hw3-Solution

Graded

Student

Chiang Yi Jie

Total Points

239 / 260 pts

Question 1

Problem 1

20 / 20 pts

✔ **+ 20 pts** Totally correct.

+ 18 pts Some minor errors.

+ 10 pts Deriving the wrong false positive/negative error or the wrong relation between false positive and negative error.

+ 5 pts Wrong answer but have some reasonable efforts.

+ 0 pts Wrong answer

Question 2

Problem 2

20 / 20 pts

✔ **+ 20 pts** Totally correct.

+ 18 pts Small typo.

+ 15 pts Correct, but express the answer in terms other than $E_{out}(g)$ and g.

+ 10 pts Correct, but it lacks some explanation for your steps.

+ 5 pts Wrong answer, but you have some reasonable efforts.

+ 0 pts Wrong answer

Question 3

Problem 3

20 / 20 pts

✔ **+ 20 pts** Correct.

+ 15 pts Generally correct, but with minor error.

+ 10 pts On the right track, but with major error.

+ 0 pts Totally Wrong.

– 2 pts Wrong selected page.

Question 4

Problem 4

20 / 20 pts

✔ **+ 20 pts** Correct.

+ 15 pts Generally correct, but with minor error.

+ 10 pts On the right track, but with major error or multiple minor error.

+ 5 pts Incomplete solution. (only derive the gradient or the expectation of x)

+ 3 pts Just consider gradient.

+ 0 pts Totally Wrong.

− 2 pts Wrong selected page.

Question 5

Problem 5

20 / 20 pts

✔ **+ 20 pts** Correct.

+ 15 pts Generally correct, but with minor error.

+ 10 pts On the right track of derivation, but with major error or multiple minor error.

+ 5 pts Wrong, but with reasonable effort.

+ 0 pts Totally Wrong.

− 2 pts Wrong selected page.

Question 6

Problem 6

20 / 20 pts

✔ **− 0 pts** Correct

− 20 pts No answer or Incorrect proof.

− 10 pts No Hessian derivative.

− 10 pts No final answer of D

− 4 pts Unclear or incorrect relationship between derivative and A_E(w)

− 4 pts Incorrect comparing a diagonal matrix with a sum form.

− 4 pts Incorrect answer of D

− 2 pts Wrong page.

− 2 pts inappropriate expression or unclear assumption

− 1 pt Miss $1/N$ in D

− 2 pts Incorrect result of first-order derivative.

Question 7

Problem 7

19 / 20 pts

− 0 pts Correct

− 6 pts No gradient, or the gradient is wrong.

− 6 pts No updating rule.

− 6 pts No or incorrect comparison with PLA.

− 4 pts Incorrect updating rule

− 2 pts Unclear or minor incorrect gradient of error function.

− 2 pts Unclear or minor incorrect updating rule.

− 2 pts Unclear or minor incorrect comparison with PLA.

✔ **− 1 pt** inappropriate expression or unclear assumption

1 ys<0?

− 18 pts Major error of the proof.

− 20 pts No solution.

− 2 pts Wrong page.

Question 8

Problem 8

20 / 20 pts

✔ **+ 20 pts** Totally correct.

+ 18 pts Almost correct.

+ 18 pts Small typo.

+ 13 pts Only consider one case of derivatives
(w.r.t. $w_k,\ k = y$), but neglect the other case that differentiate w.r.t. $w_k,\ k \neq y$

+ 7 pts Wrong answer but with some reasonable efforts.

+ 0 pts Wrong answer.

Minor typo.

2 minus

Question 9

Problem 9

20 / 20 pts

- ✔ **+ 10 pts** Correct answer
- ✔ **+ 10 pts** Correct histogram
- **+ 10 pts** Wrong answer with bell shaped histogram
- **+ 0 pts** No answer / Wrong answer
- **− 2 pts** wrong page selection

Question 10

Problem 10

20 / 20 pts

- ✔ **+ 10 pts** Correct answer
- ✔ **+ 10 pts** Correct histogram
- **+ 10 pts** Wrong answer with bell shaped histogram
- **+ 0 pts** No answer / Wrong answer
- **− 2 pts** wrong page selection
- **+ 5 pts** nice try

Question 11

Problem 11

20 / 20 pts

- ✔ **+ 10 pts** Correct answer
- ✔ **+ 10 pts** Correct scatter plot
- **+ 10 pts** Wrong answer with reasonable scatter plot
- **+ 0 pts** No answer / Wrong answer
- **− 2 pts** wrong page selection

Question 12

Problem 12

20 / 20 pts

- ✔ **+ 10 pts** Correct answer (within error range)
- ✔ **+ 5 pts** Correct scatter plot
- ✔ **+ 5 pts** Reasonable findings and explanations
- **+ 10 pts** Wrong answer with reasonable scatter plot and findings
- **+ 0 pts** No answer / Wrong answer

Question 13

Problem 13

0 / 20 pts

– 0 pts Totally correct, no logical flaw.

– 5 pts A minor logical flaw exists.

