CS 189: Introduction to Machine Learning

Fall 2017

Homework 3

Solutions by

Jinhong Du

3033483677

Question 1

(a)

Jinhong Du
jaydu@berkeley.edu

In Homework party, I worked out Question 2(i) with the help of Kaiqian Zhu – tim3212008@berkeley.edu Shengxian Wang – shengxianwang@berkeley.edu
Wediscusstheproblemtogetherandthenwritebyourown.

(b)

I certify that all solutions are entirely in my words and that I have not looked at another student's solutions. I have credited all external sources in this write up. Jinhong Du

Question 2

(a)

$$\because \quad Z \sim N(0,1)$$

$$\therefore \quad X = x, Y_{X=x} = xw + b + Z,\ f_Z(z) = \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}}$$

$$\because \quad xw + b \text{ is constant}$$

$$\therefore \quad f_{Y|X}(y|x) = f_Z(y - xw - b) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(y-xw-b)^2}{2}}, \qquad i.e.,\ Y|X = x \sim N(xw+b, 1)$$

(b)

$\because$ given $X_i = x_i$, $Y_i \sim N(x_i w + b, 1)$

$\therefore$

$$\begin{aligned} L(w, b; x_1, \cdots, x_n, y_1, \cdots, y_n) &= f_{Y_1|X_1,\cdots,Y_n|X_n}(y_1, \cdots, y_n | x_1, \cdots, x_n) \\ &= \prod_{i=1}^{n} f_{Y_i|X_i}(y_i|x_i) \\ &= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}} e^{-\frac{(y_i - x_i w - b)^2}{2}} \\ &= \frac{1}{(2\pi)^{\frac{n}{2}}} e^{-\frac{1}{2}\sum\limits_{i=1}^{n}(y_i - x_i w - b)^2} \\ \ln L &= -\frac{1}{2}\sum_{i=1}^{n}(y_i - x_i w - b)^2 - \frac{n}{2}\ln(2\pi) \end{aligned}$$

Let

$$\begin{cases} \dfrac{\partial \ln L}{\partial w} = \sum\limits_{i=1}^{n} x_i (y_i - x_i w - b) = 0 \\ \dfrac{\partial \ln L}{\partial b} = \sum\limits_{i=1}^{n} (y_i - x_i w - b) = 0 \end{cases}$$

We get

$$\begin{cases} \hat{w} = \dfrac{\overline{XY} - \sum\limits_{i=1}^{n} X_i Y_i}{\overline{X}^2 - \sum\limits_{i=1}^{n} X_i^2} \\ \hat{b} = \overline{Y} - \hat{w}\overline{X} \end{cases}$$

(c)

$$
\begin{aligned}
&\because \quad Z \sim U[-0.5, 0.5] \\
&\therefore \quad X = x, Y_{X=x} = xw + Z,\ f_Z(z) = \mathbb{I}_{[-0.5,0.5]}(z) \\
&\because \quad xw \text{ is constant} \\
&\therefore \quad f_{Y|X}(y|x) = f_Z(y - xw) = \mathbb{I}_{[-0.5,0.5]}(y - xw) = \mathbb{I}_{[-0.5+xw,0.5+xw]}(y), \\
&i.e.,\ Y|X = x \sim U[-0.5 + xw, 0.5 + xw]
\end{aligned}
$$

(d)

$\because$ given $X_i = x_i$, $Y_i \sim U[-0.5 + x_i w, 0.5 + x_i w]$

$\therefore$

$$
\begin{aligned}
L(w; x_1, \cdots, x_n, y_1, \cdots, y_n) &= f_{Y_1|X_1,\cdots,Y_n|X_n}(y_1, \cdots, y_n | x_1, \cdots, x_n) \\
&= \prod_{i=1}^{n} f_{Y_i|X_i}(y_i|x_i) \\
&= \prod_{i=1}^{n} \mathbb{I}_{[-0.5+x_i w,0.5+x_i w]}(y_i) \\
&= \prod_{i=1}^{n} \mathbb{I}_{[-0.5,0.5]}(y_i - x_i w)
\end{aligned}
$$

The values of L may be 0 or 1. Therefore, to maximize L, we should maximize the interval where $L = 1$.

$\because$ when

$$
-0.5 \leqslant \min\{y_i - x_i w\} \leqslant \max\{y_i - x_i w\} \leqslant 0.5
$$

i.e.

$$
\hat{w} \in \{-0.5 \leqslant \min\{Y_i\} - \max\{X_i\}w \leqslant \max\{Y_i\} - \min\{X_i\}w \leqslant 0.5\}
$$

we can get the MLE

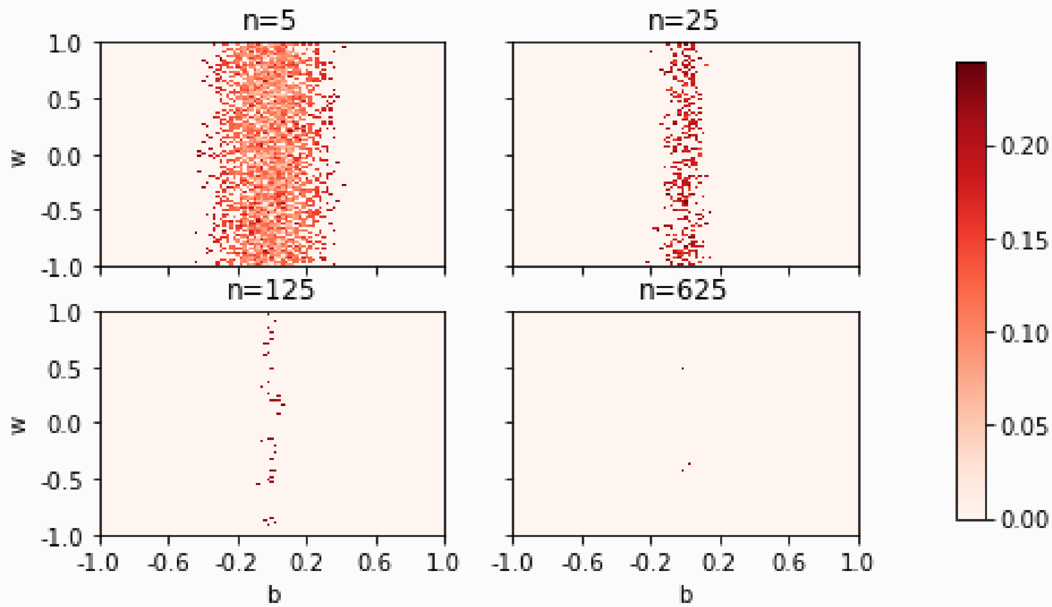
$$
\begin{cases}
\dfrac{\max\{Y_i\} - 0.5}{\min\{X_i\}} \leqslant \hat{w} \leqslant \dfrac{\min\{Y_i\} + 0.5}{\max\{X_i\}}, & \min\{X_i\} > 0 \\
\dfrac{\min\{Y_i\} + 0.5}{\max\{X_i\}} \leqslant \hat{w} \leqslant \dfrac{\max\{Y_i\} - 0.5}{\min\{X_i\}}, & \max\{X_i\} < 0 \\
\max\{\dfrac{\min\{Y_i\} - 0.5}{\max\{X_i\}}, \dfrac{\max\{Y_i\} + 0.5}{\min\{X_i\}}\} \leqslant \hat{w} \leqslant \min\{\dfrac{\max\{Y_i\} - 0.5}{\min\{X_i\}}, & \dfrac{\min\{Y_i\} + 0.5}{\max\{X_i\}}\}, \\
& \min\{X_i\} < 0 < \max\{X_i\} \\
\dfrac{\max\{Y_i\} + 0.5}{\min\{X_i\}} \leqslant \hat{w} \leqslant \dfrac{\max\{Y_i\} - 0.5}{\min\{X_i\}}, & \min\{X_i\} < \max\{X_i\} = 0 \\
\dfrac{\min\{Y_i\} - 0.5}{\max\{X_i\}} \leqslant \hat{w} \leqslant \dfrac{\min\{Y_i\} + 0.5}{\max\{X_i\}}, & 0 = \min\{X_i\} < \max\{X_i\} \\
\varnothing, & \min\{X_i\} = \max\{X_i\} = 0
\end{cases}
$$

(e)

From (d), we know that when $|\hat{Y}(x) - Y(x)| > 0.5$, then the data won't fall in the (w, b) model parameter space. So we set up w and b for some choices, and simulate the different situations.
When a pair of w and b is choosed, we have a true model. Then generate training data and add gaussian

Solution (cont.)

noise to it and fit a predict model. Then determine whether the training data satisfies the inequality.



When n get largely, the range of likelihood becomes very small because if we get more data, we are more likely to have at least one data lie out of the intervals ± 0.5.

