Yawen Han (yh3069)
COMS W4271
HW02
March 06, 2019

Problem 1

$$\text{set } \mathrm{L}(\pi, \lambda_{y,d}) = \sum_{i=1}^{n}\left[\ln p(y_i|\pi) + \sum_{d=1}^{D}\left(\ln p(\lambda_{0,d}) + \ln p(\lambda_{1,d}) + \ln p(x_{i,d}|\lambda_{y_i,d})\right)\right]$$

(a) derive $\hat{\pi}$ by maximizing $\mathrm{L}(\pi, \lambda_{y,d})$ above

As only $\ln p(y_i|\pi)$ contains the parameter π, we can maximize $\mathrm{L}(\pi, \lambda_{y,d})$ through π by only maximizing $\ln p(y_i|\pi)$, and treat $\sum_{d=1}^{D}\left(\ln p(\lambda_{0,d}) + \ln p(\lambda_{1,d}) + \ln p(x_{i,d}|\lambda_{y_i,d})\right)$ as a constant,

That is,

$$\arg\max_{\pi} \mathrm{L}(\pi, \lambda_{y,d}) = \arg\max_{\pi} \sum_{i=1}^{n}\left[\ln p(y_i|\pi) + \sum_{d=1}^{D}\left(\ln p(\lambda_{0,d}) + \ln p(\lambda_{1,d}) + \ln p(x_{i,d}|\lambda_{y_i,d})\right)\right]$$

$$= \arg\max_{\pi} \sum_{i=1}^{n} \ln p(y_i|\pi)$$

$$= \arg\max_{\pi} \sum_{i=1}^{n} (y_i \ln \pi + (1 - y_i)\ln(1 - \pi))$$

then, set

$$l(\pi) = \sum_{i=1}^{n} (y_i \ln \pi + (1 - y_i)\ln(1 - \pi))$$

$$\frac{\partial \mathrm{l}}{\partial \pi} = \frac{\partial}{\partial \pi}\sum_{i=1}^{n} (y_i \ln \pi + (1 - y_i)\ln(1 - \pi))$$

$$= \sum_{i=1}^{n}\left(\frac{y_i}{\pi} + \frac{1 - y_i}{1 - \pi}\right)$$

$$= \sum_{i=1}^{n}\left(\frac{(1-\pi)y_i - \pi(1 - y_i)}{\pi(1-\pi)}\right)$$

$$= \frac{\sum_{i=1}^{n}(y_i - \pi)}{\pi(1-\pi)} = 0$$

Thus,

$$\hat{\pi} = \frac{\sum_{i=1}^{n} y_i}{n}$$

(b) derive $\widehat{\lambda_{y,d}}$ by maximizing $\mathrm{L}(\pi, \lambda_{y,d})$ above

As $\ln p(y_i|\pi)$ does not contain the parameter $\lambda_{y,d}$, we can maximize $\mathrm{L}(\pi, \lambda_{y,d})$ through $\lambda_{y,d}$ by maximizing $\sum_{d=1}^{D}(\ln p(\lambda_{0,d}) + \ln p(\lambda_{1,d}) + \ln p(x_{i,d}|\lambda_{y_i,d}))$, and treat $\ln p(y_i|\pi)$ as a constant.

That is,

$$\arg\max_{\lambda_{y,d}} \mathrm{L}(\pi, \lambda_{y,d}) = \arg\max_{\lambda_{y,d}} \sum_{i=1}^{n} \left[\ln p(y_i|\pi) + \sum_{d=1}^{D} (\ln p(\lambda_{0,d}) + \ln p(\lambda_{1,d}) + \ln p(x_{i,d}|\lambda_{y_i,d})) \right]$$

$$= \arg\max_{\pi} \sum_{i=1}^{n} \sum_{d=1}^{D} (\ln p(\lambda_{0,d}) + \ln p(\lambda_{1,d}) + \ln p(x_{i,d}|\lambda_{y_i,d}))$$

$$= \arg\max_{\pi} \sum_{d=1}^{D} \ln p(\lambda_{0,d}) + \ln p(\lambda_{1,d}) + \sum_{i=1}^{n} \ln p(x_{i,d}|\lambda_{y_i,d})$$