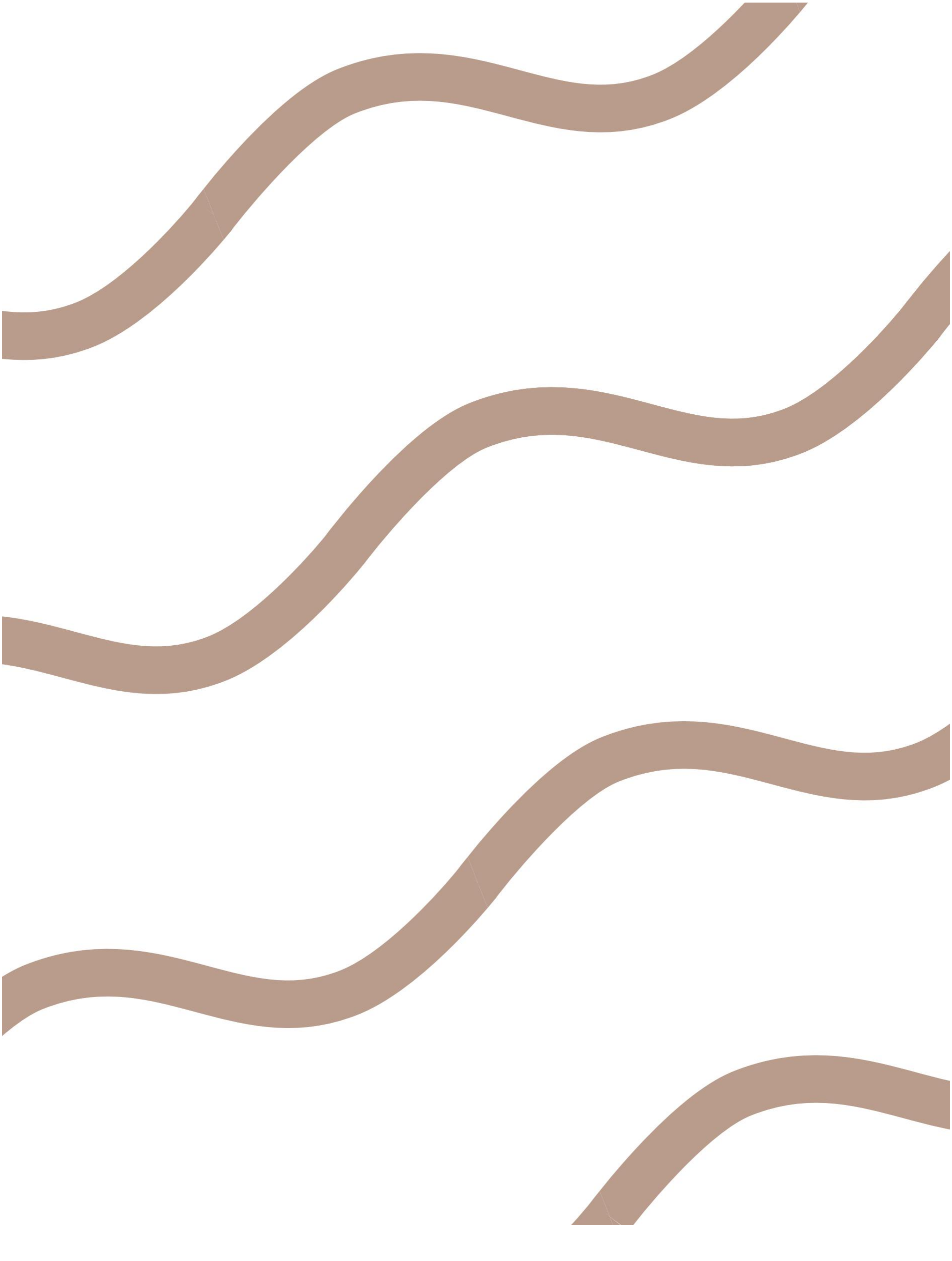
– 10 pts A logical flaw exists.

– 15 pts A major logical flaw exists.

✔ **– 20 pts** Wrong answer

3 yx, not x.

No questions assigned to the following page.



Question assigned to the following page: 1

1. (20 points) Consider a binary classification problem, where $\mathcal{Y} = \{-1, +1\}$. Assume a noisy scenario where the data is generated i.i.d. from some $P(\mathbf{x}, y)$. In class, we discussed that when the 0/1 error function (i.e. classification error) is considered, calculating the "ideal mini-target" on each $\mathbf{x}$ reveals the hidden target function of

$$f_{0/1}(\mathbf{x}) = \text{argmax}_{y\in\{-1,+1\}} P(y|\mathbf{x}) = \text{sign}\left(P(+1|\mathbf{x}) - \frac{1}{2}\right).$$

Instead of the 0/1 error, if we consider the CIA error function, where a false positive (classifying a negative example as a positive one) is 1000 times more important than a false negative, the hidden target should be changed to

$$f_{\text{CIA}}(\mathbf{x}) = \text{sign}\big(P(+1|\mathbf{x}) - \alpha\big).$$

Prove what the value of α should be.

$f_{0/1}(x) = \text{sign}\left(P(+1|x) - \frac{1}{2}\right)$, $f(x) = 1$ if $P(+1|x) > 0.5$.
$f(x) = -1$ if $P(+1|x) < 0.5$ threshold.

	$f(x)$: +1	-1
y: +1	0	1
-1	1	0

$f_{0/1}(x_n) \neq y_n$, $y_n = -1$ is 1000 times important than $y_n = +1$
↓
In Error measure, when we enter the point $(y_n, f(x_n)) = (-1, +1)$ once, equals to we enter the point 1000 times.

	$f(x)$: +1	-1
y: +1	0	1
-1	1000	0

$\rightarrow E_{out} = P(+1|x) \times [\![f(x) = -1]\!] + P(-1|x) \times [\![f(x) = +1]\!] \times 1000$

$\rightarrow P(+1|x) : P(-1|x) = 1000 : 1$

$P(+1|x) = \frac{1000}{1001}$, $\alpha = \frac{1000}{1001}$ #

Question assigned to the following page: 2

2. (20 points) Consider a binary classification task, where God gives you some noiseless data i.i.d. from an unknown distribution $P(\mathbf{x})$ and an unknown target function $f(\mathbf{x})$ that maps from $\mathcal{X}$ to $\{-1,+1\}$. After you use the data to obtain some $g(\mathbf{x})$ that suffers

$$\begin{aligned} E_{\text{out}}(g) &= \mathcal{E}_{\mathbf{x}\sim P(\mathbf{x})} [\![g(\mathbf{x}) \neq f(\mathbf{x})]\!] \text{ (here } \mathcal{E} \text{ means expectation, as shown in class slides)} \\ &= \mathbb{E}_{\mathbf{x}\sim P(\mathbf{x})} [\![g(\mathbf{x}) \neq f(\mathbf{x})]\!] \text{ (or if you like the more beautiful font } \mathbb{E} \text{ for expectation).} \end{aligned}$$

Now, assume that $g(\mathbf{x})$ is put in a noisy test environment where

$$\begin{aligned} P(y = +f(\mathbf{x})|\mathbf{x}) &= 1-\epsilon \\ P(y = -f(\mathbf{x})|\mathbf{x}) &= \epsilon. \end{aligned}$$

Derive

$$\mathbb{E}_{(\mathbf{x},y)\sim P(\mathbf{x},y)} [\![g(\mathbf{x}) \neq y]\!]$$

as a function of $E_{\text{out}}(g)$ and ϵ.

