EE5907-Assignment1-Report

This report is for the first assignment of EE5907 .It contains 4 major parts ,and each major part is for one question in the assignment .

1 Beta-binomial Naive Bayes

1.1 Algorithm and code

In this part ,a classifier based on Beta-Binomial Naive Bayes is built to handle the binarized data from given dataset and finally predict whether an email is or is not a spam.

Based on training dataset $D(x_{1:N},y_{1:N})$,in order to predict the class label y of a specific sample x ,we need to compute the posterior possibility of all potential class labels of x ,and then choose the highest one:

$$p(\tilde{y}=c|\tilde{x},D) \propto p(\tilde{y}=c|y_{1:N}) \prod_{j=1}^{D} p(\tilde{x}_j|x_{i\in c,j},\tilde{y}=c) \tag{1-1}$$

In this formula ,we drop the denominator and assume features are conditionally independent given class label.We predict the class label with the highest posterior probability ,which is called the MAP estimate.

For computation simplicity ,we use log function as it is a monotonic function ,so the final result won't change:

$$\log p(\tilde{y}=c|\tilde{x},D) \propto \log p(\tilde{y}=c|y_{1:N}) + \sum_{j=1}^{D} \log p(\tilde{x}_j|x_{i\in c,j},\tilde{y}=c) \tag{1-2}$$

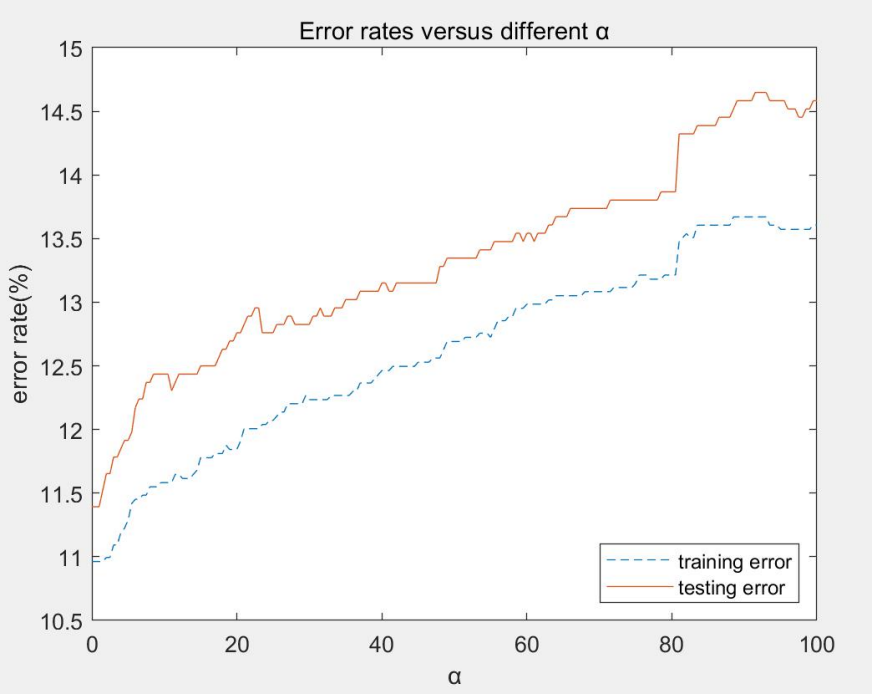
As there are many emails ,the second term of formula (1-2) can be replaced by maximum likelihood estimate(ML estimate) for the class prior distribution ,then we get:

$$\log p(\tilde{y}=c|\tilde{x},D) \propto \log p(\tilde{y}=c|\lambda^{ML}) + \sum_{j=1}^{D} \log p(\tilde{x}_j|x_{i\in c,j},\tilde{y}=c) \tag{1-3}$$

Now based on formula (1-3) ,I fit the classifier on the training data and then do prediction. My code flow chart is as follows:

1.2 Result and analysis

1.2.1 Plots of training and test error rates versus α



1.2.2 What do you observe about the training and test errors as α change?

First ,generally both the training and testing errors will increase as α increases .I thinks it is because the prior Beta(α， α) we set on the feature distribution does not match the true feature distribution ,moreover as α increases ,the influence of the prior Beta(α， α) on the final prediction will also increase .So as α increases ,the error rates also increase.

