MLFALL 2021 HW1 Sol

1. Logistic Regression

- 0.1pt would be granted for any trying to answer the problem

1-(a)

Suppose we have a logistic regression model with four features that learns the following bias and weights:
$b = 1, w_1 = -1, w_2 = 2, w_3 = -1, w_4 = 5$

Suppose the following feature values for a given example:

x_1	x_2	x_3	x_4
7	0	3	10

Function set:

$$f_{w,b}(x) = P_{w,b}(C_1|x) = \sigma(\sum_i w_i x_i + b)$$

Please calculate the logistic regression prediction for the above particular example. (The answer should be a scalar indicating the posterior probability of class C_1)

Sol:

The log-odds: $b + w_1x_1 + w_2x_2 + w_3x_3$ (0.1 pt)
is $1 + \underset{(0.1pt)}{(-1)(7) + (2)(0) + (-1)(3) + (5)(10)} = \underset{(0.1pt)}{41}$

Write down log-odd correctly – 0.1 pt
Plug in the data properly – 0.1pt
Correct answer – 0.1pt

Consequently, the logistic regression prediction for this particular example will be
$y' = \frac{1}{1+e^{-41}} = 1$

1-(b)

Given training data:

x^1	x^2	x^3	...	x^N
$\hat{y}^1 = 1$ (Class 1)	$\hat{y}^2 = 1$ (Class 1)	$\hat{y}^3 = 0$ (Class 2)	...	$\hat{y}^N = 1$ (Class 1)

Assume the data is generated so that the probability of sample x belonging to C_1 is

$$f_{w,b}(x) = P_{w,b}(C_1|x)$$

Given a set of w and b, the probability of generating the data is as follows(assuming the data is generated independently):

$$L(w,b) = f_{w,b}(x^1)f_{w,b}(x^2)(1 - f_{w,b}(x^3)) \cdots f_{w,b}(x^N)$$

Please write down the loss function $L(w,b)$ defined as the negative of the log likelihood
(Hint: Cross entropy)

Sol:

$$-lnL(w,b) = -lnf_{w,b}(x^1) - lnf_{w,b}(x^2) - ln(1-f_{w,b}(x^3))\ldots -lnf_{w,b}(x^N)$$
$$(0.1pt) \qquad\qquad\qquad\qquad (0.1pt)$$

$$= \sum_n -[\hat{y}^n lnf_{w,b}(x^n) + (1-\hat{y}^n) ln(1-f_{w,b}(x^n))]\ (0.1pt)$$

1-(c)

Derive the formula that describes the update rule of parameters in logistic regression. (e.g., $w_i \leftarrow w_i - \ldots$) (Hint: Gradient descent)

Sol:

$$\frac{\partial(-lnL(w,b))}{\partial w_i} = \sum_n -[\hat{y}^n(1-f_{w,b}(x^n))x_i^n - (1-\hat{y}^n)f_{w,b}(x^n)x_i^n]\ (0.1pt)$$

$$= \sum_n -[\hat{y}^n - \hat{y}^n f_{w,b}(x^n) - f_{w,b}(x^n) + \hat{y}^n f_{w,b}(x^n)]x_i^n$$

$$= \sum_n -(\hat{y}^n - f_{w,b}(x^n))x_i^n\ (0.1pt)$$

$$w_i \leftarrow w_i - \eta\sum_n -(\hat{y}^n - f_{w,b}(x^n))x_i^n\ (0.1pt)$$

2. Closed-Form Loss Linear Regression

- 0.1pt would be granted for any trying to answer the problem

2-(a)

Let's begin with a specific dataset

$$S = \{(x_i, y_i)\}_{i=1}^5 = \{(1, 1.5), (2, 2.4), (3, 3.5), (4, 4.1), (5, 5.3)\}$$

Please find the linear regression model $(\mathbf{w}, b) \in \mathbb{R} \times \mathbb{R}$ that minimizes the sum of squares loss

$$L_{ssq}(\mathbf{w}, b) = \frac{1}{2 \times 5}\sum_{i=1}^5 (y_i - (\mathbf{w}^T\mathbf{x}_i + b))^2$$

Sol:

$$\begin{aligned} L_{ssq}(w,x) &= \tfrac{1}{10}\textstyle\sum_{i=1}^5 (y_i - (wx_i + b))^2 = \tfrac{1}{10}\sum_{i=1}^5 (wx_i + b - y_i)^2 \\ &= \tfrac{1}{10}((w+b-1.5)^2 + (2w+b-2.4)^2 + (3w+b-3.5)^2 \\ &+ (4w+b-4.1)^2 + (5w+b-5.3)^2)\ (0.1pt) \end{aligned}$$

$$\begin{aligned} \tfrac{\partial}{\partial w}L_{ssq} &= \tfrac{1}{10}(2(w+b-1.5)\cdot 1 + 2(2w+b-2.4)\cdot 2 + 2(3w+b-3.5)\cdot 3 \\ &+ 2(4w+b-4.1)\cdot 4 + 2(5w+b-5.3)\cdot 5 = 11w + 3b - 11.94 \end{aligned}$$

$$\begin{aligned} \tfrac{\partial}{\partial b}L_{ssq} &= \tfrac{1}{10}(2(w+b-1.5)\cdot 1 + 2(2w+b-2.4)\cdot 1 + 2(3w+b-3.5)\cdot 1 \\ &+ 2(4w+b-4.1)\cdot 1 + 2(5w+b-5.3)\cdot 1) = 3w + b - 3.36 \\ &\qquad (0.1pt\ for\ the\ differentiations) \end{aligned}$$

$$\begin{aligned} 11w + 3b - 11.94 = 0 \qquad (1) \\ 3w + b - 3.36 = 0 \qquad (2) \end{aligned}$$

