ECSE-4963
Matthew Caulfield
October 16, 2020

Homework 2

1. The optimal $f(x)$ when the loss function $l(f(x), y) = (f(x) - y)^2$ is the function $f(x)$ that minimizes $l(l(f(x), y))$. The minimum of $l(l(f(x), y))$ is found from:

$$\frac{d}{df(x)} l(f(x), y) = 0$$
$$\frac{d}{df(x)} (f(x) - y)^2 = 0$$
$$2(f(x) - y) = 0$$
$$f(x) = y$$

   So the optimal value of $f(x)$ that minimizes the loss function is when $f(x) = y$.

2. For question 2 we are comparing a generative and discrimanitive classifier on the spam data set from the UC Irvine data repository. We are attempting to evaluate the results reported in the paper "On Discriminative vs. Generative Classifiers: A comparison of logistic regression and naive Bayes" by A. Ng and M. Jordan where they found that discriminative classifiers performed better then generative classifiers with large enough training data. However there is a trade off of computational complexity. I am using logisitic regression as the discriminative classifier and multivariete Gaussian naive Bayes as the gerative classifier.

   (a) The approach done to implement logistic regression classification in part i of question two was to implement logistic regression with stochastic gradient descent instead of batch gradient descent. Logistic Regression was done by implementing the logit function

$$P(y|X) = \frac{1}{1 + \exp^{-w^T x_n}}$$

   Where y is the predicted class w are the weights for each attribute and x is the input for an instance to be classified in the data set. To update the weights for instance, t, I found the gradient of the objective function. To do this I first found the error between the predicted class and the actual class then multiply the data for the instance with that error as follows.

$$\nabla E(w_t) = X^T (\frac{1}{1 + \exp^{-w^T x_n}} - y_{actual})$$

   After finding the gradient I updated the t^{th} instance of w with the gradient and a learning rate α with the following function.

$$w_{t+1} = w_t - \alpha_t \nabla E(w_t)$$

   After implementing the logistic regression algorithm I had to experimentally find a number of step sizes and iterations to do to find a logistic regression with a low error rate. I found an optimal learning rate to be $\alpha = 5 * 10^(- 6)$ and the number of iterations to

be 200. The biggest restraint to the number of iterations was processing time. To run all 100 evaluations of the 6 different training set percentages the logistic regression algorithm would need to be run $600 \times numberofiterations$ this took a very long time. With 200 iterations the total time took over 6 hours. Given more time or better optimized code I would have used more iterations.

(b) To implement Naive-Bayes with marginal univariate Gaussian distributions. Each attribute was though to have a Gaussian distribution as well as a Gaussian distribution of the attribute given each class. This allowed us to find the probability of the class given the attributes using Bayes rule.

$$P(C|x) = \frac{P(x_1|C)P(x_2|C)...P(x_n|C)}{P(C)}$$

To find each distribution, the training data was divided by class and then the mean, μ, of each attribute and the standard deviation, σ,of each attribute were found by class using the following formulas where x_k is the attribute value of the kth piece of data and n is the number of sample in the training data. .

$$\mu_{iC=j} = \frac{1}{n}\sum_{k=1}^{n} x_k$$

$$\sigma^2_{iC=j} = \frac{1}{n-1}\sum_{k=1}^{n}(x_k - \mu)^2$$

Using this information,

$$P(x_i|C = j) = \mathcal{N}(\mu_{iC=j}, \sigma^2_{iC=j})$$

$P(C)$ is found by counting the number of times $C = 0$ and $C = 1$ in the data and creates the probability of each outcome in accordance to this. Overall naive Bayes is much less computationally complex because only needs to go over the training data once to find the probability distribution of the classes and attributes. The error rate is much more consistent between the size of the training data.

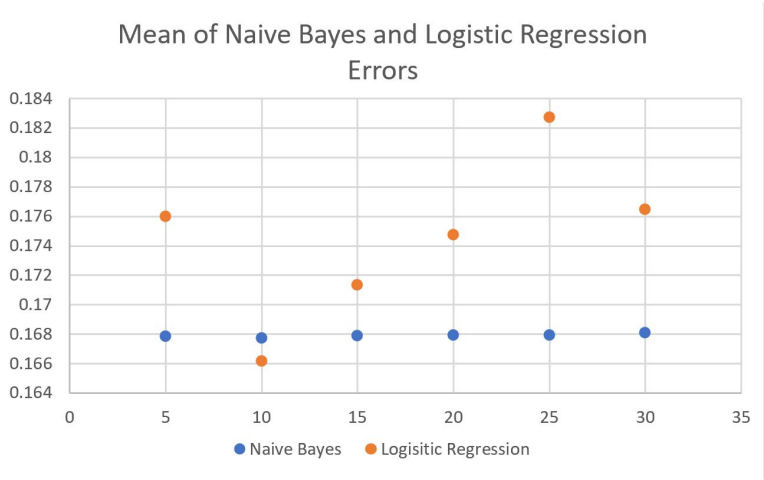

Figure 1: Mean of errors for Naive Bayes and Logistic Regression

From plotting the mean error of the logistic regression and naive Bayes we see that the error for naive Bayes is consistently around 16.8 percent regardless of the amount of training data used. The results of the logistic regression were much different with error between 16.6 to 18.2 percent. It was expected that the more training data the better logistic regression would work. However this was not the case this time. One possible explanation is over fitting. As more test data is introduced the model could be over fitting to this test data.

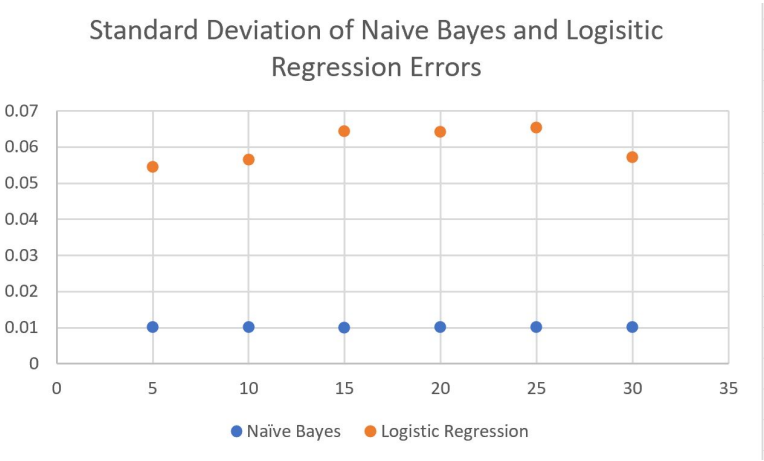

Figure 2: Standard Deviation of errors for Naive Bayes and Logistic Regression

From plotting the standard deviations of the errors over 100 trials we see that the standard deviation of naive Bayes is consistently around 0.01 while the standard deviation of the logistic regression errors is around 0.06. We see that naive Bayes gives consistent results for this data set regardless of the split but for logistic regression it is highly dependent on the training data. For this data I would suggest using naive Bayes.