Second ,the testing error is always higher than training error given the same α .

1.2.3 Training and testing error rates for α = 1，10 and 100

When α=1, training error rate is 10.9625% ,testing error rate is 11.3932%
When α=10, training error rate is 11.5823% ,testing error rate is12.4349%
When α=100 , training error rate is 13.6052%,testing error rate is14.5833%

2 Gaussian Naive Bayes

2.1 Algorithm and code

In this part ,a classifier based on Gaussian Naive Bayes is built to handle the log-transformed data from given dataset and finally predict whether an email is or is not a spam.

As in this question ,we assume features subject to Gaussian distribution ,we first need to compute the ML estimate of conditional mean(μ) and variance(σ^2) of each feature based on training data:

$$(\hat{\mu}, \hat{\sigma}^2) \stackrel{\triangle}{=} \underset{\mu,\sigma^2}{\operatorname{argmax}}\, p(x_1, \cdots, x_N|\mu, \sigma^2) \tag{2-1}$$

Based on conditional independent among features and for computation simplicity ,we then get:

$$(\hat{\mu}, \hat{\sigma}^2) = \operatorname{argmax}_{\mu,\sigma^2} \sum_{n=1}^{N} \left[-\frac{(x_n-\mu)^2}{2\sigma^2} - \log\sqrt{2\pi\sigma^2} \right] \tag{2-2}$$

Differentiate separately μ and σ ,and set to 0 ,we get:

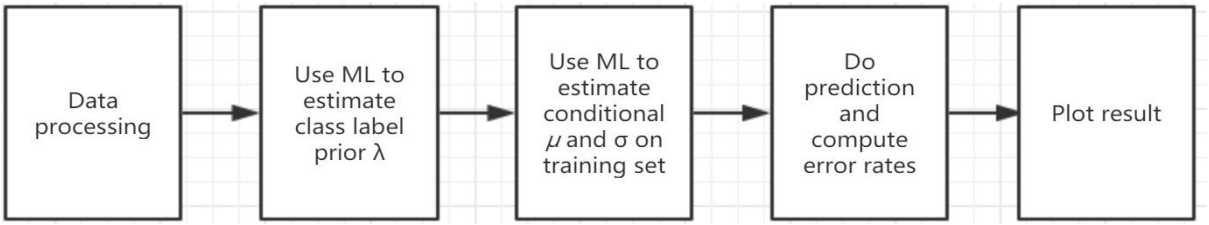
$$\frac{\partial L}{\partial \mu} = \frac{\partial}{\partial \mu}\left(\sum_{n=1}^{N} -\frac{(x_n-\mu)^2}{2\sigma^2} \right) = \sum_{n=1}^{N} \frac{(x_n-\mu)}{\sigma^2} = 0$$

$$\Longrightarrow \hat{\mu} = \frac{1}{N}\sum_{n=1}^{N} x_n \tag{2-3}$$

$$\frac{\partial L}{\partial \sigma} = \frac{\partial}{\partial \sigma}\left(\sum_{n=1}^{N} -\frac{(x_n-\mu)^2}{2\sigma^2} - N\log\sigma \right) = \sum_{n} \frac{(x_n-\mu)^2}{\sigma^3} - \frac{N}{\sigma} = 0$$

$$\Longrightarrow \hat{\sigma}^2 = \frac{1}{N}\sum_{n=1}^{N}(x_n-\mu)^2 = \frac{1}{N}\sum_{n=1}^{N}(x_n-\hat{\mu})^2 \tag{2-4}$$

As it is the same to Q1 ,I still use ML to estimate the class label prior λ , then based on formula (2-3) and (2-4) , fit the classifier on training set and do prediction. My code flow chart is as follows:



2.2 Result and analysis

2.2.1 Training and testing error rates for log-transformed data.

The training error rates is 16.6721%.

The testing error rates is 18.3594%.

3 Logistic regression

3.1 Algorithm and code

In this part ,a classifier based on logistic regression with l_2 regularization is built to handle the log-transformed data from given dataset and finally predict whether an email is or is not a spam.