解二元一次方程式可得 $w = 0.93\ b = 0.57 (0.1pt)$

2-(b)

Please find the linear regression model $(\mathbf{w}, b) \in \mathbb{R}^k \times \mathbb{R}$ that minimizes the sum of squares loss

$$L_{ssq}(\mathbf{w}, b) = \frac{1}{2N}\sum_{i=1}^{N}(y_i - (\mathbf{w}^T\mathbf{x}_i + b))^2$$

Sol:

$$令\ A = \begin{bmatrix} 1 & x_{1,1} & x_{1,2} & \dots & x_{1,k} \\ 1 & x_{2,1} & x_{2,2} & \dots & x_{2,k} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{N,1} & x_{N,2} & \dots & x_{N,k} \end{bmatrix} \quad w = \begin{bmatrix} b \\ \mathbf{w_1} \\ \vdots \\ \mathbf{w_k} \end{bmatrix} \quad y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} (0.1pt)$$

$$則\ Aw = y \quad (A^TA)w = A^Ty\ (0.1pt) \quad \begin{bmatrix} b \\ \mathbf{w} \end{bmatrix} = w = (A^TA)^{-1}A^Ty\ (0.1pt)$$

2-(c)

A key motivation for regularization is to avoid overfitting. A common choice is to add a L^2-regularization term into the original loss function

$$L_{reg}(\mathbf{w}, b) = \frac{1}{2N}\sum_{i=1}^{N}(y_i - (\mathbf{w}^T\mathbf{x}_i + b))^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2$$

where $\lambda \geq 0$ and for $\mathbf{w} = [w_1\ w_2\ \dots\ w_k]^T$, one denotes $\|\mathbf{w}\|^2 = w_1^2 + \dots + w_k^2$.

Please find the linear regression model $(\mathbf{w}, b)$ that minimizes the aforementioned regularized sum of squares loss.

$$令\ A = \begin{bmatrix} 1 & x_{1,1} & x_{1,2} & \dots & x_{1,k} \\ 1 & x_{2,1} & x_{2,2} & \dots & x_{2,k} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{N,1} & x_{N,2} & \dots & x_{N,k} \end{bmatrix} \quad w = \begin{bmatrix} b \\ \mathbf{w_1} \\ \vdots \\ \mathbf{w_k} \end{bmatrix} \quad y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$$

$L = \frac{1}{2N}(Aw - y)^2 + \frac{\lambda}{2}\|w\|^2 - \frac{\lambda b^2}{2}\ (0.1pt)$

$$\frac{\partial L}{\partial w} = \frac{1}{N}(A^TAw - A^Ty) + \lambda I w - \lambda I \begin{bmatrix} b \\ 0 \\ \vdots \\ 0 \end{bmatrix} = 0\ (0.1pt)$$

$$\begin{bmatrix} b \\ 0 \\ \vdots \\ 0 \end{bmatrix} = (A^TA)^{-1}A^Ty$$

$$\begin{bmatrix} 0 \\ w1 \\ \vdots \\ wk \end{bmatrix} = (A^TA + \lambda N I)^{-1}A^Ty$$

$$\begin{bmatrix} b \\ \mathbf{w} \end{bmatrix} = w = \left[((A^TA)^{-1} + (A^TA + \lambda N I)^{-1})A^Ty \right]\ (0.1pt)$$

3. Noise and regulation

Consider the linear model $f_{\mathbf{w},b} : \mathbb{R}^k \to \mathbb{R}$, where $\mathbf{w} \in \mathbb{R}^k$ and $b \in \mathbb{R}$, defined as

$$f_{\mathbf{w},b}(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$$

Given dataset $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$, if the inputs $\mathbf{x}_i \in \mathbb{R}^k$ are contaminated with input noise $\eta_i \in \mathbb{R}^k$, we may consider the expected sum-of-squares loss in the presence of input noise as

$$\tilde{L}_{ssq}(\mathbf{w}, b) = \mathbb{E}\left[\frac{1}{2N}\sum_{i=1}^{N}(f_{\mathbf{w},b}(\mathbf{x}_i + \eta_i) - y_i)^2\right]$$

where the expectation is taken over the randomness of input noises $\eta_1, \ldots, \eta_N$.

