EE5907 PATTERN RECOGNITION

CA 1 SPAM

Name: Yu Shixin

Matriculation Number.: A0195017E

Date:2019/2/24

Q1.Beta-binomial Naïve Bayes

- Plots of training and test error rates versus α.

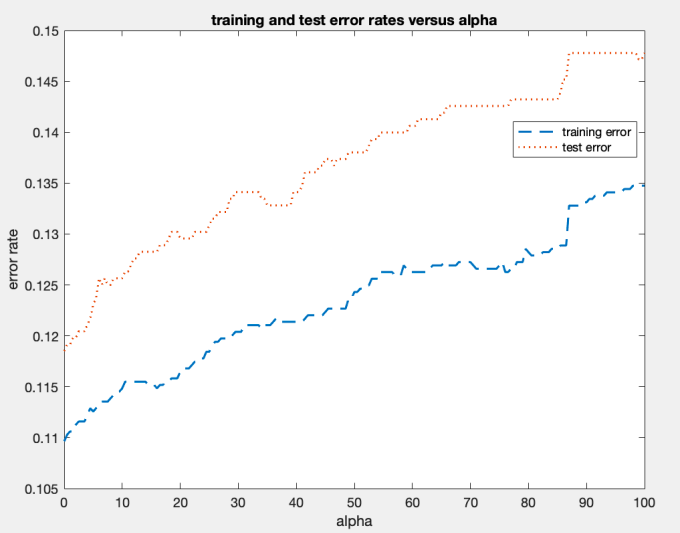

Fig.1 training and test error rates versus α

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

  We can see very clearly from the Fig.1 that

  1. With the increase of the α, the error rates of both training and test dataset show a similar upward trend, from 0.1096 to 0.1347 and from 0.1185 to 0.1478, respectively.

  2. The error rate of training dataset is always lower than that of test dataset, which means the parameters are more fit to training dataset.

  3. Although the α change from 0 to 100, but the error rate do not increase too much, with 0.251 for training dataset and 0.293 for test dataset, which means the performance of Beta-binomial Naïve Bayes classifier is good at assorting spam email.

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

	α=1	α=10	α=100
Training error rate	0.1106	0.1148	0.1347
Test error rate	0.1191	0.1257	0.1478

- Summary:

  1. Binarization for data processing. The original data is decimal, we should reform it to 1&0, which is convenient to later calculation.

2. This algorithm is derived from the Bayes' theorem

$$P(C|F_1F_2 \dots F_n) = \frac{P(F_1F_2 \dots F_n|C)P(C)}{P(F_1F_2 \dots F_n)}$$

Thereinto, $P(C)$,the class label prior λ, can be estimated using ML and use λ^{ML} as a plug-in estimator for testing.

Due to $P(F_1F_2 \dots F_n)$ is same for all category (spam and not spam in this case), it can be omitted. So, if we want to get posterior probability $P(C|F_1F_2 \dots F_n)$, we only need to calculate $P(F_1F_2 \dots F_n|C)$.

In naïve Bayes, we assume that each feature F_i is conditionally independent of every other feature F_j for $i \neq j$, given the category C. Thus, $P(F_1F_2 \dots F_n|C) = P(F_1|C)P(F_2|C) \dots P(F_n|C)$.

For each $P(F_n|C)$, we should use Laplacian correction to decrease the error, so

$$P(F_n|C) = \frac{N_n + \alpha}{N + \alpha + \alpha}$$

Later, by comparing the result of a certain sample. For example, for sample A, the result of spam(1) is larger than that of not spam(0), if the ytrain(A)=1, there is no error happening, but if the ytrain(A)=0, the number of error adds one. In the end, we judge the performance both training and test dataset by calculating the error rate.

Q2.Gaussian Naïve Bayes

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

	Training	**Testing**
error rate	0.1638	0.1816

- Summary:

1. Compared with Q1, in this part, I use log-transform method to process the dataset. We can find that error rates of binarization is always lower than that of log-transform.

2. Like Q1, the algorithm is derived from the Bayes' theorem, but Q1 is discrete random variable and Q2 is continuous random variable. By using Univariate Gaussian Distribution, we need to calculate mean and variance from training dataset.

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

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

We can find three function in MATLAB, that normpdf(), mean(), std(), to calculate normal distribution probability density, mean and variance, respectively.

Then we use training parameter as a plug-in estimator for testing, and calculate the error rate.

Q3.Logistic regression

- Plots of training and test error rates versus α.

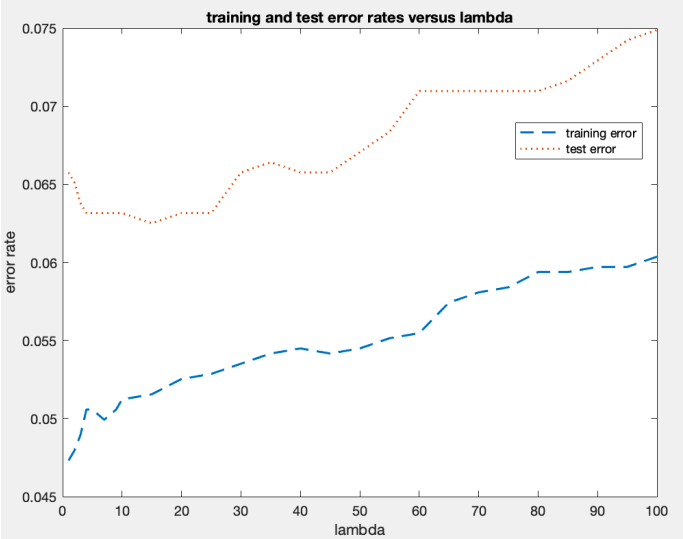

Fig.2 training and test error rates versus λ

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

  We can see very clearly from the Fig.2 that

  1. With the increase of the λ, the error rates of both training and test dataset show a similar upward trend, from 0.04731 to 0.06036 and from 0.06576 to 0.07487, respectively.