	$1-E_{out}$ $g(x)=f(x)$	E_{out} $g(x)\neq f(x)$	
$y=f(x)$		X	$\to 1-\epsilon$
$y=-f(x)$	X		$\to \epsilon$

$\to (1-E_{out})\cdot\epsilon + E_{out}(1-\epsilon)$

In ideal case $P[g(x)\neq f(x)] = E_{out}$

$P[g(x)=f(x)] = 1-E_{out}$

In noise $P[y=f(x)\,|\,x] = 1-\epsilon$

$P[y=-f(x)\,|\,x] = \epsilon$

	$g(x)=f(x)$	$g(x)\neq f(x)$	
$y=f(x)$	S_1	S_2	$1-\epsilon$
$y=-f(x)$	S_3	S_4	ϵ
P_2	$1-E_{out}$	E_{out}	

$$\begin{aligned} E_{(x,y)\sim P(x,y)}[g(x)\neq y] &= E_{x\sim p(x)}[g(x)\neq f(x)]\cdot P(y=+f(x)\,|\,x) + \\ &\quad E_{x\sim p(x)}[g(x)=f(x)]\cdot P(y=-f(x)\,|\,x) \\ &= (1-E_{out})\cdot\epsilon + E_{out}\cdot(1-\epsilon) \text{ \#} \end{aligned}$$

Question assigned to the following page: 3

3. (20 points) Consider a hypothesis set that contains hypotheses of the form $h(x) = wx$ for $x \in \mathbb{R}$. Combine the hypothesis set with the squared error function to minimize

$$E_{\text{in}}(w) = \frac{1}{N}\sum_{n=1}^{N}(h(x_n) - y_n)^2$$

on a given data set $\{(x_n, y_n)\}_{n=1}^{N}$. Derive the optimal w_{LIN} in terms of (x_n, y_n) and express the result *without* using matrix/vector notations. You can assume all denominators to be non-zero. *(Hint: This is linear regression in $\mathbb{R}$ without the added x_0.)* $\frac{\partial}{\partial w_1} = 2[ax^2 - w_1 x + (b - w_0)](1) \quad -- (w) = \int$

$h(x) = wx$

$$E_{in}(w) = \frac{1}{N}\sum_{n=1}^{N}(wx_n - y_n)^2$$

W_{LIN} exists when min $E_{in}(w)$ appears $\rightarrow$ $\nabla E_{in}(w) = 0$

$$E_{in}(w) = \frac{1}{N}\left[(wx_1 - y_1)^2 + (wx_2 - y_2)^2 + (wx_3 - y_3)^2 \cdots + (wx_n - y_n)^2\right]$$

$$= \frac{1}{N}(w^2x_1^2 - 2wx_1y_1 + y_1^2 \cdots + w^2x_n^2 - 2wx_ny_n + y_n^2)$$

$$\nabla E_{in}(w) = \frac{1}{N}(2wx_1^2 - 2x_1y_1 + 2wx_2^2 + 2x_2y_2 \cdots + 2wx_n^2 + 2x_ny_n)$$

$$= \frac{2}{N}\left(w\sum_{n=1}^{n=N} x_n^2 - \sum_{n=1}^{n=N} x_n y_n\right) = 0$$

$$\rightarrow w = \frac{\sum_{n=1}^{N} x_n y_n}{\sum_{n=1}^{N} x_n^2} \quad \#$$

Question assigned to the following page: 4

4. (20 points) Consider the target function $f(x) = ax^2 + b$. Sample x uniformly from $[0, 1]$, and use all linear hypotheses $h(x) = w_0 + w_1 \cdot x$ to approximate the target function with respect to the squared error. For any given (a, b), derive the weights (w_0^*, w_1^*) of the optimal hypothesis as a function of (a, b).

$$E_{in}(w) = \frac{1}{N}\sum_{n=1}^{N}\left(f(x_n) - h(x_n)\right) = \frac{1}{N}\sum_{n=1}^{N}\left(ax_n^2 + b - w_1x_n - w_0\right)^2$$

since x uniformly from $[0,1]$, $E_{in}(w_0, w_1) = \int_0^1 (ax^2 - w_1x + b - w_0)^2\,dx$

$$\min E_{in}(w_0, w_1) \Leftrightarrow \nabla E_{in}(w_0, w_1) = 0$$

$$\begin{cases} \frac{\partial}{\partial w_0} = \int_0^1 (ax^2 - w_1x + b - w_0)\cdot 2\cdot(-1)\,dx = 0 \\ \frac{\partial}{\partial w_1} = \int_0^1 (ax^2 - w_1x + b - w_0)\cdot 2\cdot(-x)\,dx = 0 \end{cases}$$

$$\begin{cases} \int_0^1 (ax^2 - w_1x + b - w_0)\,dx = 0 \\ \int_0^1 (ax^3 - w_1x^2 + bx - w_0x)\,dx = 0 \end{cases}$$

$$\begin{cases} \frac{1}{3}ax^3 - \frac{1}{2}w_1x^2 + (b - w_0)x \Big|_0^1 = 0 \\ \frac{1}{4}ax^4 - \frac{1}{3}w_1x^3 + \frac{1}{2}(b - w_0)x^2 \Big|_0^1 = 0 \end{cases}$$

