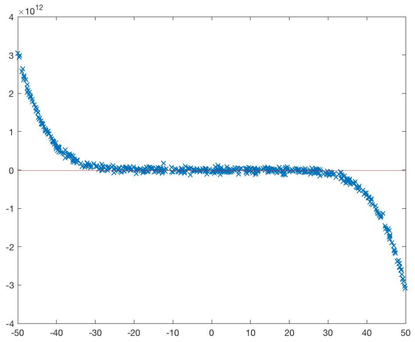
Homework 1

Yunian Pan

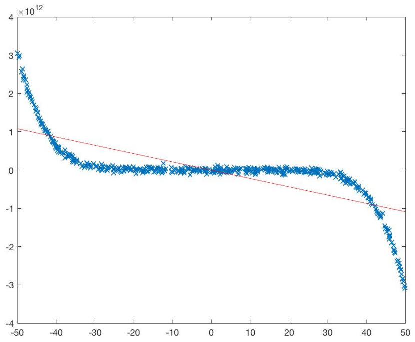
September 21, 2018

1 Problem 1

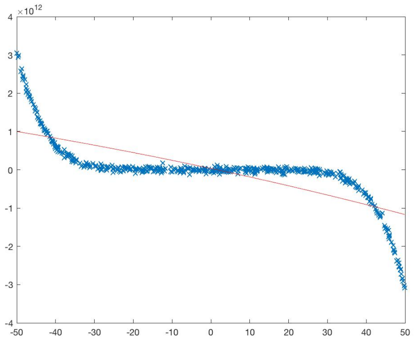
Select some typical values of d from {1, . . . , 40}, the fitting lines are shown as below:



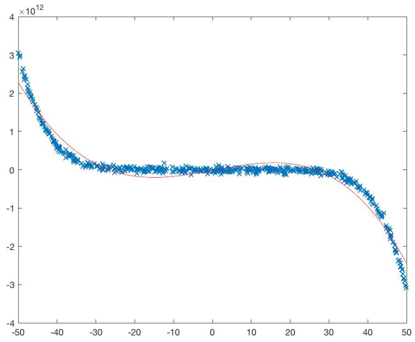
(a) d=1



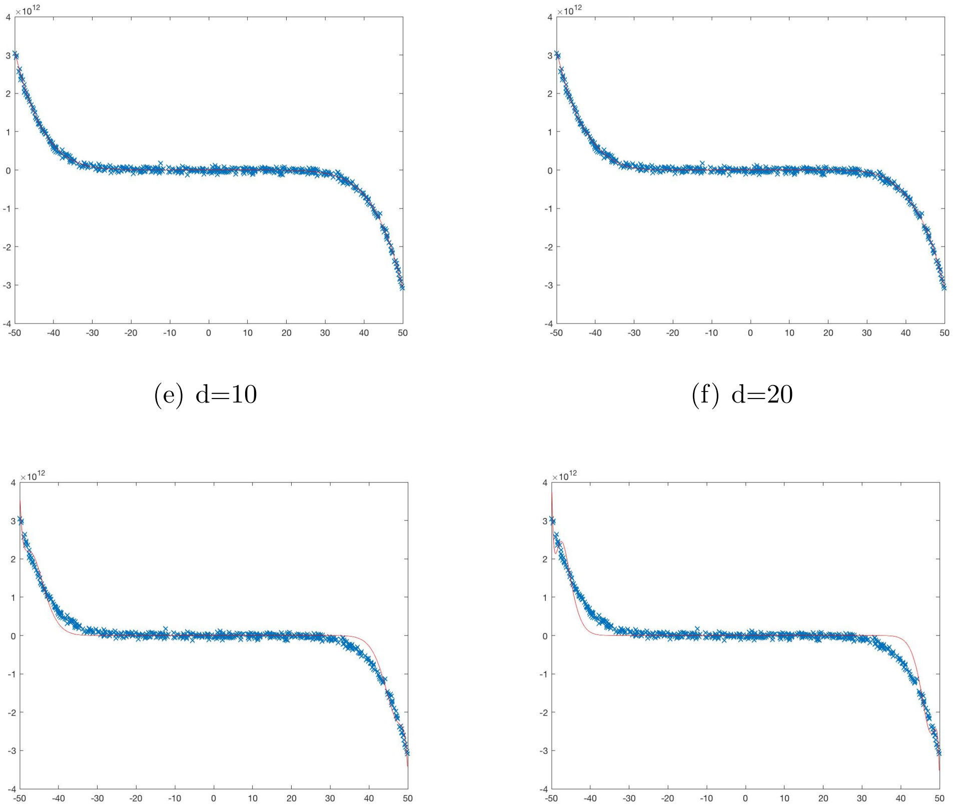
(b) d=2



(c) d=3



(d) d=5



(e) d=10

(f) d=20

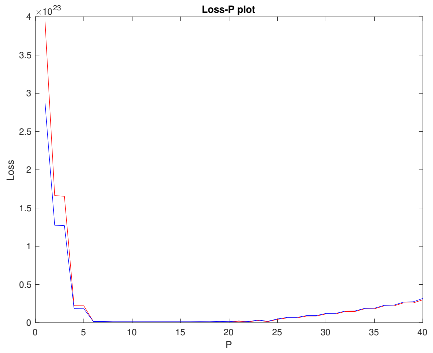
(g) d=30

(h) d=40

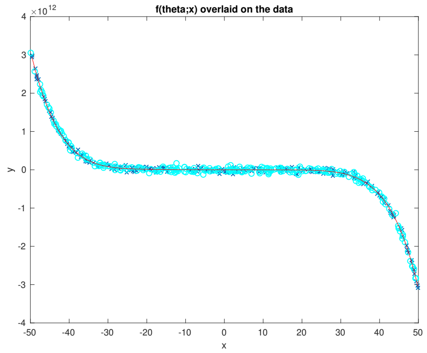
Obviously the functions from 1(e) and 1(f) best fit the data set.Through cross-validation, we can get the best d 11, which is corresponded to the lowest loss on testing set, as shown in 1(i) and 1(j):

2 Problem 2

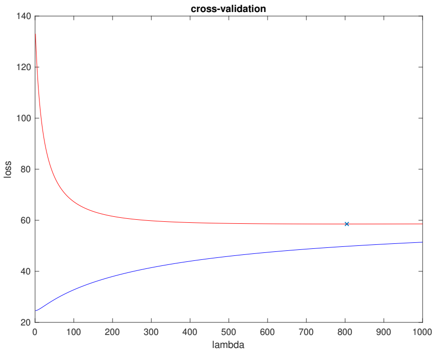
Use regularized risk minimization to get models, and select the best $\lambda = 804$ corresponded to the lowest testing loss, which is shown in $1(k)$ and $1(l)$



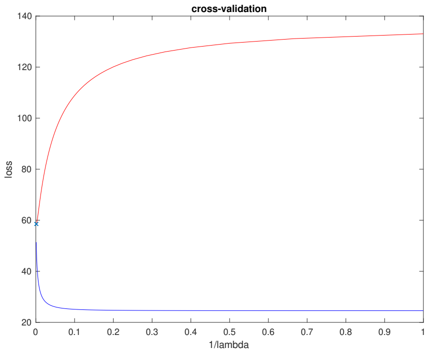
(i) loss-d



(j) overlaid



(k) λ



(l) $\frac{1}{\lambda}$

3 Problem 3

3.1

Proof:

$$\begin{aligned} g(-z) &= \frac{1}{1+e^{z}} \\ &= \frac{e^{-z}}{1+e^{-z}} \\ &= 1 - \frac{1}{1+e^{-z}} \\ &= 1 - g(z) \end{aligned}$$

3.2

Proof: suppose: y = g(z)

$$\begin{aligned}
g^{-1}(y) &= \ln \frac{y}{1-y} \\
&= \ln y - \ln (1-y) \\
&= \ln g(z) - \ln (1-g(z)) \\
&= \ln (\frac{1}{1+e^{-z}}) - \ln (\frac{e^{-z}}{1+e^{-z}}) \\
&= 0 - \ln (1-e^{-z}) + \ln (1+e^{-z}) + \ln e^{-z} \\
&= z
\end{aligned}$$

Therefore, its inverse is given by $g^{-1}(y) = \ln (y/(1-y))$.

4 Problem 4

For the function:

$$f(\mathbf{x};\theta) = (1+\exp(-\theta^\intercal \mathbf{x}))^{-1}$$

with the loss function:

$$L = (1-y_i)\log(1-f(x_i,\theta)) - y_i \log(f(x_i,\theta))$$

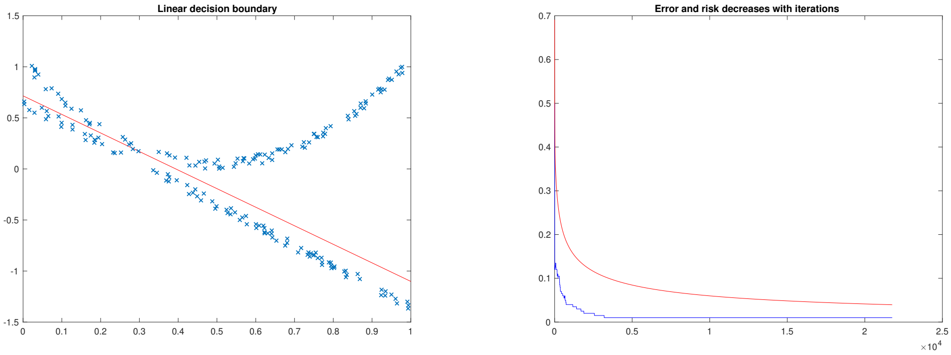
We were supposed to solve the gradient descent with derivation: $\nabla_\theta R = 0$

