CSIE 5432 — Machine Learning Foundations

Name: 李吉昌 **Homework 3**

Student Number: r08922a27 **Due Date:** November 20 2020, 13:00

Linear Regression

1. Answer: [b]

代入 $\sigma = 0.1, d = 11$, 則 $\mathbb{E}_{\mathcal{D}}[E_{in}(\mathbf{W_{lin}})] = 0.01 \cdot (1 - \frac{12}{N})$, 代入不同 N 可得下列表格:

N	25	30	35	40	45
$\mathbb{E}_{\mathcal{D}}[E_{in}(\mathbf{W_{lin}})]$	0.0052	0.006	0.0066	0.007	0.0073

$N = 30$ 為最小值使得 $\mathbb{E}_{\mathcal{D}}[E_{in}(\mathbf{W_{lin}})]$ 不少於 0.006。

2. Answer: [a]

根據 Lecture 9 slides 第 11 頁, $X^TXw = X^T\mathbf{y}$, 當 $X^TXw = 0$ 時, $w^TX^TXw = \|Xw\|^2 = 0$, 則 $Xw = 0$ 必成立, 得 $\mathcal{N}(X^TX) \subseteq \mathcal{N}(X^T)$; 當 $Xw = 0$ 時, $X^T(Xw) = X^T(0) = 0$, 則 $X^TXw = 0$ 必成立, 得 $\mathcal{N}(X^T) \subseteq \mathcal{N}(X^TX)$。綜上所述, 因為 $\mathcal{N}(X^T) = \mathcal{N}(X^TX)$, 因此可推至 $\mathcal{C}(X^T) = \mathcal{C}(X^TX)$, $\mathcal{C}(X^T)$ 的任一向量 $\mathbf{X^Ty}$, 一定可以找到對應的 w 使得 $X^TXw = X^T\mathbf{y}$, 當 X^TX 可逆時, 表示 w 存在唯一解; 當 X^TX 不可逆時, w 存在多組解。
根據 Lecture 9 slides 第 15 頁, 當 w 有解時, $E_{in}(\mathbf{w}) = \frac{1}{N}\|(I - XX^\dagger)\mathbf{y}\| \neq 0$, 故僅有「There exists at least one solution for the normal equation.」敘述正確。

3. Answer: [c]

H 為 X 的 column space 的投影矩陣, $H\mathbf{y}$ 為 X 的 column space 中離 $\mathbf{y}$ 最近的點, [a], [b] 和 [d] 僅對 column 做 scaling 的運算, 展開的空間是一樣的, 不影響 $\mathbf{y}$ 投影到 column space 的結果。令 $X = \begin{pmatrix} 1 & 1 \\ 4 & 2 \\ 0 & 3 \end{pmatrix}$, 經過 [c] 的運算得 $X' = \begin{pmatrix} 1 & 1 \\ 2 & 1 \\ 0 & 1 \end{pmatrix}$, $X(X^TX)^{-1}X^T = \begin{pmatrix} \frac{13}{157} & \frac{36}{157} & \frac{24}{157} \\ \frac{36}{157} & \frac{148}{157} & \frac{-6}{157} \\ \frac{24}{157} & \frac{-6}{157} & \frac{153}{157} \end{pmatrix}$, $X'(X'^TX')^{-1}X'^T = \begin{pmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{5}{6} & \frac{-1}{6} \\ \frac{1}{3} & \frac{-1}{6} & \frac{5}{6} \end{pmatrix}$, 運算前後 H 結果不相同, [c] 會改變 H 結果。

Likelihood and Maximum Likelihood

4. Answer: [e]

(1) 因為 head 代表的 y_n 為 1, 另外一面是 0, $\nu = \frac{1}{N}\sum_{n=1}^N y_n = \frac{1}{N}\sum_{n=1}^N [\![y_n = 1]\!]$, ν 亦為 sample 到 head 的 fraction, 描述正確。

(2) 由題意得, likelihood$(\hat{\theta}) = \prod_{n=1}^N \hat{\theta}^{y_n}(1-\hat{\theta})^{1-y_n} = \hat{\theta}^{\sum_{n=1}^N y_n}(1-\hat{\theta})^{N-\sum_{n=1}^N y_n}$,
令 $L = \hat{\theta}^{\nu N}(1-\hat{\theta})^{N-\nu N}$, 對 $\hat{\theta}$ 取微分為 0:

$\frac{d}{d\hat{\theta}}\{\text{likelihood}(\hat{\theta})\} = \nu N \cdot \hat{\theta}^{\nu N-1}(1-\hat{\theta})^{N-\nu N} + (N-\nu N)\hat{\theta}^{\nu N}(1-\hat{\theta})^{N-\nu N-1} = 0,$

$\nu N \cdot \frac{L}{\hat{\theta}} = (\nu N - N) \cdot \frac{L}{1-\hat{\theta}}$, 同除以 L, N 並移項整理得, $\hat{\theta} = \nu$
當 $\hat{\theta} = \nu$ 時, likelihood$(\hat{\theta})$ 為極大值, 描述正確。

(3) $\nabla E_{in}(\hat{y}) = \frac{d}{d\hat{y}}\{\frac{1}{N}\sum_{n=1}^{N}(\hat{y}-y_n)^2\} = \frac{1}{N}\sum_{n=1}^{N} 2\cdot(\hat{y}-y_n) = 2\cdot(\hat{y}-\nu) = 0$, 得 $\hat{y} = \nu$,
當 $\hat{y} = \nu$ 時, $E_{in}(\hat{y})$ 為極小值, 描述正確。

(4) 由 (3) 得 $\nabla E_{in}(\hat{y}) = 2\cdot(\hat{y}-\nu)$, 代入 $\hat{y} = 0$, $\nabla E_{in}(\hat{y}) = -2\nu$, $2\nu = -\nabla E_{in}(\hat{y})$, 描述正確。

5. Answer: [a]

任意 y_n 的機率為 $P(y_n|\theta) = \frac{1}{\theta}$, 因為 θ 未知, 令 $\hat{\theta}$ 為估計 θ 的隨機變數。根據 likelihood 定義, N 筆 sample 形成的 likelihood 為 $\prod_{n=1}^{N} P(y_n|\hat{\theta}) = (\frac{1}{\hat{\theta}})^N$。

Gradient and Stochastic Gradient Descent

6. Answer: [b]

在 point-wise 的情況, 參數更新式可寫作 $\mathbf{w_{t+1}} = \mathbf{w_t} + \eta \cdot [\![y_n \neq \text{sign}(\mathbf{w_t}^T\mathbf{x_n})]\!] y_n\mathbf{x_n}$, 根據 Lecture 10 slides 第 33 頁, $\nabla err(\mathbf{w}, \mathbf{x}, y) = -[\![y_n \neq \text{sign}(\mathbf{w_t}^T\mathbf{x_n})]\!] y_n\mathbf{x_n}$。
$\max(0, -y\mathbf{w}^T\mathbf{x})$ 在 $y = \text{sign}(\mathbf{w}^T\mathbf{x}$ 的情況, 數值為 0, 則 $\nabla err(\mathbf{w}, \mathbf{x}, y) = 0$; 在 $y \neq \text{sign}(\mathbf{w}^T\mathbf{x})$ 的情況, 數值為 $-y\mathbf{w}^T\mathbf{x}$, 則 $\nabla err(\mathbf{w}, \mathbf{x}, y) = \frac{d}{d\mathbf{w}}\{-y\mathbf{w}^T\mathbf{x}\} = -y\frac{d}{d\mathbf{w}}\{\mathbf{w}^T\mathbf{x}\} = -y\mathbf{x}$。
綜上所述, $\nabla err(\mathbf{w}, \mathbf{x}, y) = -[\![y_n \neq \text{sign}(\mathbf{w_t}^T\mathbf{x_n})]\!] y_n\mathbf{x_n} = \nabla \max(0, -y\mathbf{w}^T\mathbf{x})$, 因為微分結果相同, $\max(0, -y\mathbf{w}^T\mathbf{x})$ 可作為 $err(\mathbf{w}, \mathbf{x}, y)$。

7. Answer: [a]

$$
\begin{aligned}
-\nabla err_{exp}(\mathbf{w}, \mathbf{x_n}, y_n) &= -\frac{d}{d\mathbf{w}}\{\exp(-y_n\mathbf{w}^T\mathbf{x_n})\} = -\exp(-y_n\mathbf{w}^T\mathbf{x_n})\frac{d}{d\mathbf{w}}\{-y_n\mathbf{w}^T\mathbf{x_n}\} \\
&= \exp(-y_n\mathbf{w}^T\mathbf{x_n})y_n\frac{d}{d\mathbf{w}}\{\mathbf{w}^T\mathbf{x_n}\} = y_n\mathbf{x_n}\exp(-y_n\mathbf{w}^T\mathbf{x_n})
\end{aligned}
$$

