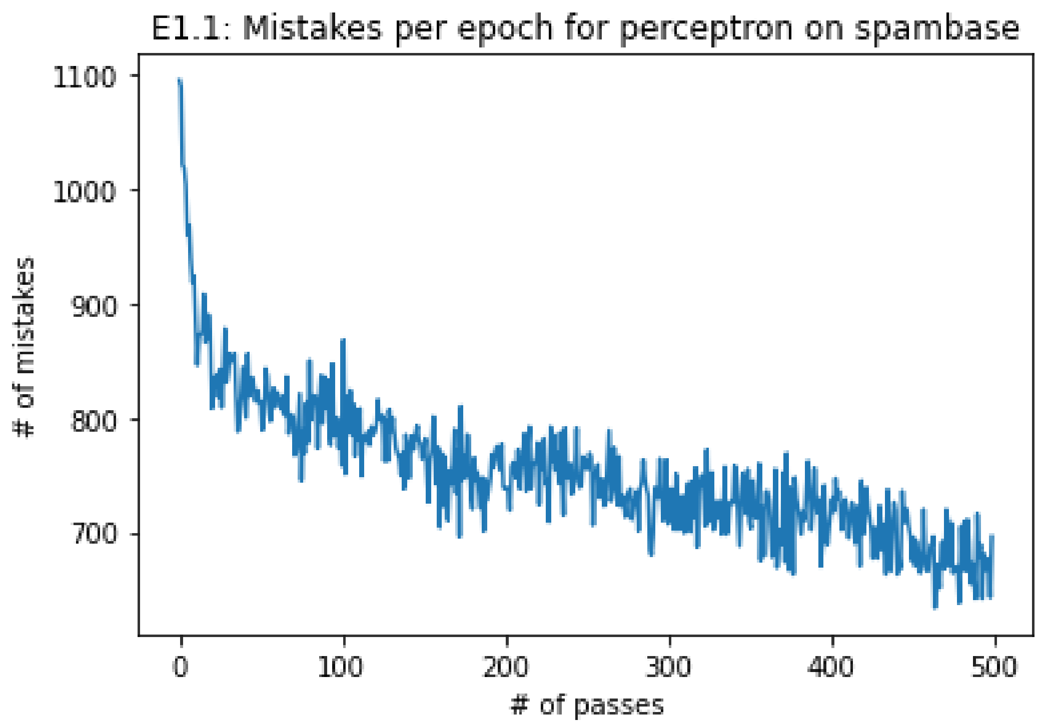
CS680 – Assignment 1

Ali ElSaid (20745892)

E1.Q1:



E1.Q2:

```
# Training errors: 139
# Testing errors: 145
```

E1.Q3:

```
# Training errors: 111
# Testing errors: 148
```

E1.Q4:

E1.Q4 Consider a term from the summation:

$$\max\{0, -y_i(\langle x_i, w\rangle)\} = f(w)$$

this term is
$$\begin{cases} 0 & \text{, when } y_i(\langle x_i,w\rangle) \text{ is positive, i.e. the prediction is right} \\ -y_i(\langle x_i, w\rangle) & \text{, when the prediction is wrong} \end{cases}$$

- So this term can be seen as the penalty assigned when a wrong prediction is made.
- Taking ~~its~~ its derivative by the rule for $\frac{d}{dw} f(w) = \max_k f_k(w)$

then
$$\frac{d}{dw} f(w) = g(w) = \begin{cases} 0 & \text{, when prediction correct} \\ -y_i x_i & \text{, when prediction wrong} \end{cases}$$

Comparing that gradient to the ~~update~~ update ~~of~~ the algorithm makes we find that they are exactly the same; i.e. when the prediction is correct, ~~t~~ no update happens. But when the prediction is wrong, the algorithm updates $w \Leftarrow w - (-y_i x_i)$.

$\therefore$ we see that the update is $w \Leftarrow w - g(w)$.

