HW3

1~5 bacea
6~10 babbb
11~15 eebdc
16~20 cbabd

B06504016
林家宏

1 $0.1^2\left(1-\frac{|H+1|}{N}\right) \geq 0.006 \Rightarrow N \geq 30 \Rightarrow$ Choose (b).

2 Let $\hat{y} = \text{proj}_{col.X}\, y$, $\hat{y}$ is in the column space of X.
Hence, $\hat{y} = Xw$.
We must can find an $\hat{y} \in$ column X such that $y - \hat{y}$ is orthogonal to column space of X. Therefore, $X^T(\hat{y}-y)=0 \Rightarrow X^T(Xw-y)=0 \Rightarrow X^TXw = X^Ty$. There exist at least one solution for this equation. $\Rightarrow$ Choose (a).

3
Let $X = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$ $H = \begin{bmatrix} 1 \\ 2 \end{bmatrix}\left([1\ \ 2]\begin{bmatrix} 1 \\ 2 \end{bmatrix}\right)^{-1}[1\ \ 2] = \frac{1}{5}\begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix}$

Let's look at (c), X will become $X' = \begin{bmatrix} 1 \\ 2\times\frac{1}{2} \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

$X' = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ $H' = \begin{bmatrix} 1 \\ 1 \end{bmatrix}\left([1\ \ 1]\begin{bmatrix} 1 \\ 1 \end{bmatrix}\right)^{-1}[1\ \ 1] = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \neq H$

Choose (c)

4 ① $\Pr(|v-\theta|>\varepsilon) \leq 2\exp(-2\varepsilon^2 N)$ for all $N \in \mathbb{N}$ and $\varepsilon > 0$.
Hoeffding's Inequality. $\Rightarrow$ True

② Likelihood $= f(\theta) = \theta^k(1-\theta)^{N-k}$, where k is the number of flipping result being head.

$f'(\theta) = \frac{k}{\theta}\theta^k(1-\theta)^{N-k} + \frac{k-N}{1-\theta}\theta^k(1-\theta)^{N-k}$

$= \left(\frac{k}{\theta} - \frac{N-k}{1-\theta}\right)f(\theta) = 0$

$\Rightarrow \frac{k}{\theta} = \frac{N-k}{1-\theta} \Rightarrow \theta = \frac{k}{N} = v \Rightarrow v$ maximize likelihood in $[0,1]$
$\Rightarrow$ True

③ $E_{in}'(\hat{y}) = \frac{1}{N}\sum_{n=1}^{N} 2(\hat{y}-y_n) = 2\left(\hat{y} - \sum_{i=1}^{N}\frac{y_n}{N}\right) = 2(\hat{y}-v) = 0$

$\Rightarrow \hat{y} = v \Rightarrow v$ minimize $E_{in}(\hat{y})$ over all $\hat{y} \in \mathbb{R}$. $\Rightarrow$ True

④
$-\nabla E_{in}(\hat{y}) = -2(\hat{y}-v) \Rightarrow -2(\hat{y}-v)\big|_{\hat{y}=0} = 2v$
$\Rightarrow 2v$ is the negative gradient direction $-\nabla E_{in}(\hat{y})$ at $\hat{y}=0$.
$\Rightarrow$ True $\Rightarrow$ 4 statements are true. $\Rightarrow$ Choose (e)

5 Uniform distribution $[0, \hat{\theta}]$.

If $\hat{\theta} \geq \max(y_1, y_2 \cdots y_N)$, p.d.f. $f_Y(y) = \frac{1}{\hat{\theta}}$

$P(Y_1 = y_1, Y_2 = y_2, \cdots Y_n = y_n) = f_Y(y_1) \times f_Y(y_2) \times \cdots f_Y(y_n) = \left(\frac{1}{\hat{\theta}}\right)^n \Rightarrow (a)$

6 Let's look at (b), $err(w, x, y) = \max(0, -y w^T x)$

If $y \neq sign(w^T x) \Rightarrow y w^T x > 0 \Rightarrow -\nabla err(w, x, y) = y x$

If $y \neq sign(w^T x) \Rightarrow y w^T x < 0 \Rightarrow -\nabla err(w, x, y) = 0$

The error function is consistent with the update rule.