8. Answer: [b]

代入 $\mathbf{w} = \mathbf{u} + \mathbf{v}$, $E(\mathbf{w}) = E(\mathbf{u} + \mathbf{v}) \approx E(\mathbf{u}) + \mathbf{b}_E(\mathbf{u})^T\mathbf{v} + \frac{1}{2}\mathbf{v}^T\mathbf{A}_E(\mathbf{u})\mathbf{v}$

$$
\begin{aligned}
\frac{\partial}{\partial\mathbf{v}}\{E(\mathbf{u}) + \mathbf{b}_E(\mathbf{u})^T\mathbf{v} + \frac{1}{2}\mathbf{v}^T\mathbf{A}_E(\mathbf{u})\mathbf{v}\} &= 0 + \frac{\partial}{\partial\mathbf{v}}\{\mathbf{b}_E(\mathbf{u})^T\mathbf{v}\} + \frac{\partial}{\partial\mathbf{v}}\{\frac{1}{2}\mathbf{v}^T(\mathbf{A}_E(\mathbf{u})\mathbf{v})\} \\
&= \mathbf{b}_E(\mathbf{u}) + \frac{1}{2}(\frac{\partial}{\partial\mathbf{v}}\{\mathbf{v}\}\mathbf{A}_E(\mathbf{u})\mathbf{v} + \frac{\partial}{\partial\mathbf{v}}\{\mathbf{A}_E(\mathbf{u})\mathbf{v}\}\mathbf{v}) \\
&= \mathbf{b}_E(\mathbf{u}) + \frac{1}{2}(\mathbf{A}_E(\mathbf{u}) + \mathbf{A}_E(\mathbf{u})^T)\mathbf{v}, \\
&\mathbf{A}_E(\mathbf{u}) \text{ 為正定矩陣, } \mathbf{A}_E(\mathbf{u}) \text{ 對稱性為必要條件, } \mathbf{A}_E = \mathbf{A}_E^T \\
&= \mathbf{b}_E(\mathbf{u}) + \mathbf{A}_E(\mathbf{u})\mathbf{v}
\end{aligned}
$$

當 $\nabla_\mathbf{v}E(\mathbf{w}) = 0$ 時, $-\mathbf{b}_E(\mathbf{u}) = \mathbf{A}_E(\mathbf{u})\mathbf{v}$, 得 $\mathbf{v} = -\mathbf{A}_E(\mathbf{u})^{-1}\mathbf{b}_E(\mathbf{u})$。

9. Answer: [b]

$$\nabla_{\mathbf{w_t}} E(\mathbf{w_t}) \approx \frac{\partial}{\partial \mathbf{w_t}}\{E(\mathbf{u}) + \mathbf{b}_E(\mathbf{u})^T\mathbf{w_t} - \mathbf{b}_E(\mathbf{u})^T\mathbf{v} + \frac{1}{2}[\mathbf{w_t}^T\mathbf{A}_E(\mathbf{u})\mathbf{w_t} + \mathbf{v}^T\mathbf{A}_E(\mathbf{u})\mathbf{v}]\}$$

$$= \frac{\partial}{\partial \mathbf{w_t}}\{\mathbf{b}_E(\mathbf{u})^T\mathbf{w_t}\} + \frac{\partial}{\partial \mathbf{w_t}}\{\frac{1}{2}\mathbf{w_t}^T\mathbf{A}_E(\mathbf{u})\mathbf{w_t}\}$$,微分過程和上題一樣, 可得下面結果,

$$= \mathbf{b}_E(\mathbf{u}) + \mathbf{A}_E(\mathbf{u})\mathbf{w_t}$$

$\mathbf{A}_E(\mathbf{w_t}) = \nabla_{\mathbf{w_t}}[\nabla_{\mathbf{w_t}} E(\mathbf{w_t})] \approx \frac{\partial}{\partial \mathbf{w_t}}\{\mathbf{b}_E(\mathbf{u}) + \mathbf{A}_E(\mathbf{u})\mathbf{w_t}\} = 0 + \mathbf{A}_E(\mathbf{u})^T = \mathbf{A}_E(\mathbf{u})$ [$\because$ 對稱性]