(f)

$$
\begin{aligned}
f_W(w) &= \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{w^2}{2\sigma^2}} \\
f(x_1, \cdots, x_n, y_1, \cdots, y_n; w) &= \prod_{i=1}^{n} f(x_1, y_1; w) \\
&= \prod_{i=1}^{n} f_Z(y_i - x_i W; W = w) \\
&= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}} e^{-\frac{(y_i - x_i w)^2}{2}} \\
&= \frac{1}{(2\pi)^{\frac{n}{2}}} e^{-\frac{1}{2}\sum\limits_{i=1}^{n}(y_i - x_i w)^2} \\
f(x_1, \cdots, x_n, y_1, \cdots, y_n; w) f_W(w) &= \frac{1}{\sigma(2\pi)^{\frac{n+1}{2}}} e^{-\frac{1}{2}\sum\limits_{i=1}^{n}(y_i - x_i w)^2 - \frac{w^2}{2\sigma^2}} \\
\int_R f(x_1, \cdots, x_n, y_1, \cdots, y_n; w) f_W(w) \mathrm{d}w &= \frac{e^{-\frac{1}{2}\sum\limits_{i=1}^{n} y_i^2 + \frac{1}{2}\frac{\left(\sum\limits_{i=1}^{n} x_i y_i\right)^2}{\sum\limits_{i=1}^{n} x_i^2 + \frac{1}{\sigma^2}}}}{\sigma(2\pi)^{\frac{n+1}{2}}} \int_R e^{-\frac{1}{2}\left(\sum\limits_{i=1}^{n} x_i^2 + \frac{1}{\sigma^2}\right)\left(w - \frac{\sum\limits_{i=1}^{n} x_i y_i}{\sum\limits_{i=1}^{n} x_i^2 + \frac{1}{\sigma^2}}\right)^2} \mathrm{d}w \\
&= \frac{e^{-\frac{1}{2}\sum\limits_{i=1}^{n} y_i^2 + \frac{1}{2}\frac{\left(\sum\limits_{i=1}^{n} x_i y_i\right)^2}{\sum\limits_{i=1}^{n} x_i^2 + \frac{1}{\sigma^2}}}}{\sigma(2\pi)^{\frac{n}{2}}\sqrt{\sum\limits_{i=1}^{n} x_i^2 + \frac{1}{\sigma^2}}}
\end{aligned}
$$

Solution (cont.)

$$
\begin{aligned}
f(w|x_1,\cdots,x_n,y_1,\cdots,y_n) &= \frac{f(x_1,\cdots,x_n,y_1,\cdots,y_n;w)f_W(w)}{\int_R f(x_1,\cdots,x_n,y_1,\cdots,y_n;w)f_W(w)\mathrm{d}w} \\
&= \frac{\sqrt{\sum\limits_{i=1}^n x_i^2+\frac{1}{\sigma^2}}}{\sqrt{2\pi}} e^{\sum\limits_{i=1}^n x_iy_iw-\frac{1}{2}\sum\limits_{i=1}^n x_i^2w^2-\frac{w^2}{2\sigma^2}-\frac{\left(\sum\limits_{i=1}^n x_iy_i\right)^2}{2\sum\limits_{i=1}^n x_i^2+\frac{2}{\sigma^2}}} \\
&= \frac{\sqrt{\sum\limits_{i=1}^n x_i^2+\frac{1}{\sigma^2}}}{\sqrt{2\pi}} e^{-\frac{1}{2}\left(\sum\limits_{i=1}^n x_i^2+\frac{1}{\sigma^2}\right)(w-\frac{\sum\limits_{i=1}^n x_iy_i}{\sum\limits_{i=1}^n x_i^2+\frac{1}{\sigma^2}})^2}
\end{aligned}
$$

$\therefore$

$$
W|X_1=x_1,\cdots,X_n=x_n,Y_1=y_1,\cdots,Y_n=y_n \sim N\left(\frac{\sum\limits_{i=1}^n x_iy_i}{\sum\limits_{i=1}^n x_i^2+\frac{1}{\sigma^2}}, \frac{1}{\sum\limits_{i=1}^n x_i^2+\frac{1}{\sigma^2}}\right)
$$

$\therefore$

$$
E(W|X_1=x_1,\cdots,X_n=x_n) = \frac{\sum\limits_{i=1}^n x_iy_i}{\sum\limits_{i=1}^n x_i^2+\frac{1}{\sigma^2}}
$$

(g)

MLE

Let

$$
\begin{gathered}
y = \begin{pmatrix} y_1 & y_2 & \cdots & y_n \end{pmatrix}^T \in \mathbb{R}^n \\
x = \begin{pmatrix} x_1^T & x_2^T & \cdots & x_n^T \end{pmatrix}^T \in \mathbb{R}^{n\times d} \\
Y = \begin{pmatrix} Y_1 & Y_2 & \cdots & Y_n \end{pmatrix}^T \in \mathbb{R}^n \\
X = \begin{pmatrix} X_1^T & X_2^T & \cdots & X_n^T \end{pmatrix}^T \in \mathbb{R}^{n\times d} \\
Z = \begin{pmatrix} Z_1 & Z_2 & \cdots & Z_n \end{pmatrix}^T \in \mathbb{R}^n
\end{gathered}
$$

$\because$ given $X_i = x_i \in \mathbb{R}^d$, $Y_i \sim N(w^Tx_i, 1)$

$\therefore$

$$
\begin{aligned}
L(w;x_1,\cdots,x_n,y_1,\cdots,y_n) &= f_{Y_1|X_1,\cdots,Y_n|X_n}(y_1,\cdots,y_n|x_1,\cdots,x_n) \\
&= \prod_{i=1}^n f_{Y_i|X_i}(y_i|x_i) \\
&= \prod_{i=1}^n \frac{1}{\sqrt{2\pi}} e^{-\frac{(y_i-w^Tx_i)^2}{2}} \\
&= \frac{1}{(2\pi)^{\frac{n}{2}}} e^{-\frac{1}{2}(y-xw)^T(y-xw)} \\
\ln L &= -\frac{1}{2}(y-xw)^T(y-xw) - \frac{n}{2}\ln(2\pi)
\end{aligned}
$$

Let

$$
\frac{\partial \ln L}{\partial w} = x^Txw - 2x^Ty = 0
$$

Solution (cont.)

We get

$$\hat{w} = (X^T X)^{-1} X^T Y$$

OLS

$$\begin{aligned} Y &= Xw + Z \\ w_{OLS} &= \arg\min_w \|Y - Xw\|_2^2 \\ &= (X^T X)^{-1} X^T Y \end{aligned}$$

Therefore the maximum likelihood estimator for w is the solution to a least squares problem

(h)

$\because \quad Y_i | X = x_i, W = w \sim N(w^T x_i, 1), W \sim N(0, \sigma^2 I_{d\times d})$

$$\begin{aligned} P(Y_i|x_i, w_i) &= \frac{1}{(2\pi)^{\frac{1}{2}}} e^{-\frac{1}{2}(y_i - w^T x_i)^2} \\ &= \frac{1}{(2\pi)^{\frac{1}{2}}} e^{-\frac{|x_i|^2}{2}(w - y_i x_i)^2} \\ P(Y|x, w) &= \frac{1}{(2\pi)^{\frac{n}{2}}} e^{-\frac{1}{2}\sum\limits_{i=1}^{n}(y_i - w^T x_i)^2} \\ P(w) &= \frac{1}{(2\pi\sigma^2)^{\frac{d}{2}}} e^{-\frac{1}{2\sigma^2} w^T w} \\ \int_w P(Y|X, w)P(w|X)\mathrm{d}w &= \int_w \frac{1}{(2\pi)^{\frac{n+d}{2}}\sigma^d} e^{-\frac{1}{2}\left(\sum\limits_{i=1}^{n}|x_i|^2 + \frac{1}{\sigma^2}\right) w'^T w'} \mathrm{d}w \\ &= \frac{\sqrt{\pi}\left(\sum\limits_{i=1}^{n}|x_i|^2 + \frac{1}{\sigma^2}\right)}{(2\pi)^{\frac{n}{2}}\sigma^d} e^{-\frac{1}{2}\sum\limits_{i=1}^{n} y_i^2 + \frac{1}{2}\frac{\sum_{i=1}^{n} x_i y_i}{\sum_{i=1}^{n} x_i^2 + \frac{1}{\sigma^2}}} \end{aligned}$$

where

$$\begin{aligned} w' &= w - \frac{\sum\limits_{i=1}^{n} y_i x_i}{\sum\limits_{i=1}^{n}|x_i|^2 + \frac{1}{\sigma^2}} \\ P(w|X, Y) &= \frac{P(Y|X, w)P(w|X)}{\int_w P(Y|X, w)P(w|X)\mathrm{d}w} \\ &= \frac{1}{(2\pi)^d \left(\sum\limits_{i=1}^{n}|x_i|^2 + \frac{1}{\sigma^2}\right)} e^{-\frac{\left(\sum_{i=1}^{n}|x_i|^2 + \frac{1}{\sigma^2}\right)}{2} w'^T w'} \end{aligned}$$

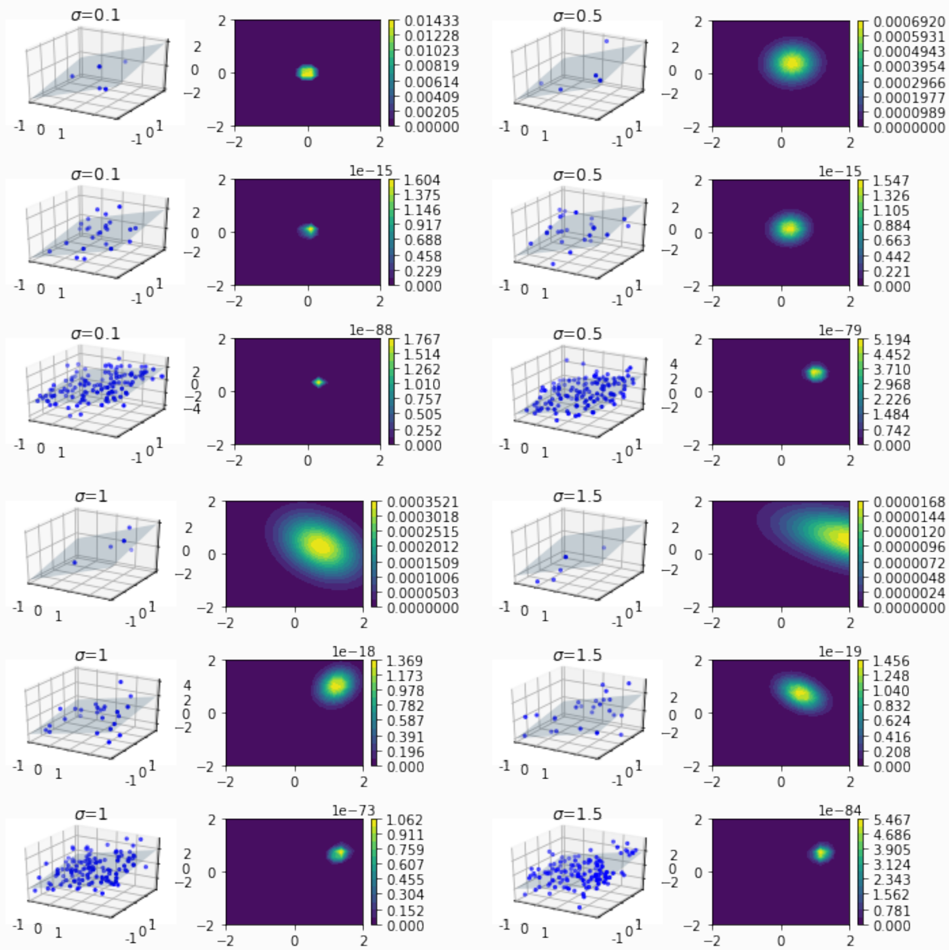
$\therefore \quad W | X_1 = x_1, \cdots, X_n = x_n, Y_1 = y_1, \cdots, Y_n = y_n \sim N\left(\frac{\sum_{i=1}^{n} y_i x_i}{\sum_{i=1}^{n}|x_i|^2 + \frac{1}{\sigma^2}}, \sum\limits_{i=1}^{n}|x_i|^2 + \frac{1}{\sigma^2}\right)$

$\therefore$

$$E(W | X_1 = x_1, \cdots, X_n = x_n, Y_1 = y_1, \cdots, Y_n = y_n) = \frac{\sum\limits_{i=1}^{n} y_i x_i}{\sum\limits_{i=1}^{n}|x_i|^2 + \frac{1}{\sigma^2}}$$

(i)

With small σ or large n, the posteriori probability that samples are close to their mean point will be greater. The more the data are, the bigger ratio that data lie in a certain range of the data center. The small σ also implies that the datas are close to each other.



