EE5907 Pattern Recognition

Assignment 1

Liu Xingyu
A0116430W

National University of Singapore

Electrical and Computer Engineering

29 Feb 2020

Q1. Beta-binomial Naive Bayes (24%)

Since it is Beta-binomial Naive Bayes, so the we assume all data samples are independent and all the 57 features are binary, either 0 or 1. And the result is also binary, 0 and 1. 0 represents not spam emails and 1 represents spam emails. In the experiment, we will test different α in Beta(α, α) distribution for features. α = {0, 0.5, 1, 1.5, 2, …, 100}. Fit the classifier and get the training and testing errors for different α values.

Naive Bayes classifier can be estimated by following equation:

$$p(\tilde{y} = c \mid \tilde{x}, D) \propto logp(\tilde{y} = c \mid \lambda^{ML}) + \sum_{j=1}^{D} log\, p(\tilde{x_j} \mid x_{i \in cj}, \tilde{y} = c)$$

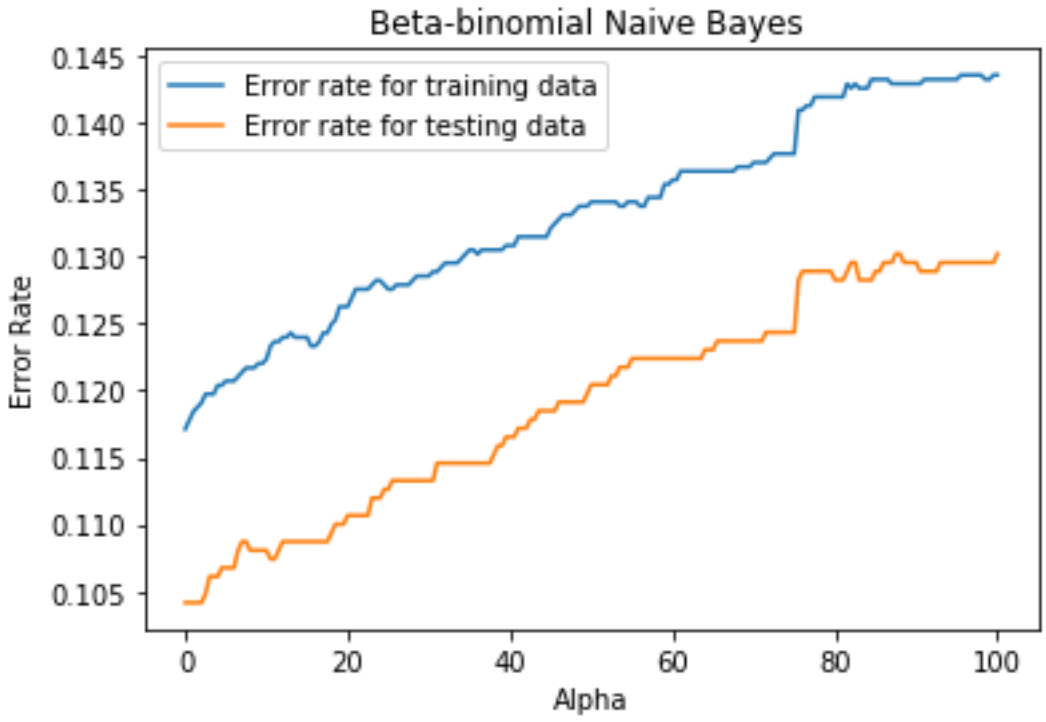
$log\, p(\tilde{y} = 0 \mid \lambda^{ML}) = \lambda^{ML}$ and $log\, p(\tilde{y} = 1 \mid \lambda^{ML}) = 1 - \lambda^{ML}$
λ can be estimated using ML (Maximum likelihood). λ^{ML} can be calculated by N_1 / N where N_1 denotes the total number of class 1 samples while N denotes the total number of training samples.

For features, using posterior predictive. $log\mathrm{p}(\tilde{x_j} \mid x_{i \in cj}, \tilde{y} = c)$ can be estimated that

$$p(\tilde{x} = 1 \mid D) = \frac{N_1 + a}{N + a + b}$$

Where N_1 denotes the number that feature c is 1 and N denotes the number of class 1. In this assignment, a = b = α. And we will calculate all the $log\, p(\tilde{x} = 1 \mid D)$ and $logp(\tilde{x} = 0 \mid D)$ for all features. Then summing the prior $log\, p(\tilde{y} = 0 \mid \lambda^{ML})$ and $log\, p(\tilde{y} = 1 \mid \lambda^{ML})$ to get $p(\tilde{y} = 1 \mid \tilde{x}, D)$ and $p(\tilde{y} = 0 \mid \tilde{x}, D)$. Finally compare the probability, the predicting result will be the one with higher probability. Then repeat the same experiment using different α.

- Plots of training and test error rates versus α



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

We can see that as α increases, the error rate for both training data and testing data increases. When α = 0, the posterior predictive $p(\tilde{x} = 1 \mid D) = N_1 / N$ which is the training data sample itself. When α increase, the $p(\tilde{x} = 1 \mid D)$ will be more dominated by the value α which will cause larger error rate.

