Naive Bayes for Location Prediction*

Bowen Zhang, Yiquan Li, Honghao Wang, Ximin Huang

***Abstract*— Naive Bayes Classifier is one of the simplest probabilistic machine learning models that is used for classification task. It has found wide use in our real life, including in text classification, and among some common applications like real-time and multi-class prediction. In this project, we will illustrate the mathematical foundations of Naive Bayes Classifier and demonstrate the application of the model to our self-collected data set.**

I. INTRODUCTION

As the number of coronavirus cases arise rapidly in the United States, many universities and schools have decided to close out their campuses since early spring this year. However, there are many challenges that schools have to face with online classes. One of the challenges is the discrepancy of student locations and the different time zones students are dealing with. In this project, we will study Naive Bayes classifier [2] and predict the variation in student locations for south campus students at UCLA. Predicted student locations are classified as near UCLA (in the Westside Region), in the U.S. but not near UCLA, or outside the states. We will implement the Naive Bayes algorithm in building a probabilistic model based on a set of training data collected from Fall 2020 Math 156 students. Given the advantages of Naive Bayes, it allows us to work with small amounts of test data and to perform the task easily. In our model, we assume the independence of each variable we will be collecting and apply the method of Maximum Likelihood to the training data. At the end of the day, we will compare and measure the variation of predicted student locations with actual student locations.

The benefits of using a Naive Bayes for a classification task are many. The main two advantages of Naive Bayes are that it is very simple to implement and fast, and it requires a little training data to achieve better results than other classifiers. Besides the simplicity, Naive Bayes algorithm can both handle continuous and discrete data. We called Naive Bayes "naive" because it always assumes that all features are either independent or conditional independent. Even though this assumption usually does not hold true in real world data, Naive Bayes classifier still works unexpectedly well and even better compared to other models in real world situations. And because of the assumption made, it's going to converge quicker than discriminative models [4]. These features and advantages are especially beneficial to us since our survey data may not be sufficiently large.

Bayes theorem was initially founded by Thomas Bayes, which Naive Bayes Classifier is based on later. The Bayes theorem is one of the most fundamental theorems in the subject of statistics. It has found wide use to make predictions given certain evidence. It is undeniable that Bayes made a huge impact in probability theory, his friend Richard Price took his ideas very seriously and made significant contributions to edit, correct and publish Bayes' work [1] after Bayes passed away [3]. In general, Naive Bayes Classifier is a simple probabilistic classifier based on Bayes' theorem with independent assumptions.

Under the Naive Bayes Algorithm, there are three types of Naive Bayes Classifier: Multinomial Naive Bayes, Bernoulli Naive Bayes, and Gaussian Naive Bayes. When dealing with continuous data, we use Gaussian Naive Bayes and assume that the values are sampled from a Gaussian distribution. Where in Multinomial Naive Bayes and Bernoulli Naive Bayes, the chosen predictors take up discrete values. By comparing these three types of classifiers, we can determine which one of the classifiers is most suitable for a project based on the dataset acquired.

A. Notation

We let y be the class variable, which represents the values of classes, and $X = (x_1, x_2, x_3, \ldots, x_n)$ represent the features (predictors). Each x_i represents the value of the corresponding predictor.

B. Organization

In what follows, we present the mathematical foundations of Naive Bayes Classifier and illustrate an application to our self-collected data set. In Section II, we illustrate the formulation of Naive Bayes, get into the detail of three types of Naive Bayes Classifier and compare them to each other. In Section III, we begin with survey data (self-collected) introduction and data cleaning. Then we apply Naive Bayes to our refined survey data set. Finally, in Section IV, we offer some conclusions.

II. MATHEMATICAL FOUNDATIONS

In this section, we present the mathematical foundation of Naive Bayes. First, we illustrate the derivation of the Naive Bayes formulation and how it is mathematically solved. We then compare the three Naive Bayes classifiers and analysis the features of data that each classifier prefers to.

A. Mathematical Formulation

The goal of Naive Bayes is to find the class label y that can maximize the posterior probability $P(y|X)$ in which $X = (x_1, x_2, \ldots, x_n)$ is a vector of feature values. It is a technique for constructing classifiers. Roughly speaking,

*This work was not supported by any organization

Naive Bayes classifier assign class labels y to problem instances represented by vectors of feature (predictor) values X. We assume that the value of a particular feature is independent of the value of any other, and all the predictors have an equal effect on the outcome.