Question 3

(a)

$\because \quad X_1, \cdots, X_n$ iid

$\therefore \quad EX_1 = \cdots = EX_n = \mu$, $E\hat{X} = \hat{\mu}$

1.

$$
\begin{aligned}
E\left[\frac{X_1 + \cdots + X_n}{n} - \mu\right] &= \frac{1}{n}\sum_{i=1}^{n} EX_i - \mu \\
&= EX_1 - \mu \\
&= 0
\end{aligned}
$$

2.

$$
\begin{aligned}
E\left[\frac{X_1 + \cdots + X_n}{n+1} - \mu\right] &= \frac{1}{n+1}\sum_{i=1}^{n} EX_i - \mu \\
&= \frac{n}{n+1}EX_1 - \mu \\
&= -\frac{1}{n+1}\mu
\end{aligned}
$$

3.

$$
\begin{aligned}
E\left[\frac{X_1 + \cdots + X_n}{n+n_0} - \mu\right] &= \frac{1}{n+n_0}\sum_{i=1}^{n} EX_i - \mu \\
&= \frac{n}{n+n_0}EX_1 - \mu \\
&= -\frac{n_0}{n+n_0}\mu
\end{aligned}
$$

4.

$$
E(0 - \mu) = -\mu
$$

(b)

1.

$$
\begin{aligned}
Var\hat{X} &= Var\left(\frac{X_1 + \cdots + X_n}{n}\right) \\
&= \frac{1}{n^2}\sum_{i=1}^{n} VarX_i \\
&= \frac{\sigma^2}{n}
\end{aligned}
$$

Solution (cont.)

2.

$$\begin{aligned} Var\hat{X} &= Var\left(\frac{X_1+\cdots+X_n}{n+1}\mu\right) \\ &= \frac{1}{(n+1)^2}\sum_{i=1}^{n} VarX_i \\ &= \frac{n\sigma^2}{(n+1)^2} \end{aligned}$$

3.

$$\begin{aligned} Var\hat{X} &= Var\left(\frac{X_1+\cdots+X_n}{n+n_0}\right) \\ &= \frac{1}{(n+n_0)^2}\sum_{i=1}^{n} VarX_i \\ &= \frac{n\sigma^2}{(n+n_0)^2} \end{aligned}$$

4.

$$Var(0) = 0$$

(c)

1.

$$\begin{aligned} MSE &= Var\hat{X} + [E(\hat{X}-\mu)]^2 \\ &= \frac{\sigma^2}{n} + 0 \\ &= \frac{\sigma^2}{n} \end{aligned}$$

2.

$$\begin{aligned} MSE &= Var\hat{X} + [E(\hat{X}-\mu)]^2 \\ &= \frac{n\sigma^2}{(n+1)^2} + \left(\frac{\mu}{n+1}\right)^2 \end{aligned}$$

3.

$$\begin{aligned} MSE &= Var\hat{X} + [E(\hat{X}-\mu)]^2 \\ &= \frac{n\sigma^2}{(n+n_0)^2} + \left(\frac{n_0\mu}{n+n_0}\right)^2 \end{aligned}$$

Solution (cont.)

4.

$$
\begin{aligned}
MSE &= Var\hat{X} + [E(\hat{X} - \mu)]^2 \\
&= 0 + (-\mu)^2 \\
&= \mu^2
\end{aligned}
$$

(d)

By add terms 0, we have

1. $n_0 = 0$
2. $n_0 = 1$
3. $n_0 = n_0$
4. $n_0 = \infty$

(e)

$$
\alpha_{\min} = \arg\min_{\alpha} \left[\frac{n\sigma^2}{(n+\alpha n)^2} + \left(\frac{\alpha n \mu}{n + \alpha n} \right)^2 \right]
$$

$$
\begin{aligned}
f(\alpha) &= \frac{n\sigma^2}{(n+\alpha n)^2} + \left(\frac{\alpha n \mu}{n + \alpha n} \right)^2 \\
&= \frac{\sigma^2 + n\alpha^2\mu^2}{n(1+\alpha)^2} \\
f'(\alpha) &= \frac{2\alpha n \mu^2 (1+\alpha) - 2(\sigma^2 + n\alpha^2\mu^2)}{n(1+\alpha)^3} \\
&= \frac{2(\alpha n \mu^2 - \sigma^2)}{n(1+\alpha)^3} \\
&= 0
\end{aligned}
$$

We get

$$
\alpha = \frac{\sigma^2}{n\mu^2}
$$

$\because \quad f'(\alpha) > 0$ when $\alpha < \dfrac{\sigma^2}{n\mu^2}$, $f'(\alpha) < 0$ when $\alpha > \dfrac{\sigma^2}{n\mu^2}$

$\therefore \quad \alpha_{\min} = \dfrac{\sigma^2}{n\mu^2}$

(f)

$\because \quad \alpha_{\min} = \dfrac{\sigma^2}{n\mu^2} \to \infty$ when $\sigma \to \infty, \mu \to 0$

$\therefore \quad$ when μ is close to 0 and σ is large, $\alpha_{\min}$ will be very large

(g)

Let $X' = X - \mu_0$, we have $EX' = EX - \mu_0 = \mu - \mu_0 \approx 0$

(h)

When α increases, the bias will decrease and the variance will increase.
In ridge regression, when λ increases, the bias may funcuate first and always goes up at the end. However, the variance will decrease when λ increases.
Therefore, α and λ seem to have same function to the regression problem.
We should choose λ such that it can best minimize the sum of bias and variance instead of only the bias or only the variance.

Question 4

(a)

OLS

$$\begin{aligned}\hat{X} &= \arg\min_x \|Y - Ax\|_2^2 \\ &= (A^TA)^{-1}A^TY \\ y^* &= Ax^* \\ Y &= Ax^* + W\end{aligned}$$

$\therefore$

$$\begin{aligned}\|A\hat{X} - y^*\|_2^2 &= \|A(A^TA)^{-1}A^TY - Ax^*\|_2^2 \\ &= \|A(A^TA)^{-1}A^T(Ax^* + W) - Ax^*\|_2^2 \\ &= \|A(A^TA)^{-1}(A^TA)x^* + A(A^TA)^{-1}A^TW - Ax^*\|_2^2 \\ &= \|Ax^* + A(A^TA)^{-1}A^TW - Ax^*\|_2^2 \\ &= \|A(A^TA)^{-1}A^TW\|_2^2\end{aligned}$$

(b)

From (a), we have

$$\begin{aligned}\|A\hat{X} - y^*\|_2^2 &= \|A(A^TA)^{-1}A^TW\|_2^2 \\ &= (A(A^TA)^{-1}A^TW)^TA(A^TA)^{-1}A^TW \\ &= W^TA[(A^TA)^{-1}]^T(A^TA)(A^TA)^{-1}A^TW \\ &= W^TA[(A^TA)^{-1}]^TA^TW \\ &= W^TU\Sigma V^T[(V\Sigma^TU^TU\Sigma V^T)^{-1}]^TV\Sigma U^TW \\ &= W^TU\Sigma V^T[(V\Sigma^T\Sigma V^T)^{-1}]^TV\Sigma U^TW \\ &= W^TU\Sigma V^T(V\Sigma^T)^{-T}(\Sigma V^T)^{-T}V\Sigma U^TW \\ &= W^TU\Sigma V^T(\Sigma V^T)^{-1}(V\Sigma)^{-1}V\Sigma U^TW \\ &= W^TUU^TW \\ &= \|U^TW\|_2^2\end{aligned}$$

(c)

Proof.

Solution (cont.)

