CSED442 Spring 2018 Homework [number]

POVIS ID: seongtae0205
Name: Seongtae Kim

By turning in this assignment, I agree by the POSTECH honor code and declare that all of this is my own work.

Problem 1

(a)

$$\mathbf{w} \leftarrow \mathbf{w} - (1)(-1)[1 - \mathbf{w} \cdot \phi(x)y > 0]\phi(x)y \tag{1}$$

$$\mathbf{w} = [1, 2, 1, 1, 1, 0]$$

(b) Let $f : \mathbf{w_1}x_{not} + \mathbf{w_2}x_{good} + \mathbf{w_3}x_{bad}$, and $\phi(x) = [x_{not}, x_{good}, x_{bad}]$. My Dataset is $\phi(x_1) = [1, 1, 0], \quad \phi(x_2) = [1, 0, 1], \quad \phi(x_3) = [0, 1, 0], \quad \phi(x_4) = [0, 0, 1]$.
The true result of Dataset is $y_1 = -1,\ y_2 = 1,\ y_3 = 1,\ y_4 = -1$. We can solve this problem by using Matrix.

$$\begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \mathbf{w_1} \\ \mathbf{w_2} \\ \mathbf{w_3} \end{pmatrix} = \begin{pmatrix} -1 \\ 1 \\ 1 \\ -1 \end{pmatrix}$$

By back substitution, we can get $\mathbf{w_3} = -1,\ \mathbf{w_2} = 1,\ \mathbf{w_1} = 2$ from the last 3 equations. However, at the first equation, $\mathbf{w_1} + \mathbf{w_2} = 2 + 1 = 3 \neq -1$. Therefore, there is no solution satisfies all equations. This means that all linear classifiers using the above 3 words features must occur error in my dataset.

x_{not} is relative to x_{good} and x_{bad}. Therefore, we need cross-terms to decribe feature. The new feature vector $\phi(x) = [x_{good},\ x_{bad},\ x_{not} \cdot x_{good},\ x_{not} \cdot x_{bad}]$.

Problem 2

(a)

$$Loss(x, y, \mathbf{w}) = (\sigma(\mathbf{w} \cdot \phi(x)) - y)^2$$

(b)

$$\begin{aligned}
\sigma(z) &= (1 + e^{-z}))^{-1} \\
\sigma'(z) &= e^{-z}(1 + e^{-z}))^{-2} \\
&= \sigma(z)(1 - \sigma(z))
\end{aligned}$$

$$\begin{aligned}
\nabla_{\mathbf{w}} Loss(x, y, \mathbf{w}) &= 2(\phi(x)\sigma'(\mathbf{w} \cdot \phi(x))(\sigma(\mathbf{w} \cdot \phi(x)) - y) \\
&= 2\phi(x)p(1 - p)(p - y)
\end{aligned}$$

(c) When y=0, we have to minimize $2\phi(x)p^2(1-p)$. Then we can differentiate that formula by p.

$$\begin{aligned}
2\phi(x)\{2p(1 - p) - p^2\} &= 0 \\
2\phi(x)p(2 - 3p) &= 0
\end{aligned}$$

$0 < p < 1$, So we have to check when p=0, p=$\frac{2}{3}$ and p=1.

$$\begin{aligned}
\sigma(\mathbf{w} \cdot \phi(x)) &= 0 \\
\sigma(\mathbf{w} \cdot \phi(x)) &= \frac{2}{3} \\
\sigma(\mathbf{w} \cdot \phi(x)) &= 1
\end{aligned}$$

$$\begin{aligned}
\| \nabla_{\mathbf{w}} Loss(x, y, \mathbf{w}) \|_{y=0} &= 0 \\
\| \nabla_{\mathbf{w}} Loss(x, y, \mathbf{w}) \|_{y=\frac{2}{3}} &= \frac{8}{27} \| \phi(x) \| \\
\| \nabla_{\mathbf{w}} Loss(x, y, \mathbf{w}) \|_{y=1} &= 0
\end{aligned}$$

Therefore, when p=1 and p=0, the gradient has minimized value.

$$\begin{aligned}
\mathbf{w} &= -\infty \\
\mathbf{w} &= \infty
\end{aligned}$$

(d) When p=$\frac{2}{3}$, the gradient has maximized value.

$$\| \nabla_{\mathbf{w}} Loss(x, y, \mathbf{w}) \|_{y=\frac{2}{3}} = \frac{8}{27} \| \phi(x) \|$$

Problem 3

(a) programming

(b) programming

(c) programming

(d) 1. === home alone goes hollywood , a funny premise until the kids start pulling off stunts not even steven spielberg would know how to do . besides , real movie producers aren't this nice . Truth: -1, Prediction: 1 [WRONG]
Reason: There is too much positive weight on "funny" (0.4).

2.=== a perfectly competent and often imaginative film that lacks what little lilo stitch had in spades – charisma . Truth: 1, Prediction: -1 [WRONG]
Reason: There is too much negative weight on "lacks" (-0.56)

3.=== a heady , biting , be-bop ride through nighttime manhattan , a loquacious videologue of the modern male and the lengths to which he'll go to weave a protective cocoon around his own ego . Truth: 1, Prediction: -1 [WRONG]
Reason: There is too much positive weight on "ride" (0.87).

4. === 'it's painful to watch witherspoon's talents wasting away inside unnecessary films like legally blonde and sweet home abomination , i mean , alabama . ' Truth: -1, Prediction: 1 [WRONG]
Reason: There is too much positive weight on "sweet" (0.55).

5. === wickedly funny , visually engrossing , never boring , this movie challenges us to think about the ways we consume pop culture . Truth: 1, Prediction: -1 [WRONG]
Reason: Reason: There is too much positive weight on "culture" (0.46).