Before we jump right into the Naive Bayes Classifier algorithm, it's essential to understand why the Naive Bayes is called to be "naive". It is because of the assumption that all features are independent. In other words, it assumes that the occurrence of a certain feature (event) does not affect the occurrence of any other features. For example, suppose that you are assigned to identify the type of fruit based on three features: color, taste, and shape. Now you will probably identify a fruit as orange if it has orange color, tastes sour and looks spherical. Even if these three features are indeed dependent on each other or upon the existence of the other features, the Naive Bayes classifier will consider all three properties to individually contribute to the probability that this fruit is an orange. Now we know that Naive Bayes is "naive" because it does not consider dependence between each feature which will easily lead to work with redundant data.

Naive Bayes classifier is a kind of probabilistic classifier based on applying Bayes' Theorem with strong independence assumptions, so based on Bayes' theorem, we decompose the conditional probability as

$$P(y|X) = \frac{p(X|y)P(y)}{P(X)}.$$

Substituting for X, then we get

$$P(y|x_1, x_2, ..., x_n) = \frac{P(x_1, x_2, ..., x_n|y)P(y)}{P(x_1, x_2, ..., x_n)}.$$

Now, the numerator of the fraction is equivalent to the joint probability $P(x_1, x_2, ..., x_n, y)$. Using the chain rule, the joint probability is then

$$\begin{aligned}
&P(x_1, x_2, ..., x_n, y) \\
&\quad = P(x_1|x_2, x_3, ..., x_n, y)P(x_2, x_3, ..., x_n, y) \\
&\quad = P(x_1|x_2, x_3, ..., x_n, y)P(x_2|x_3, x_4, ..., x_n, y) \\
&\qquad P(x_3, x_4..., x_n, y) \\
&\quad = ... \\
&\quad = P(x_1|x_2, x_3, ..., x_n, y)P(x_2|x_3, x_4, ..., x_n, y)... \\
&\qquad P(x_{n-1}|x_n, y)P(x_n|y)P(y) \\
&\quad = (\prod_{i=1}^{n-1} P(x_i|x_{i+1}, ...x_n, y))P(x_n|y)P(y).
\end{aligned}$$

Since all the features in X are mutually independent, $P(x_i|x_{i+1}, ..., x_n, y) = P(x_i|y)$. So, now

$$\begin{aligned}
&P(x_1, x_2, ..., x_n, y) \\
&\quad = (\prod_{i=1}^{n-1} P(x_i|y))P(x_n|y)P(y) \\
&\quad = P(y)\prod_{i=1}^{n} P(x_i|y).
\end{aligned}$$

Similarly, because of the conditional independence assumption, the denominator of the fraction is then

$$P(x_1, x_2, ..., x_n) = P(x_1)P(x_2)...P(x_n) = \prod_{i=1}^{n} P(x_i).$$

So now the conditional distribution of the class variable y becomes

$$P(y|x_1, x_2, ..., x_n) = \frac{P(y)\prod_{i=1}^{n} P(x_i|y)}{\prod_{i=1}^{n} P(x_i)}.$$

Notice that the denominator only depends on $x_1, x_2, ...x_n$ and does not depend upon y. Thus if the feature variables are decided, $P(x)$ will be a constant. So we have the proportionality

$$P(y|x_1, x_2, ..., x_n) \propto P(y)\prod_{i=1}^{n} P(x_i|y).$$

Thus, maximize the $P(y)\prod_{i=1}^{n} P(x_i|y)$ is equivalent to maximize the posterior probability $P(y|x_1, x_2, ..., x_n)$.

Now, the optimization problem we seek to solve is

$$\arg\max_y P(y)\prod_{i=1}^{n} P(x_i|y).$$

To solve this maximization problem, we can obtain the values for each probability by looking at the dataset and substitute them into the function, or compute the probability based on the assumed distribution of features.

B. Comparison of the Three Naive Bayes Classifier

There are three types of Naive Bayes classifiers – Multinomial Naive Bayes, Bernoulli Naive Bayes, and Gaussian Naive Bayes.

Multinomial Naive Bayes is used for discrete counts. Predictors take discrete values and each predictor can have more than two values. So if the feature variables are all discrete or categorical, one can usually choose Multinomial Naive Bayes. For example, Multinomial Naive Bayes is mostly used for document classification problem, in which we count how often a word occurs in the document.

Bernoulli Naive Bayes is similar to the Multinomial Naive Bayes, but the predictors are Boolean variables. So for $X = (x_1, x_2, ..., x_n)$, each x_i only takes up two values – "true"(1) or "false"(0). For example, in document classification problem, the value of x_i means whether the corresponding word occurs in the document.

Gaussian Naive Bayes is used when the predictors are not discrete and take up a continuous value. And we assume that the values of the predictors follow a Gaussian distribution. In this case, the probability distribution of x_i given a class y is

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$

where μ_y is the mean values in x_i associated with class y, and σ_y^2 is the variance of the values in x_i associated with class y.

