Tutorials on Bayes and Naive Bayes Classifiers

Prof. Le Wang

Contents

1 What is Bayes Classifers? 1

2 When and why does Bayes Classifiers work? 1

3 Bayes Classifiers with Predictors (Covariates) 2

4 Bayes Classifiers with Multiple Predictors (Covariates) 3

5 Naive Bayes Classifiers with Multiple Predictors (Covariates) 4

5.1 Motivation . . . 4

5.2 Example . . . 4

5.2.1 Term A . . . 5

5.2.2 Term B . . . 5

5.2.3 Term C . . . 6

5.2.4 Putting them together . . . 7

5.2.5 Prediction . . . 7

6 Further thoughts and discussions 8

1 What is Bayes Classifers?

Lets first introduce the method intuitively. When we have a discrete outcome, Y, **Bayes Classifiers** is simply an algorithm that predicts the outcome value to be the value with the highest probability.

Lets consider an example, say, we would like to predict the survival outcome of an individual in the `titanic` dataset. In this case, the outcome `Survived` can take on only two values 0 (not survived) or 1 (survived). Prior to making our prediction, we need to first calculate the probability for each possible value of our outcome (i.e., the **distribution** of `Survived`):

$$\Pr[\text{Survived} = 0], \quad \Pr[\text{Survived} = 1]$$

```
prop.table(table(titanic_train$Survived))
```

```
##
##         0         1
## 0.6161616 0.3838384
```

Based on the distribution obtained, we can see that $\Pr[\text{Survived} = 0]$ is the largest, and hence pick 0 as our prediction for every individual when no further information is given. That is it!

2 When and why does Bayes Classifiers work?

If our goal is to minimize the error rate of a prediction, that is, how many times that we make an error, any error, Bayes classifier is the optimal choice. When do we make an error with our prediction? Whenver our

outcome is NOT equal to our prediction. In our running example, if we use 0 as our prediction of survival status, we make an error for any individual that survived.

$$\text{Error} = \begin{cases} 1 & \text{if Survived} \neq 0 \\ 0 & \text{if Survived} = 0 \end{cases}$$

More compactly, we can use an indicator function or variable to show whether we make an error when our prediction is 0:

$$\mathbb{I}[\text{Survive} \neq 0]$$

where it equals one when the argument inside the bracket holds true, zero otherwise. We just need to count how many times we will make an error. Well, only when Survive is not 0.

$$\Pr[\text{Survive} \neq 0] = 1 - \Pr[\text{Survive} = 0]$$

Similarly, if we use 1 as our prediction, the instances of any errors is

$$\Pr[\text{Survive} \neq 1] = 1 - \Pr[\text{Survive} = 1]$$

You can immediately see that if we pick the value with the highest probability, then the error rate is actually the smallest.

In fact, the pattern holds true for any discrete outcomes that can take on two or more values. If you use one of the possible outcome values, y, as our prediction for any discrete outcome, Y, the error rate for this particular prediction is always

$$\Pr[Y \neq y] = 1 - \Pr[Y = y]$$

In other words, the error rate is inversely related with the probability of y. That's why given the objective function to minimize the error rate, our optimal algorithm is Bayes classifier, always picking the value with the highest probability. Note, also, that the error rate is also called **expected error rate** since

$$\Pr[Y \neq y] = \mathbb{E}[\mathbb{I}[Y \neq y]]$$

3 Bayes Classifiers with Predictors (Covariates)

Now, lets refine our prediction to take into account more information. For example, if people in different classes have different survival rates in the `titanic` dataset, then we should take into account this information in our prediction. And obviously, the prediction should change for people from different classes. The algorithm remains the same, with a slight modification. **Bayes Classifiers** is simply an algorithm that predicts the outcome value to be the value with the highest probability **among the same class** (or the subgroup). We just need to find out the outcome value with the highest probability among those people for whom we would like to predict. In order to do this, we need the information on **conditional distribution**, as opposed to **unconditional/marginal distribution** above.

For example, for people from the first class, the (conditional) distribution of survival status is

$$\Pr[\text{Survived} = 0 \mid \text{Pclass} = 1], \quad \Pr[\text{Survived} = 1 \mid \text{Pclass} = 1]$$

```
prop.table(table(titanic_train$Survived[Pclass == 1]))
```

```
##
##         0         1
## 0.3703704 0.6296296
```

Based on the distribution obtained, we can see that $\Pr[\text{Survived} = 1 \mid \text{Pclass} = 1]$ is the largest, and hence pick 1 as our prediction for every individual from the first class.

We can similarly do that for all other classes.

```
prop.table(table(titanic_train$Survived[Pclass == 2]))
```

```
##
##         0         1
## 0.5271739 0.4728261
```

```
prop.table(table(titanic_train$Survived[Pclass == 3]))
```

```
##
##         0         1
## 0.7576375 0.2423625
```

And you can clearly see that the prediction varies with the class. For the second and third classes, your prediction will be 0 (not survived). The reasoning behind this algorithm is exactly the same as above: our prediction minimizes the error rates **among the subgroup**.