For each dimension d,

$$\ln p(\lambda_{0,d}) = \ln \lambda_{0,d} - \lambda_{0,d} - \ln \Gamma(2),$$
$$\ln p(\lambda_{1,d}) = \ln \lambda_{1,d} - \lambda_{1,d} - \ln \Gamma(2),$$

as the conditional independence assumption states that features are independent of each other given the class,

$$\sum_{i=1}^{n} \ln p(x_{i,d}|\lambda_{y_i,d}) = \sum_{i=1}^{n} \ln p(x_{i,d}|\lambda_{0,d})\, 1\{y_i = 0\} + \ln p(x_{i,d}|\lambda_{1,d}) 1\{y_i = 1\}$$

$$= \sum_{i=1}^{n} (x_{i,d} \ln(\lambda_{0,d}) - \lambda_{0,d} - \ln(x_{i,d}!))\, 1\{y_i = 0\} +$$
$$(x_{i,d} \ln(\lambda_{0,d}) - \lambda_{0,d} - \ln(x_{i,d}!)\, 1\{y_i = 1\})$$

then, set

$$l(\lambda_{y,d}) = \ln p(\lambda_{0,d}) + \ln p(\lambda_{1,d}) + \sum_{i=1}^{n} \ln p(x_{i,d}|\lambda_{y_i,d})$$
$$= \ln \lambda_{0,d} - \lambda_{0,d} - \ln \Gamma(2) + \ln \lambda_{1,d} - \lambda_{1,d}$$
$$- \ln \Gamma(2) + \sum_{i=1}^{n} (x_{i,d} \ln(\lambda_{0,d}) - \lambda_{0,d} - \ln(x_{i,d}!))\ 1\{y_i = 0\}$$
$$+ (x_{i,d} \ln(\lambda_{0,d}) - \lambda_{0,d} - \ln(x_{i,d}!)\, 1\{y_i = 1\})$$

for $\widehat{\lambda_{0,d}}$:

$$\frac{\partial \mathrm{l}}{\partial \lambda_{0,d}} = \frac{\partial}{\partial \lambda_{0,d}} l(\lambda_{y,d})$$
$$= \frac{1}{\lambda_{0,d}} - 1 + \sum_{i=1}^{n} \left(\frac{x_{i,d}}{\lambda_{0,d}} - 1 \right) 1\{y_i = 0\}$$

$$= \frac{1 + \sum_{i=1}^{n} x_{i,d} 1\{y_i = 0\}}{\lambda_{0,d}} - \left(1 + \sum_{i=1}^{n} 1\{y_i = 0\}\right) = 0$$

thus,

$$\widehat{\lambda_{0,d}} = \frac{1 + \sum_{i=1}^{n} x_{i,d} 1\{y_i = 0\}}{1 + \sum_{i=1}^{n} 1\{y_i = 0\}}$$

Similarly,
for $\widehat{\lambda_{1,d}}$:

$$\widehat{\lambda_{1,d}} = \frac{1 + \sum_{i=1}^{n} x_{i,d} 1\{y_i = 1\}}{1 + \sum_{i=1}^{n} 1\{y_i = 1\}}$$

In conclusion,
For each dimension d,

$$\widehat{\lambda_{y,d}} = \frac{1 + \sum_{i=1}^{n} x_{i,d} 1\{y_i = y\}}{1 + \sum_{i=1}^{n} 1\{y_i = y\}}$$

with y ∈ {0,1}

Problem 2

(a)

Naïve Bayes Classifier:

the Bayes classifier observes a new x_0 and predicts y_0 as

$$y_0 = \arg\max_y p(y_0 = y|\pi) \prod_{d=1}^{D} p\left(x_{0,d}|\lambda_{y,d}\right)$$

