Introduction to Probability Theory for Data Science
EAS 595: Class Project
Bayesian Classifier

Boobalaganesh Ezhilan
School of Engineering and Applied Sciences
State University of New York at Buffalo
Buffalo, United States
boobalag@buffalo.edu

Jaideep Reddy
School of Engineering and Applied Sciences
State University of New York at Buffalo
Buffalo, United States
jaideepreddy3112@gmail.com

***Abstract*—The Naive Bayesian classifier is based on Bayes theorem with the independence assumptions between predictors. A Naive Bayesian model is easy to build, with no complicated iterative parameter estimation which makes it particularly useful for very large data set. It assigns predefined class labels to problem instances. There are tens of classifiers based on just naive Bayes classifiers and are the building blocks of algorithms ranging from logistic regression to perceptron that are at the heart of neural networks.**

***Index Terms*—classifier, matrix, classification accuracy, multivariate**

I. Introduction

This paper documents the experiment involving 1000 participants where they performed five different tasks (C1, C2, C3, C4 and C5) and two measurements (F1 and F2) were recorded for each participant. It has been given that P(F1/Ci) and P(F2/Ci) follows normal distribution. For this problem we will be using Bayesian classifier that would predict the tasks performed given the scores F1 or F2 values(i.e whether the participants falls under which class). We will also build another Bayesian classifier for multivariate classification where we will predict the task performed given the multi-normally distributed data (Z1, F2) where Z1 is the observation wise normalized value of F1. We will analyze these four cases:

- Case 1 : X = F1
- Case 2 : X = Z1 (normalized F1)
- Case 3 : X = F2
- Case 4 : X = $\begin{vmatrix} Z1 & F2 \end{vmatrix}$

II. Model

In all of the methods we will be using 100 observations to determine the mean and variance of the normally distributed P(Fi/Ci). Then we will assume the rest of the 900 observation follows the same distribution and use Bayes' theorem to predict the class (C1, C2, C3, C4, C5) corresponding to the measurement. For univariate case (Case 1, Case 2 and Case 3) our variable is F1(F2, Z1) and we assume that our variable is continuous, whose PDF, given class Ci is normal with mean and variance as calculated using test set for each class. Using Bayes' theorem we can write:

$$\begin{aligned} P(Ci|Fi) &\propto p(Fi|Ci)P(Ci) \\ &\propto N(Fi;\mu,\sigma)P(Ci) \end{aligned}$$

In the above equation $P(Ci)$ will have the same value for all the classes so we can just compare the value of $N(Fi;\mu,\sigma)P(Ci)$ against all the class to the class on which Fi fall. We will also calculate the classification accuracy and classification error to justify our model.

III. ANALYSIS

A. Case 1

In this case we will build a Bayesian classifier for the case where X = F1. The first task is to calculate the mean and variance from the training set i.e. first 100 observations to build our model. After we run our model on MATLAB we get a classification accuracy of 53.00% and classification error rate of 47.00% on the test data.

B. Case 2

This is the case where X = Z1 i.e. the normalized F1 matrix. To normalize we subtract each value in each observation with the observation mean and divide by the observation standard deviation. Let's plot a scatter plot between Z1 and F2 to see the class distribution. Figure 1 is a plot between Z1 and F1 and we can see that most of the classes overlap and does not seem to be independent. It violates the core assumption of the Bayes' theorem that the variables are independent. This is one of the reason why we are not getting a good classification accuracy from our model. In Figure 2 we can clearly see that the classes are well separated which indicates the independence of the classes.

We execute the same model with the normalized F1 i.e. Z1 and get a classification accuracy of 88.31% and a classification error rate of 11.69%. We get an accuracy improvement of 35.31% just by normalizing the data. This shows how important it is to bring down the variables on the same scale before processing it.