4 Bayes Classifiers with Multiple Predictors (Covariates)

If we have multiple predictors, we can continue the same logic. For example, if we would like to predict the survival status of female passengers in the first class. We can estimate the conditional distribution as follows:

$$\Pr[\text{Survived} = 0 \mid \text{Pclass} = 1, \text{Sex} = \text{Female}], \quad \Pr[\text{Survived} = 1 \mid \text{Pclass} = 1, \text{Sex} = \text{Female}]$$

```
prop.table(table(titanic_train$Survived[Sex == "female" & Pclass == 1]))
```

```
##
##          0          1
## 0.03191489 0.96808511
```

We can see survival equal to one has the highest probability, and thus our prediction is 1 for female passengers from the first class.

There are two alternative ways to estimate the information using (1) subset or (2) tidy approach. The first approach takes two steps (to select the subset and then estimate the distribution for this subset).

```
# Select Subset
sub.sample <- subset(titanic_train, Sex == "female" & Pclass == 1 )

prop.table(table(sub.sample$Survived))
```

```
##
##          0          1
## 0.03191489 0.96808511
```

We can also use the tidy approach

```
titanic_train %>%
  filter(Sex == "female" & Pclass == 1) %>%
  group_by(Survived) %>%
  summarize(n = n()) %>%
  mutate(prop=n/sum(n))
```

```
## # A tibble: 2 x 3
##   Survived     n   prop
##      <int> <int>  <dbl>
## 1        0     3 0.0319
## 2        1    91 0.968
```

5 Naive Bayes Classifiers with Multiple Predictors (Covariates)

5.1 Motivation

So far so good. The Bayes classifier above is based on the conditional distribution and easy to apply. However, we cannot always apply our Bayes Classifiers in every situation, especially when there are many predictors. The reason is that in the case of many predictors, the subgroup can be too narrowly defined or even not well defined.

Consider an example. Suppose that I would like to predict the unemployment outcome for a guy called Le Wang who is a 39-year-old male, was born in Guangdong, China, but now live in Vermont. It is a pretty straightforward exercise if we can obtain the conditional distribution of unemployment outcome for all the people called Le Wang who is a 39-years-old male, was born in Guangdong, China, but now live in Vermont.

$$\begin{aligned}&\Pr[\text{Unemployment} = 0 \mid \text{Name } = \text{ Le Wang}, \text{Age } = 39,\\ &\text{Gender } = \text{Male}, \text{ Place of Birth } = \text{ Guangdong },\\ &\text{Place of Residence } = \text{ Vermont }]\end{aligned}$$

$$\begin{aligned}&\Pr[\text{Unemployment} = 1 \mid \text{Name } = \text{ Le Wang}, \text{Age } = 39,\\ &\text{Gender } = \text{Male}, \text{ Place of Birth } = \text{ Guangdong },\\ &\text{Place of Residence } = \text{ Vermont }]\end{aligned}$$

In practice, this may not be feasible at all since there may not be any people meeting this requirement. This will definitely be the case when any of the predictors is continuous (we will make this more clear later, but can you think of why?). This means that we cannot just use the algorithm above to select the subsample defined by the criteria, estimate the distribution, and pick the most likely value. Instead, we will have to invoke some assumptions to help us to estimate the distribution, which is called **conditional independence assumption**.

5.2 Example

To see how this works, lets continue with our example of predicting the survival status of female passengers from the first class. As we have seen above, we can directly estimate the conditional distribution, so this example is simply illustrative to focus on ideas, rather than the actual merits.

$$\Pr[\text{Survived} = 0 \mid \text{Pclass} = 1, \text{Sex} = \text{Female}], \quad \Pr[\text{Survived} = 1 \mid \text{Pclass} = 1, \text{Sex} = \text{Female}]$$

Let's first apply the **Bayes Rule**.

$$\Pr[\text{Survived} = 0 \mid \text{Pclass} = 1, \text{Sex} = \text{Female}] = \frac{\Pr[\text{Pclass} = 1, \text{Sex} = \text{Female} \mid \text{Survived} = 0] \cdot \Pr[\text{Survived} = 0]}{\Pr[\text{Pclass} = 1, \text{Sex} = \text{Female}]}$$

We need three terms:

$$\begin{aligned}
&\textbf{Term A:} = \Pr[\text{Survived} = 0] \\
&\textbf{Term B:} = \Pr[\text{Pclass} = 1, \text{Sex} = \text{Female} \mid \text{Survived} = 0] \\
&\textbf{Term C:} = \Pr[\text{Pclass} = 1, \text{Sex} = \text{Female}]
\end{aligned}$$