得 $\mathbf{A}_E(\mathbf{w_t}) = \mathbf{A}_E(\mathbf{u})$, 其中 $\mathbf{u}$ 為更新前的參數點, $\mathbf{u} = \mathbf{w_{t-1}}$, 則 $\mathbf{A}_E(\mathbf{w_t}) = \mathbf{A}_E(\mathbf{w_{t-1}})$。由上式可知每次參數更新時 $\mathbf{A}_E$ 不會被參數變化而影響。

求取 linear regression 的 $\mathbf{A}_E$ 無需考慮時間點, 任一參數 $\mathbf{w}$ 求得的 $\mathbf{A}_E(\mathbf{w})$ 和 Newton's method 得到的 $\mathbf{A}_E(\mathbf{w_t})$ 相同。根據 Lecture 9 slides 的第 10 頁, $\nabla_w E_{in}(\mathbf{w}) = \frac{2}{N}(X^TX\mathbf{w} - X^T\mathbf{y})$, $\mathbf{A}_E(\mathbf{w}) = \nabla_w[\nabla_w E_{in}(\mathbf{w})] = \frac{d}{d\mathbf{w}}\{\frac{2}{N}(X^TX\mathbf{w} - X^T\mathbf{y})\} = \frac{2}{N}\frac{d}{d\mathbf{w}}\{((X^TX)\mathbf{w}\} + 0 = \frac{2}{N}(X^TX)^T = \frac{2}{N}X^TX$

Multinomial Logistic Regression

10. Answer: [b]

$$\frac{\partial}{\partial W_{ik}}\{err(W,\mathbf{x},y)\} = \frac{\partial}{\partial W_{ik}}\{-[\![y=k]\!]\ln h_k(\mathbf{x})\} = -[\![y=k]\!]\frac{\partial}{\partial W_{ik}}\{\mathbf{w_k}^T\mathbf{x} - ln(\sum_{i=1}^{K}\exp{(\mathbf{w_i}^T\mathbf{x})})\}$$

$$= -([\![y=k]\!]x_i - \frac{1}{\sum_{i=1}^{K}\exp{(\mathbf{w_i}^T\mathbf{x})}}\frac{\partial}{\partial W_{ik}}\{\sum_{i=1}^{K}\exp{(\mathbf{w_i}^T\mathbf{x})}\})$$

$$= -([\![y=k]\!]x_i - \frac{\exp{(\mathbf{w_k}^T\mathbf{x})}}{\sum_{i=1}^{K}\exp{(\mathbf{w_i}^T\mathbf{x})}}x_i) = (h_k(\mathbf{x}) - [\![y=k]\!])x_i$$

11. Answer: [e]

當 $y_n = 2, y'_n = +1$ 時, $h_2(\mathbf{x_n}) = \dfrac{\exp{(\mathbf{w_2}^{*T}\mathbf{x})}}{\exp{(\mathbf{w_1}^{*T}\mathbf{x})} + \exp{(\mathbf{w_2}^{*T}\mathbf{x})}} = \dfrac{1}{\exp{(-(\mathbf{w_2}^* - \mathbf{w_1}^*)^T\mathbf{x})} + 1}$;

當 $y_n = 1, y'_n = -1$ 時, $h_1(\mathbf{x_n}) = \dfrac{\exp{(\mathbf{w_1}^{*T}\mathbf{x})}}{\exp{(\mathbf{w_1}^{*T}\mathbf{x})} + \exp{(\mathbf{w_2}^{*T}\mathbf{x})}} = \dfrac{\exp{(-(\mathbf{w_2}^* - \mathbf{w_1}^*)^T\mathbf{x})}}{\exp{(-(\mathbf{w_2}^* - \mathbf{w_1}^*)^T\mathbf{x})} + 1}$

$$= 1 - \frac{1}{\exp{(-(\mathbf{w_2}^* - \mathbf{w_1}^*)^T\mathbf{x})} + 1} = 1 - h_2(\mathbf{x_n})$$

令 $\mathbf{w_2}^* - \mathbf{w_1}^*$ 為 binary logistic regression 之解, $\nabla E_{in}(\mathbf{w_2}^* - \mathbf{w_1}^*) = \frac{1}{N}\sum_{n=1}^{N}\theta(-y_n(\mathbf{w_2}^* - \mathbf{w_1}^*)^T\mathbf{x_n})(-y_n\mathbf{x_n})$。