$$\begin{cases} \frac{1}{3}a - \frac{1}{2}w_1 + b - w_0 = 0 \\ \frac{1}{4}a - \frac{1}{3}w_1 + \frac{1}{2}b - \frac{1}{2}w_0 = 0 \end{cases}$$

$$\begin{cases} w_0 + \frac{1}{2}w_1 = \frac{1}{3}a + b \\ \frac{1}{2}w_0 + \frac{1}{3}w_1 = \frac{1}{4}a + \frac{1}{2}b, \quad w_0 + \frac{2}{3}w_1 = \frac{1}{4}a + b \end{cases}$$

$$\rightarrow \frac{1}{6}w_1 = \frac{1}{6}a, \quad w_1^* = a \ \#$$

$$w_0^* = -\frac{1}{6}a + b \ \#$$

Question assigned to the following page: 5

5. (20 points) Consider running linear regression on $\{(\mathbf{x}_n, y_n)\}_{n=1}^N$, where $\mathbf{x}_n$ includes the constant dimension $x_0 = 1$ as usual. For simplicity, you can assume that X^TX is invertible. Assume that the unique (why :-)) solution $\mathbf{w}_{\text{LIN}}$ is obtained after running linear regression on the data above. Now, consider an output transformation $W_{LIN} = (X^TX)^{-1}X^Ty$

$$y'_n = ay_n + b.$$

for some given constants (a, b). Run linear regression on $\{(\mathbf{x}_n, y'_n)\}_{n=1}^N$ to obtain the unique solution $\mathbf{w}'_{\text{LIN}}$. Derive $\mathbf{w}'_{\text{LIN}}$ as a function of $\mathbf{w}_{\text{LIN}}$ and (a, b).

$$W_{LIN} = (X^TX)^{-1}X^Ty \quad \text{——} \ ①$$

$$E_{in}(w') = \frac{1}{N}\| \mathbb{X} w' - (ay+b) \|^2 = \frac{1}{N}\left(w'^T\mathbb{X}^T\mathbb{X}w' - 2w'^T\mathbb{X}^T(ay+b) + (ay+b)^T(ay+b) \right)$$

$$\nabla E_{in}(w') = \frac{1}{N}(2\mathbb{X}^T\mathbb{X}w' - 2\mathbb{X}^T(ay+b) + 0)$$

W_{LIN}' happens when $\nabla E_{in} = 0$: $2\mathbb{X}^T\mathbb{X}w' = 2\mathbb{X}^T(ay+b)$, $W_{LIN}' = (\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T(ay+b) = (\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T \cdot ay + (\mathbb{X}^T\cdot\mathbb{X})^{-1}\mathbb{X}^T\cdot b$

$$\left[\begin{array}{l} a(\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^Ty = a \cdot W_{LIN} \\ (\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T \cdot b \end{array}\right.$$

since $\mathbb{X} \cdot aW_{LIN} = ay$ $\quad \begin{bmatrix} 1 & x_{11} & x_{12} & x_{13} \cdots x_{1n} \\ 1 & & & x_{2n} \\ \vdots & & & x_{3n} \\ & & & \vdots \\ 1 & & & x_{nn} \end{bmatrix} \times a \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} = a \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$

all bs comes from $(W_{LIN}')_1$

$\mathbb{X} \cdot W_{LIN}' = ay+b$ $\quad \begin{bmatrix} 1 & x_{11} & x_{12} & x_{13} \cdots & x_{1n} \\ \vdots & & & & \vdots \\ 1 & x_{n1} & \cdots & & x_{nn} \end{bmatrix} \times \begin{bmatrix} aw_1+b \\ aw_2 \\ aw_3 \\ \vdots \\ aw_n \end{bmatrix} = \begin{bmatrix} ay_1+\underline{b} \\ ay_2+\underline{b} \\ \vdots \\ ay_n+b \end{bmatrix}$

$$\downarrow$$

$$a\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} + b\begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$\rightarrow W_{LIN}' = aW_{LIN} + b\begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$, where $\begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$ is a $n \times 1$ matrix. #

Question assigned to the following page: 6

6. (20 points) Let $E(\mathbf{w})\colon \mathbb{R}^d \to \mathbb{R}$ be a function. Denote the gradient $\mathbf{b}_E(\mathbf{w})$ and the Hessian $\mathrm{A}_E(\mathbf{w})$ by

$$\mathbf{b}_E(\mathbf{w}) = \nabla E(\mathbf{w}) = \begin{bmatrix} \frac{\partial E}{\partial w_1}(\mathbf{w}) \\ \frac{\partial E}{\partial w_2}(\mathbf{w}) \\ \vdots \\ \frac{\partial E}{\partial w_d}(\mathbf{w}) \end{bmatrix}_{d\times 1} \quad \text{and } \mathrm{A}_E(\mathbf{w}) = \begin{bmatrix} \frac{\partial^2 E}{\partial w_1^2}(\mathbf{w}) & \frac{\partial^2 E}{\partial w_1 \partial w_2}(\mathbf{w}) & \cdots & \frac{\partial^2 E}{\partial w_1 \partial w_d}(\mathbf{w}) \\ \frac{\partial^2 E}{\partial w_2 \partial w_1}(\mathbf{w}) & \frac{\partial^2 E}{\partial w_2^2}(\mathbf{w}) & \cdots & \frac{\partial^2 E}{\partial w_2 \partial w_d}(\mathbf{w}) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 E}{\partial w_d \partial w_1}(\mathbf{w}) & \frac{\partial^2 E}{\partial w_d \partial w_2}(\mathbf{w}) & \cdots & \frac{\partial^2 E}{\partial w_d^2}(\mathbf{w}) \end{bmatrix}_{d\times d}.$$

Then, the second-order Taylor's expansion of $E(\mathbf{w})$ around $\mathbf{u}$ is:

$$E(\mathbf{w}) \approx E(\mathbf{u}) + \mathbf{b}_E(\mathbf{u})^T(\mathbf{w}-\mathbf{u}) + \frac{1}{2}(\mathbf{w}-\mathbf{u})^T \mathrm{A}_E(\mathbf{u})(\mathbf{w}-\mathbf{u}).$$

$b_E(u)^T \cancel{(w-u)}$
$= -(w-u)^T A_E(u) \cancel{(w-u)}$

Suppose $\mathrm{A}_E(\mathbf{u})$ is positive definite. The optimal direction $\mathbf{v}$ such that $\mathbf{w} \leftarrow \mathbf{u} + \mathbf{v}$ minimizes the right-hand-side of the Taylor's expansion above is simply $-(\mathrm{A}_E(\mathbf{u}))^{-1}\mathbf{b}_E(\mathbf{u})$.