We first specify a discriminative model *p(y|x,w)* ,then based on given training set ,we want find the *w* that satisfy:

$$\hat{w} = \underset{w}{\operatorname{argmax}}\, p(y_{1:N}|x_{1:N}, w) \tag{3-1}$$

Use log function and assume samples are independent ,we get:

$$\hat{w} = \underset{w}{\operatorname{argmin}} - \sum_{i=1}^{N} \log p(y_i|x_i, w) \triangleq \underset{w}{\operatorname{argmin}}\, NLL(w) \tag{3-2}$$

So now our goal is to find a *w* which can minimize *NLL(w)* ,the *NLL(w)* can be expressed in this way:

$$\begin{aligned}
&\log p(y_i = 1|x_i, w) = \log \frac{1}{1 + \exp(-w^T x_i)} = \log \mu_i \\
&\log p(y_i = 0|x_i, w) = \log(1 - p(y_i = 1|x_i, w)) = \log(1 - \mu_i) \\
&NLL(w) = -\sum_{i=1}^{N} \log p(y_i|x_i, w) = -\sum_{i=1}^{N} [y_i \log \mu_i + (1 - y_i) \log(1 - \mu_i)]
\end{aligned} \tag{3-3}$$

Then we calculate derivatives(gradient matrix and hessian matrix):

$$\begin{aligned}
g &= \frac{d}{dw} NLL(w) = \sum_{i=1}^{N} (\mu_i - y_i) x_i = X^T(\mu - y) \\
H &= \frac{d}{dw} g(w)^T = \sum_{i=1}^{N} \mu_i(1 - \mu_i) x_i x_i^T = X^T S X,
\end{aligned} \tag{3-4}$$

To free the decision boundary from origin and reduce overfitting ,we add a bias term and some constrains on the model(but not on the bias term).Here we use l_2 regularization.Then the gradient matrix and hessian matrix become:

$$\begin{aligned}
g_{reg}(\mathbf{w}) &= g(\mathbf{w}) + \lambda \begin{pmatrix} 0_{1\times 1} \\ w_{D\times 1} \end{pmatrix} \\
H_{reg}(\mathbf{w}) &= H(\mathbf{w}) + \lambda \begin{pmatrix} 0_{1\times 1} & \cdots \\ \vdots & I_{D\times D} \end{pmatrix}
\end{aligned}$$

row is all zero

column is all zero

(3-5)

And the *NLL(w)* becomes:

$$NLL_{reg}(\mathbf{w}) = NLL(\mathbf{w}) + \frac{1}{2}\lambda \mathbf{w}^T \mathbf{w} \tag{3-6}$$

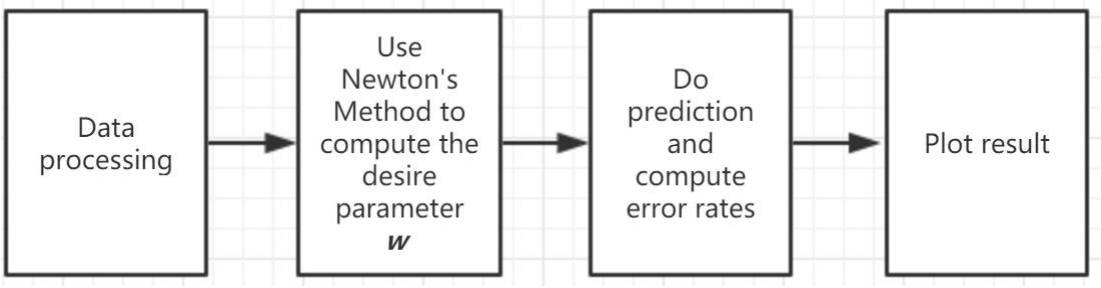
In formula (3-5) and (3-6) ,the bold ***w*** means *w* with a bias term and the subscript *reg* means with regularization.We can deem λ as the degree of regularization.