  2. The error rate of training dataset is always lower than that of test dataset, which means the parameters are more fit to training dataset.

  3. Although the λ change from 0 to 100, but the error rates do not increase too much, with 0.01305 for training dataset and 0.00911 for test dataset, which means the performance of Logistic Regression classifier is good at assorting spam email. Unlikely Q1, the change of training dataset is bigger than that of testing dataset.

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

	λ=1	λ=10	λ=100
Training error rate	0.0473	0.0512	0.0604
Test error rate	0.0658	0.0632	0.0749

- Summary:

  1. Compared with Q1 and Q2, in this part, we can find that error rates by Logistic Regression is always lower than that by Naïve Bayes.

  2. For Logistic Regression, we can not specify joint distribution $p(x, y)$. So in order to predict ytest, we need plug $\widehat{w}$ (xtrain) into posterior $p(ytest = c|xtest, \widehat{w})$.

$$\widehat{w} = argmin_w\ NLL(w) \text{ (negative log likelihood)}$$

$$NLL(w) = -\sum_{i=1}^{N}[y_i log\mu_i + (1 - y_i)\log(1 - \mu_i)], y_i = 1\ or\ 0$$

Then using Newton's method to update the w

$$g_{D\times 1} = X^T_{D\times N}(\mu - Y)_{N\times 1}$$

$$H = X^T_{D\times N} S_{N\times N} X_{N\times D}$$

Thereinto, S is diagonal matrix, i-th diagonal is $\mu_i(1 - \mu_i)$

Later include bias term, which concatenate 1 to start of x_i, so the feature vector is $D + 1$ and w is $(D + 1) \times 1$ vector, whose first element is now bias term.

Then following this step:

1. replacing $x_{i(D)}$ with $x_{i(D+1)}$, $w_{(D)}$ with $w_{(D+1)}$
2. initialize by $w = \overrightarrow{0_{D+1}}$
3. repeat until convergence $w_{k+1} = w_k - H_k^{-1} g_k$

Last but not least, implementing Logistic Regression with l_2 Regulation.

$$NLL_{reg}(w) = NLL(w) + \frac{1}{2}\lambda w^T w, (w\ without\ first\ element)$$

New gradient and hessian

$$g_{reg}(w) = g(w) + \lambda w$$

$$H_{reg}(w) = H(w) + \lambda I, I\ is\ a\ (D+1) \times (D+1)\ identity\ matrix$$

WARNNING! DO NOT include Bias in l_2 Regulation.

Q4.K-Nearest Neighbors

- Plots of training and test error rates versus K.

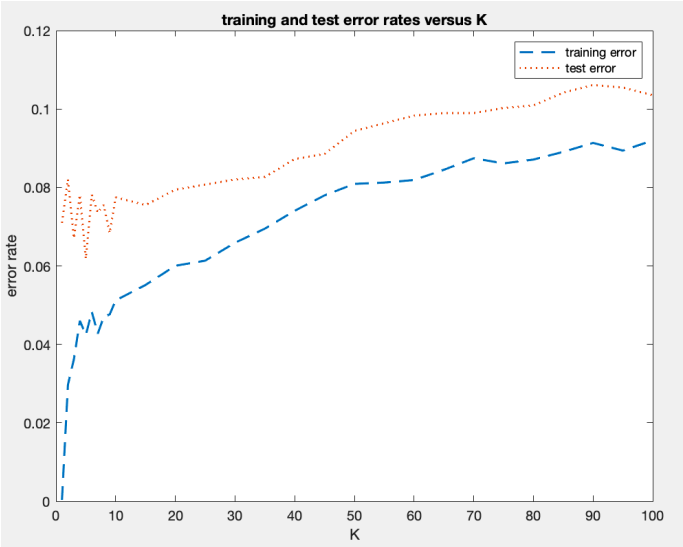

Fig.3 training and test error rates versus K

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

  We can see very clearly from the Fig.3 that

  1. With the increase of the K, the error rates of both training and test dataset show an upward trend, from 0.0003 to 0.0920 and from 0.0710 to 0.1035, respectively.

  2. The error rate of training dataset is always lower than that of test dataset, which means the parameters are more fit to training dataset.

  3. Although the K change from 0 to 100, but the error rate do not increase too much, with 0.0197 for training dataset and 0.0325 for test dataset, which means the performance of K-Nearest Neighbors classifier is good at assorting spam email. But unlikely others, the increase of K impacts training dataset more than test dataset.

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

	K=1	K=10	K=100
Training error rate	0. 0003	0.0512	0. 0920
Test error rate	0. 0710	0.0775	0. 1035

- Summary:

  1. For KNN algorithm, we should suppose volume V around x capture K sample, K_c were from class $y = c$.

     The joint probability $p(x, y = c) = \frac{k_c/N}{V}$

     Posterior $p(y = c|x) = \frac{k_c}{K}$

     Use Euclidean method to measure distance

$$dist(a, b) = \sqrt{\sum_{j=1}^{D} |a_j - b_j|^2}$$

Q5.Survey

For this assignment, I estimate 45 hours to complete both report and MATLAB program.