$\boxed{\textit{Hint: Homework 0! :-)}}$

Now, consider minimizing $E_{\text{in}}(\mathbf{w})$ in logistic regression problem with Newton's method on a data set $\{(\mathbf{x}_n, y_n)\}_{n=1}^N$ with the cross-entropy error function for E_{in}:

$w_1x_1 + w_2x_2 \cdots w_nx_n$

$$E_{\text{in}}(\mathbf{w}) = \frac{1}{N}\sum_{n=1}^N \ln(1 + \exp(-y_n \mathbf{w}^T \mathbf{x}_n)).$$

$C \;{-1}\; \Sigma \ln\left(1 + \frac{1}{h(x)}\right)$

For any given $\mathbf{w}_t$, let

$$h_t(\mathbf{x}) = \frac{1}{1 + \exp(\mathbf{w}_t^T \mathbf{x})}.$$

Express the Hessian $\mathrm{A}_E(\mathbf{w}_t)$ with $E = E_{\text{in}}$ as $\mathrm{X}^T\mathrm{D}\mathrm{X}$, where D is an N by N diagonal matrix. Derive what D should be in terms of h_t, $\mathbf{w}_t$, $\mathbf{x}_n$, and y_n.

24

$$h_t(x) = \frac{1}{1+\exp(w_t^T x)} \quad , \quad E_{in} = \frac{1}{N}\sum_{n=1}^{N} \ln(1+\exp(-y_n w^T x_n))$$

$$h_t(-x) = \frac{1}{1+\exp(-w_t^T x)} = 1 - h_t(x)$$

$$b_E(w) = \nabla E_{in} = \frac{1}{N}\sum_{n=1}^{N} \frac{\exp(-y_n x_n)}{1+\exp(-y_n x_n)}(-y_n x_{n,i}) = \frac{1}{N}\sum_{n=1}^{N}(1-h_t(y_n x_n))$$

$$A_E(w) = \nabla b_E(w) = \frac{1}{N}\sum_{n=1}^{N} \frac{\exp(-y_n x_n)(-y_n x_{nj})(1+\exp(-y_n x_n)) - \exp^2(-y_n x_n)(-y_n x_{n,j})}{(1+\exp(-y_n x_n))^2} \cdot (-y_n x_{n,i})$$

$$= \frac{1}{N}\sum_{n=1}^{N} \frac{\exp(-y_n x_n)(-y_n x_{nj})(-y_n x_{n,i})[1+\exp(-y_n x_n) - \exp(-y_n x_n)]}{(1+\exp(-y_n x_n))^2}$$

$$= \frac{1}{N}\sum_{n=1}^{N} \frac{\exp(-y_n x_n)(y_n x_{nj})(y_n x_{n,i})}{(1+\exp(-y_n x_n))^2} = X^T D X.$$

$$D_{i,j} = \frac{1}{N} h_t(x_n y_n)(1-h_t(x_n y_n)) \cdot y_{n,i} \cdot y_{n,j}$$

$$= \frac{1}{N}(1-h_t(x_n))h_t(x_n)\, y_{ni} \cdot y_{nj}$$

since $y_{n,i} \cdot y_{n,j} = 1$ (D diagonal), $D_{ij} = \frac{1}{N}\begin{bmatrix} h_t(x_1)(1-h_t(x_1)) & & & \\ & h_t(x_2)(1-h_t(x_2)) & & \\ & & h_t(x_3)(1-h_t(x_3)) & \\ & & & \ddots \end{bmatrix}$

Question assigned to the following page: [7]

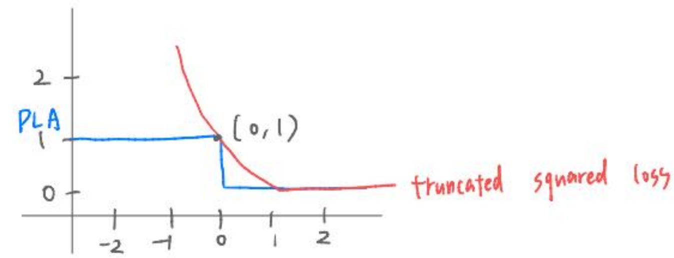
7. (20 points) The truncated squared loss

$$\text{err}(s, y) = (\max(0, 1 - ys))^2$$

can be easily shown to be an upper bound on the 0/1 error. Assume that s is generated from a linear scoring function $s = \mathbf{w}^T\mathbf{x}$ like Page 3/25 of Lecture 11. Derive a "perceptron learning algorithm" by applying SGD on the truncated squared loss. Compare the resulting algorithm with the original PLA. Discuss the similarities and differences using 5 to 10 sentences.