Note that **Term C** can be obtained if we know **Term A** and **Term B**. Due to the law of total probability, the denominator, **Term C**, is given by

$$\begin{aligned}
\Pr[\text{Pclass} = 1, \text{Sex} = \text{Female}] = &\Pr[\text{Pclass} = 1, \text{Sex} = \text{Female} \mid \text{Survived} = 0] \cdot \Pr[\text{Survived} = 0] \\
&+ \Pr[\text{Pclass} = 1, \text{Sex} = \text{Female} \mid \text{Survived} = 1] \cdot \Pr[\text{Survived} = 1]
\end{aligned}$$

You can immediately see that to obtain the denominator, we need the following (the product of **Term A** and **Term C**):

$$\Pr[\text{Pclass} = 1, \text{Sex} = \text{Female} \mid \text{Survived} = 0]$$

$$\Pr[\text{Pclass} = 1, \text{Sex} = \text{Female} \mid \text{Survived} = 1]$$

Below we discuss how we esitmate each term.

5.2.1 Term A

It is very straightforward to estimate $\Pr[\text{Survived} = 0]$, which is nothing but counting how many people who did not survived (in percentage).

```
prior <- prop.table(table(Survived))
prior
```

```
## Survived
##         0         1
## 0.6161616 0.3838384
```

5.2.2 Term B

$$\Pr[\text{Pclass} = 1, \text{Sex} = \text{Female} \mid \text{Survived} = 0]$$

As mentioned above, these two terms become very difficult to estimate when the number of predictors increase and perhaps do not even exist! Here is where the magic comes from. We will impose an assumption called **conditional independence assumption**: we will assume that class and gender are not related to each other in any way once we among the same survival status. Among those who survived, class and gender are not related. Among those who did not survive, class and gender are not related. Mathematically,

$$\text{Pclass} \perp \text{Sex} \quad | \quad \text{Survived}$$

Remember that when two variables are conditionally independent with each other

$$p(x, y \mid z) = p(x \mid z) \cdot p(y \mid z)$$

Therefore, the distribution that we want now becomes

$$\Pr[\text{Pclass} = 1, \text{Sex} = \text{Female} \mid \text{Survived} = 0] = \Pr[\text{Pclass} = 1 \mid \text{Survived} = 0] \cdot \Pr[\text{Sex} = \text{Female} \mid \text{Survived} = 0]$$

Instead of estimating the left-hand-side term directly, we can estimate

$$\Pr[\text{Pclass} = 1 \mid \text{Survived} = 0]$$

This is nothing but finding out how many first-class passengers among those who did not survive

```
cond0.Pclass <- prop.table(table(Pclass[Survived==0]))
cond0.Pclass
```

```
##
##         1         2         3
## 0.1457195 0.1766849 0.6775956
```

$$\Pr[\text{Sex} = \text{Female} \mid \text{Survived} = 0]$$

This is just to find out how many female passengers among those who did not survive

```
cond0.sex <- prop.table(table(Sex[Survived==0]))
cond0.sex
```

```
##
##   female     male
## 0.147541 0.852459
```

So, among those who did not survive, the probability of female passengers is given by the product of these two terms:

$$\Pr[\text{Pclass} = 1, \text{Sex} = \text{Female} \mid \text{Survived} = 0] = \Pr[\text{Pclass} = 1 \mid \text{Survived} = 0] \cdot \Pr[\text{Sex} = \text{Female} \mid \text{Survived} = 0]$$

```
(cond0.Pclass["1"]*cond0.sex["female"])
```

```
##         1
## 0.0214996
```

Similarly, we can estimate

$$\Pr[\text{Pclass} = 1, \text{Sex} = \text{Female} \mid \text{Survived} = 1] = \Pr[\text{Pclass} = 1 \mid \text{Survived} = 1] \cdot \Pr[\text{Sex} = \text{Female} \mid \text{Survived} = 1]$$

```
cond1.Pclass <- prop.table(table(Pclass[Survived==1]))
cond1.sex <- prop.table(table(Sex[Survived==1]))
cond1.Pclass["1"]*cond1.sex["female"]
```

```
##        1
## 0.270921
```

5.2.3 Term C

Now we have all we need to calculate the denominator

$$\begin{aligned}\Pr[\text{Pclass} = 1, \text{Sex} = \text{Female}] = {} & \Pr[\text{Pclass} = 1, \text{Sex} = \text{Female} \mid \text{Survived} = 0] \cdot \Pr[\text{Survived} = 0] \\ & + \Pr[\text{Pclass} = 1, \text{Sex} = \text{Female} \mid \text{Survived} = 1] \cdot \Pr[\text{Survived} = 1]\end{aligned}$$

```
den <- (cond0.Pclass["1"]*cond0.sex["female"])*prior["0"]+(cond1.Pclass["1"]*cond1.sex["female"])*prior
den
```

```
##         1
## 0.1172371
```