The answer is (b).

7 $-\nabla err_{exp}(W, X_n, y_n) = -\frac{\partial err_{exp}(W, X_n, y_n)}{\partial W_i} = y_n X_n \exp(-y_n w^T X_n)$

$\Rightarrow$ Choose (a)

8 $E(w) \approx E(u) + b_E(u)^T V + \frac{1}{2} V^T A_E(u) V$, $\boxed{V = W - U}$

$\nabla E(w) = \frac{\partial E(w)}{\partial V} = b_E(u) + A_E(u) V = 0 \Rightarrow V = -(A_E(u))^{-1} b_E(u)$

$\Rightarrow$ Choose (b)

9 According to the slide of lecture 9

$\nabla E_{in}(w) = \frac{1}{N}(2AW - 2b)$

$\nabla(\nabla E_{in}(w)) = \nabla E(w) \times (\nabla E_{in}(w))^T = \nabla\left(\frac{2}{N}(w^T A^T - b^T)\right)$

$= \frac{2}{N} A^T = \frac{2}{N} X^T X \Rightarrow$ Choose (b)

10 $\frac{\partial err(w, x, y)}{\partial W_{ik}} = -\frac{1}{h_y(X)} \times \frac{1}{\left(\sum_{i=1}^{K} e^{w_i^T x}\right)^2} \times \left[[\![y=k]\!] \sum_{i=1}^{K} e^{w_i^T x} e^{w_k^T x} - x_i e^{w_k^T x} e^{w_y^T x} \right]$

$= \frac{-1}{h_y(x)} \left[[\![y=k]\!] x_i h_k(x) - x_i h_k(x) h_y(x) \right]$

$= (h_k(x) - [\![y=k]\!]) x_i \Rightarrow$ Choose (b)

CHA SHIN

11 $P(y=1|x) = \dfrac{e^{w_1^{*T}x}}{e^{w_1^{*T}x}+e^{w_2^{*T}x}} = 1-p$ $\quad p(y=2|x) = \dfrac{e^{w_2^{*T}x}}{e^{w_1^{*T}x}+e^{w_2^{*T}x}} = p$

$\qquad\qquad\qquad\qquad\qquad\qquad \overset{\shortparallel}{P(y'=-1|x)} \qquad\qquad\qquad\qquad\qquad\qquad \overset{\shortparallel}{P(y'=1|x)}$

$$\theta(w^Tx) = \frac{1}{1+e^{-w^Tx}} = p = \frac{1}{1+e^{(w_1^{*T}-w_2^{*T})x}}$$

$\Rightarrow w = w_2^{*T} - w_1^{*T}$

13 $\left(\bigcup_{k=1}^{d} H_k\right) \Rightarrow 2dN$ seperations

These seperations have to shatter 2^N result

$$2^N \leq 2dN \Rightarrow N \leq \log_2 2dN = 1+\log_2 d + \log_2 N \leq 1+\frac{N}{2}+\log_2 d$$

when $N \geq 4$

$\Rightarrow \frac{N}{2} \leq 1+\log_2 d \Rightarrow N \leq 2(1+\log_2 d) \Rightarrow$ Choose (b)

CHA SHIN

14. Please use “python hw3_14.py hw3_train.txt” to execute.

```
import sys
import numpy as np
from numpy.linalg import inv
from numpy.linalg import multi_dot
in_filename = sys.argv[1]
fin = open(in_filename, 'r')
data = np.loadtxt(fin)
padding = np.full((1000, 1), 1)
fin.close()
data = np.append(padding, data, axis = 1)

x = data[0:1000 , :11]
y = data[0:1000 , 11:12]
xt = x.transpose()
xr = np.dot(xt, x)
xin = inv(xr)
Wlin = multi_dot([xin, xt, y])
reg = np.dot(x, Wlin)
err = reg - y
error = np.vdot(err, err)/1000
print(error)
```