PLA: $\text{error}_{0/1}(s, y) = [\,\text{sign}(s) \neq y\,]$

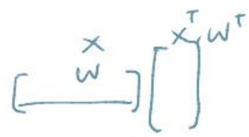
$W_{t+1} = W_t + 1 \cdot [\,y_n \neq \text{sign}(s)\,]\,(y_n \cdot x_n)$



SGD: $\text{error}(s, y) = (\max(0, 1 - y_s))^2$

$W_{t+1} = W_t + \eta \cdot (-\nabla \text{error}(y, w^T, x))$

$= W_t + \eta \cdot (-\nabla (\max(0, 1 - ys))^2)$



• if $ys > 0$, $\text{error}(s, y) = 0$

$0 < ys < 1$, $\text{error}(s, y) > \text{error}_{0/1}(s, y)$

$ys = -1$, $\text{error}(s, y) = \text{error}_{0/1}(s, y)$

$(\nabla \max(0, 1 - ys))^2 = \nabla(1 - W^T X y_n)^2 = \nabla(W X^T y_n - 1)^2$

$= \nabla(X \cdot W^T W \cdot X^T - 2 X W^T y_n + 1)^2$

$= 2(X^T X \cdot W - y_n X) = 2(s - y_n)X = (1 - ys) X y_n$

$W_{t+1} = W_t + 2\eta\,(1 - ys) X_n y_n$ #

PLA always updates with $\eta = -1$ when $[\![y_n \neq \text{sign}(s)]\!]$; while SGD updates with altered η when $\max(0, 1 - ys) > 0$. When $ys > 1$, both PLA and SGD won't update (error = 0), when $0 < ys < 1$, SGD updates (PLA doesn't), when $ys < 0$, $W^{PLA}_{t+1} = W^{PLA} + x_n y_n$, $W^{SGD}_{t+1} = W^{SGD}_t + 2\eta(1 - ys) x_n y_n$ (updates depend on $x_n y_n$, alterable).

If $ys \ll 0$, updates larger; $ys < 0$, updates less.
However, eventually how much to update depend on η, which is usually smaller than 0.1

Consequently, SGD usually upates with flexible and smaller changes, PLA offers constant and unstable steps. In error measure, Error (SGD) > Error (PLA), that is, SGD can also be viewed as upper bound of PLA.

Question assigned to the following page: 8

Multinomial Logistic Regression

8. In Lecture 11, we solve multiclass classification by OVA or OVO decompositions. One alternative to deal with multiclass classification is to extend the original logistic regression model to Multinomial Logistic Regression (MLR). For a K-class classification problem, we will denote the output space $\mathcal{Y} = \{1, 2, \cdots, K\}$. The hypotheses considered by MLR can be indexed by a matrix

$$\mathrm{W} = \begin{bmatrix} | & | & \cdots & | & \cdots & | \\ \mathbf{w}_1 & \mathbf{w}_2 & \cdots & \mathbf{w}_k & \cdots & \mathbf{w}_K \\ | & | & \cdots & | & \cdots & | \end{bmatrix}_{(d+1)\times K},$$

that contains weight vectors $(\mathbf{w}_1, \cdots, \mathbf{w}_K)$, each of length $d+1$. The matrix represents a hypothesis

$$h_y(\mathbf{x}) = \frac{\exp(\mathbf{w}_y^T \mathbf{x})}{\sum_{i=1}^K \exp(\mathbf{w}_i^T \mathbf{x})}$$

that can be used to approximate the target distribution $P(y|\mathbf{x})$ for any $(\mathbf{x}, y)$. MLR then seeks for the maximum likelihood solution over all such hypotheses. For a given data set $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$ generated i.i.d. from some $P(\mathbf{x})$ and target distribution $P(y|\mathbf{x})$, the likelihood of $h_y(\mathbf{x})$ is proportional to $\prod_{n=1}^N h_{y_n}(\mathbf{x}_n)$. That is, minimizing the negative log likelihood is equivalent to minimizing an $E_{\text{in}}(\mathrm{W})$ that is composed of the following error function

$$\mathrm{err}(\mathrm{W}, \mathbf{x}, y) = -\ln h_y(\mathbf{x}) = -\sum_{k=1}^K [\![y = k]\!] \ln h_k(\mathbf{x}).$$

Consider minimizing $E_{\text{in}}(\mathrm{W}) = \frac{1}{N}\sum_{n=1}^N \mathrm{err}(\mathrm{W}, \mathbf{x}_n, y_n)$ with gradient descent. Derive $\nabla E_{\text{in}}(\mathrm{W})$. Your result should simply be a matrix with the same size as W. *(Note: the hypothesis that transforms the scores $\{\mathbf{w}_i^T\mathbf{x}\}_{i=1}^K$ to $h_y(\mathbf{x})$ is often called a softmax function in (multiclass) deep learning.)*

$$\frac{\partial}{\partial W_1} \sum_{i=1}^{k} \exp(W_i^T x_n)$$

$$= \frac{\partial}{\partial W_1} \left(\exp(W_1^T x_n) + \exp(W_2^T x_n) \cdots \right)$$

$$= \exp(W_1^T x_n) \begin{bmatrix} x_{n,1} \\ x_{n,2} \\ \vdots \\ x_{n,n} \end{bmatrix} = \exp(W_1^T x_n)\, x_n$$

$$E_{in}(w) = \frac{-1}{N} \sum_{n=1}^{N} \ln h_y(x)$$

$$\nabla E_{in}(w) = -\frac{1}{N} \sum_{n=1}^{N} \frac{\partial}{\partial w} \ln \frac{\exp(W_i^T x_n)}{\sum_{i=1}^{k} \exp(W_i^T x_n)} = -\frac{1}{N} \sum_{n=1}^{N} \frac{\sum_{i=1}^{k} \exp(W_i^T x_n)}{\exp(W_i^T x_n)} \times \frac{\exp(W_i^T x_n)\, x_n \sum_{i=1}^{k} \exp(W_i^T x_n) - \exp(W_i^T x_n) \exp(W_k^T x_n)\, x_n}{\left(\sum_{i=1}^{k} \exp(W_i^T x_n) \right)^2}$$

$$= -\frac{1}{N} \sum_{n=1}^{N} \frac{\cancel{\sum_{i=1}^{k} \exp(W_i^T x_n)}}{\cancel{\left(\sum_{i=1}^{k} \exp(W_i^T x_n) \right)^2}} \times \frac{\cancel{\exp(W_i^T x_n)}\, x_n \sum_{i=1}^{k} \exp(W_i^T x_n) - \cancel{\exp(W_i^T x_n)} \exp(W_k^T x_n)\, x_n}{\cancel{\exp(W_i^T x_n)}}$$