6. === rain is a small treasure , enveloping the viewer in a literal and spiritual torpor that is anything but cathartic . Truth: 1, Prediction: -1 [WRONG]
Reason: There are less crucial positive weighted words in the review.

7. === patchy combination of soap opera , low-tech magic realism and , at times , ploddingly sociological commentary . Truth: -1, Prediction: 1 [WRONG]
Reason: There is too much positive weight on "magic" (0.5) and "realism"(0.42).

8. === the best thing i can say about this film is that i can't wait to see what the director does next . Truth: 1, Prediction: -1 [WRONG]
Reason: There is too much negative weight on "can't" (-0.44).

9. === only in its final surprising shots does rabbit-proof fence find the authority it’s looking for . Truth: -1, Prediction: 1 [WRONG]
Reason: There is too much positive weight on ”does” (0.6).

10. === . . . standard guns versus martial arts cliche with little new added . Truth: -1, Prediction: 1 [WRONG]
Reason: There is too much positive weight on ”new” (0.37).

I think the main reason is the unbalance of the weight value of the words. And the other main reason is that it doesn’t consider the effect of negative words. For example, even though the sentence “I can’t wait” has actually positive meaning, because of the word “can’t”, this sentence was predicted by negative sentence.

(e) programming

(f) I did test3b2() with extractCharacterFeatures function.

```
train error = 0.457231288689, dev error = 0.478615644344        when n=1
train error = 0.337929093979, dev error = 0.419527293191        when n=2
train error = 0.00281373100732, dev error = 0.318232976927        when n=3
train error = 0.0, dev error = 0.283342712437        when n=4
train error = 0.0, dev error = 0.270680922904        when n=5
train error = 0.0, dev error = 0.271525042206        when n=6
train error = 0.000281373100732, dev error = 0.270962296005        when n=7
train error = 0.000562746201463, dev error = 0.292909397862        when n=8
train error = 0.000844119302195, dev error = 0.308384918402        when n=9
train error = 0.000844119302195, dev error = 0.337647720878        when n=10
```

The cases which have the smallest test error are when n is between 4 and 8. Actually most of the words consist of 4 to 8 characters. I believe that the reason why this result came out is that a number of extracted characters is similar with the word size.

Problem 4

(a) 1.

$$z_i \leftarrow arg \min_{k=1,\dots,K} \parallel \phi(x_i) - \mu_k \parallel^2$$

$$\mu_k \leftarrow \frac{1}{|\{i : z_i = k\}|} \sum_{i:z_i=k} \phi(x_i)$$

$$\text{for } \phi(x_1), arg\ min(1, 13)$$
$$\text{z}_1 = 1$$
$$\text{for } \phi(x_2), arg\ min(2, 10)$$
$$\text{z}_2 = 1$$
$$\text{for } \phi(x_3), arg\ min(9, 5)$$
$$\text{z}_3 = 2$$
$$\text{for } \phi(x_4), arg\ min(13, 1)$$
$$\text{z}_4 = 2$$

$$\mu_1 = [0,\ 0.5] \quad \mu_2 = [2,\ 1]$$

$$\text{for } \phi(x_1), arg\ min(0.25, 5)$$
$$\text{z}_1 = 1$$
$$\text{for } \phi(x_2), arg\ min(0.25, 4)$$
$$\text{z}_2 = 1$$
$$\text{for } \phi(x_3), arg\ min(4.25, 1)$$
$$\text{z}_3 = 2$$
$$\text{for } \phi(x_4), arg\ min(6.25, 1)$$
$$\text{z}_4 = 2$$

$$\mu_1 = [0,\ 0.5] \quad \mu_2 = [2,\ 1]$$

The result is $\mu_1 = [0,\ 0.5] \quad \mu_2 = [2,\ 1]$, and $z_1 = 1,\ z_2 = 1,\ z_3 = 2,\ z_4 = 2$

2.

$$\text{for } \phi(x_1), arg\ min(2, 4)$$
$$\text{z}_1 = 1$$
$$\text{for } \phi(x_2), arg\ min(5, 1)$$
$$\text{z}_2 = 2$$

$$\text{for } \phi(x_3), arg\ min(2,8)$$
$$\text{z}_3 = 1$$
$$\text{for } \phi(x_4), arg\ min(10,4)$$
$$\text{z}_4 = 2$$

$$\mu_1 = [1,\ 0] \quad \mu_2 = [1,\ 1.5]$$

$$\text{for } \phi(x_1), arg\ min(1,3.25)$$
$$\text{z}_1 = 1$$
$$\text{for } \phi(x_2), arg\ min(2,1.25)$$
$$\text{z}_2 = 2$$
$$\text{for } \phi(x_3), arg\ min(1,3.25)$$
$$\text{z}_3 = 1$$
$$\text{for } \phi(x_4), arg\ min(5,1.25)$$
$$\text{z}_4 = 2$$

$$\mu_1 = [1,\ 0] \quad \mu_2 = [1,\ 1.5]$$

The result is $\mu_1 = [1,\ 0] \quad \mu_2 = [1,\ 1.5]$, and $z_1 = 1,\ z_2 = 2,\ z_3 = 1,\ z_4 = 2$

(b) programming

(c) STEP 1: For each point i=1, ... ,n :

$$z_i \leftarrow arg \min_{k=1,\ldots,K} \parallel \phi(x_i) - \mu_k \parallel^2$$

And if there exists the pair (i, j) in S, then do

$$z_j \leftarrow z_i$$

And for j, do not execute the first one. Just do $z_j \leftarrow z_i$.

STEP 2:

$$\mu_k \leftarrow \frac{1}{|\{i : z_i = k\}|} \sum_{i:z_i=k} \phi(x_i)$$