E1.Q5:

E1.Q5 when $C=2$: consider the two cases for y_i & the corresponding term inside the summation

when $y_i=1 \rightarrow$ term is: $\max\left\{\langle x_i, w_1\rangle - \langle x_i, w_1\rangle, \langle x_i, w_2\rangle - \langle x_i, w_1\rangle\right\}$

$$= \max\{0, \langle x_i, w_2\rangle - \langle x_i, w_1\rangle\}$$

when $y_i=2 \rightarrow$ term is: $\max\{\langle x_i, w_2\rangle - \langle x_i, w_2\rangle, \langle x_i, w_1\rangle - \langle x_i, w_2\rangle\}$

$$= \max\{0, \langle x_i, w_1\rangle - \langle x_i, w_2\rangle\}$$

~~Since we~~

This term should be equal to the similar term in the binary case

So, equating both we get:

& if we map $y_i = 1 \longrightarrow y_i^{(2)} = -1$

&

$$\underbrace{y_i = 2}_{\text{when } C=2} \longrightarrow \underbrace{y_i^{(2)} = 1}_{\text{in binary perceptron}}$$

then, when when $y_i=1$ in our problem & $y_i^{(2)} = -1$ in the binary perceptron

$$\langle x_i, w_2\rangle - \langle x_i, w_1\rangle = -(\underset{y_i^{(2)}}{-1})(\langle x_i, w\rangle)$$

$\rightarrow \langle x_i, w_2 - w_1\rangle = \langle x_i, w\rangle$, which implies

$$w = w_2 - w_1$$

E1.Q6:

```
# Training errors: 126
# Testing errors: 132
```

E1.Q6 Suppose we predict $\hat{y}_i$

So, by the definition of the prediction

$$\langle x_i, w_{\hat{y}_i} \rangle > \langle x_i, w_k \rangle \quad \forall k \neq \hat{y}_i$$

$$\text{So, } \langle x_i, w_{\hat{y}_i} \rangle - \langle x_i, w_{y_i} \rangle > \langle x_i, w_k \rangle - \langle x_i, w_{y_i} \rangle \quad \forall k \neq \hat{y}_i$$

Consider 2 cases:

Case 1: $\hat{y}_i = y_i$, i.e. prediction is correct.

then $\langle x_i, w_{\hat{y}_i} \rangle - \langle x_i, w_{y_i} \rangle$

$$= \langle x_i, w_{\hat{y}_i} \rangle - \langle x_i, w_{\hat{y}_i} \rangle = 0 > \langle x_i, w_k \rangle - \langle x_i, w_{\hat{y}_i} \rangle \quad \forall k \neq \hat{y}_i$$

& So, the max term is equal to zero.

i.e. No penalty since we predicted correctly.

Case 2: $\hat{y}_i \neq y_i$, i.e. wrong Prediction.

& $\langle x_i, w_{\hat{y}_i} \rangle - \langle x_i, w_{y_i} \rangle > 0$ & we incur Penalty for wrong Prediction.

To get the algorithm & update, we find the gradient of the term $f(w)$ inside the summation. From the definition of $f'(w)$ when $f(w) = \max_k f_k(w)$ we get $\longrightarrow$

if Correct Prediction

$$\nabla_w O = 0$$, So no update

if incorrect Prediction

$$\frac{\partial}{\partial w_k} \langle x_i, w_{\hat{y}_i} \rangle - \langle x_i, w_{y_i} \rangle = \begin{cases} 0 & \text{, when } k \neq \hat{y}_i \text{ \& } k \neq y_i \\ x_i & \text{, when } k = \hat{y}_i \\ -x_i & \text{, when } k = y_i \end{cases}$$

So we do the following update:

$$w_{\hat{y}_i} \leftarrow w_{\hat{y}_i} - x_i \quad \& \quad w_{y_i} \leftarrow w_{y_i} + x_i$$ (Padded w & x_i)

Algorithm:

Input: $X \in \mathbb{R}^{d \times n}$, $y \in \{1, 2, -, c\}^n$,

$w = [w_1, ---, w_c]$ ~~[illegible]~~ $= 0_{d \times c}$, ie. each $w_j = 0_d$ $\forall j \in \{1, ..., c\}$

$b = [b_1, ---, b_c] = 0_c$, i.e. Each $b_j = 0$ $\forall j \in \{1, ---, c\}$

Max_Pass $\in \mathbb{N}$

Output: $W \in \mathbb{R}^{d \times c}$, $b \in \mathbb{R}^c$, mistake

for $t = 1, 2, ---,$ Max_Pass do
    mistake$(t) \leftarrow 0$
    for $i = 1, 2, --, n$ do
        $\hat{y}_i = \underset{k=1,-,c}{\text{argmax}} \langle x_i, w_k \rangle + b_k$
        if $\hat{y}_i \neq y_i$ then:
            $w_{\hat{y}_i} \leftarrow w_{\hat{y}_i} - x_i$, $b_{\hat{y}_i} \leftarrow b_{\hat{y}_i} - 1$
            $w_{y_i} \leftarrow w_{y_i} + x_i$, $b_{y_i} \leftarrow b_{y_i} + 1$
            mistake$(t) \leftarrow$ mistake$(t) + 1$