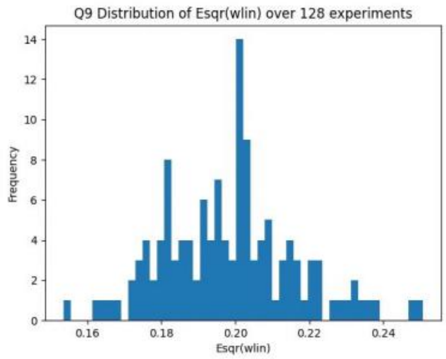
$$= -\frac{1}{N} \sum_{n=1}^{N} \frac{\left(\sum_{i=1}^{k} \exp(W_i^T x_n) - \exp W_k^T x_n \right)}{\sum_{i=1}^{k} \exp(W_i^T x_n)} \cdot x_n = \frac{1}{N} \sum_{n=1}^{N} \left(h_k(x) - 1 \right) x_n$$

if $j \neq k$, $-\frac{1}{N} \sum_{n=1}^{N} \frac{\sum_{i=1}^{k} \exp(W_i^T x_n)}{\exp(W_y^T x_n)} \times \frac{-\exp(W_y^T x_n) \exp(W_k^T x_n)}{\sum_{i=1}^{k} \exp(W_i^T x_n)^2}\, x_n = \frac{1}{N} \sum_{i=1}^{N} h(x) \cdot x_n$

$\rightarrow$ $\nabla E_{in}(W) = -\frac{1}{N}$ [$\sum_{n=1}^{N} (h_1(x$ [illegible] $y = 1]\!]\, x_n$ | $\sum_{n=1}^{N} (h_2(x_2) - [\![y = 2]\!]\, x_n$ | $\cdots$]

Question assigned to the following page: 9

9. (20 points, *) Implement the linear regression algorithm taught in the lecture. Run the algorithm for 128 times, each with a different random seed for generating the two data sets above. Plot a histogram to visualize the distribution of $E_{\text{in}}^{\text{sqr}}(\mathbf{w}_{\text{LIN}})$, where $E_{\text{in}}^{\text{sqr}}$ denotes the *averaged* squared error over N examples. What is the median $E_{\text{in}}^{\text{sqr}}$ over the 128 experiments?

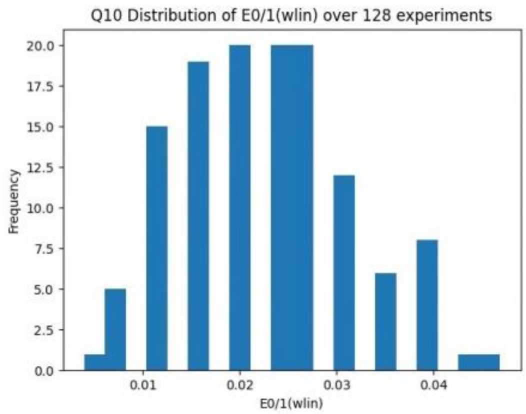


median = 0.199638 #

Question assigned to the following page: 10

10. (20 points, *) Following the previous problem, plot a histogram to visualize the distribution of $E_{\text{in}}^{0/1}(\mathbf{w}_{\text{LIN}})$, where $E_{\text{in}}^{0/1}$ denotes the averaged 0/1 error over N examples (i.e. using $\mathbf{w}_{\text{LIN}}$ for binary classification). What is the median $E_{\text{in}}^{0/1}$ over the 128 experiments?

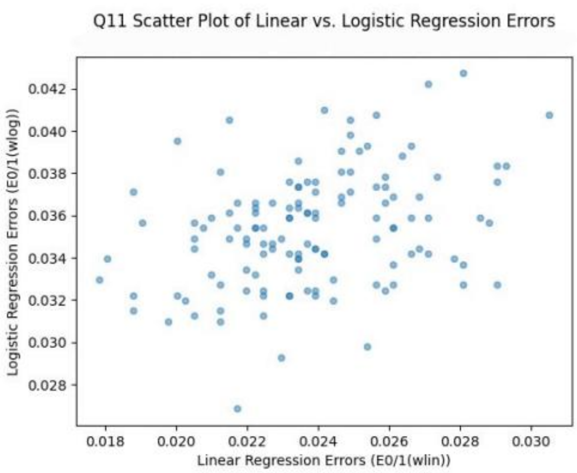
(Note: You can choose to run 128 *new experiments in this problem, or just re-use the* 128 *hypotheses* $\mathbf{w}_{\text{LIN}}$ *and test data sets obtained from the previous problem.)*



median = 0.02343 #

Question assigned to the following page: 11

11. (20 points, *) Consider two algorithms. The first one, $\mathcal{A}$, is the linear regression algorithm above. The second one $\mathcal{B}$ is logistic regression, trained with fixed learning rate gradient descent with $\eta = 0.1$ for $T = 500$ iterations, starting from $\mathbf{w}_0 = \mathbf{0}$. Run the algorithms on the same $\mathcal{D}$, and record $[E_{\text{out}}^{0/1}(\mathcal{A}(\mathcal{D})), E_{\text{out}}^{0/1}(\mathcal{B}(\mathcal{D}))]$. Repeat the process for 128 times, each with a different random seed for generating the training and test data sets above. Plot a scatter plot for $[E_{\text{out}}^{0/1}(\mathcal{A}(\mathcal{D})), E_{\text{out}}^{0/1}(\mathcal{B}(\mathcal{D}))]$. What is the median of $E_{\text{out}}^{0/1}(\mathcal{A}(\mathcal{D}))$ and what is the median of $E_{\text{out}}^{0/1}(\mathcal{B}(\mathcal{D}))$?