We use Newton's Method to find the desired ***w*** , first we initialize ***w*** as a zero vector , then we repeat computing:

$$\mathbf{w}_{k+1} = \mathbf{w}_k - H_k^{-1} g_k \tag{3-7}$$

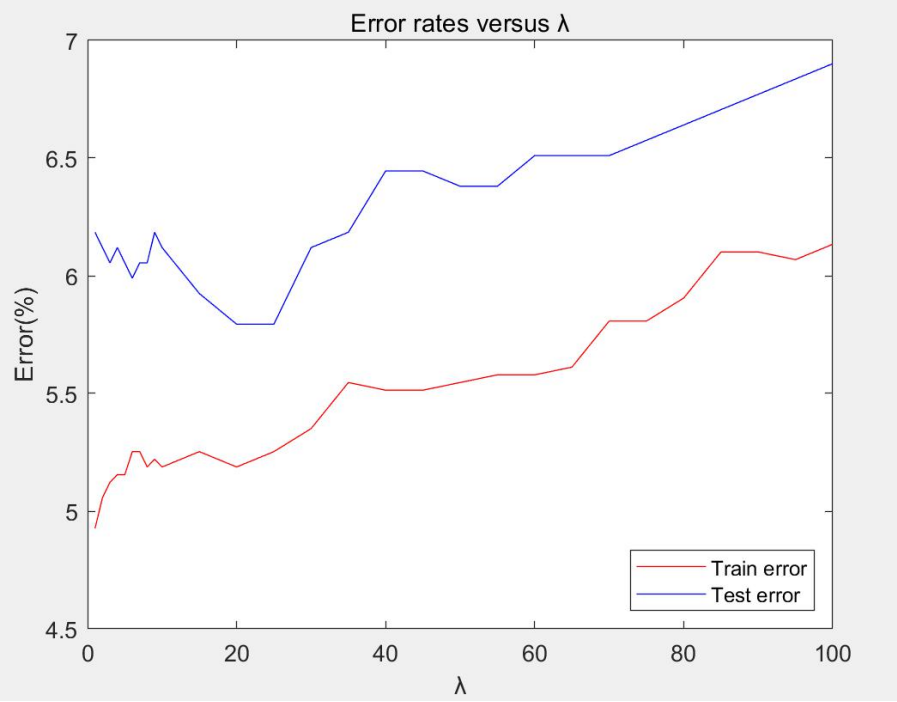
Until convergence. In my code ,it will be seen as convergence if the difference of $NLL_{reg}(\boldsymbol{w})$ between 2 successive iteration is less than a threshold(0.01).

Finally I use the ***w*** to do prediction and my code flow chart is as follows:



3.2 Result and analysis

3.2.1 Plots of training and test error rates versus λ



3.2.2 What do you observe about the training and test error as λ changes?

For training set , the error will generally increase as λ increases.

For test set ,when λ is less than 20, the error will remains or decrease slightly as λ increases . And when λ gets bigger than 20 , the error will basically increase as λ increases.

Test error is always higher than training error given a specific λ.

3.2.3 Training and testing error rates for λ = 1, 10 and 100.

When λ = 1，training error rate is 4.9266%,test error rate is 6.1849%

When λ = 10，training error rate is 5.1876%,test error rate is 6.1198%

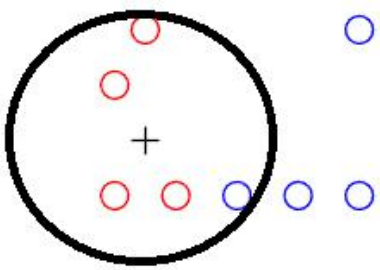
When λ = 100，training error rate is 6.1338%,test error rate is 6.9010%

4 K-Nearest Neighbors

4.1 Algorithm and code

In this part ,I will implement a KNN classifier to handle the log-transformed data from given dataset with Euclidean distance as the measurement of distance between samples.

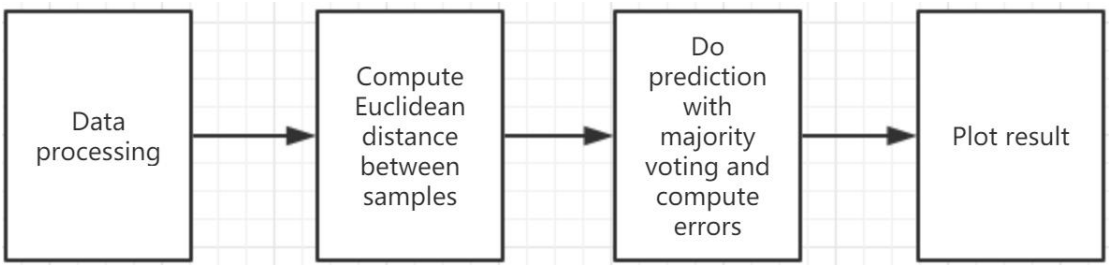
The basic idea behind this classifier is very simple and straightforward(so I think there is no need to write any formula): to predict the class label of a specific sample, we collect its K nearest surrounding training samples. Then use majority voting to do prediction.A simple illustration is shown below:



In this figure red circle and blue circle represent training samples of 2 different class labels .We want to predict the class label of the test sample‘+’ .If we set K = 5 ,then it is easy to see that there are 4 red circles among its 5 nearest neighbors ,so we predict this ‘+’ as the same class label to red circle.

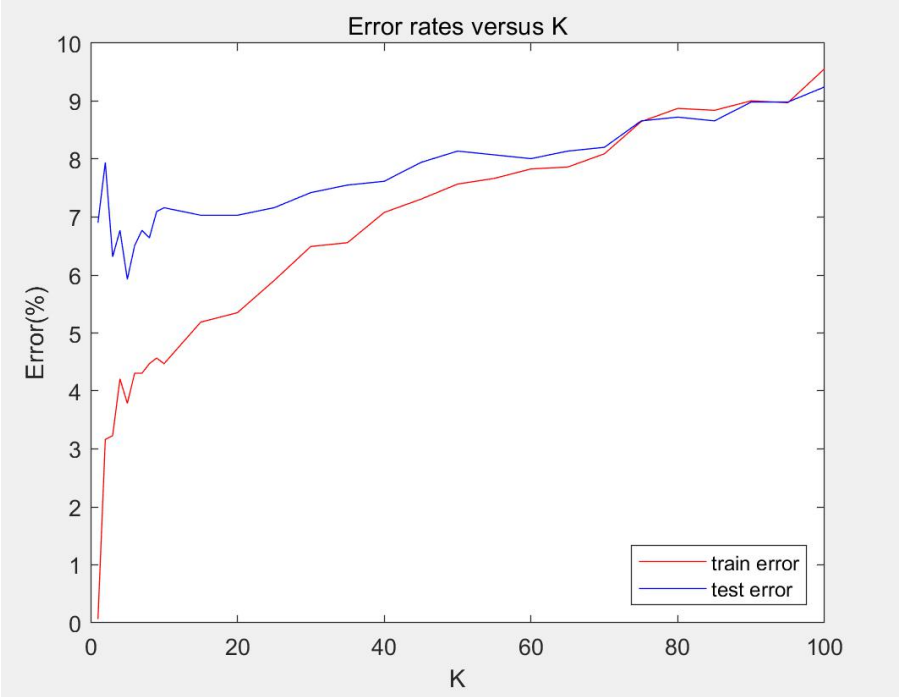
What we should pay attention to is that if K is an even number ,and among the K nearest neighbors of a test sample ,there are half number of training samples from class 0 and another half number of training samples from class 1, in this case ,I will predict this test sample to be class 1.

My code flow chart is as follows:



4.2 Result and analysis

4.2.1 Plots of training and test errors versus K



4.2.2 What do you observe about the training and test errors as K change?

For training set , in general the error will increase as K increases ,and theoretically the error will be 0 when K = 1 , because the Euclidean distance between every training sample and itself is 0 ,or you can say that the 1 nearest neighbor of every training sample is itself ,so there will be no error when K = 1.

For test set , when K is about less than 10 , the error will fluctuate as K increases. When K is bigger than 10 ,the error will generally increase as K increases.

Mostly test error is bigger than training error ,when K is very small ,the gap between test error and training error is very big , as K increases ,the gap gets smaller and after K is more than 70 , the test error line and training error line

cross with each other and the test error and training error are very closed to each other.

4.2.3 Training and test error rates for K = 1, 10, 100.

When K = 1, training error rate is 0.0653%, test error rate is 6.9010%.
When K = 10, training error rate is 4.4698%, test error rate is 7.1615%.
When K = 100, training error rate is 9.5595%, test error rate is 9.2448%.

5 Survey

Roughly ,I spent 3 hours on Q1 ,3 hours on Q2 ,10 hours on Q3 and 2 hours on Q4. So I spent nearly 20 hours to finish this assignment.