當 $N = 1, y'_n = +1$ 時,
$\nabla E_{in}(\mathbf{w_2}^* - \mathbf{w_1}^*) = \theta(-(\mathbf{w_2}^* - \mathbf{w_1}^*)^T\mathbf{x_n})(-\mathbf{x_n}) = h_1(\mathbf{x_n})(-\mathbf{x_n})|_{\mathbf{w_1}^*,\mathbf{w_2}^*}$。
在 $y'_n = +1$ 的條件下, $\mathbf{w_1}^*$ 為最佳解, 使得 $h_1(\mathbf{x_n}) = 0$, 則 $\nabla E_{in}(\mathbf{w_2}^* - \mathbf{w_1}^*) = 0 \cdot (-\mathbf{x_n}) = 0$。

當 $N = 1, y'_n = -1$ 時,
$\nabla E_{in}(\mathbf{w_2}^* - \mathbf{w_1}^*) = \theta((\mathbf{w_2}^* - \mathbf{w_1}^*)^T\mathbf{x_n})(\mathbf{x_n}) = h_2(\mathbf{x_n})(\mathbf{x_n})|_{\mathbf{w_1}^*,\mathbf{w_2}^*}$。
在 $y'_n = -1$ 的條件下, $\mathbf{w_2}^*$ 為最佳解, 使得 $h_2(\mathbf{x_n}) = 0$, 則 $\nabla E_{in}(\mathbf{w_2}^* - \mathbf{w_1}^*) = 0 \cdot (\mathbf{x_n}) = 0$。

綜上所述, 在所有 y'_n 的情況下, $\nabla E_{in}(\mathbf{w_2}^* - \mathbf{w_1}^*) = \frac{1}{N}\sum_{n=1}^{N} 0 = 0$, 得 $\mathbf{w_2}^* - \mathbf{w_1}^*$ 為 binary logistic regression 之最佳解。

Nonlinear Transformation

12. Answer: [e]

choice	[a]	[b]	[c]	[d]	[e]
$err_{0/1}$	0.1429	0.1429	0.4286	0.5714	0

13. Answer: [b]

令 $\mathcal{H}_k$ 的參數為 (w_0, w_k), boundary 等式為 $w_0 + w_k \cdot x_k = 0$, 分類的門檻值為 $x_k = -\frac{w_0}{w_k}$, 若資料有 N 筆, 在第 k 維度有 $N-1$ 個間隔可以插入, 等價於第 k 維度上 Decision Stump 的 hypothesis, 不討論全為 positive/negative 的 2 種情況, 間隔數量 $N-1$ 個以及考慮對稱性數量為兩倍, 得 dichotomy 的數量最多可以有 $2(N-1)$ 個, 令 $\cup_{k=1}^{d}\mathcal{H}_k$ 的 growth function 為 $m_{\mathcal{H}}(N)$, 如果 d 個維度的 $\cup_{k=1}^{d}\mathcal{H}_k$ 的交集只有全為 positive/negative 的 2 種情況, 則 dichotomy 的數量為 $2(N-1)\cdot d+2$, 若有其他交集的可能 dichotomy 的數量只會變小, 得 $m_{\mathcal{H}}(N) \le 2(N-1)\cdot d+2$。
選項大小排序為 $2(d^2+1) \ge 2(d\log_2 d+1) \ge 2(d+1) \ge 2(\log_2 d+1) \ge 2(\log_2\log_2 d+1)$, 若右方為 VC upper bound, 左方必為更寬鬆的 VC upper bound, 最右邊是 VC upper bound 的選項即為 tightest upper bound。

當 $N = 2(\log_2\log_2 d+1)$ 時:

每個維度至少存在一邊界可以得到 2 種 dichotomy, 則 d 個維度至少會有 2d 種 dichotomy, 考慮全為 positive/negative 的 2 種情況, $m_{\mathcal{H}}(N) \ge 2d+2 > 2d$, 可以確定 $d_{vc} > \log_2(2d)$。若 $2(\log_2\log_2 d+1) < \log_2(2d)$ 恆成立, 代表 $2(\log_2\log_2 d+1)$ 必小於 d_{vc}。

$2(\log_2\log_2 d+1) < \log_2(2d), 2\log_2\log_2 d - \log_2 d + 1 < 0,$ 令$t = \log_2 d,$