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

α	1	10	100
Training	0.118434	0.122349	0.143556
Testing	0.104167	0.108073	0.130208

Output of code:

```
Training error rates for alpha 1 is  0.11843393148450244
Training error rates for alpha 10 is  0.12234910277324633
Training error rates for alpha 100 is  0.14355628058727568
Testing error rates for alpha 1 is  0.10416666666666667
Testing error rates for alpha 10 is  0.10807291666666667
Testing error rates for alpha 100 is  0.13020833333333334
```

Q2. Gaussian Naive Bayes (24%)

As Gaussian Naive Bayes distribution is used for the data samples. The data will be processed as log transformation. And 0.1 will be added after log transformation to avoid log 0.

The prior estimation will be the same as Q1 Beta-binomial Naïve Bayes. But Gaussian distribution will be used for posterior predictive.

For Gaussian distribution, ML estimation of mean $\hat{\mu} = \frac{1}{N}\sum_{n=1}^{N} x_n$ and variance $\hat{\sigma}^2 = \frac{1}{N}\sum_{n=1}^{N}(x_n - \hat{\mu})^2$.

And probability can be calculated using $Pr(x) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp[-0.5(x-\mu)]^2/\sigma^2]$
After training process, please refer to below results of error rate for both training and testing data.

Error rate for training data is: 0.16574225122349104
Error rate for testing data is: 0.17122395833333334

Output of code:

```
Error rate for training data is:  0.16574225122349104
Error rate for testing data is:  0.17122395833333334
```

Q3. Logistic regression (24%)

As Logistic regression with l_2 regularization is used in this section, regularization parameter value λ = {1, 2,…, 9, 10, 15, 20,…, 95, 100} will be used. The training data and testing data samples will be used log transformation and 0.1 will be added to avoid log 0 similar to Q2.

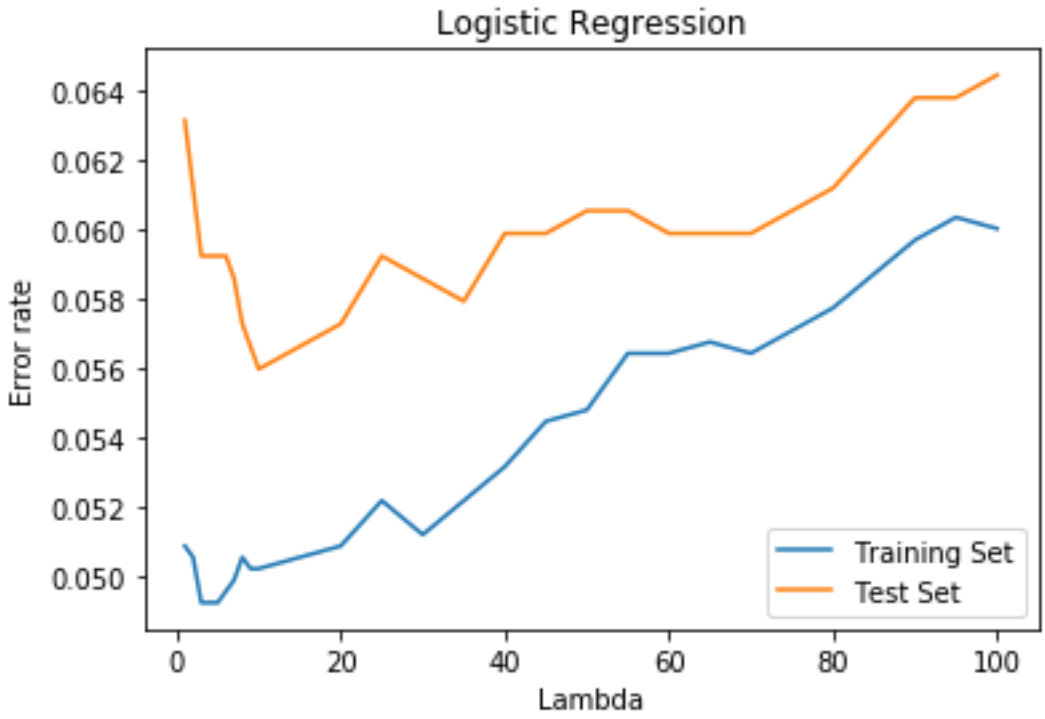
By comparing $p(y = 1|\boldsymbol{x}) = \frac{1}{1+e^{-\boldsymbol{w}^T\boldsymbol{x}}}$ and $p(y = 0|\boldsymbol{x}) = \frac{1}{1+e^{\boldsymbol{w}^T\boldsymbol{x}}}$. The label will be the one with higher probability. Then repeat the same experiment using different regularization parameter value λ.

To get the $(y = 1|x)$ and $p(y = 0|x)$. We need to get the weight $\boldsymbol{w}$.

$$NLL_{reg}(\boldsymbol{w}) = NLL(\boldsymbol{w}) + \frac{1}{2}\lambda \boldsymbol{w}^T\boldsymbol{w}$$

Use Newton's method to minimize the negative likelihood $NLL_{reg}(\boldsymbol{w})$ to get the $\boldsymbol{w}$.