Let $U = \begin{pmatrix} u_1 & u_2 & \cdots & u_d \end{pmatrix}$ where $u_i \in \mathbb{R}^n$, $\|u_i\|_2^2 = 1$

$$\begin{aligned}
\frac{1}{n}E\left[\|A\hat{X} - y^*\|_2^2\right] &= \frac{1}{n}E[\|U^T W\|_2^2] \\
&= \frac{1}{n}E\left[\left\| \begin{pmatrix} u_1^T \\ u_2^T \\ \vdots \\ u_d^T \end{pmatrix}_{d\times n} W \right\|_2^2\right] \\
&= \frac{1}{n}E\left[\left\| \begin{pmatrix} u_1^T W \\ u_2^T W \\ \vdots \\ u_d^T W \end{pmatrix} \right\|_2^2\right] \\
&= \frac{1}{n}E\left[\sum_{i=1}^{d}(u_i^T W)^2\right] \\
&= \frac{\sigma^2}{n}\sum_{i=1}^{d}E(u_i^T W_0)^2 \\
&= \frac{\sigma^2}{n}\sum_{i=1}^{d}\{Var(u_i^T W) + [E(u_i^T W)]^2\} \\
&= \frac{\sigma^2}{n}\sum_{i=1}^{d}\|u_i\|_2^2 \\
&= \frac{\sigma^2}{n}d
\end{aligned}$$

□

(d)

$$A = \begin{bmatrix} x_1^0 & x_1^1 & \cdots & x_1^d \\ x_2^0 & x_2^1 & \cdots & x_2^d \\ \vdots & \vdots & \vdots & \vdots \\ x_n^0 & x_n^1 & \cdots & x_n^d \end{bmatrix}$$

$$x^* = \begin{pmatrix} m_0 & m_1 & \cdots & m_d \end{pmatrix}^T$$

Therefore $y^* = Ax^* + W$. From (c) we have:
When $d = D + 1$, average squared error is

$$\frac{\sigma^2(D+1)}{n} < \epsilon$$

We have

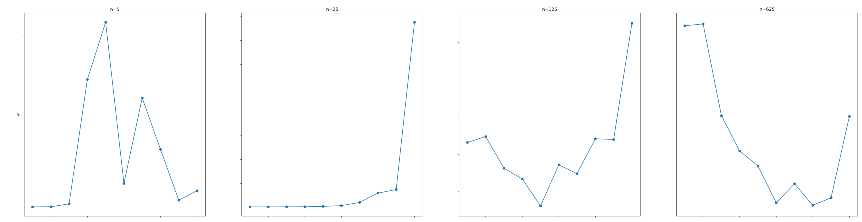
$$n > \frac{\sigma^2(D+1)}{\epsilon}$$

Therefore, when we increase model complexity, i.e. D, to bound the average square error we have to increase number of samples n

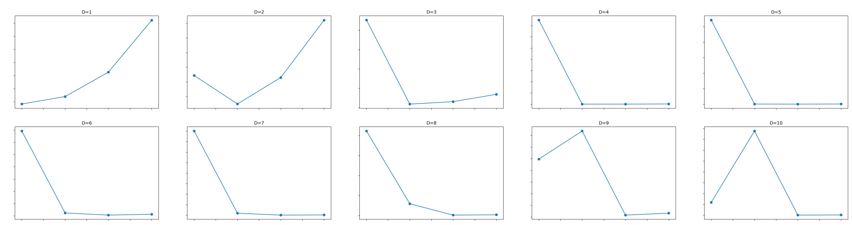
(e)

Given $n = \{5, 25, 125, 625\}$ and $D = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$

When D increase, the training error may first increase but decrease finally:



When n increase, the traning error may first increase but decrease finally:



Question 5

(a)

$$\begin{aligned}\hat{\pi} &= \min_x \|X\pi - U\|_F^2 \\ &= \min_x \|\pi^T X - U^T\|_F^2\end{aligned}$$

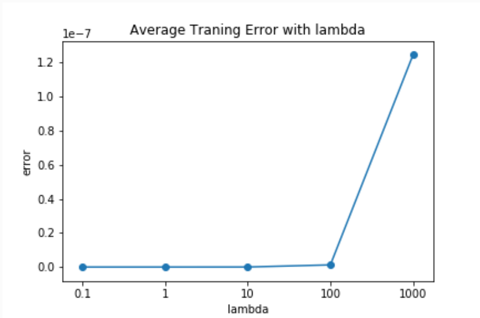
From HW1 5 (d), the solution is

$$\begin{aligned}\hat{\pi}^T &= U^T X (X^T X)^{-1} \\ \hat{\pi}^T X^T X &= U^T X \\ X^T X \hat{\pi} &= X^T U\end{aligned}$$

By using $np.linalg.solve()$, I get **LinAlgError: Singular Matrix**.
It is because that X is a 91×2700 matrix, i.e. $rank(X) \leqslant 91$, and therefore $rank(X^T X) \leqslant 91$. However $X^T X$ is a 2700×2700 matrix. So $X^T X$ is sigular matrix.

(b)

The code is attached at the end.



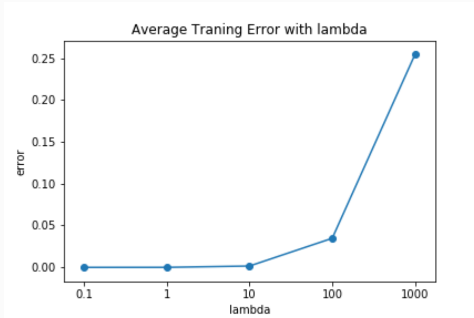
When $\lambda = 0.1$ we have the least training error.

(c)

λ	training error
0.1	3.2557474989148846e-07
1	2.9105122907682248e-05
10	0.0015903814573038761
100	0.034773122042375752
1000	0.25440296146797026

The code is attached at the end.

Solution (cont.)



Again $\lambda = 0.1$ minimize the training error.

Standardizing the states imporving the training process. The large traning error turns to be very small.

(d)

Without standardization,

$$k_1 = 52711693.3276$$

With standardization,

$$k_2 = 444.725931711$$

Standardization improves the loss function by reducing the magnitude of the eigenvalues. All states are now scaled between -1 to 1 and the loss also become smaller.

(e)

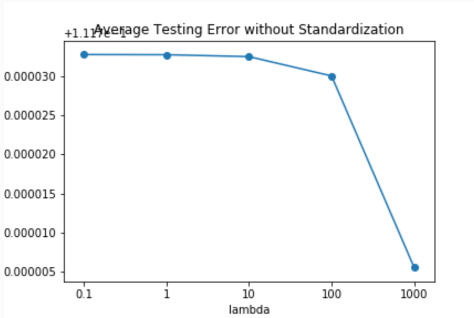
Loss without standardization

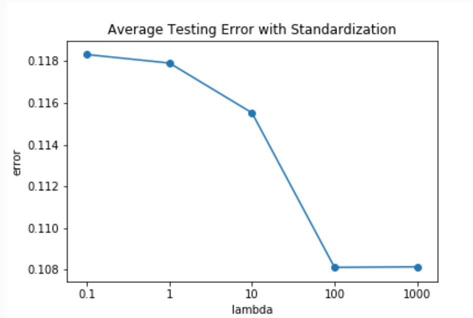
λ	testing error
0.1	0.11173275797988971
1	0.1117327356116609
10	0.11173248999880003
100	0.11173002786555813
1000	0.1117055646108573

Loss with standardization

λ	testing error
0.1	0.1183268483625583
1	0.11791898072707883
10	0.11552877144355782
100	0.10811083659462012
1000	0.10813777058742725

Solution (cont.)





When $\lambda = 100$ and with standardization, the loss of test data is smallest.
In ridge regression, when λ increases, the bias may funcuate first and always goes up at the end. However, the variance will decrease because it restrict the changes of π.
Therefore, we should choos λ such that it can best minimize the sum of bias and variance instead of only the bias or only the variance.

Question 6

QuestionLet X and Y be independent random variables taking values in $\mathbb{N}$, such that

$$\mathbb{P}(X=k|X+Y=n)=\binom{n}{k}p^k(1-p)^{n-k}$$

for some $0<p<1$ and all $0\leqslant k\leqslant n$. Show that X and Y have Poisson distribution.
(I choose this problem because the common question is given 2 independent poisson random variables to prove the conditional probability equation above. However, the contraty is true too.)
Solution

Proof.

$\because$

$$\begin{aligned}P(X=k|X+Y=n)&=\frac{P(X=k,X+Y=n)}{P(X+Y=n)}\\&=\frac{P(X=k)P(Y=n-k)}{P(X+Y=n)}\end{aligned}$$

$\therefore$

$$\begin{aligned}\frac{P(X=k+1|X+Y=n)}{P(X=k|X+Y=n)}&=\frac{P(X=k+1)P(Y=n-k-1)}{P(X=k)P(Y=n-k)}\\&=\frac{\binom{n}{k+1}p^{k+1}(1-p)^{n-k-1}}{\binom{n}{k}p^k(1-p)^{n-k}}\\&=\frac{n-k}{k+1}\frac{p}{1-p}\end{aligned}$$

$\therefore$

$$\begin{aligned}\frac{P(X=k|X+Y=n-1)}{P(X=k-1|X+Y=n-1)}&=\frac{P(X=k)P(Y=n-k-1)}{P(X=k-1)P(Y=n-k)}\\&=\frac{\binom{n-1}{k}p^k(1-p)^{n-k-1}}{\binom{n-1}{k-1}p^{k-1}(1-p)^{n-k}}\\&=\frac{n-k}{k}\frac{p}{1-p}\end{aligned}$$

$\therefore$

$$\begin{aligned}\frac{\frac{P(X=k+1)P(Y=n-k-1)}{P(X=k)P(Y=n-k)}}{\frac{P(X=k)P(Y=n-k-1)}{P(X=k-1)P(Y=n-k)}}&=\frac{k}{k+1}\\(k+1)\frac{P(X=k+1)}{P(X=k)}&=k\frac{P(X=k)}{P(X=k-1)}\\&=\frac{P(X=1)}{P(X=0)}\end{aligned}$$

Solution (cont.)

