Faculty of Engineering

EE5907: Pattern Recognition
Assignment 1 Report
for assignment deadline: 11.59pm, Monday, Sep 30th, 2019

Name: Han Jie
Matriculation Number: A0116448A

Q1. Beta-binomial Naïve Bayes

Question Description:

Beta-binomial Naïve Bayes classifier is used for this question. This implies the assumptions that all the data samples are independent and all the features are binary, 0 or 1. The context of the question, all the questions for this assignment actually, is to classify spam and non-spam emails given data samples of 57 features. So class 0 represent non-spam and 1 represents spam emails.
The question asks to try different hyperparameter α for Beta(α, α) and observe the training and test errors as α changes.

Approach:

Since it makes use of Beta-binomial Naïve Bayes, the data should be binary data, data binarization is performed immediately after reading the data from file.
Given λ can be estimated using ML and use λ^{ML} as a plug-in estimator for the testing. λ^{ML} can be easily computed by count the class 0 and class 1 samples in the training data set. $\lambda^{ML} = N_1 / N$, while N_1 denotes the total number of class1 samples while N denotes the total number of training samples.
The general equation for Naïve Bayes classifier using ML plug-in estimator after log-transformation can be written as:

$$\mathrm{p}(\tilde{y} = c \mid \tilde{x}, D) \propto \log \mathrm{p}(\tilde{y} = c \mid \lambda^{ML}) + \sum_{j=1}^{D} log\ \mathrm{p}\left(\tilde{x}_{\mathrm{j}} \left| x_{i \in cj}, \tilde{y} = c\right.\right)$$

Equation 1

The first is fairly straightforward, $log\, p(\tilde{y} = 0 \mid \lambda^{ML}) = \lambda^{ML}$ and $log\, p(\tilde{y} = 1 \mid \lambda^{ML}) = 1 - \lambda^{ML}$.
The second term need to use the equation:

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

Equation 2

Similar to the method of getting λ^{ML}, N_1 and N can be obtained by counting the class labels. However, N_1 and N denote different meaning.
N denotes the number of times that class 1 appears.
N_1 denotes the number of times that feature c is 1.
a and b denote two hyperparameters of Beta distribution, both being α for this question.
This should sum over all features for both classes to get $p(\tilde{y} = c \mid \tilde{x}, D)$.
By comparing $p(\tilde{y} = 0 \mid \tilde{x}, D)$ and $p(\tilde{y} = 1 \mid \tilde{x}, D)$, the class with larger value is estimated to be the predicted class, 0 or 1.
The computation can be accelerated by using matrix multiplication by Python package numpy.
The above procedure should be repeated for different value of α, the hyperparameter, and the graphs of α versus error rates for both training and test data are to be plotted for comparison.

Results:

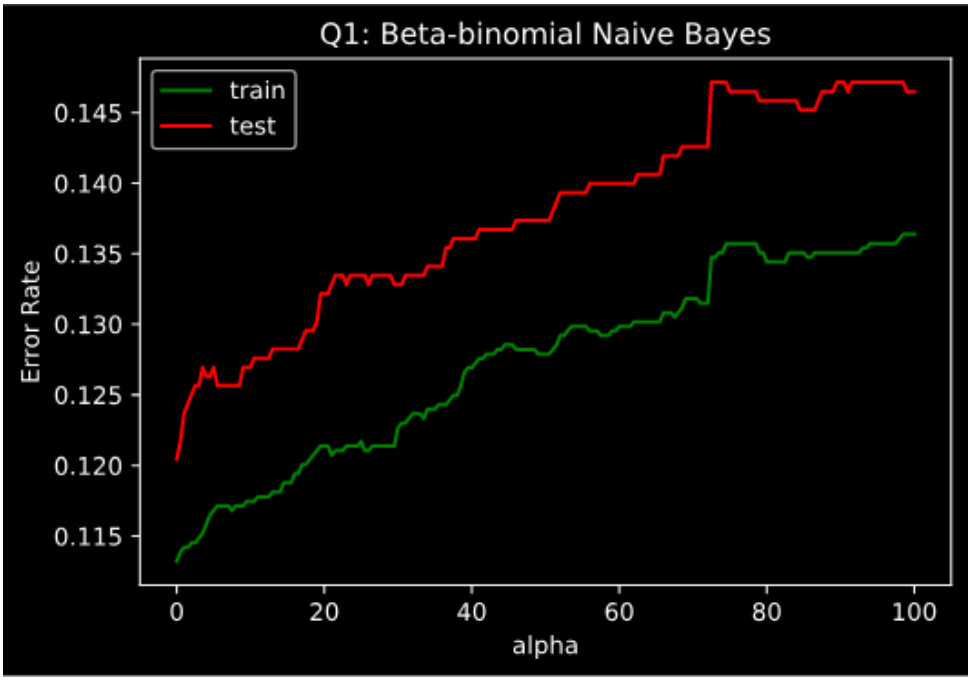

Figure 1 Q1: Beta-binomial Naive Bayes

Figure 1 shows the trend between error rates and α, as α goes from 0 to 100, the error rates for both training set and test set increases.
This trend can be explained by Equation 2.
Consider two extreme cases, when α goes to 0, $p(\tilde{x} = 1 \mid D) = \frac{N_1}{N}$, which is purely determined by the data sample itself. While α goes to infinity, $p(\tilde{x} = 1 \mid D) = \frac{\alpha}{2\alpha} = \frac{1}{2}$. It means as α increases, the result is dominated by α instead of training data itself, so the error rates increase as well.

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

α	1	10	100
Training Set	0.11419	0.11746	0.13638
Test Set	0.12370	0.12695	0.14648

Q2. Gaussian Naïve Bayes

Question Description:

Similar to Q1, the only difference is to model Gaussian Naïve Bayes, which means Univariate Gaussian Distribution is assumed.
For this question, Maximum Likelihood Estimation of mean and variance is used as a plug-in estimator for testing.

Approach:

Since Gaussian Naïve Bayes is used for this question, the data sets should be preprocessed using log-transformation. 0.1 is added to all the data samples, to avoid log 0.
Same as Q1, Equation 1 is still valid for Q2 and the first term can still be obtained by computing λ^{ML} from training labels.
However, second sum term is different. Instead of using Beta-binomial model, Gaussian is assumed. Same as getting N_1 and N in Q1, this question requires the computation of mean μ and variance σ^2, which can be estimated using Maximum Likelihood.

$$\hat{\mu} = \frac{1}{N}\sum_{n=1}^{N} x_n$$

Equation 3

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

Equation 4

Similar to Equation 2, we can get Gaussian distribution from ML:

$$p(\tilde{x} \mid \hat{\mu}, \hat{\sigma}^2) = \mathcal{N}(\tilde{x} \mid \hat{\mu}, \hat{\sigma}^2)$$

Same procedures of iterating over both classes and all samples as Q1 can be applied to Q2 as well.

Results:

Error Rate on Training Data: 0.16802610114192496

Error Rate on Test Data: 0.16341145833333334

Q3. Logistic Regression

Question Description:

This question

Approach:

Results:

Q4. K-Nearest Neighbors

Question Description:

This question

Approach:

Results:

Q5. Survey