CSE6740: Computational Data Analysis **(Due: 06/28/2020)**

Submission Assignment #3

Instructor: Prof Xie *Name:* Student name(s):Jian Pang, *Netid:*

Problem 1 (10+10+10 points)

(1) Solution

$$\begin{aligned}\frac{dl}{d\theta} &= \sum_{i=1}^{m}\{\frac{exp(-\theta x_i)x_i}{1+exp(-\theta x_i)} + (y_i-1)x_i\} \\ &= \sum_{i=1}^{m} x_i(y_i - 1 + \frac{exp(-\theta x_i)}{1+exp(-\theta x_i)}) \\ &= \sum_{i=1}^{m} x_i(y_i - \frac{1}{1+exp(-\theta x_i)}) \\ &= \sum_{i=1}^{m} x_i(y_i - \hat{y})\end{aligned} \tag{0.1}$$

Where $\hat{y}$ is the predicted value.

To perform the gradient descent function,

Randomly initialize θ_0

While $||\theta_{t+1} - \theta_t|| > \epsilon$,

$\theta_{t+1} = \theta_t + \gamma \sum_{i=1}^{m} x_i(y - \hat{y})$

(2) Solution

Randomly shuffle the dataset

Randomly sample a batch of t data points $S_k = [x_1, ..., x_t], k = 1, 2, 3...$

For each k, apply the gradient descent function:

While $||\theta_{t+1} - \theta_t|| > \epsilon$,

$\theta_{t+1} = \theta_t + \gamma \sum_{i=1}^{t} x_i(y - \hat{y})$

(3) Solution

$$\begin{aligned}\frac{d^2l}{d\theta^2} &= \frac{dl}{d\theta}\sum_{i=1}^{m} x_i(y_i - \hat{y}) \\ &= \frac{dl}{d\theta}\sum_{i=1}^{m} x_i y_i - x_i\hat{y} \\ &= \frac{dl}{d\theta}\sum_{i=1}^{m} x_i\hat{y} \\ &= \frac{dl}{d\theta}\sum_{i=1}^{m} x_i \frac{1}{1+exp(-\theta x_i)} \\ &= \frac{dl}{d\theta}\sum_{i=1}^{m} \frac{-x_i^2\ exp(-\theta x_i)}{[1+exp(-\theta x_i)]^2} \leqslant 0\end{aligned} \tag{0.2}$$

The second order derivative is proved to be less or equal than zero. This proves that the original log-likelihood function is concave. Therefore, we should have a point which has the optimal maximized value.

Problem 2 (10+10+10 points)

0.0.1 Part One

(a)

Pred/Label	0	1
0	14	1
1	0	19

Table 1: Naive Bayes

Pred/Label	0	1
0	14	1
1	0	19

Table 2: KNN

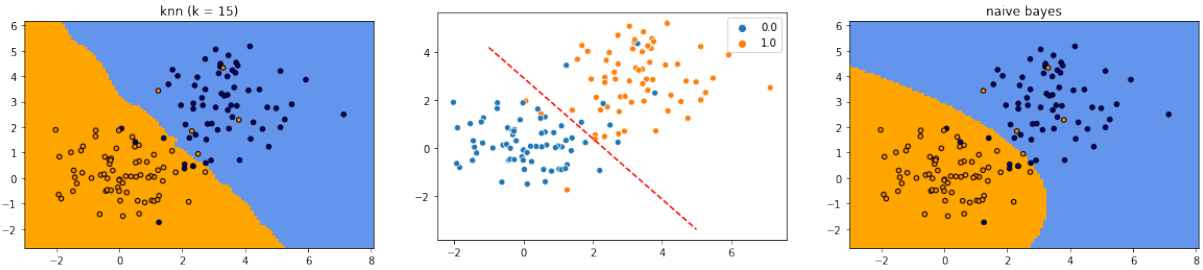
Pred/Label	0	1
0	14	1
1	0	19

Table 3: Logistic Regression

We can see that the three classifiers perform identically here. Only one divorce case has been predicted as "not divorce" here. A total hit rate of 97.0%. The step size of the logistic regression is set at 0.1, and the number of neighbors for KNN is 11. I tried again with different numbers of neighbors for the KNN and the results were still the same. However, for logistic regression, as I change the step size down to 0.02, the hit rate falls to 88.2%. I think this might be due to the property that it is a linear classifier, so it is very sensitive to step size, and as I decrease the step size, it might stuck at a local optimal which is why it has a worse hit rate.