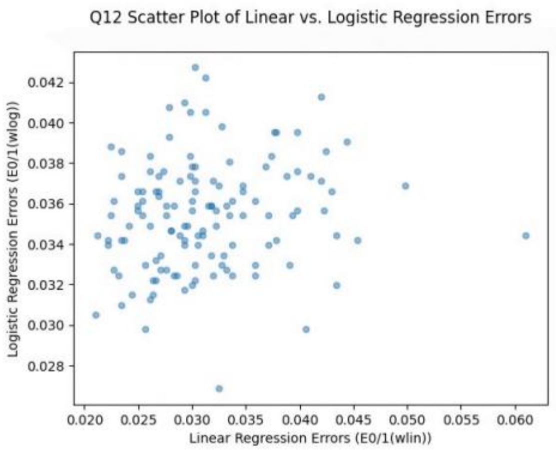


$E_{out}^{0/1}(A(D))$ median : 0.0235595703125

$E_{out}^{0/1}(B(D))$ median : 0.035400390625

Question assigned to the following page: 12

12. (20 points, *) Following the previous problem, in addition to the 256 examples in $\mathcal{D}$, add 16 outlier examples generated from the following process to your training data (but not to your test data). All outlier examples will be labeled $y = +1$ and $\mathbf{x} = [1, x_1, x_2]$ where (x_1, x_2) comes from a normal distribution of mean $[0, 6]$ and covariance $\begin{bmatrix} 0.1 & 0 \\ 0 & 0.3 \end{bmatrix}$. Name the new training data set $\mathcal{D}'$. Run the algorithms on the same $\mathcal{D}'$, and record $[E_{\text{out}}^{0/1}(\mathcal{A}(\mathcal{D}')), E_{\text{out}}^{0/1}(\mathcal{B}(\mathcal{D}'))]$. Repeat the process for 128 times, each with a different random seed for generating the training and test data sets above. Plot a scatter plot for $[E_{\text{out}}^{0/1}(\mathcal{A}(\mathcal{D}')), E_{\text{out}}^{0/1}(\mathcal{B}(\mathcal{D}'))]$. What is the median of $E_{\text{out}}^{0/1}(\mathcal{A}(\mathcal{D}'))$ and what is the median of $E_{\text{out}}^{0/1}(\mathcal{B}(\mathcal{D}'))$? Compare your results to the previous problem. Describe your findings.



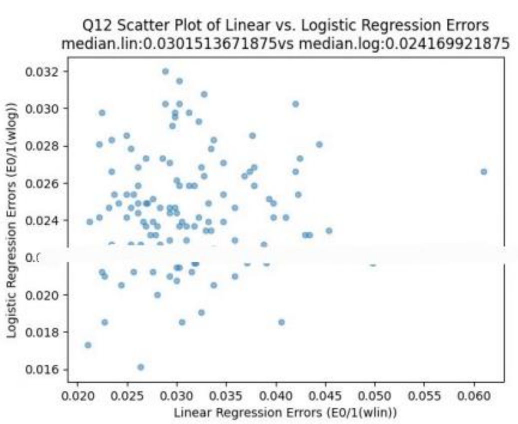
$E_{out}^{0/1}(A(D'))$: median.lin:0.0301513671875

$E_{out}^{0/1}(B(D'))$: median.log:0.035400390625

Median of linear regression algorithm becomes larger than training without outlier data, while median of logistic algorithms doesn't change much.

Since linear regression make error ^2, when we use noise in training data, it may sacrifice original best g, modify itself to reduce the error from noice when training.

The plot also shows that Eout(log) doesn't increase with Eout(Lin), but in most of cases Eout(log) is bigger than Eout(Lin), so I run another plot to see if iteration=20000 (which means it walks more steps to modify), Eout'(log) : 0.0241 the consequence indicates Eout(log) becomes smaller than Eout(Lin) .That is, if we increase iteration, logistic have the ability to perform better that linear.



Question assigned to the following page: 13

13. (Bonus 20 points) When using Newton's method for solving logistic regression, as discussed in Problem 6, each update $\mathbf{v}$ is calculated by

$$\mathbf{v} = -(\mathrm{X}^T\mathrm{D}\mathrm{X})^{-1}\nabla E_{\text{in}}(\mathbf{w}_t)$$

when $(\mathrm{X}^T\mathrm{D}\mathrm{X})$ is invertible. In linear regression, when $\mathrm{X}^T\mathrm{X}$ is invertible, the optimal

$$\mathbf{w}_{\text{LIN}} = (\mathrm{X}^T\mathrm{X})^{-1}\mathrm{X}^T\mathbf{y}.$$

If we can express $-\nabla E_{\text{in}}(\mathbf{w}_t)$ as some $\tilde{\mathrm{X}}^T\tilde{\mathbf{y}}$, and $\mathrm{X}^T\mathrm{D}\mathrm{X}$ as some $\tilde{\mathrm{X}}^T\tilde{\mathrm{X}}$, then each iteration of Newton-solving logistic regression is performing an internal linear regression! State the internal linear regression problem—in particular, what are $\tilde{\mathrm{X}}$ and $\tilde{\mathbf{y}}$?

13. $h_t(x) = \dfrac{1}{1+\exp(w^T x)}$, $D = \begin{bmatrix} h_t(x_1)(1-h_t(x_1)) & & & \\ & h_t(x_2)(1-h_t(x_2)) & & \\ & & \ddots & \\ & & & h_t(x_n)(1-h_t(x_n)) \end{bmatrix}$

① $X^T D X = X^T \sqrt{D} \cdot \sqrt{D}^T X$, $\tilde{X} = \sqrt{D} X$ #

② $-\nabla E_{in}(W_t) = \frac{1}{N}\sum_{n=1}^{N} y_n x_n (1-h_t(x_n)) = X^T [y(1-h_t(x))]$

$\frac{1}{N}(X^T\sqrt{D})(\sqrt{D}^{-1} y(1-h_t(x))) = \tilde{X}^T\tilde{y}$, $\tilde{X}^T y = X^T \sqrt{D}^T \tilde{y} = X^T \sqrt{D}\tilde{y}$

$\tilde{y} = \frac{1}{N}\sqrt{D}^{-1} y(1-h_t(x))$ #