according to problem1,

$$p(y_0 = y|\pi) = \hat{\pi}^{y_0}(1 - \hat{\pi})^{(1-y_0)}$$

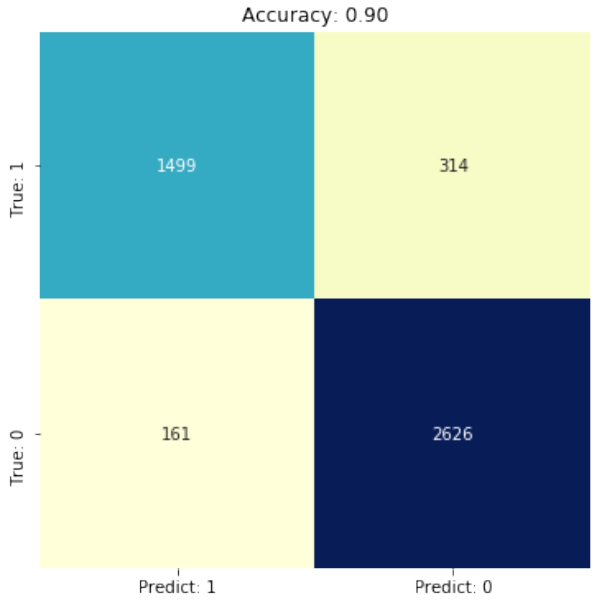
and

$$\prod_{d=1}^{D} p\left(x_{0,d}|\lambda_{y,d}\right) = \prod_{d=1}^{D} p\left(x_{0,d}|y\right) \cdot p\left(\lambda_{y,d}\right)$$

$$= \prod_{d=1}^{D} \frac{\left(\widehat{\lambda_{y,d}}\right)^{x_{0,d}} \cdot e^{-\widehat{\lambda_{y,d}}}}{x_{0,d}!} \cdot \frac{\beta^{\alpha}}{\Gamma(\alpha)} \cdot \widehat{\lambda_{y,d}}\, e^{-\widehat{\lambda_{y,d}}}$$

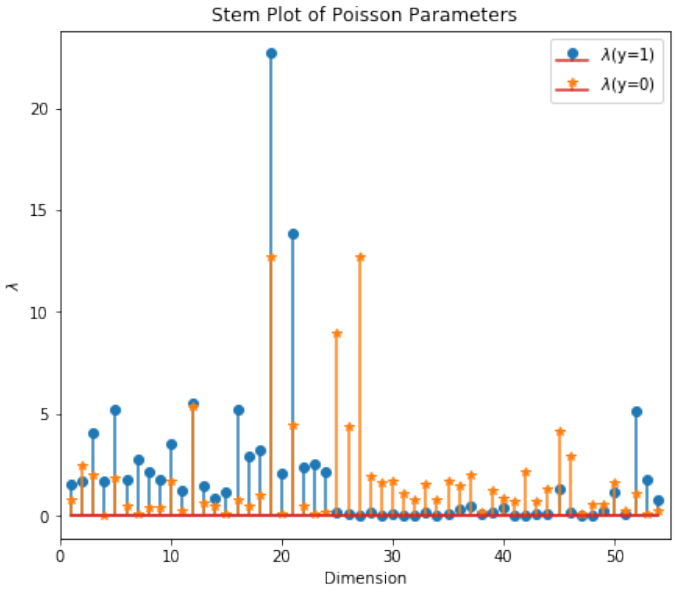
$$= \prod_{d=1}^{D} \frac{\beta^{\alpha}}{\Gamma(\alpha) \cdot x_{0,d}!} \left(\widehat{\lambda_{y,d}}\right)^{1+x_{0,d}} \cdot e^{-2\widehat{\lambda_{y,d}}}$$

using the above formula to predict y_0 for the test set, and the confusion matrix of the Naïve Bayes Classifier shown as below:



According to the confusion matrix above, TP=1499, FN=314, FP=161, TN=2626. As the title shows, the accuracy of Naïve Bayes Classifier is 0.90.

(b)
Stem plot:



Observation:

I. For both dimension 16 and 52, $\lambda_{y=1}$ is obviously greater than $\lambda_{y=0}$.
II. According to the README file, dimension 16 is word "free", $\lambda_{y=1} = 5.2136, \lambda_{y=0} = 0.7397, and\ the\ difference\ is\ |\lambda_{y=1} - \lambda_{y=0}| = 4.4739$

III. According to the README file, dimension 52 is word "!", $\lambda_{y=1} = 5.1296, \lambda_{y=0} = 1.0961, and\ the\ difference\ is\ |\lambda_{y=1} - \lambda_{y=0}| = 4.0335$