- Plots of training and test error rates versus λ



- What do you observe about the training and test errors as λ change?

We can see that in generally, when λ increases, the error rate firstly will decrease and then start to increase. The error rate may drop a bit when λ increases but in generally, the error rate will increases.

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

λ	1	10	100
Training	0.050897	0.050245	0.060033
Testing	0.063151	0.055990	0.064453

Output of code:

```
lambda = 1
training error: 0.05089722675367048
testing error: 0.06315104166666663
lambda = 10
training error: 0.050244698205546445
testing error: 0.05598958333333337
lambda = 100
training error: 0.060032626427406144
testing error: 0.064453125
```

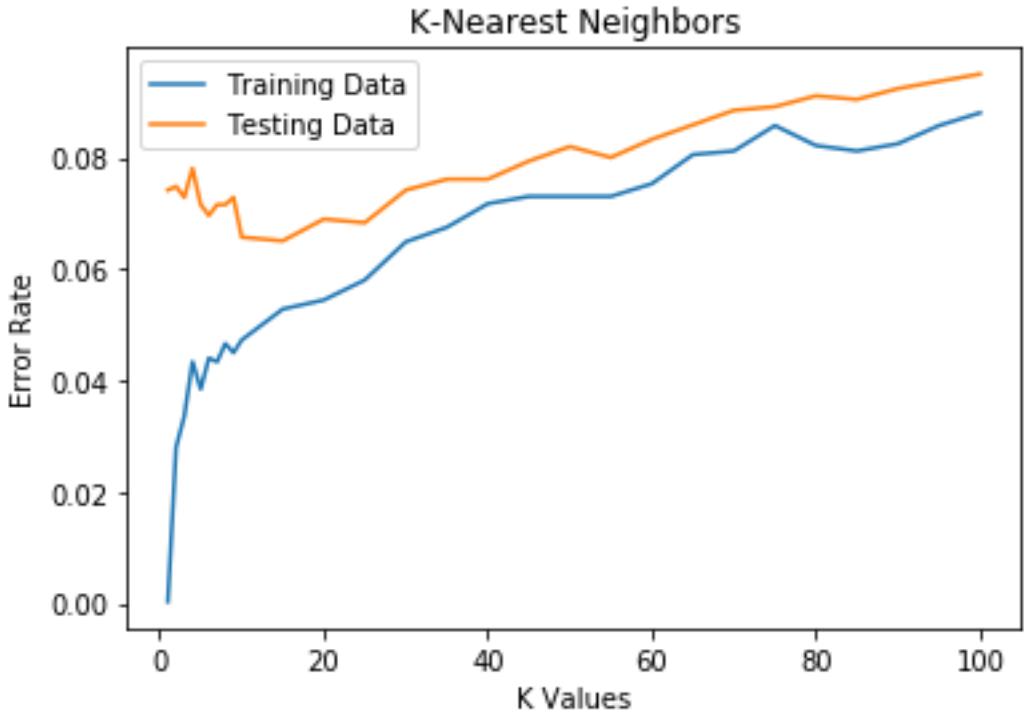
Q4. K-Nearest Neighbors (24%)

As K-Nearest Neighbors is used in this section. Different values of K = {1,2,⋯ ,9,10,15,20,⋯ ,95,100} will be used. The training data and testing data samples will be used log transformation and 0.1 will be added to avoid log 0 similar to Q2.

Minkowski $dist(a, b) = \left(\sum_{j=1}^{D} |a_j - b_j|^p\right)^{1/p}$. Euclidean if $p = 2$. In this section, Euclidean distance will be used. We get the Euclidean distance array that store the Euclidean distance between training samples and training samples, training samples and testing samples. Sort the distance array and get the index of the data samples in ascending order.

For each experiment of K, we choose the first K elements of the index array which indicates the K number of nearest neighbors for each sample data. Then count the number of each labels in the K number of nearest neighbors. Then the sample data will belong to the label that has more labels in the K number of nearest neighbors. That is how the new sample data is predicted. Then repeat the same experiment using different values of K.

- Plots of training and test error rates versus K



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

We can see that when K increases, the training data error rate increases rapidly from very small value until a moderately stable value. But the testing data error firstly decreases a bit and then increases in a moderate rate.

- Training and testing error rates for K = 1, 10 and 100.

K	1	10	100
Training	0.000326	0.047308	0.088091
Testing	0.074219	0.065755	0.095052

Output of code:

```
K = 1
training error: 0.0003262642740619902
testing error: 0.07421875
K = 10
training error: 0.04730831973898858
testing error: 0.06575520833333333
K = 100
training error: 0.08809135399673736
testing error: 0.09505208333333333
```

Q5. Survey (4%)

It took me about maybe 30 – 50 hours. I can't remember a specific number, so let me choose the average 40 hours.
This assignment is quite good in my opinion. The lectures are sometimes too abstract for me. And this assignment help me have a better understanding for the theory. I hope we can have more real examples in lectures. Thanks.