Now assume the input noises $\eta_i = [\eta_{i,1}\ \eta_{i,2}\ \ldots \eta_{i,k}]$ are random vectors with zero mean $\mathbb{E}[\eta_{i,j}] = 0$, and the covariance between components is given by

$$\mathbb{E}[\eta_{i,j}\eta_{i',j'}] = \delta_{i,i'}\delta_{j,j'}\sigma^2$$

where $\delta_{i,i'} = \begin{cases} 1 & \text{, if } i = i' \\ 0 & \text{, otherwise.} \end{cases}$ denotes the Kronecker delta.

Please show that

$$\tilde{L}_{ssq}(\mathbf{w}, b) = \frac{1}{2N}\sum_{i=1}^{N}(f_{\mathbf{w},b}(\mathbf{x}_i) - y_i)^2 + \frac{\sigma^2}{2}\|\mathbf{w}\|^2$$

That is, minimizing the expected sum-of-squares loss in the presence of input noise is equivalent to minimizing noise-free sum-of-squares loss with the addition of a L^2-regularization term on the weights.

- Hint: $\|\mathbf{x}\|^{\mathbf{2}} = \mathbf{x^T x} = \mathbf{Trace(xx^T)}$.

Sol:

$$\tilde{L}_{ssq}(\mathbf{w}, b) = \mathbb{E}\left[\frac{1}{2N}\sum_{i=1}^{N}(w_i^T(x_i + \eta_i) + b - y_i)^2\right]$$

$$= \frac{1}{2N}\sum_{i=1}^{N}\left(\mathbb{E}\left[(w_i^T x_i + b - y_i)^2\right] + 2(w_i^T x_i + b - y_i)\mathbb{E}\left[w_i^T\eta_i\right] + \mathbb{E}\left[(w_i^T\eta_i)^2\right]\right)$$

(0.3pts)

$$\because \mathbb{E}\left[w_i^T\eta_i\right] = 0 \quad and \quad \mathbb{E}\left[(w_i^T\eta_i)^2\right] = \sigma^2\|\mathbf{w^2}\|$$

(0.4pts)

$$\tilde{L}_{ssq}(\mathbf{w}, b) = \frac{1}{2N}\sum_{i=1}^{N}((f_{\mathbf{w},b}(\mathbf{x}_i) - y_i)^2 + 0 + \sigma^2\|\mathbf{w^2}\|)$$

$$= \mathbf{\frac{1}{2N}\sum_{i=1}^{N}(f_{w,b}(x_i) - y_i)^2 + \frac{\sigma^2}{2}\|w\|^2}$$

(0.3pts)

4. Kaggle Hacker

Suppose you have trained $K+1$ models $g_0, g_1, \cdots g_K$, and in particular $g_0(\mathbf{x}) = 0$ is the zero function.

Assume the testing dataset is $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, where you only know x_i while y_i is hidden. Nevertheless, you are allowed to observe the sum of squares testing error

$$e_k = \frac{1}{N}\sum_{i=1}^{N}(g_k(\mathbf{x}_i) - y_i)^2, \ \ k = 0, 1, \cdots K$$

Ofcourse, you know $s_k = \frac{1}{N}\sum_{i=1}^{N}(g_k(\mathbf{x}_i))^2$.

4-(a)

Please express $\sum_{i=1}^{N} g_k(\mathbf{x}_i)y_i$ in terms of $N, e_0, e_1, \cdots, e_K, s_1, \cdots, s_K$. Prove your answer.

- Hint: $e_0 = \frac{1}{N}\sum_{i=1}^{N} y_i^2$

Sol:

$e_k = \frac{1}{N}\sum_{i=1}^{N}(g_k(x_i) - y_i)^2 = \frac{1}{N}\sum_{i=1}^{N}((g_k(x_i)^2 - 2g_k(x_i)y_i + y_i^2)) = S_k - \frac{2}{N}\sum_{i=1}^{N} g_k(x_i)y_i + e_0$ (0.2pts)

$\sum_{i=1}^{N} g_k(x_i)y_i = \frac{N}{2}(S_k - e_k + e_0)$ (0.3pts)

4-(b)

For the given $K+1$ models in the previous problem, explain how to solve

$\min_{\alpha_1,\cdots\alpha_K} L_{test}(\sum_{k=1}^{K}\alpha_k g_k) = \min[\frac{1}{N}\sum_{i=1}^{N}(\sum_{k=1}^{K}\alpha_k g_k(\mathbf{x}_i) - y_i)^2]$, and obtain the optimal weights $\alpha_1, \cdots \alpha_K$.

Sol:

$$\text{令}\ A = \begin{bmatrix} g_1(x_1) & g_2(x_1) & \dots & g_K(x_1) \\ g_1(x_2) & g_2(x_2) & \dots & g_K(x_2) \\ \vdots & \vdots & \vdots & \vdots \\ g_1(X_N) & g_2(x_N) & \dots & g_K(x_N) \end{bmatrix} \alpha = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_K \end{bmatrix} y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$$

$$L_{test}(f) = \frac{1}{N}(G\alpha - y)^T(G\alpha - y)$$

(0.1pts)

$$\frac{\partial L}{\partial \alpha} = \frac{1}{N}G^T(G\alpha - y) = 0$$

(0.2pts)

$$\alpha = (G^TG)^{-1}G^Ty$$

(0.2pts)