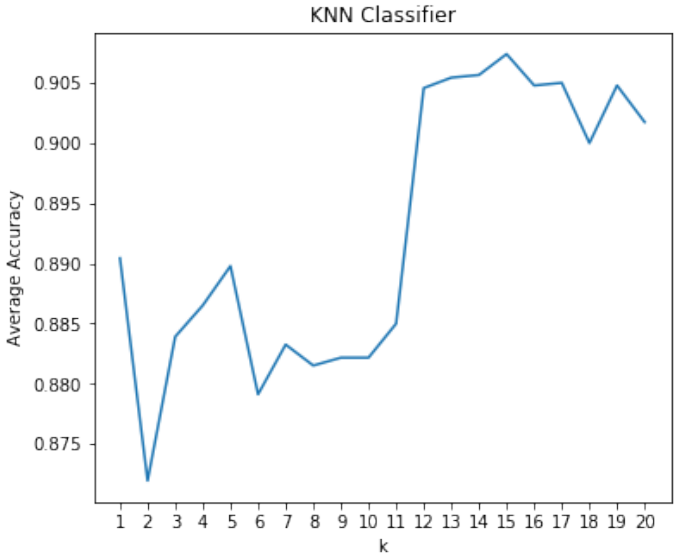
In conclusion, the email with words “free” or “!” is more likely to be classified as “spam” emails than “non-spam” emails.

(c)

KNN-classifier:

Develop the KNN classifier using ℓ_1 distance as $d_{\ell_1}(\vec{x}, \vec{y}) = \sum_{i=1}^{n} |x_i - y_i|$, and predict the test data as the majority class in its k nearest neighbors.

plot the prediction accuracy as a function of k



According to the plot above, the accuracy of the developed KNN-classifier is increasing with k at first, then decreasing when k gets larger. The accuracy is maximized at k =16, with maximum accuracy = 0.9074.

(d)

Steepest Ascent algorithm:

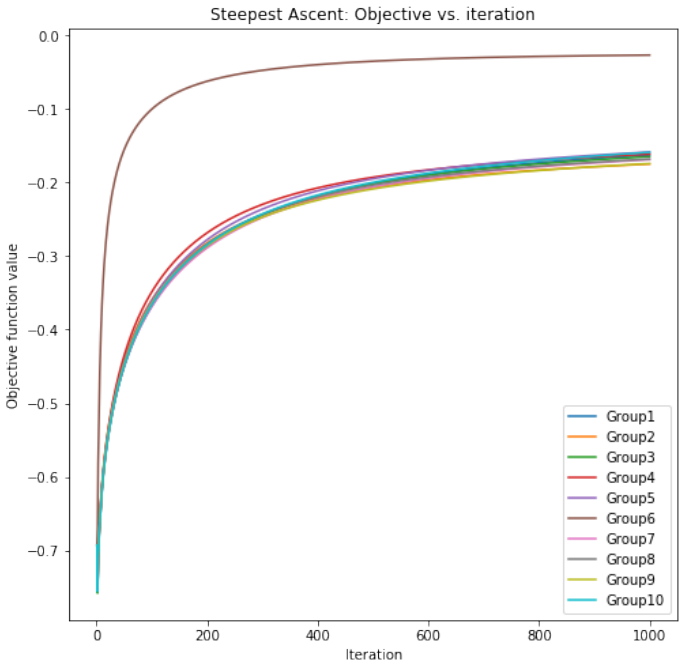
In the python code, the gradient was built using the following formula:

$$\nabla_w \mathcal{L} = \sum_{i=1}^{n} \left(1 - \sigma_i(y_i \cdot w)\right) y_i x_i$$

and w was updated each iteration by

$$w^{(t+1)} = w^{(t)} + \eta \nabla_w \mathcal{L}$$

plot the logistic regression objective training function per iteration for each of the 10 training runs (1000 iterations in total)



According to the plot above, the objective function is improved iteration by iteration, and achieve a stable stage after 500 iterations.