We will choose the suitable classifier for our project based on the features of our dataset.

```
> summary(data)
                  Current_Location  Year_of_Study Severity_at_Hometown
 Foreign country or region  : 8     Freshman: 1    1: 9
 Other place but inside U.S.:15     Junior  :10    2:15
 Westside Region of LA      :15     Senior  :27    3:14


 Restriction_for_Entering_US         Family_in_US
 No            : 7           No            :14
 Not applicable:18           Not applicable:13
 Yes           :13           Yes           :11

        Frequency_of_Going_Outdoor Number_of_COVID_Tests
3 to 5 times per week :16          0:22
More frequently       : 4          1: 5
Twice or less per week:18          2: 5
                                   4: 4
                                   5: 1
                                   6: 1
```

Fig. 1. Summary output of full data in R.

```
> (priors <- prop.table(table(data$Current_Location)))

Foreign country or region Other place but inside U.S.      Westside Region of LA
                0.2105263                 0.3947368                 0.3947368
> (locations <- names(priors))
[1] "Foreign country or region"   "Other place but inside U.S." "Westside Region of LA"
```

Fig. 2. Computed P(y).

```
# Main Program of NB
survey_nb <- function(data){
  out <- apply(data, 1, function(row1){
    mat1 <- matrix(nrow = length(row1) - 1, ncol = 3)
    for (i in seq_len(length(row1))[-1]){
      mat1[i-1,] <- c(mean(data[,i][data[,1] == locations[1]] == row1[i]),
                      mean(data[,i][data[,1] == locations[2]] == row1[i]),
                      mean(data[,i][data[,1] == locations[3]] == row1[i]))
    }
    apply(mat1, 2, prod) * priors / sum(apply(mat1, 2, prod) * priors)
  })
  max.col(t(out))
}
```

Fig. 3. Code for computing and using the value of P(X|y).

III. EXPERIMENTS

In this section, we apply Multinomial Naive Bayes to our data collected from a survey for UCLA Fall 2020 Math 156 students and some other south campus students in UCLA.

A. Data Description

The entire dataset obtained from the survey contains 38 observations and 7 variables. A screenshot of its summary in R is shown in Figure 1. An important thing to be noticed is that since all variables are treated as discrete, they only contain levels that are presented in the survey responses. For example, "Year of Study" does not have Sophomore as its level and neither does "Number of Covid Tests" have three times per week as its level. To make the code unanimous for all variables, we used three levels 1, 2 and 3 which stand for mild, medium and severe severeness respectively to discretize the infection rate of COVID-19 by the end of May 2020 in every subject's hometown.

Among seven variables, "Current Location" is the target variable, so it has to be included in the dataset for prediction all the time. "Restriction for Entering US" and "Family in US" are dependent on each other, since one who answered "Not Applicable" on one variable was likely to have the same response on the other, so one of two variables has to be eliminated from the dataset in order to satisfy the assumption of independence in Naive Bayes.

B. Implementation

Based on the maximum a posteriori(MAP) decision rule, we apply the Naive Bayes classifier to our dataset.

Step 1: Compute $\mathbf{P}(\mathbf{y})$ in $MAP(y) = \max \frac{P(X|y)\mathbf{P}(\mathbf{y})}{P(X)}$.

To compute priors $P(y)$, where y is one of the *locations*, the *table* function is applied to the column of target variable to return a frequency table and then the *prop.table* function is applied to convert the frequency table into the density distribution table, which contains the priors. The result is shown in Figure 2.

Step 2: Compute $\mathbf{P}(\mathbf{X}|\mathbf{y})$ in $MAP(y) = \max \frac{\mathbf{P}(\mathbf{X}|\mathbf{y})P(y)}{P(X)}$.

To compute likelihoods $P(X|y)$, where X stands for a condition as any possible combination of values in variables, conditional probabilities of variables taking certain values for a row entry are calculated at first. And according to Naive Bayes, conditional probabilities for all row entries with respect to each variable are multiplied together in order to get the likelihoods. See how we compute and use $P(X|y)$ in Figure 3.

Step 3: Compute $\mathbf{P}(\mathbf{X})$ in $MAP(y) = \max \frac{P(X|y)P(y)}{\mathbf{P}(\mathbf{X})}$.

To compute marginal probability $P(X)$, the dot product of likelihoods and priors are taken and then summed for all elements in order to derive the marginal probability. See how we compute and use $P(X)$ in Figure 4.

Finally, the results of all components are plugged into $MAP(y)$ to get the values of predictions.

C. Model Accuracy

After the predictions are obtained from the algorithm, two helper functions *correct_rate* and *optim_combn* are defined for computing the rate of correct predictions according to a dataset with a certain combination of variables and finding the optimal combination of variables that gives the highest rate respectively. See Figure 5 for the definition of the two helper functions.