15. Please use “python hw3_15.py hw3_train.txt” to execute.

```
import sys
import numpy as np
import random
import math
from numpy.linalg import inv
from numpy.linalg import multi_dot
in_filename = sys.argv[1]
fin = open(in_filename, 'r')
data = np.loadtxt(fin)
padding = np.full((1000, 1), 1)
fin.close()
data = np.append(padding, data, axis = 1)
x = data[0:1000 , :11]
y = data[0:1000 , 11:12]
xt = x.transpose()
```

```
xr = np.dot(xt, x)
xin = inv(xr)
Wlin = multi_dot([xin, xt, y])
reg = np.dot(x, Wlin)
err = reg - y
error = np.vdot(err, err)/1000
w = np.zeros((1, 11))
def SGDRegression(w, x, y):
    j = 0
    e = 1
    while e > 1.01 * error:
        k = random.randint(0, 999)
        xn = x[k, :11]
        yn = y[k]
        check = np.vdot(w, xn)
        w = w + 0.001 * 2 * (yn - check) * xn
        j = j+1
        r = np.dot(x, w.transpose())
        er = r - y
        e = np.vdot(er, er)/1000
    return j
iternum = 0
for i in range(1000):
    num = SGDRegression(w, x, y)
    iternum = iternum + num
print(iternum/1000)
```

16. Please use “python hw3_16.py hw3_train.txt” to execute.

```
import sys
import numpy as np
import random
import math
from numpy.linalg import inv
from numpy.linalg import multi_dot
in_filename = sys.argv[1]
fin = open(in_filename, 'r')
data = np.loadtxt(fin)
padding = np.full((1000, 1), 1)
fin.close()
```

```
data = np.append(padding, data, axis = 1)
x = data[0:1000 , :11]
y = data[0:1000 , 11:12]
w = np.zeros((1, 11))
def SGDLogistic(w, x, y):
    e = 0
    for i in range(500):
        k = random.randint(0, 999)
        xn = x[k, :11]
        yn = y[k]
        check = np.vdot(w, xn)
        w = w + 0.001 * yn * xn * (1 / (1 + math.exp(yn * check)))
    for i in range(1000):
        xn = x[i, :11]
        yn = y[i]
        e = e + np.log(((1 + math.exp(-yn * np.vdot(w, xn)))))
    return e/1000
aver = 0
for i in range(1000):
    num = SGDLogistic(w, x, y)
    aver = aver + num
print(aver/1000)
```

17. Please use “python hw3_17.py hw3_train.txt” to execute.

```
import sys
import numpy as np
import random
import math
from numpy.linalg import inv
from numpy.linalg import multi_dot
in_filename = sys.argv[1]
fin = open(in_filename, 'r')
data = np.loadtxt(fin)
padding = np.full((1000, 1), 1)
fin.close()
data = np.append(padding, data, axis = 1)
x = data[0:1000 , :11]
y = data[0:1000 , 11:12]
xt = x.transpose()
```

```
xr = np.dot(xt, x)
xin = inv(xr)
Wlin = multi_dot([xin, xt, y])
w = Wlin.transpose()
def SGDLogistic(w, x, y):
    e = 0
    for i in range(500):
        k = random.randint(0, 999)
        xn = x[k, :11]
        yn = y[k]
        check = np.vdot(w, xn)
        w = w + 0.001 * yn * xn * (1 / (1 + math.exp(yn * check)))
    for i in range(1000):
        xn = x[i, :11]
        yn = y[i]
        e = e + np.log(((1 + math.exp(-yn * np.vdot(w, xn)))))
    return e/1000
aver = 0
for i in range(1000):
    num = SGDLogistic(w, x, y)
    aver = aver + num
print(aver/1000)
```

18. Please use “python hw3_18.py hw3_train.txt hw3_test.txt” to execute.

```
import sys
import numpy as np
from numpy.linalg import inv
from numpy.linalg import multi_dot
in_filename1 = sys.argv[1]
in_filename2 = sys.argv[2]
fin = open(in_filename1, 'r')
data = np.loadtxt(fin)
padding1 = np.full((1000, 1), 1)
padding2 = np.full((3000, 1), 1)
fin.close()
fin = open(in_filename2, 'r')
test = np.loadtxt(fin)
data = np.append(padding1, data, axis = 1)
test = np.append(padding2, test, axis = 1)
```

```
x = data[0:1000 , :11]
y = data[0:1000 , 11:12]
xt = x.transpose()
xr = np.dot(xt, x)
xin = inv(xr)
Wlin = multi_dot([xin, xt, y])
xtest = test[0:3000 , :11]
ytest = test[0:3000 , 11:12]
reg1 = np.dot(x, Wlin)
reg2 = np.dot(xtest, Wlin)
err1 = 0
err2 = 0
for i in range(1000):
    if reg1[i][0] * y[i][0] <0:
        err1 = err1 + 1
for i in range(3000):
    if reg2[i][0] * ytest[i][0] <0:
        err2 = err2 + 1
print(abs(err1-err2/3)/1000)
fin.close()
```