(e)

Newton's Method:

In the python code, the gradient and hessian were built using the following formula:

$$\nabla_w \mathcal{L} = \sum_{i=1}^{n} \left(1 - \sigma_i(y_i \cdot w)\right) y_i x_i$$

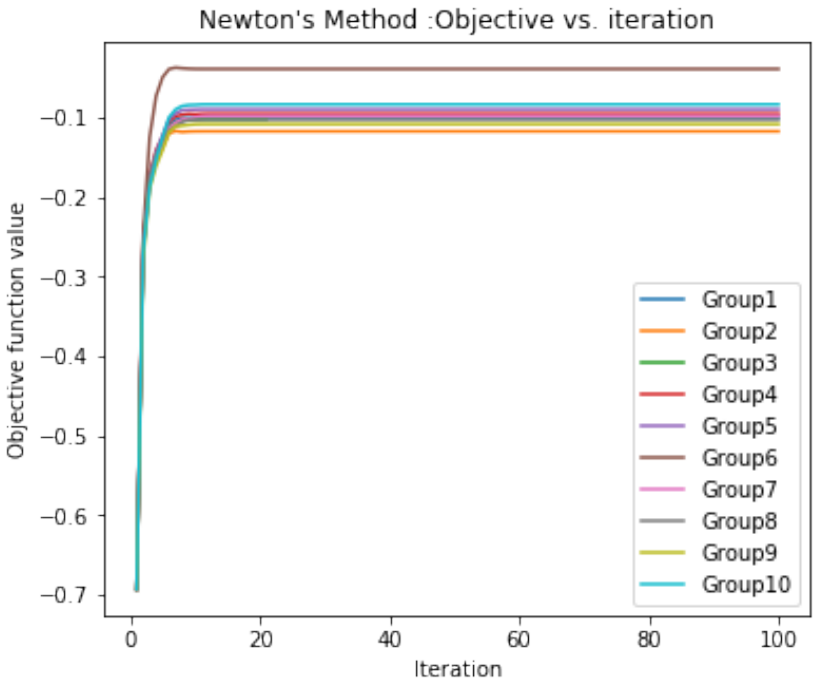
$$\nabla_w^2 \mathcal{L} = \nabla_w \left(\sum_{i=1}^{n} \left(1 - \sigma_i(y_i \cdot w)\right) y_i x_i \right)$$

$$= -\sum_{i=1}^{n} \sigma_i(y_i \cdot w)\left(1 - \sigma_i(y_i \cdot w)\right) x_i x_i^T$$

and w was updated each iteration by

$$w^{(t+1)} = w^{(t)} - (\nabla_w^2 \mathcal{L})^{-1} \nabla_w \mathcal{L}$$

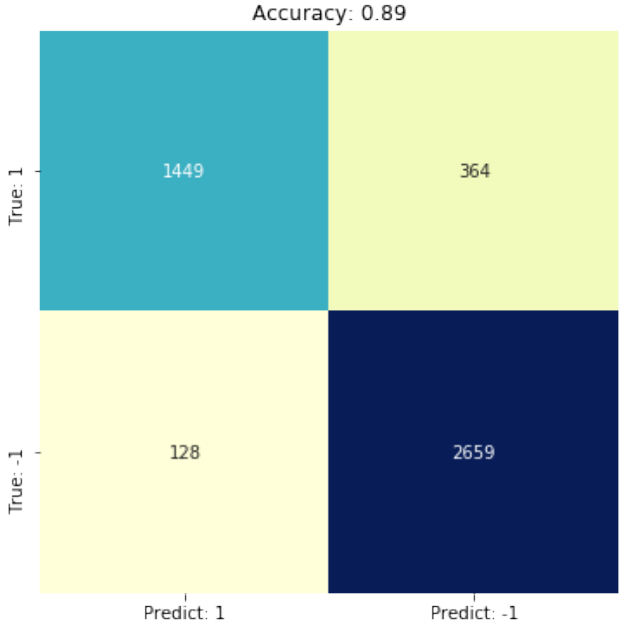
plot the logistic regression objective training function per iteration for each of the 10 training runs (100 iterations in total)



According to the plot above, the objective function is improved iteration by iteration, and achieve a stable stage after 10 iterations.

(f)

Newton's Method: Confusion matrix



According to the confusion matrix above, TP=1449, FN=364, FP=128, TN=2659. As the title shows, the accuracy of Newton's Method is 0.89.