$\frac{d}{dt}\{2\log_2 t - t + 1\} = \frac{1}{\ln 2}\cdot(\frac{2}{t}-1),$ 令 $\frac{1}{\ln 2}\cdot(\frac{2}{t}-1) \le 0,$ 則 $\frac{2}{t}-1 \le 0,$

得 $t \ge 2$ 時, $d \ge 4$, $2\log_2\log_2 d - \log_2 d + 1$ 為嚴格遞減,

因此, 在 $d \ge 5$ 時, $2\log_2\log_2 d - \log_2 d + 1 < 0$ 恆成立。

在 $d \ge 5$ 時, $2\log_2\log_2 d - \log_2 d + 1 < 0$ 恆成立, 即 $2(\log_2\log_2 d+1) < d_{vc}$, 得 $2(\log_2\log_2 d+1)$ 並非 VC upper bound。

當 $N = 2(\log_2 d+1)$ 時:

根據 Lecture 7 slides 第 5 頁, 若 N 為 d_vc 的 upper bound, 則必符合 $m_{\mathcal{H}}(N) \le 2^N$。

$$\begin{cases} 2^N = 2^{2(\log_2 d+1)} = 4d^2 \\ m_{\mathcal{H}}(N) = 2(2(\log_2 d+1)-1)\cdot d+2 = 4(\log_2 d)\cdot d+2d+2 \end{cases}$$

$4(\log_2 d)\cdot d+2d+2 \le 4d^2,$ 得 $2(\log_2 d)\cdot d+d-2d^2+1 \le 0$

由題意, 當$d \ge 4$ 時, $\log_2 d \le \frac{d}{2},$ 則 $2(\log_2 d)\cdot d+d-2d^2+1 \le d-d^2+1$

$\begin{cases} d-d^2+1 \text{ 為圓弧向下, 當 } d \ge \frac{1}{2} \text{ 時, 為嚴格遞減} \\ \text{當 } d=4 \text{ 時, } d-d^2+1 = -11(<0) \end{cases}$ 得 $d \ge 4$ 時, $d-2d^2+1 \le 0$ 皆成立。

$\begin{cases} \text{當 } d \ge 4 \text{ 時, } d-2d^2+1 \le 0 \\ d-2d^2+1 \ge 2(\log_2 d)\cdot d+d-2d^2+1 \end{cases}$ 則 $d \ge 4$ 時, $2(\log_2 d)\cdot d+d-2d^2+1 \le 0$ 必成立,
得 $2(\log_2 d+1)$ 為 tightest upper bound。