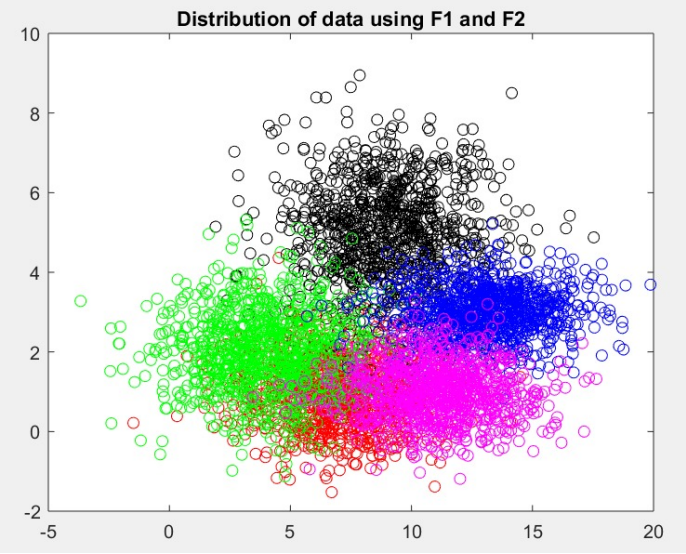

Fig. 1. Example of a figure caption.

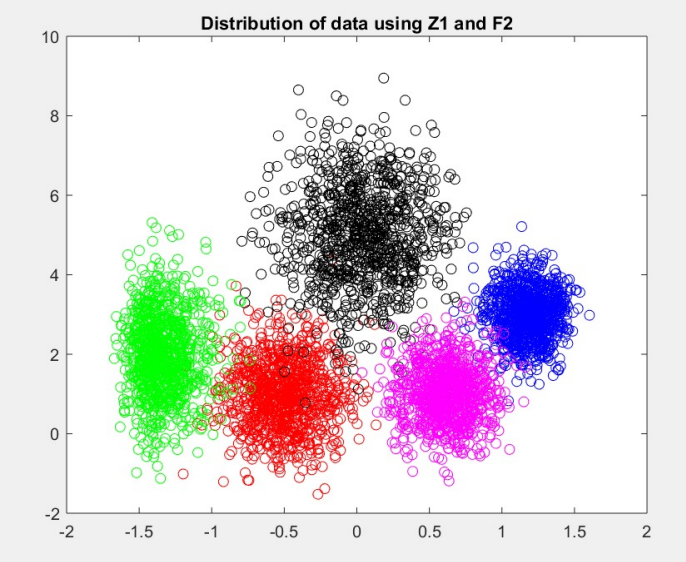

Fig. 2. Example of a figure caption.

C. Case 3

Here we will build the classifier for X = F2. We do the same steps as we did in Case 1. The classification accuracy is 55.08% and classification error rate of 44.89%. This is again the case where we are not normalizing the data and hence are getting a poor classification.

D. Case 4

In this part we will build a multivariate classifier where we will have both Z1 and F2 as variable. We calculate the mean and co-variance matrix from the test set to feed into the multivariate normal PDF. The model classifies 97.98% of the test data correctly. This is an impressive result and is because our classes are well separated and satisfies the basic assumption of the Bayesian classifier.

Conclusion

Normalization of data is important in improving the model performance. Multivariate Classifier has better performance due to encapsulating more information from the data. The sample mean and variance is enough to give us a good approximation of population.

Acknowledgment

Thanks to Professor Abani Patra and Professor Esfahani for there devoted guidance as this project is done as the part of Introduction to Probability Theory for Data science at the State University of New York at Buffalo.

References

[1] Probability and Statistics for Engineering and the Sciences (9th Edition).
[2] Introduction to Probability (2nd Ed.) by D.P. Bertsekas and J. N.Tsitsiklis (Athena Scientific) Devore
[3] https://en.wikipedia.org/wiki/Bayes
[4] https://www.saedsayad.com/naive$_b ayesian.htm$