After applying Naive Bayes on all possible subsets of variables, we can see from Figure 6 that the optimal com-

```
# Main Program of NB
survey_nb <- function(data){
  out <- apply(data, 1, function(row1){
    mat1 <- matrix(nrow = length(row1) - 1, ncol = 3)
    for (i in seq_len(length(row1))[-1]){
      mat1[i-1,] <- c(mean(data[,i][data[,1] == locations[1]] == row1[i]),
                      mean(data[,i][data[,1] == locations[2]] == row1[i]),
                      mean(data[,i][data[,1] == locations[3]] == row1[i]))
    }
    apply(mat1, 2, prod) * priors / sum(apply(mat1, 2, prod) * priors)
  })
  max.col(t(out))
}
```

Fig. 4. Code for computing and using the value of P(X).

```
correct_rate <- function(data){
  res <- survey_nb(data)
  mean(data[,1] == locations[res])
}

optim_combn <- function(nidx2t5 = 3, idx6n7 = 6){
  combn1 <- combn(c(2,3,4,5), nidx2t5, simplify = FALSE)
  rates <- numeric(length(combn1))
  for (i in seq_len(length(combn1))){
    rates[i] <- correct_rate(data[,c(1,combn1[[i]],idx6n7)])
  }
  list(combn = c(1,combn1[[which.max(rates)]],idx6n7),
       rate = max(rates))
}
```

Fig. 5. Definitions of the two helper functions *correct_rate* and *optim_combn*.

```
# Run NB on all possible subsets of variables
vars_lst <- list()
rates <- numeric(12)
i <- 0
for (nidx2t5 in c(4, 3, 2, 1)){
  for (idx6n7 in list(6, 7, NULL)){
    i <- i + 1
    res <- optim_combn(nidx2t5, idx6n7)
    vars_lst[[i]] <- res[[1]]
    rates[[i]] <- res[[2]]
  }
}
```

```
> max(rates)
[1] 0.8947368
> vars_lst[[which.max(rates)]]
[1] 1 2 4 5 6
> names(data)[vars_lst[[which.max(rates)]][-1]
[1] "Year_of_Study"              "Frequency_of_Going_Outdoor"
[3] "Number_of_COVID_Tests"      "Restriction_for_Entering_US"
```

Fig. 6. The optimal combination of variables are "Year of Study", "Frequency of Going Outdoor", "Number of COVID Tests" and "Restriction for Entering US" with the rate of correctly predicting the location as nearly 90%.

bination of variables are "Year of Study", "Frequency of Going Outdoor", "Number of COVID Tests" and "Restriction for Entering US" with the rate of correctly predicting the location as nearly 90%.

IV. CONCLUSIONS

In this manuscript, we have described and applied the popular classification technique, Naive Bayes. Based on the independent assumptions that Naive Bayes Classifier makes, we have derived the formulation for Naive Bayes and demonstrate how it can be applied to both discrete and continuous data. We have compared among Multinomial Naive Bayes, Bernoulli Naive Bayes, and Gaussian Naive Bayes.

In our experiments, we start out by collecting data from a survey for UCLA Math 156 students and some other south campus UCLA students. And we have chosen 6 features to be our predictor variables in order to predict the current location of UCLA students into one of the three categories. We then use the survey questions as our training dataset and apply Naive Bayes Classifier to develop our model. During our model development, we modify our dataset into categorical variables and clean the predictor values.

In the end, we decide to use 4 of the 6 features to predict the location of a UCLA student who enrolled in Math 156 in which the combination yields the best precision rate of about 89.5%. At the end of the day, we successfully apply the Naive Bayes Classifier to predict the current location of UCLA students who enrolled in Math 156 into one of the three categories. We believe that the model will be beneficial for faculties to arrange office hours and exams' time range.

REFERENCES

[1] Thomas Bayes. An essay towards solving a problem in the doctrine of chances. by the late rev. mr. bayes, f. r. s. communicated by mr. price, in a letter to john canton, a. m. f. r. s. *Philosophical Transactions*, 53:370–418, 1763.

[2] David J. Hand and Keming Yu. Idiot's bayes: Not so stupid after all? *International Statistical Review / Revue Internationale De Statistique*, 69(3):385–398, 2001.

[3] Holy Python. Naive Bayes Classifier History. Https://holypython.com/nbc/naive-bayes-classifier-history/. Accessed: 2020-12-10.

[4] N. Sharma. Understanding the mathematics behind naive bayes. Https://heartbeat.fritz.ai/understanding-the-mathematics-behind-naives-bayes-ab6ee85f50d0. Accessed: 2020-12-10.