	0.3359	0.2246
$Q_f = 10$	64.73	0.7717	0.4714	$Q_f = 10$	1.3191	0.4879	0.4076
$Q_f = 15$	240.4	1.1	0.5645	$Q_f = 15$	1.3920	0.5140	0.4094
$Q_f = 20$	284.7	1.268	0.4575	$Q_f = 20$	1.4687	0.5066	0.3674
$var = 2$	$N = 40$	$N = 80$	$N = 120$	$var = 2$	$N = 40$	$N = 80$	$N = 120$
$Q_f = 5$	1840	1.925	0.4002	$Q_f = 5$	1.3748	0.4369	0.3064
$Q_f = 10$	335.7	1.266	0.5315	$Q_f = 10$	1.7841	0.5443	0.3647
$Q_f = 15$	512.3	1.444	0.6503	$Q_f = 15$	1.6043	0.5421	0.4328
$Q_f = 20$	738.7	2.116	0.538	$Q_f = 20$	1.5136	0.5893	0.3965

Problems

2. LFD Exercise 4.5

(a)

A softer order constraint is $\sum_{q=0}^{Q} w_q^2 \leq C$, which is stated in the textbook. As the Tikhonov regularizer is a more general constraint regularizer, we can consider it as an identity matrix, that is I and that would give as an equation that is exactly the same as $\sum_{q=0}^{Q} w_q^2 \leq C$ in the textbook.

$$w^T I^T I w = \begin{pmatrix} w_0 & w_1 & w_2 & ... \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & ... \\ 0 & 1 & 0 & ... \\ 0 & 0 & 1 & ... \\ . & . & . & ... \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & ... \\ 0 & 1 & 0 & ... \\ 0 & 0 & 1 & ... \\ . & . & . & ... \end{pmatrix} \begin{pmatrix} w_0 \\ w_1 \\ w_2 \\ . \\ . \\ . \end{pmatrix} = \sum_{q=0}^{Q} w_q^2 \tag{3}$$

(b)

In order to get the $(\sum_{q=0}^{Q} w_q)^2 \leq C$, the translation of the equation is that summing all the element in the weight vector and then take the square of it, the result is required to less than or equal to C. To obtain this result, Tikhonov regularizer should be as follows,

$$w^T \Gamma^T \Gamma w = \begin{pmatrix} w_0 & w_1 & w_2 & ... \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & ... \\ 1 & 0 & 0 & ... \\ 1 & 0 & 0 & ... \\ . & . & . & ... \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & ... \\ 0 & 0 & 0 & ... \\ 0 & 0 & 0 & ... \\ . & . & . & ... \end{pmatrix} \begin{pmatrix} w_0 \\ w_1 \\ w_2 \\ . \\ . \\ . \end{pmatrix} = (\sum_{q=0}^{Q} w_q)^2 \tag{4}$$

In the equation above, Γ should be a matrix with the first element in all columns are all 1, and the result would be $(\sum_{q=0}^{Q} w_q)^2$

Problems

3. LFD Problem 4.8

First, $E_{aug}(w) = E_{in} + \lambda w^T w$, so the gradient is

$$\nabla E_{aug}(w) = \nabla E_{in}(w) + 2\lambda w$$

so the gradient descent update becomes

$$w(t+1) \leftarrow w(t) - \eta \nabla E_{aug}(w(t)) = (1 - 2\eta\lambda)w(t) - \eta \nabla E_{in}(w(t))$$

Problems

4. LFD Problem 4.25

(a)

No, in this cases, since we will get different size of validation set K_m there are no guarantees that we can get the VC-bound we obtained when using the same validation set for all models.

(b)

As explained in the theory, since the validation model H_{val} was obtained before looking at the data in the validation set, the process of model selection is equivalent to learning a hypothesis from H_{val} using the data in D_{val}. In this case, we can apply the VC-bound for finite hypothesis sets.

(c)

We've already known from the Hoeffding inequality and from the part (b) that for each $m = 1, ..., M$,

$$P[E_{out}(m) - E_{val}(m) > \epsilon] \leq e^{-\epsilon^2 K_m}$$

for all ϵ ¿ 0. And using the union bound method, we can get,

$$P[E_{out}(m^*) - E_{val}(m^*) > \epsilon] \leq P[E_{out}(1) - E_{val}(1) > \epsilon] + ... + P[E_{out}(M) - E_{val}(M) > \epsilon] \leq \sum_{m=1}^{M} e^{-\epsilon^2 K_m}$$

Now, we let $k(\epsilon) = -\frac{1}{2\epsilon^2} \ln(\frac{1}{M} \sum_{m=1}^{M} e^{-2\epsilon^2 K_m})$, we can get the following,

$$Me^{-2\epsilon^2 k(\epsilon)} = Me^{\ln(\frac{1}{M} \sum_{m=1}^{M} e^{-2\epsilon^2 K_m})} = \sum_{m=1}^{M} e^{-2\epsilon^2 K_m}$$

Finally we obtain,

$$P[E_{out}(m^*) > E_{val}(m^*) + \epsilon] \leq Me^{-2\epsilon^2 k(\epsilon)}$$

Moreover, we may note that $k(\epsilon) \geq 0$ since $-2\epsilon^2 K_m \leq 0$, this implies that $e^{-2\epsilon^2 K_m} \leq 1$, and so $\frac{1}{M} \sum_{m=1}^{M} e^{-2\epsilon^2 K_m} \leq 1$ and finally $k(\epsilon) \geq 0$

Problems

5. LFD Problem 5.4

(a)

The problem here is that we need $N = 12500$ days (50 years) of data. Although we opted to fix the $M = 500$, there's a data snooping involved in this choice since these 500 stocks were also selected beforehand, by the definition of the S&P 500, which is looking at the whole data set, then selected the largest companies stock to form the portfolio. Moreover, for many of the 50000 stocks we do not have the full 12500 days of data but much less in many circumstances.

As stated above, the correct M should be $M = 50000$, the number of stocks that has been traded over the last 50 years. In this conditions, we get

$$P[E_{in} - E_{out} > 0.02] \leq 2 \cdot 50000 \cdot e^{-2 \cdot 12500 \cdot 0.02^2} \approx 4.53999$$

(b)

As we elaborated in part(a), we cannot conclude with any certainty that buying and holding stocks is a good strategy since we only based on 500 stocks that were selected beforehand. The issue here is called data snooping.

What we would be able to say something about the performance of buy and hold trading is if we considered all 50000 stocks that are currently trading in the market, not just the 500 largest companies selected beforehand by evaluating the company's value.

Collaboration Statement

I didn't collaborate with anyone in this assignment.