E2.Q1:

E2.Q1

$$L(Y|X)$$

likelihood function : $P(y_1, \ldots, y_n | x_1, \ldots, x_n) = \prod_{i=1}^{n} P(y_i | x_i)$

since we assume indep.

$$\ell_n(Y|X) = -\log L(Y|X) = -\log \prod_{i=1}^{n} P(y_i | x_i)$$

$$= -\sum_{i=1}^{n} \log P(y_i | x_i) = -\sum_{i=1}^{n} \log\left(\exp(M(x_i) y_i - \lambda(x_i)) q(y_i)\right)$$

$$= \boxed{-\sum_{i=1}^{n} M(x_i) y_i - \lambda(x_i) + \log q(y_i)}$$

where $\ell_n(Y|X)$ is the negative log-likelihood function

E2.Q2:

E2.Q2 Plugging into ℓ_n

we get $\ell_n(w) = -\sum_{i=1}^{n} w^T x_i\, y_i - \lambda(x_i) + \log q(y_i)$

the gradient
$$\nabla_w \ell_n(w) = -\sum_{i=1}^{n} x_i y_i - \nabla_w \lambda(x_i)$$

& $\nabla_w \lambda(x_i) = \nabla_w \log \int_y \left(e^{w^T x_i \cdot y}\right) q(y)\, dy$,

$$= \frac{\nabla_w \int_y \left(e^{w^T x_i \cdot y}\right) q(y)\, dy}{\int_y \exp(w^T x_i \cdot y)\, q(y)\, dy} \quad , \text{ Since } \frac{\partial}{\partial w} \log f(w) = \frac{\frac{\partial}{\partial w} f(w)}{f(w)}$$

$$= \frac{\int_y \nabla_w \exp(\ldots)\, q(y)\, dy}{\int_y \exp(\ldots)\, q(y)\, dy} \quad , (\ldots) = w^T x_i \cdot y$$

$$= \frac{x_i \int_y y \left(e^{w^T x_i \cdot y}\right) q(y)\, dy}{e^{\lambda(x_i)}} \quad , \text{ since } \lambda(x_i) = \log \int_y e^{w^T x_i \cdot y} q(y)\, dy$$

$$= x_i \int_y y\, e^{w^T x_i y} \cdot e^{-\lambda(x_i)} \cdot q(y)\, dy = x_i \int_y y\, e^{w^T x_i y - \lambda(x_i)} q(y)\, dy$$

$$= x_i \int_y y\, P(y|x_i)\, dy = \boxed{x_i \cdot E(y|x_i)}$$

So the gradient part $\nabla_w \lambda(x_i) = x_i\, E(y|x_i)$

& $\boxed{\nabla_w \ell_n(w) = -\sum_{i=1}^{n} x_i \left(y_i - E(y|x_i)\right)}$

E2.Q3:

E2.Q3

$$y^2 - 2y\nu(x) + \nu(x)^2$$

$$P(y|x) = \frac{1}{\sqrt{2\pi}} \exp\left(- \frac{(y - \nu(x))^2}{2}\right)$$

$$= \underbrace{\frac{\exp(-y^2/2)}{\sqrt{2\pi}}}_{q(y)} \exp\Big(y \underbrace{\nu(x)}_{\mu(x)} - \underbrace{\frac{\nu(x)^2}{2}}_{\lambda(x)}\Big)$$

Plugging into $\ell_n(\omega)$

$$\boxed{\ell_n(\omega)} = - \sum_{i=1}^{n} \omega^T x_i \cdot y_i - \frac{(\omega^T x_i)^2}{2} + \log\left(\frac{1}{\sqrt{2\pi}}\right) - \frac{y^2}{2}$$

$$\boxed{\nabla_\omega \ell_n(\omega)} = - \sum_{i=1}^{n} x_i (y_i) - x_i^T \omega x_i$$

$$= - \sum_{i=1}^{n} x_i (y_i - x_i^T \omega)$$

E2.Q4:

E2.Q4

$P(Y=y|x) = [\nu(x)]^{y}[1-\nu(x)]^{1-y}$, taking $\exp(\log(\ ))$

$\rightarrow\ = \exp(y \cdot \underbrace{\log(\frac{\nu(x)}{1-\nu(x)})}_{\eta(x)} + \underbrace{\log(1-\nu(x))}_{\lambda(x)})$