5.2.4 Putting them together

The numerator is given by

$$\Pr[\text{Pclass} = 1, \text{Sex} = \text{Female} \mid \text{Survived} = 0] \cdot \Pr[\text{Survived} = 0]$$

```
num0 <- (cond0.Pclass["1"]*cond0.sex["female"])*prior["0"]
num0
```

```
##          1
## 0.01324723
```

Finally, we can put everything together to obtain

$$\Pr[\text{Survived} = 0 \mid \text{Pclass} = 1, \text{Sex} = \text{Female}] = \frac{\Pr[\text{Pclass} = 1, \text{Sex} = \text{Female} \mid \text{Survived} = 0] \cdot \Pr[\text{Survived} = 0]}{\Pr[\text{Pclass} = 1, \text{Sex} = \text{Female}]}$$

```
num0/den
```

```
##         1
## 0.1129952
```

We can also obtain this for the probability of survival for this subgroup using the conditional independence assumption. we can put everything together to obtain

$$\Pr[\text{Survived} = 1 \mid \text{Pclass} = 1, \text{Sex} = \text{Female}] = \frac{\Pr[\text{Pclass} = 1, \text{Sex} = \text{Female} \mid \text{Survived} = 1] \cdot \Pr[\text{Survived} = 1]}{\Pr[\text{Pclass} = 1, \text{Sex} = \text{Female}]}$$

```
num1 <- (cond1.Pclass["1"]*cond1.sex["female"])*prior["1"]
num1/den
```

```
##         1
## 0.8870048
```

5.2.5 Prediction

In other words, the conditional distribution **under the conditional independence assumption** is given by

$$\Pr[\text{Survived} = 0 \mid \text{Pclass} = 1, \text{Sex} = \text{Female}] = 0.1129952$$
$$\Pr[\text{Survived} = 1 \mid \text{Pclass} = 1, \text{Sex} = \text{Female}] = 0.8870048$$

Once we have this information, we can pick the most likely outcome to be our prediction, which is 1. In this case, the **Naive Bayes Classifier** produces the same prediction as **Bayes Classifer** above. But there are some subtle differences.

```r
prop.table(table(titanic_train$Survived[Sex == "female" & Pclass == 1]))
```

```
##
##          0          1
## 0.03191489 0.96808511
```

6 Further thoughts and discussions

Two questions for you to think about.

1. **Question** With increasing availability of big data (more data points), what will happen to the need of the Naive Bayes Classisfier?
2. **Question** Which approach is preferred, Bayes Classifer vs Naive Bayes Classifier? Why?

Here we discuss only **discrete** predictors. In practice, we will have continuous predictors, which can drastically complicate estimation. For example, for a continuous variable we know that the probability is equal to zero! Therefore, we need to impose further assumption in our estimation. The built-in package `naivebayes` does this for you.

```r
# Load the naivebayes package
library(naivebayes)
```

```
## naivebayes 0.9.6 loaded
```

```r
# estimate the Naive Bayes Model using the training data set
model <- naive_bayes(factor(Survived) ~ Pclass + Sex, data =titanic_train)
model
```

```
##
## ================================== Naive Bayes ==================================
##
##  Call:
## naive_bayes.formula(formula = factor(Survived) ~ Pclass + Sex,
##     data = titanic_train)
##
## ---------------------------------------------------------------------------------
##
## Laplace smoothing: 0
##
## ---------------------------------------------------------------------------------
##
##  A priori probabilities:
##
##         0         1
## 0.6161616 0.3838384
##
## ---------------------------------------------------------------------------------
##
##  Tables:
##
## ---------------------------------------------------------------------------------
##  ::: Pclass (Gaussian)
## ---------------------------------------------------------------------------------
##
## Pclass         0         1
```

```
##   mean 2.5318761 1.9502924
##   sd   0.7358050 0.8633206
##
## ------------------------------------------------------------------------------
##  ::: Sex (Bernoulli)
## ------------------------------------------------------------------------------
##
## Sex              0         1
##   female 0.1475410 0.6812865
##   male   0.8524590 0.3187135
##
## ------------------------------------------------------------------------------
```

Now let's generate a new dataset for which we can generate prediction(s)

```
  # We need to make sure the variables of the same type as the original data
  newdata <- data.frame(Pclass = as.integer(1), Sex = as.character("female"))
  newdata$Sex <- as.character(newdata$Sex)

 str(newdata)
```

```
## 'data.frame':    1 obs. of  2 variables:
##  $ Pclass: int 1
##  $ Sex   : chr "female"
```

We now apply our model to this new dataset to obtain the prediction.

```
# Generate Prediction
predict(model, newdata = newdata)
```

```
## [1] 1
## Levels: 0 1
```