The KNN on the other hand, is more robust because it considers multiple neighbors and assign classes based on actual existing information, rather than based on the estimates from previous iterations like logistic regression. Naive Bayes is also more robust due to similar reasons as long as the points are randomly distributed, and the feature dimensions do not have dependencies with each other, then assigning the class based on the normal distribution pdf should yield a pretty accurate result.

(b)



(a) KNN (b) Logistic Regression (c) Naive Bayes

Figure 1: Decision Boundaries of Three Classifiers

The different decision boundaries are shown above. Logistic regression is very obvious as it looks, which is a linear boundary. KNN due to its nature in the algorithm, takes into account neighboring points, and the neighboring points are randomly distributed, thus the boundary can be a little bit zig-zaggy. Naive bayes strictly follows the pdf curve, and because the pdf curve is smooth, so here the boundary is very smooth as well.

0.0.2 Part Two

In order to perform the naive bayes classification on the MNIST dataset, I tweaked the function used in the divorce dataset a little bit, which is illustrated in the python code.

Here, the hit rate for the three classifiers are:

- Naive Bayes: total hit rate = 0.88, 0 hit rate = 0.99, 1 hit rate = 0.75
- KNN: total hit rate = 0.99, 2 hit rate = 0.98, 6 hit rate = 0.99
- Logistic Regression: total hit rate = 0.97, 2 hit rate = 0.99, 6 hit rate = 0.95

Overall, KNN algorithm has the best performance, with the highest overall hit rate, as well as the highest hit rate across both classes. The reason here might due to the KNN uses the distance measure to avoid the redundancies in the data.

Problem 3 (10+10+10+10 points)

(1) Solution

$Pr(y = 0) = \frac{3}{7}$, $Pr(y = 1) = \frac{4}{7}$

(2) Solution

For spam messages:

$$Spam\ X_i = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{0.3}$$

$$Non - Spam\ X_i = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{0.4}$$

(3) Solution

$L(\theta_{0,k}) = l(\theta_{0,k}) + \mu_1(\sum_{k=1}^{15} \theta_{0,k} - 1)$

$L(\theta_{1,k}) = l(\theta_{1,k}) + \mu_2(\sum_{k=1}^{15} \theta_{1,k} - 1)$

$$\nabla L(\theta_{0,k}) = \sum_{i=1}^{m} \sum_{k=1}^{d} \frac{X_k^i}{\theta_{0,k}} + \mu_1 \tag{0.5}$$

$$\nabla L(\theta_{1,k}) = \sum_{i=1}^{m} \sum_{k=1}^{d} \frac{X_k^i}{\theta_{1,k}} + \mu_2 \tag{0.6}$$

When the two gradient functions equal to zero, we should be able to get the optimal point for maximum log-likelihood. Therefore,

$\sum_{i=1}^{m} \sum_{k=1}^{d} \frac{X_k^i}{\theta_{0,k}} = -\mu_1$

$\sum_{i=1}^{m} \sum_{k=1}^{d} \frac{X_k^i}{\theta_{1,k}} = -\mu_2$

For each feature k in either class, we have:

$$\sum_{i=1}^{m} \frac{X_k^i}{\theta_{y^{(i)},k}} = -\mu_1 \tag{0.7}$$

For example, $y^{(i)} = 0$,

$$\theta_{0,k} = \frac{\sum_{i=1}^{m} X_k^i}{-\mu_1} \tag{0.8}$$

$$\sum_{k=1}^{15} \theta_{0,k} = \sum_{k=1}^{15} \frac{\sum_{i=1}^{m} X_k^i}{-\mu_1} = 1 \tag{0.9}$$

Plug in the values of X_k^i, and then solve for $\mu_1 = -9$. Using the same methodology, I derive the μ_2 value for the "class 1" , which is -15.

Next, plug in the μ_1 and μ_2 values into equation (0, 8), I get: $\theta_{0,1} = \frac{3}{9}$, $\theta_{0,7} = \frac{1}{9}$, $\theta_{1,1} = \frac{1}{15}$, $\theta_{1,15} = \frac{1}{15}$,

(3) Solution

$Pr(today\ is\ secret = 0) = \frac{3}{7} * \frac{1}{9} * \frac{1}{9} * \frac{1}{3} = \frac{1}{567}$

$Pr(today\ is\ secret = 1) = \frac{4}{7} * \frac{1}{15} * \frac{1}{15} * \frac{1}{15} = \frac{4}{23625}$

$Pr(today\ is\ secret = 0) > Pr(today\ is\ secret = 1)$

The mail is more likely to be a spam.