& $q(y) = 1$

$\eta(x) = \log(\frac{\nu(x)}{1-\nu(x)})$ & $\eta(x) = w^T x$

So $\nu(x) = \frac{e^{\eta(x)}}{1+e^{\eta(x)}} = \frac{e^{w^T x}}{1+e^{w^T x}}$

Plugging into $\lambda(x) = -(\log(1-\nu(x))) = -(\log(1-\frac{e^{w^T x}}{1+e^{w^T x}}))$

$= \log(1+e^{w^T x})$

& $\ell_n(w) = -\sum_{i=1}^{n} w^T x_i y_i - \log(1+e^{w^T x})$

& $\nabla_w \ell_n(w) = -\sum_{i=1}^{n} x_i y_i - \frac{e^{w^T x}}{1+e^{w^T x}} \cdot x_i = -\sum_{i=1}^{n} x_i(y_i - \nu(x_i))$

E2.Q5:

$\boxed{\text{E2.Q5}}$

$$P(Y=y|x) = \frac{[\nu(x)]^y}{y!} e^{-\nu(x)}$$, taking $\exp(\log(\ ))$

$$\to = \exp\Big(y \cdot \underbrace{\log \nu(x)}_{\mu(x)} - \underbrace{\nu(x)}_{\lambda(x)}\Big) \cdot \underbrace{\frac{1}{y!}}_{q(y)}$$

E3.Q1:

E3.Q1 If we treat it as normal multiclassification, then we will count any wrong prediction to be the same no matter what. e.g. in the letter grade example if the actual grade is A, we would like to penalize more for a prediction of C than B since B is "closer" to A because of the ordering. However that will not happen if we do normal multiclassification.

E3.Q2:

E3.Q2 if we just encode the labels as $\{1, 2, \dots c\}$ and run normal regression, we will basically be telling the model that the quantitative difference between label 1 & 2 is the same as between 2 & 3 for example. i.e. the predictors have to change equivalenty to go from 1 to 2 & from 2 to 3.

And this assumption will not be a sound one often in real life, since the quantitative difference between labels might actually be different.

E3.Q3:

E3.Q3

replacing the max with the dual variable

$$\rightarrow \min_{w,\, b_1 \le b_2 \le \cdots \le b_{c-1}} \max_{0 \le \alpha \le C} \overbrace{\frac{1}{2}\|w\|_2^2 + \sum_{k=1}^{c-1} \sum_{i=1}^{n} \alpha_{ki} \Big[1 - \big([\![y_i = k]\!] - [\![y_i = k+1]\!] \big) \big(\langle x_i, w \rangle + b_k \big) \Big]}^{f(w, b)}$$

~~differentiating~~ switching min with max & differentiating we get.

$$\frac{\partial}{\partial w} f(w, b) = w - \sum_{k=1}^{c-1} \sum_{i=1}^{n} \alpha_{ki} \big([\![y_i = k]\!] - [\![y_i = k+1]\!] \big) x_i = 0$$

$$\rightarrow w = \underline{\qquad\qquad}$$

$$\frac{\partial}{\partial b_j} = 0 + \sum_{i=1}^{n} \alpha_{ji} \big([\![\ ]\!] - [\![\ ]\!] \big) = 0, \ \forall j \in \{1, \ldots, c-1\}$$

Plugging back into Primal Problem. we get the dual problem

E3.Q4:

E3.Q4 In the original formulation the penalty is

1 if $y_i \neq k$ and $y_i \neq k+1$ for a given point

however a better formulation would be ~~to punish~~

~~all the points~~, for a given point, to punish all the

planes represented by b_k's that are supposed to be

"above" it if $k \leq y_i - 1$, since all planes that

have $y_i \geq k+1 \iff k \leq y_i - 1$ should be "below" the point

& vice versa for $y_i \leq k$, the planes should be
"above" it

So the formulation would be:

$$\min_{w, b_1 \leq b_2 \leq \dots \leq b_{c-1}} \frac{\lambda}{2} \|w\|_2^2 + \sum_{k=1}^{c-1} \sum_{i=1}^{n} \max\left\{0, 1 - \left([\![y_i \leq k]\!] - [\![y_i \geq k+1]\!] \right) \left(\langle x_i, w \rangle + b_k \right) \right\}$$

, this will punish all the planes in the wrong place

for any given point.