$$\begin{aligned}
\nabla_\theta R &= \frac{1}{N}\sum_{i=1}^{N}\left[\frac{(1-y_i)}{1-f(x_i;\theta)} - \frac{y_i}{f(x_i;\theta)}\right] f^{'}(x_i;\theta) \\
&= \frac{1}{N}\sum_{i=1}^{N}\left[\frac{(1-y_i)(1+e^{-\theta^\intercal x_i})}{e^{-\theta^\intercal x_i}} - y_i(1+e^{-\theta^\intercal x_i})\right] \frac{x_i e^{-\theta^\intercal x_i}}{(1+e^{-\theta^\intercal x_i})^2} \\
&= \frac{1}{N}\sum_{i=1}^{N}\left[\frac{(1-y_i)x_i}{1+e^{-\theta^\intercal x_i}} - \frac{y_i x_i e^{-\theta^\intercal x_i}}{1+e^{-\theta^\intercal x_i}}\right] \\
&= \frac{1}{N}\sum_{i=1}^{N}\left[(1-y_i)x_i - y_i x_i e^{-\theta^\intercal x_i}\right](1+e^{-\theta^\intercal x_i})^{-1}
\end{aligned}$$

However, this equation is too complicated to solve analytically, with some numerical method we can let the results of parameters approach to the best model.

Set the iteration to be 100000, with tolerance $\epsilon = 0.001$ and steps $= 1$, the 2D X binary boundary given by $\theta_1 \times x_1 + \theta_2 \times x_2 + \theta_3 = 0$ and the classification error and empirical risk obtained from the process are shown as $1(m)$ and $1(n)$.

The model obtained is $\theta = [48.8850, 26.8971, -19.2476]$.

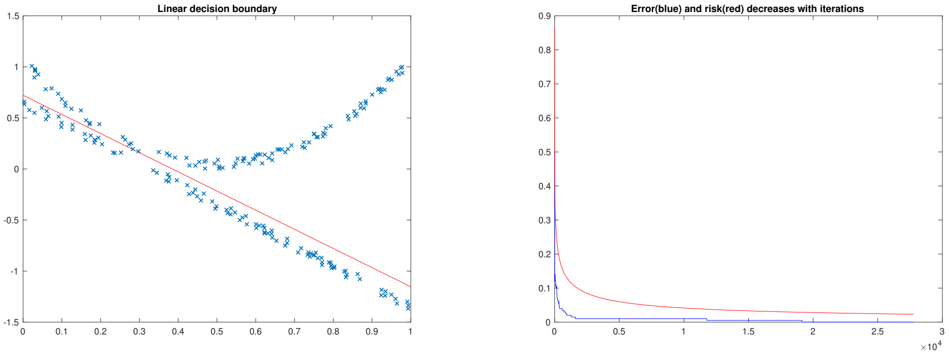


(m) binary boundary 1

(n) error and risk 1

Set the iteration to be 200000, with tolerance $\epsilon = 0.001$ and steps $= 2$, the model will be as shown as $1(o)$ and $1(p)$.

The model obtained is $\theta = [68.6228, 36.5855, -26.4124]$.



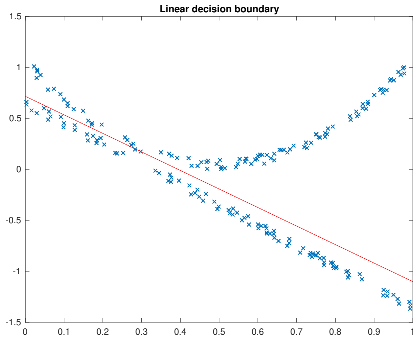
(o) binary boundary 2

(p) error and risk 2

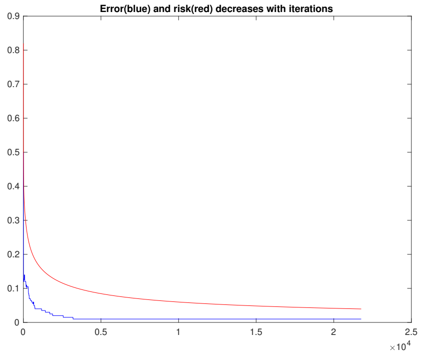
Set the iteration to be 500000, with tolerance $\epsilon = 0.001$ and steps $= 1$, the model will be as shown as $1(q)$ and $1(r)$.

The model obtained is $\theta = [48.8852, 26.8971, -19.2477]$, so the iteration doesn't affect the final results as it's the same as the first case.

Since the whole data set is used for training, the model may be overfitting when the conditions change.



(q) binary boundary 3



(r) error and risk 3

5 annotion

The matlab source file *polyregcv.m* is for problem 1, it calls the function *polyreg()* provided by *polyreg.m* which is downloaded from the website, the *multivariable.m* and *regrimin.m* are for problem 2, and the *LRGD.m* is for problem 4.