19. Please use “python hw3_19.py hw3_train.txt hw3_test.txt” to execute.

```
import sys
import numpy as np
from numpy.linalg import inv
from numpy.linalg import multi_dot
def Qtransform(A, Q):
    a = A[0:A.shape[0], 0:1]
    b = A[0:A.shape[0], 1:11]
    c = A[0:A.shape[0], 11:12]
    for i in range(Q):
        paste = np.power(b, i+1)
        a = np.concatenate((a, paste), axis=1)
    a = np.concatenate((a, c), axis=1)
    return a
in_filename1 = sys.argv[1]
in_filename2 = sys.argv[2]
fin = open(in_filename1, 'r')
```

```
data = np.loadtxt(fin)
padding1 = np.full((1000, 1), 1)
padding2 = np.full((3000, 1), 1)
fin.close()
fin = open(in_filename2, 'r')
test = np.loadtxt(fin)
data = np.append(padding1, data, axis = 1)
test = np.append(padding2, test, axis = 1)
data = Qtransform(data, 3)
test = Qtransform(test, 3)
x = data[0:1000 , 0:31]
y = data[0:1000 , 31:32]
xt = x.transpose()
xr = np.dot(xt, x)
xin = inv(xr)
Wlin = multi_dot([xin, xt, y])
xtest = test[0:3000 , 0:31]
ytest = test[0:3000 , 31:32]
reg1 = np.dot(x, Wlin)
reg2 = np.dot(xtest, Wlin)
err1 = 0
err2 = 0
for i in range(1000):
    if reg1[i][0] * y[i][0] <0:
        err1 = err1 + 1
for i in range(3000):
    if reg2[i][0] * ytest[i][0] <0:
        err2 = err2 + 1
print(abs(err1-err2/3)/1000)
fin.close()
```

20. Please use “python hw3_20.py hw3_train.txt hw3_test.txt” to execute.

```
import sys
import numpy as np
from numpy.linalg import inv
from numpy.linalg import multi_dot
def Qtransform(A, Q):
    a = A[0:A.shape[0], 0:1]
    b = A[0:A.shape[0], 1:11]
```

```
    c = A[0:A.shape[0], 11:12]
    for i in range(Q):
        paste = np.power(b, i+1)
        a = np.concatenate((a, paste), axis=1)
    a = np.concatenate((a, c), axis=1)
    return a
in_filename1 = sys.argv[1]
in_filename2 = sys.argv[2]
fin = open(in_filename1, 'r')
data = np.loadtxt(fin)
padding1 = np.full((1000, 1), 1)
padding2 = np.full((3000, 1), 1)
fin.close()
fin = open(in_filename2, 'r')
test = np.loadtxt(fin)
data = np.append(padding1, data, axis = 1)
test = np.append(padding2, test, axis = 1)
data = Qtransform(data, 10)
test = Qtransform(test, 10)
x = data[0:1000 , 0:101]
y = data[0:1000 , 101:102]
xt = x.transpose()
xr = np.dot(xt, x)
xin = inv(xr)
Wlin = multi_dot([xin, xt, y])
xtest = test[0:3000 , 0:101]
ytest = test[0:3000 , 101:102]
reg1 = np.dot(x, Wlin)
reg2 = np.dot(xtest, Wlin)
err1 = 0
err2 = 0
for i in range(1000):
    if reg1[i][0] * y[i][0] <0:
        err1 = err1 + 1
for i in range(3000):
    if reg2[i][0] * ytest[i][0] <0:
        err2 = err2 + 1
print(abs(err1-err2/3)/1000)
```

```
fin.close()
```