Let $\dfrac{P(X=1)}{P(X=0)} = a$, then $\forall\ k \in \mathbb{N}$

$$\begin{aligned} P(X=k+1) &= \frac{a}{k+1} P(X=k) \\ &= \cdots \\ &= \frac{a^{k+1}}{(k+1)!} P(X=0) \end{aligned}$$

$\because$

$$\sum_{k=0}^{\infty} P(X=k) = 1$$

$\therefore$

$$\begin{aligned} \sum_{k=0}^{\infty} \frac{a^k}{k!} P(X=0) &= e^a P(X=0) \\ &= 1 \end{aligned}$$

$\therefore$

$$P(X=0) = e^{-a}$$

$\therefore \quad \forall\ k \in \mathbb{N}$,

$$P(X=k) = \frac{a^k}{k!} e^{-a}$$

i.e. $X \sim P(a)$

$\because$

$$\frac{P(X=k)P(Y=n-k-1)}{P(X=k-1)P(Y=n-k)} = \frac{n-k}{k+1} \frac{p}{1-p}$$

$\therefore$

$$\begin{aligned} \frac{P(Y=n-k-1)}{P(Y=n-k)} &= \frac{\frac{a^{k-1}}{(k-1)!} e^{-a}}{\frac{a^k}{k!} e^{-a}} \frac{n-k}{k+1} \frac{p}{1-p} \\ &= \frac{k}{a} \frac{n-k}{k+1} \frac{p}{1-p} \end{aligned}$$

i.e.

$$\begin{aligned} P(Y=k) &\xlongequal{n=2k} \frac{a(k+1)}{k^2} \frac{1-p}{p} P(Y=k-1) \\ &\cdots \\ &= \frac{a^k(k+1)}{k!} \left(\frac{1-p}{p}\right)^k P(Y=0) \end{aligned}$$

$\therefore \quad$ by

$$\sum_{k=0}^{\infty} P(Y=k) = 1$$

we have $P(Y=0) = e^{-a\frac{1-p}{p}}$, $P(Y=k) = \dfrac{\left(a\frac{1-p}{p}\right)^k}{k!} e^{-2a\frac{1-p}{p}}$, i.e. $Y \sim P\left(a\frac{1-p}{p}\right)$ $\square$

HW3

September 15, 2017

1 Qestion 1

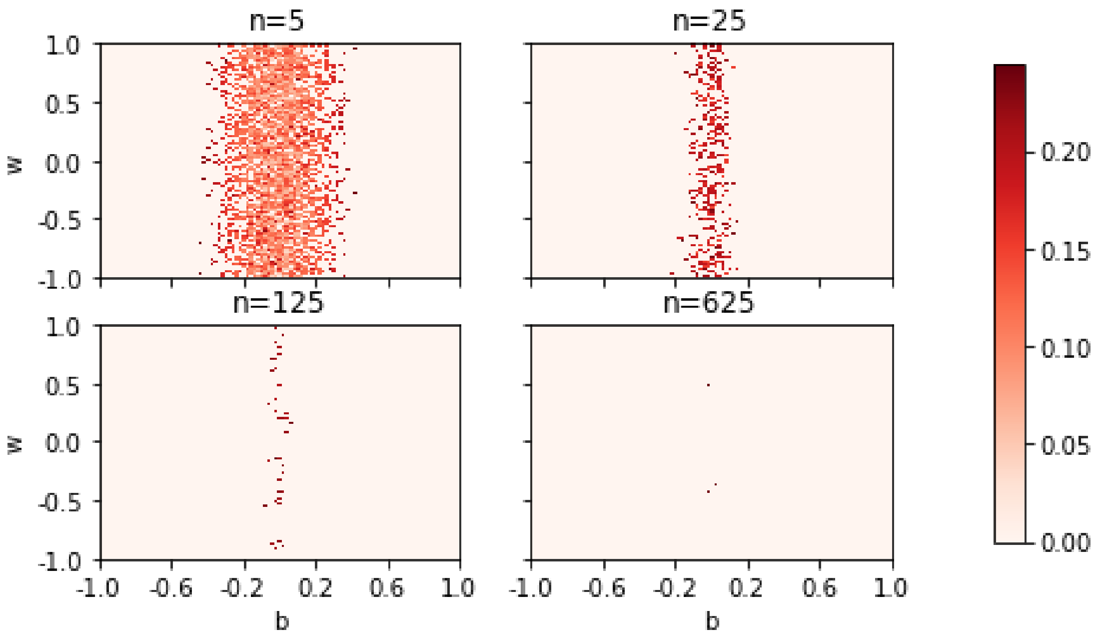
(e)

```
In [4]: import numpy as np
        import random
        import matplotlib.pyplot as plt
        %matplotlib inline
```

Generate data

```
In [51]: W = np.linspace(-1,1,100,endpoint=True)
         B = np.linspace(-1,1,100,endpoint=True)

         N = [5, 25, 125, 625]
         fig = plt.figure()

         for n in range(len(N)):
             ax = fig.add_subplot(2,2,n+1)
             S = np.zeros((len(W),len(W)))
             for w in range(len(W)):
                 for b in range(len(B)):
                     Z = [random.uniform(-0.5,0.5) for _ in range(N[n])]
                     x = np.linspace(-1,1,N[n],endpoint=True)
                     y = W[w]*x+B[b]+Z
                     A = np.vstack([x, np.ones(len(x))]).T
                     what = np.linalg.lstsq(A,y-B[b])[0]
                     if np.all(np.abs(A.dot(what.T)-y)<=0.5):
                         S[w][b] = np.mean(np.abs(A.dot(what.T)-y))
             ax.set_title('n=%s'%N[n])
             if n==0 or n==2:
                 ax.set_yticklabels(np.linspace(-1,1,5,endpoint=True))
                 ax.set_ylabel('w')
             else:
                 ax.set_yticklabels([])
             if n==2 or n==3:
                 ax.set_xticklabels(np.linspace(-1,1,6,endpoint=True))
                 ax.set_xlabel('b')
```

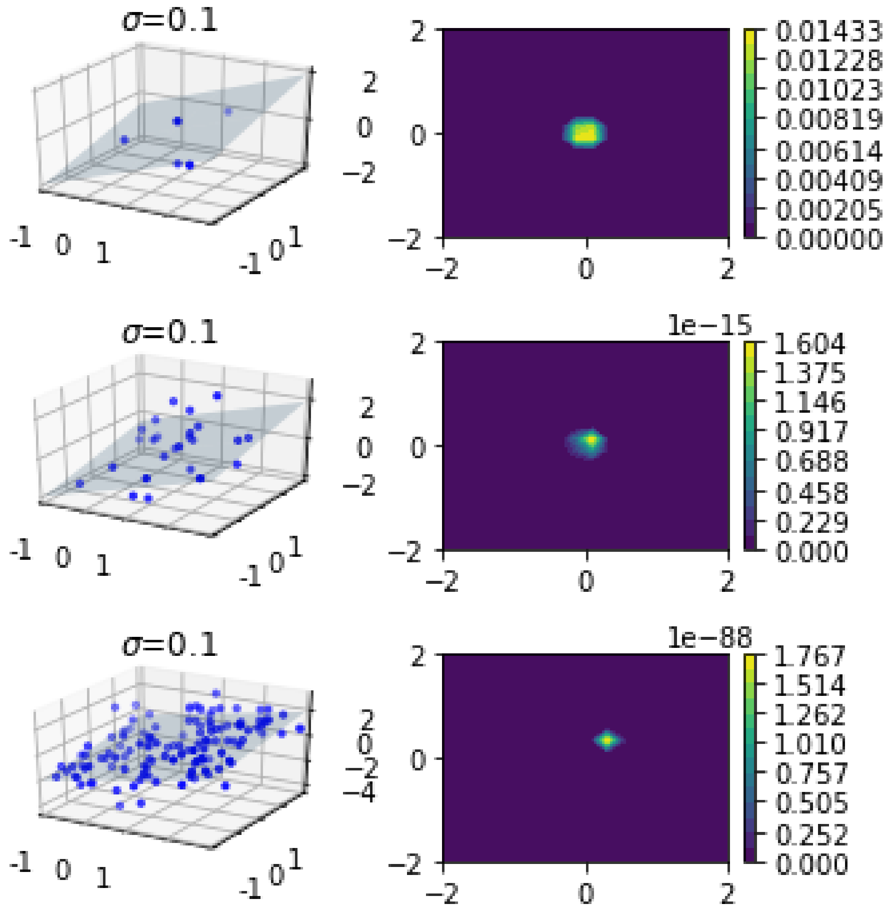
```
        else:
            ax.set_xticklabels([])

        plt.pcolor(S,cmap=plt.cm.Reds)
    cbar_ax = fig.add_axes([1.0, 0.15, 0.03, 0.7])
    plt.colorbar(cax=cbar_ax)
    plt.savefig('1e.png')
    plt.show()
    plt.close()
```