Experiment

程式碼實作細節如下, 可以透過 parser 的 −−tra_path/−−tst_path 設置訓練資料和測試資料的路徑

```
python code.py --tra_path hw3_train.dat --tst_path hw3_test.dat
```

```python
import numpy as np
import random
import argparse

'''Define Function'''


def get_data(path, bias=1.0, transform=None):
    X = []
    for x in open(path, 'r'):
        x = x.strip().split('\t')
        x = [float(v) for v in x]
        X.append([bias] + x)

    X = np.array(X)
    X, Y = np.array(X[:, :-1]), np.array(X[:, -1])

    if transform is not None:
        X = transform(X)

    return X, Y


def get_wLIN(X, Y):
    X_plus = np.matmul(np.linalg.inv(np.matmul(X.T, X)), X.T)
    return np.matmul(X_plus, Y)


def sigmoid(s):
    return 1 / (1 + np.exp(-s))


def sign(s):
    s = np.sign(s)
    s[s == 0] = -1
    return s


def Q_transform(X, Q=3):
    return np.hstack([X]+[X[:, 1:]**q for q in range(2, Q+1)])


def err(w, X, Y, mode='sqr'):
    Y_pred = np.matmul(X, w)
```

```
    if mode == 'sqr':
        return ((Y_pred - Y)**2).mean()
    elif mode == 'ce':
        return -np.log(sigmoid(Y * Y_pred)).mean()
    elif mode == '0/1':
        Y_pred = sign(Y_pred)
        return (Y.astype(int) != Y_pred.astype(int)).mean()


def SGD(X, Y, lr, w_init=None, step_num=1000000, mode='sqr'):
    def random_pick(X, Y):
        idx = random.randint(0, X.shape[0] - 1)
        return X[idx:idx+1], Y[idx:idx+1]

    def grad_func(w, X, Y, mode):
        batch_size = X.shape[0]
        if mode == 'sqr':
            return -(2 / batch_size) * np.matmul(X.T, np.matmul(X, w) - Y)
        elif mode == 'ce':
            return np.mean(sigmoid(-Y * np.matmul(X, w)).reshape(-1, 1) * (Y.reshape(-1, 1) * X), axi

    def update_w(w, x, y, lr):
        return w + lr * grad_func(w, x, y, mode)

    if mode == 'sqr':
        wLIN = get_wLIN(X, Y)
        E_in_sqr_LIN = err(wLIN, X, Y, mode='sqr')

    # initialization
    step = 0
    w = np.zeros(X.shape[1:]) if w_init is None else w_init

    # training
    while step < step_num:
        x, y = random_pick(X, Y)
        w = update_w(w, x, y, lr)
        step += 1

        # check early stopping
        if mode == 'sqr':
            E_in_sqr = err(w, X, Y, mode='sqr')
            if E_in_sqr <= 1.01 * E_in_sqr_LIN:
                break
    return w, step


def main():
    '''Parsing'''
    parser = argparse.ArgumentParser(
        description='Argument Parser for MLF HW3.')
```

```python
parser.add_argument('--tra_path', default='hw3_train.dat')
parser.add_argument('--tst_path', default='hw3_test.dat')
args = parser.parse_args()

# load data
X_tra, Y_tra = get_data(args.tra_path)
X_tst, Y_tst = get_data(args.tst_path)

'''Answer questions'''
print('RUNNING Q14...')
wLIN = get_wLIN(X_tra, Y_tra)
print('Answer of Q14 : {:.4f}\n'.format(
    err(wLIN, X_tra, Y_tra, mode='sqr')))

print('RUNNING Q15...')
update_num_list = []
for _ in range(1000):
    _, update_num = SGD(X_tra, Y_tra, lr=0.001)
    update_num_list.append(update_num)
print('Answer of Q15 : {:.4f}\n'.format(np.mean(update_num_list)))

print('RUNNING Q16...')
ce_loss_list = []
for _ in range(1000):
    w, _ = SGD(X_tra, Y_tra, lr=0.001, step_num=500, mode='ce')
    ce_loss = err(w, X_tra, Y_tra, mode='ce')
    ce_loss_list.append(ce_loss)
print('Answer of Q16 : {:.4f}\n'.format(np.mean(ce_loss_list)))

print('RUNNING Q17...')
wLIN = get_wLIN(X_tra, Y_tra)
ce_loss_list = []
for _ in range(1000):
    w, _ = SGD(X_tra, Y_tra, lr=0.001, w_init=wLIN,
               step_num=500, mode='ce')
    ce_loss = err(w, X_tra, Y_tra, mode='ce')
    ce_loss_list.append(ce_loss)
print('Answer of Q17 : {:.4f}\n'.format(np.mean(ce_loss_list)))

print('RUNNING Q18...')
print('Answer of Q18 : {:.4f}\n'.format(
    abs(err(wLIN, X_tst, Y_tst, mode='0/1') - err(wLIN, X_tra, Y_tra, mode='0/1'))))

print('RUNNING Q19...')
X_tra_Q = Q_transform(X_tra, Q=3)
X_tst_Q = Q_transform(X_tst, Q=3)
wLIN_Q = get_wLIN(X_tra_Q, Y_tra)
print('Answer of Q19 : {:.4f}\n'.format(abs(
    err(wLIN_Q, X_tst_Q, Y_tst, mode='0/1') - err(wLIN_Q, X_tra_Q, Y_tra, mode='0/1'))))

print('RUNNING Q20...')
```

```
    X_tra_Q = Q_transform(X_tra, Q=10)
    X_tst_Q = Q_transform(X_tst, Q=10)
    wLIN_Q = get_wLIN(X_tra_Q, Y_tra)
    print('Answer of Q20 : {:.4f}\n'.format(abs(
        err(wLIN_Q, X_tst_Q, Y_tst, mode='0/1') - err(wLIN_Q, X_tra_Q, Y_tra, mode='0/1'))))


if __name__ == "__main__":
    main()
```

14. [d]	15. [c]	16. [c]	17. [b]	18. [a]	19. [b]	20. [d]
0.6053	1889.73	0.5691	0.5028	0.3227	0.3737	0.4467