1. Introduction

In this lab we will illustrate Naive Bayes using a very simple and very small data example. We will calculate conditional probabilities by hand and also using an R function. This will give us confidence in that function when we apply it to complicated and very big data later on.

2. Create example data

Let’s create a data frame with example data. There are two variables type which identifies an email message as ham or spam, and viagra which identifies whether or not the word “viagra” is contained in the message.

train <- data.frame(class=c("spam","ham","ham","ham"),

viagra=c("yes","no","no","yes"))

train

## class viagra

## 1 spam yes

## 2 ham no

## 3 ham no

## 4 ham yes

3. Calculate conditional probabilities by hand

Suppose we see a message that has the word “viagra” in it. What is the probability that it is a spam?

4. Estimating naive Bayes model

We will use the naiveBayes() function which is part of e1071 package. There two main arguments of the function. The first is the formula that lists the variable to predict and a list of predictors. The second is the data. In our case we predict class using the variable viagra. The function returns a ‘model’ object.

**library**(e1071)

classifier <- naiveBayes(class ~ viagra,train)

classifier

##

## Naive Bayes Classifier for Discrete Predictors

##

## Call:

## naiveBayes.default(x = X, y = Y, laplace = laplace)

##

## A-priori probabilities:

## Y

## ham spam

## 0.75 0.25

##

## Conditional probabilities:

## viagra

## Y no yes

## ham 0.6666667 0.3333333

## spam 0.0000000 1.0000000

The classifier object contains prior probabilities of different values of the predictors (viagra) given different values of the predicted variable (class). For example P(viagra=no|ham)=0.67(two thirds of ham messages have no ‘viagra’ in them), P(viagra=no|spam)=0 (no spam messages without ‘viagra’). What we are interested in, however, is the probability of spam given certain value of the predictor variables.

5. Using estimated model to calculate conditional probability

We need to create a test data set that contains those certain values of our predictor variables. For example, let’s create a test data set that has the variable viagra which takes on value yes.

test <- data.frame(viagra=c("yes"))

test$viagra <- factor(test$viagra, levels=c("no","yes"))

test

## viagra

## 1 yes

Our test data has just one variable with one observation. This is because we only have one predictor variable viagra, and we want to predict type given only one value of viagra variable, viagra=yes. Notice that we needed to specify that the variable viagra is a factor that can take two values. Without this the classifier would not interpret the test data correctly. Note that the factor levels have to be specified in the same order as in the training dataset. Normally, factor levels are sorted alphabetically.

So, now we are ready to feed our data to our classifier object. For this we use the function predict which takes as arguments a ‘model’ object and test data. We also specify that we want the fiction to return ‘raw’ predictions which in the case of naive Bayes ‘model’ object means conditional probabilities.

prediction <- predict(classifier, test ,type="raw")

prediction

## ham spam

## [1,] 0.5 0.5

Fantastic, we got R to compute the probability of spam given that a message includes the word “viagra.” P(spam|viagra=yes)=0.5P(spam|viagra=yes)=0.5. Reassuringly, this is the same as what we computed by hand.

6. Doing the same with *two* predictors (what makes Naive Bayes naive)

Let’s add variable meet to our spam data. This variable indicates whether or not the word “meet” appears in the message.

train <- data.frame(type=c("spam","ham","ham","ham"),

viagra=c("yes","no","no","yes"),

meet=c("yes","yes","yes", "no"))

train

## type viagra meet

## 1 spam yes yes

## 2 ham no yes

## 3 ham no yes

## 4 ham yes no

7. Calculate conditional probabilities by hand

Suppose we see a message that has the word “viagra” and the word “meet” in it. What is the probability that it is a spam?

8. Estimating Naïve Bayes model

We have two predictors of type viagra and meet.

**library**(e1071)

classifier <- naiveBayes(type ~ viagra + meet,train)

classifier

##

## Naive Bayes Classifier for Discrete Predictors

##

## Call:

## naiveBayes.default(x = X, y = Y, laplace = laplace)

##

## A-priori probabilities:

## Y

## ham spam

## 0.75 0.25

##

## Conditional probabilities:

## viagra

## Y no yes

## ham 0.6666667 0.3333333

## spam 0.0000000 1.0000000

##

## meet

## Y no yes

## ham 0.3333333 0.6666667

## spam 0.0000000 1.0000000

9. Using estimated model to calculate conditional probability

Our test data set has two variables viagra and meet with both variables taking on value yes.

test <- data.frame(viagra=c("yes"), meet=c("yes"))

test$viagra <- factor(test$viagra, levels=c("no","yes"))

test$meet <- factor(test$meet, levels=c("no","yes"))

test

## viagra meet

## 1 yes yes

We are ready to feed our test data to our classifier object.

prediction <- predict(classifier, test ,type="raw")

prediction

## ham spam

## [1,] 0.4 0.6

The naive Bayes algorithm says that given that the message contains words “viagra” and “meet” the probability of it being a spam is 0.6. This is DIFFERENT from the true conditional probability we calculated by hand above. Why?

10. What makes naive Bayes naive?

The naive Bayes algorithm makes the assumption that the predictors are independent. For example, in our case it assumes that the probability that a message contains “viagra” given that it is spam is independent of whether or not the message contains “meet”, i.e. P(viagra=yes,meet=yes|spam)=P(viagra=yes|spam)⋅P(meet=yes|spam)

This assumption greatly simplifies the numerator in the conditional probability formula. Ignoring the denominator for a moment we have: P(spam|viagra=yes,meet=yes)∝P(viagra=yes|spam)⋅P(meet=yes|spam)⋅P(spam)=1⋅1⋅1/4=1/4

Similarly, P(ham|viagra=yes,meet=yes)∝P(viagra=yes|ham)⋅P(meet=yes|ham)⋅P(ham)=1/3⋅2/3⋅3/4=1/6

These numbers are proportional to probabilities, to get the actual probabilities we scale each number by the sum of the two numbers, getting P(spam|viagra=yes,meet=yes)=1/41/4+1/6=3/5=0.6

P(spam|viagra=yes,meet=yes)=1/41/4+1/6=3/5=0.6 and P(ham|viagra=yes,meet=yes)=1/61/4+1/6=3/5=0.4

P(ham|viagra=yes,meet=yes)=1/61/4+1/6=3/5=0.4 which is what the computer calculated.

Intuitively, in our case the naive Bayes understates the fact that ‘meet’ and ‘viagra’ go together in spam messages. The true probability is 1 but naive Bayes only gives us 0.6.