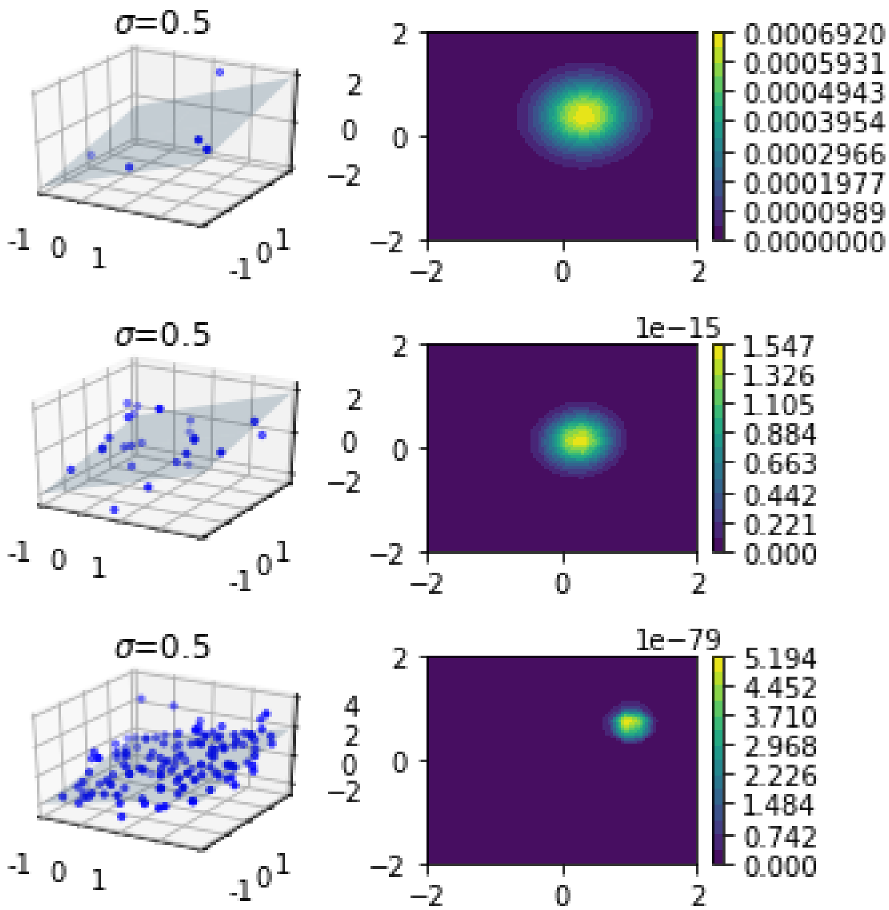


2 Question 2

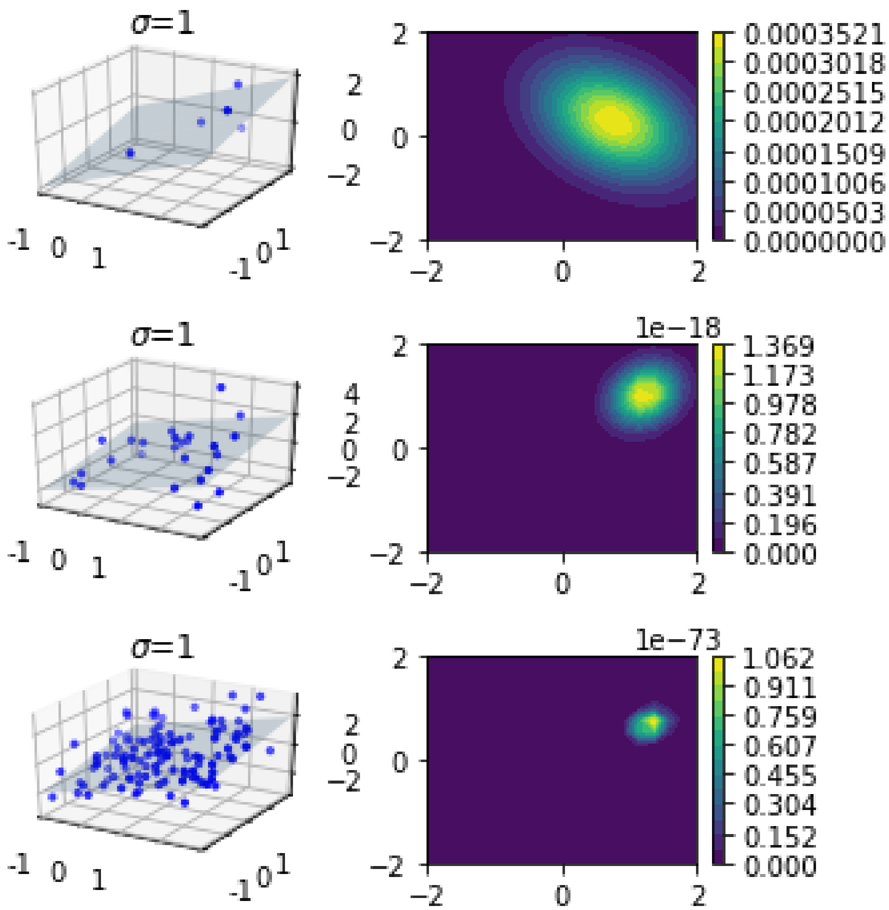
In [155]:
```
import matplotlib.ticker as ticker
from scipy.stats import norm
from mpl_toolkits.mplot3d import Axes3D
Xrange = [-1,1]
Nrange = [5, 25, 125]
STDrange = [0.1, 0.5, 1, 1.5]

wreal = np.asmatrix([1.164523, 0.733131]).T
for windex, wstd in enumerate(STDrange):
    fig = plt.figure(figsize=(5, 5))
    for index, N in enumerate(Nrange):
        Z = np.random.normal(0, 1, (N, 1))
        axs = fig.add_subplot(len(Nrange), 2, index*2+ 1, projection='3d')
```

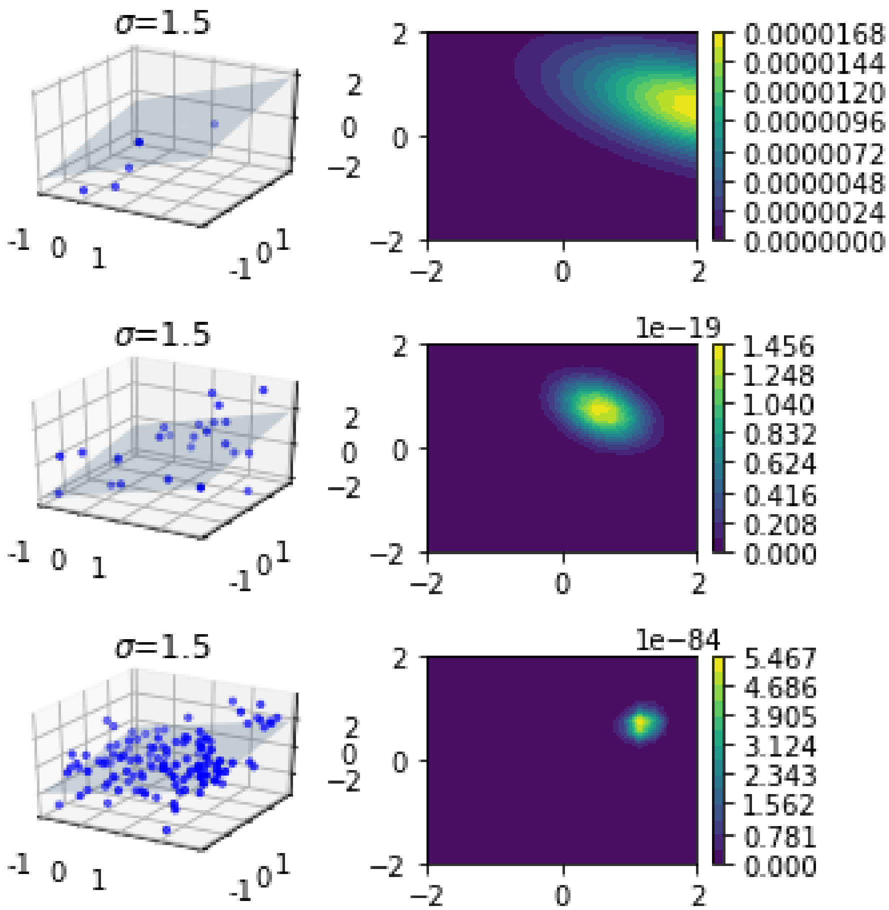
```
        axs.set_title('$\sigma$=%s'%wstd)
        # plot real weight w (the plane)
        x1, x2 = np.meshgrid(range(-1,2,1), range(-1,2,1))
        y=wreal.item(0) * x1 + wreal.item(1) * x2
        axs.plot_surface(x1, x2, y, alpha=0.2)

        # plot the training data
        X = (Xrange[1]-Xrange[0]) * np.random.random_sample((2, N)) + Xrange[0]
        Y = X.T.dot(wreal) + Z
        axs.scatter(X[0], X[1], np.asarray(Y), s=5, marker='o',  color='b')
        axs.set_xticklabels(range(-1,2,1))
        axs.set_yticklabels(range(-1,2,1))
        axs.set_xlim([-1,1])
        axs.set_ylim([-1,1])

        # Contour Diagram
        axs = fig.add_subplot(len(Nrange), 2, index * 2 + 2)
        w1, w2 = np.meshgrid(np.linspace(-2,2, 20), np.linspace(-2,2, 20))
        P = np.ones(w1.shape)
        for i in range(0, P.shape[0]):
            for j in range(0, P.shape[1]):
                w = np.asmatrix([w1.item(i,j), w2.item(i,j)]).T
                P[i, j] *= np.prod(np.vectorize(norm.pdf)(Y-X.T.dot(w)))
                P[i, j] *= np.prod(np.vectorize(norm.pdf)(w / wstd) / wstd)
        levels = np.linspace(0, np.max(P), 15)
        cs = axs.contourf(w1, w2, P, levels=levels)
        fig.colorbar(cs, ax=axs, boundaries=levels)
    plt.tight_layout()
plt.savefig('2i.png',dpi=300)
plt.show()
```


σ=0.1
0.01433
0.01228
0.01023
0.00819
0.00614
0.00409
0.00205
0.00000
σ=0.1
1e−15
1.604
1.375
1.146
0.917
0.688
0.458
0.229
0.000
σ=0.1
1e−88
1.767
1.514
1.262
1.010
0.757
0.505
0.252
0.000



σ=0.5
σ=0.5
σ=0.5
0.0006920
0.0005931
0.0004943
0.0003954
0.0002966
0.0001977
0.0000989
0.0000000
1e−15
1.547
1.326
1.105
0.884
0.663
0.442
0.221
0.000
1e−79
5.194
4.452
3.710
2.968
2.226
1.484
0.742
0.000


σ=1
σ=1
σ=1
0.0003521
0.0003018
0.0002515
0.0002012
0.0001509
0.0001006
0.0000503
0.0000000
1e−18
1.369
1.173
0.978
0.782
0.587
0.391
0.196
0.000
1e−73
1.062
0.911
0.759
0.607
0.455
0.304
0.152
0.000

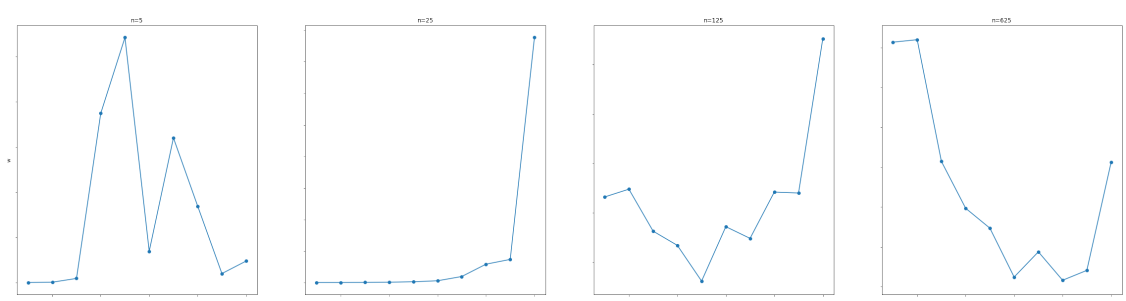


3 Qustion 4

```
In [ ]: m = 1
        c = 1
        N = [5, 25, 125, 625]
        D = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
        error = np.zeros((len(N),len(D)))
        a = 200
        for n in range(len(N)):
            for d in range(len(D)):
                for _ in range(a):
                    x = np.random.uniform(-1,1,(N[n],))
                    w = np.random.normal(-1,1,(N[n],))
                    A = np.polyfit(x+w, x+1, D[d])
                    error[n][d] += np.linalg.norm(np.polyval(A,x)-np.polyval(A,np.ones(N[n])))
                error[n][d] /= a

In [115]: error
```

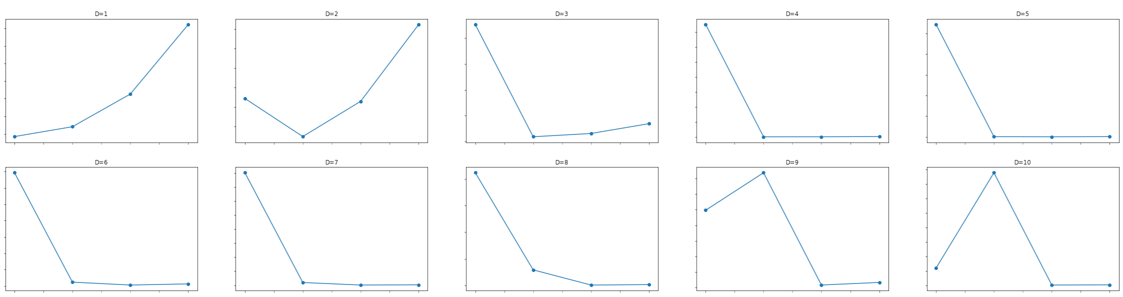
```
Out[115]: array([[  8.46187235e-01,   3.44158664e+00,   4.52792894e+01,
                    1.87368281e+03,   2.71390508e+03,   3.46387278e+02,
                    1.60186221e+03,   8.46031620e+02,   9.93409260e+01,
                    2.38107288e+02],
                 [  1.40662765e+00,   1.49218925e+00,   1.85801001e+00,
                    2.91866370e+00,   5.75070514e+00,   1.19692916e+01,
                    3.90536724e+01,   1.17156507e+02,   1.47474794e+02,
                    1.55330884e+03],
                 [  3.26401697e+00,   3.29549316e+00,   3.12561751e+00,
                    3.06812305e+00,   2.92367558e+00,   3.14379162e+00,
                    3.09646964e+00,   3.28378647e+00,   3.27991723e+00,
                    3.90347227e+00],
                 [  7.21381197e+00,   7.21998193e+00,   6.91426523e+00,
                    6.79669982e+00,   6.74675778e+00,   6.62404245e+00,
                    6.68740861e+00,   6.61600031e+00,   6.64081335e+00,
                    6.91187029e+00]])
```

```
In [130]: fig = plt.figure(figsize=(40,10))
          for n in range(len(N)):
              ax1 = fig.add_subplot(1,len(N),n+1)
              if n==0:
                  ax1.set_yticklabels(np.linspace(0,10,1,endpoint=True))
                  ax1.set_ylabel('w')
              else:
                  ax1.set_yticklabels([])
              ax1.scatter(np.arange(1,len(D)+1),error[n])
              ax1.set_xticklabels(np.linspace(1,5,1,endpoint=True))
              ax1.set_title('n=%s'%N[n])
              plt.plot(np.arange(1,len(D)+1),error[n])
          plt.savefig('4e1.png',dpi=300)
          plt.show()
```




```
In [129]: fig = plt.figure(figsize=(40,10))
          for d in range(len(D)):
              ax2 = fig.add_subplot(2,len(D)/2,d+1)
              if n==0:
```

```
                ax2.set_yticklabels(np.linspace(0,10,1,endpoint=True))
                ax2.set_ylabel('w')
            else:
                ax2.set_yticklabels([])
            ax2.scatter(np.arange(1,len(N)+1),error.T[d])
            ax2.set_xticklabels(np.linspace(1,5,1,endpoint=True))
            ax2.set_title('D=%s'%D[d])
            plt.plot(np.arange(1,len(N)+1),error.T[d])
        plt.savefig('4e2.png',dpi=300)
        plt.show()
```



4 Qustion 5

```
In [179]: import pickle
          with open('./data/x_train.p','rb') as f:
              x_train = pickle.load(f,encoding='latin1')
          with open('./data/y_train.p','rb') as f:
              y_train = pickle.load(f,encoding='latin1')
```

```
In [180]: print(np.shape(x_train[0]))
          print(len(x_train))
          N = np.size(x_train[0])
          print(N)
```

```
(30, 30, 3)
91
2700
```

```
In [185]: X = np.reshape(np.array(x_train,dtype=np.float64),(len(x_train),2700),order='F')
          print(np.shape(X))
```

```
(91, 2700)
```

```
In [220]: U = np.array(y_train,dtype=np.float64)
          print(np.shape(U))
```

```
(91, 3)
```

```
In [221]: pi = np.linalg.solve(X.T.dot(X),X.T.dot(U))
```

```
        ---------------------------------------------------------------------------

        LinAlgError                               Traceback (most recent call last)

        <ipython-input-221-6ea30d3a882a> in <module>()
    ----> 1 pi = np.linalg.solve(X.T.dot(X),X.T.dot(U))


        /anaconda/lib/python3.6/site-packages/numpy/linalg/linalg.py in solve(a, b)
        373     signature = 'DD->D' if isComplexType(t) else 'dd->d'
        374     extobj = get_linalg_error_extobj(_raise_linalgerror_singular)
    --> 375     r = gufunc(a, b, signature=signature, extobj=extobj)
        376
        377     return wrap(r.astype(result_t, copy=False))


        /anaconda/lib/python3.6/site-packages/numpy/linalg/linalg.py in _raise_linalgerror_singu
         88
         89 def _raise_linalgerror_singular(err, flag):
    ---> 90     raise LinAlgError("Singular matrix")
         91
         92 def _raise_linalgerror_nonposdef(err, flag):


        LinAlgError: Singular matrix
```

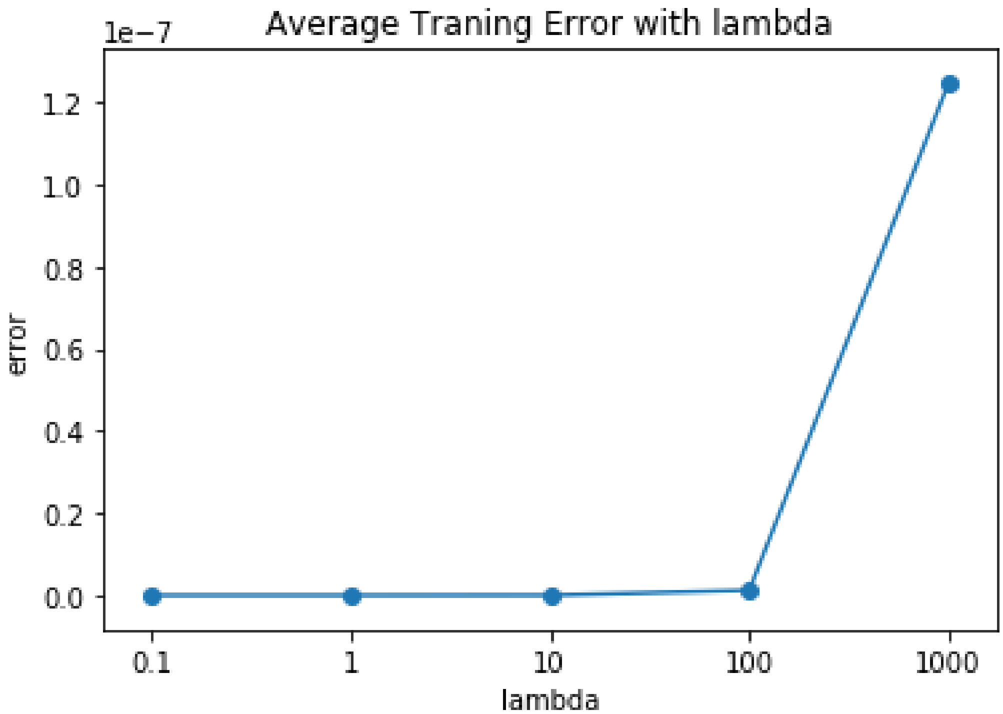
```
In [188]: def regreession_multi(X,y,lamb):

              n1, n2 = np.shape(X)
              A = np.linalg.solve(X.T.dot(X)+lamb*np.identity(n2),X.T.dot(y))
              yhat = X.dot(A)

              Rmean =  np.linalg.norm(yhat-y)**2/n1

              return {'A':A,'train_error':Rmean}
```

```
In [189]: lam = [0.1,1,10,100,1000]
          result = []
          for i in lam:
              result += [regreession_multi(X,U,i)]
          train_error = [result[i]['train_error'] for i in range(len(lam))]
```

In [190]: print(train_error)

```
[1.256702272055278e-15, 1.2566924193803558e-13, 1.2565944010143906e-11, 1.255615416610616e-09, 1
```

In [191]:
```
fig = plt.figure()
ax = fig.add_subplot(111)
ax.set_xticks(np.arange(1,len(lam)+1,1))
ax.set_xticklabels(lam)
ax.set_xlabel('lambda')
ax.set_ylabel('error')
plt.title('Average Traning Error with lambda')
plt.plot(np.arange(1,len(lam)+1,1),train_error)
plt.scatter(np.arange(1,len(lam)+1,1),train_error)
plt.savefig('b2')
plt.show()
```

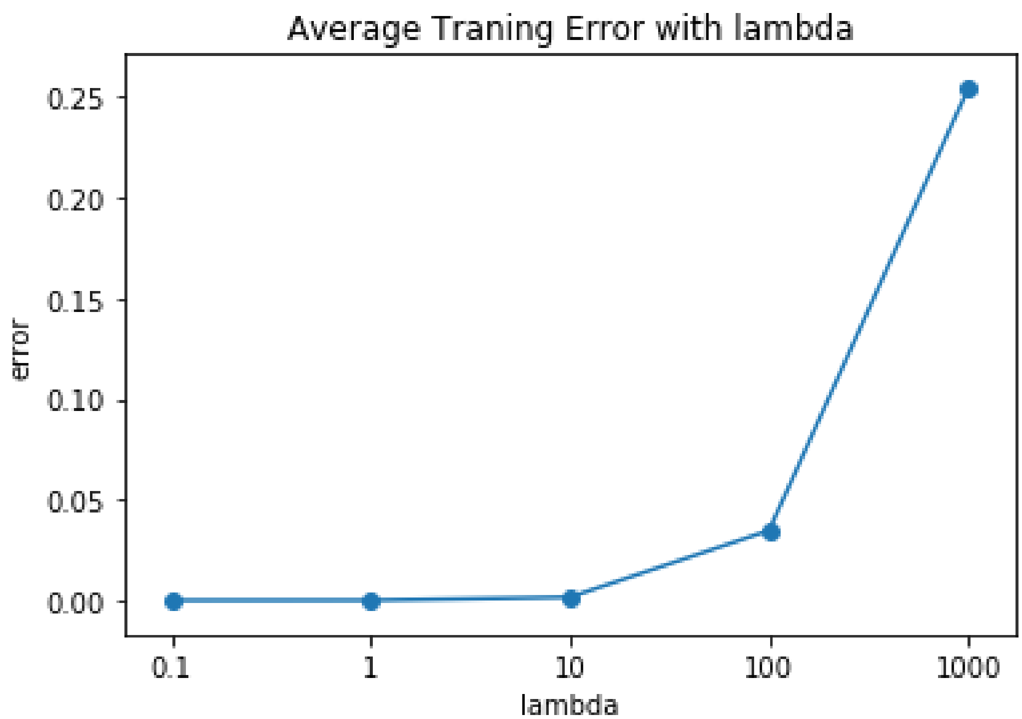


In [195]:
```
X2 = X/255.*2-1
result2 = []
for i in lam:
    result2 += [regreession_multi(X2,U,i)]
train_error2 = [result2[i]['train_error'] for i in range(len(lam))]
```

In [196]: print(train_error2)

```
[3.2557474989148846e-07, 2.9105122907682248e-05, 0.0015903814573038761, 0.034773122042375752, 0.
```

```
In [197]: fig = plt.figure()
          ax = fig.add_subplot(111)
          ax.set_xticks(np.arange(1,len(lam)+1,1))
          ax.set_xticklabels(lam)
          ax.set_xlabel('lambda')
          ax.set_ylabel('error')
          plt.title('Average Traning Error with lambda')
          plt.plot(np.arange(1,len(lam)+1,1),train_error2)
          plt.scatter(np.arange(1,len(lam)+1,1),train_error2)
          plt.savefig('b3')
          plt.show()
```



5 (d)

```
In [198]: n1, n2 = np.shape(X)
          eig1, _ = np.linalg.eig(X.T.dot(X)+ 100*np.identity(n2))
          k1 = max(eig1)/min(eig1)
          print(k1)
```

```
(52711693.3276+0j)
```

```
In [206]: n12, n22 = np.shape(X2)
          eig2, _ = np.linalg.eig(X2.T.dot(X2)+ 100*np.identity(n22))
          k1 = max(eig2)/min(eig2)
          print(k1)
```

```
(444.725931711+0j)
```

6 (f)

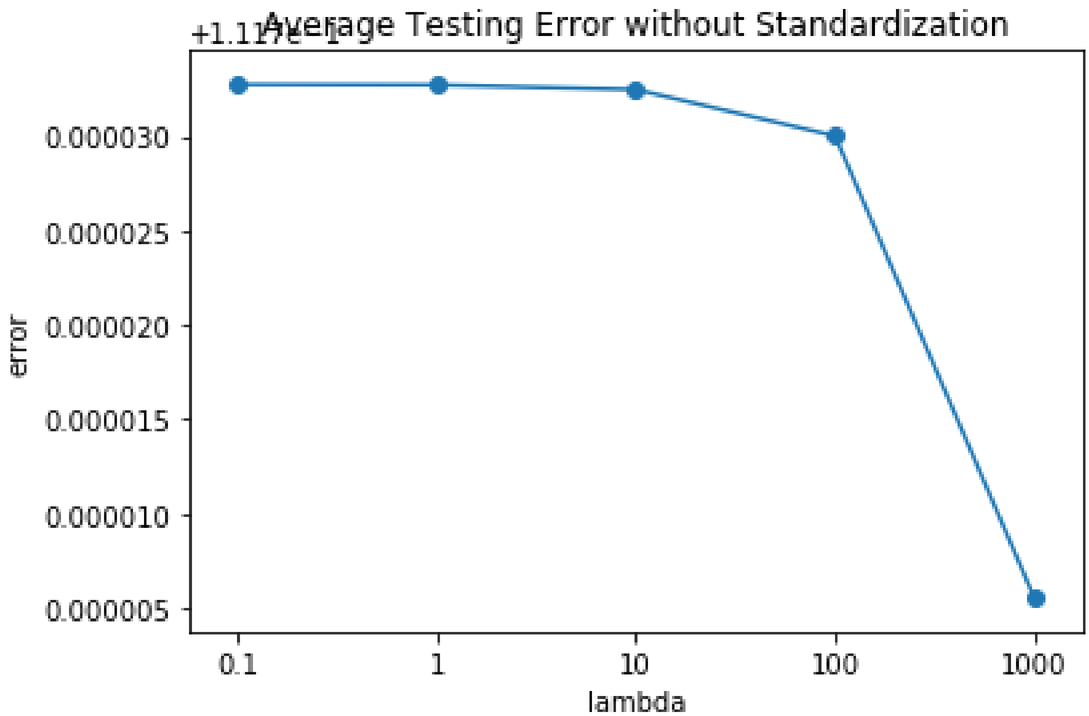
```
In [207]: with open('./data/x_test.p','rb') as f:
              x_test = pickle.load(f,encoding='latin1')
          with open('./data/y_test.p','rb') as f:
              y_test = pickle.load(f,encoding='latin1')
```

```
In [215]: Xtest = np.reshape(np.array(x_test,dtype=np.float64),(len(x_test),2700),order='F')
          ytest = np.array(y_test,dtype=np.float64)
```

```
In [216]: A = [result[i]['A'] for i in range(len(lam))]
          Rmeantest = []
          n1test, n2test = np.shape(Xtest)
          for i in range(len(A)):
              yhat = Xtest.dot(A[i])
              Rmeantest += [np.linalg.norm(yhat-ytest)/n1test]
          print(Rmeantest)
```

```
[0.11173275797988971, 0.1117327356116609, 0.11173248999880003, 0.11173002786555813, 0.1117055646
```

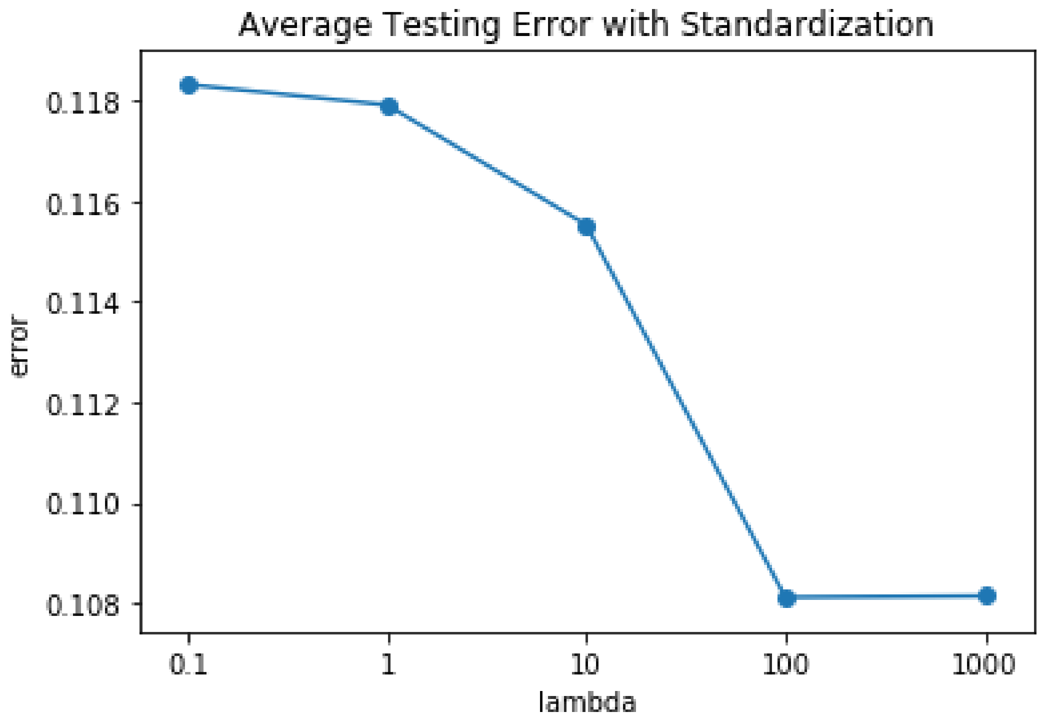
```
In [217]: fig = plt.figure()
          ax = fig.add_subplot(111)
          ax.set_xticks(np.arange(1,len(lam)+1,1))
          ax.set_xticklabels(lam)
          ax.set_xlabel('lambda')
          ax.set_ylabel('error')
          plt.title('Average Testing Error without Standardization')
          plt.plot(np.arange(1,len(lam)+1,1),Rmeantest)
          plt.scatter(np.arange(1,len(lam)+1,1),Rmeantest)
          plt.savefig('f1')
          plt.show()
```




```
In [218]: A2 = [result2[i]['A'] for i in range(len(lam))]
          Rmeantest2 = []
          Xtest2 = Xtest/255.*2-1
          n1test, n2test = np.shape(Xtest)
          for i in range(len(A)):
              yhat = Xtest2.dot(A2[i])
              Rmeantest2 += [np.linalg.norm(yhat-ytest)/n1test]
          print(Rmeantest2)
```

```
[0.1183268483625583, 0.11791898072707883, 0.11552877144355782, 0.10811083659462012, 0.1081377705
```

```
In [219]: fig = plt.figure()
          ax = fig.add_subplot(111)
          ax.set_xticks(np.arange(1,len(lam)+1,1))
          ax.set_xticklabels(lam)
          ax.set_xlabel('lambda')
          ax.set_ylabel('error')
          plt.title('Average Testing Error with Standardization')
          plt.plot(np.arange(1,len(lam)+1,1),Rmeantest2)
          plt.scatter(np.arange(1,len(lam)+1,1),Rmeantest2)
          plt.savefig('f2')
          plt.show()
```


Average Testing Error with Standardization
error
0.118
0.116
0.114
0.112
0.110
0.108
0.1
1
10
100
1000
lambda