Class 5 Assignment – Naïve Bayes

Brian Mallari

Q1. Bayes theorem shows us how to turn P(E|H) to P(H|E), with E=Evidence and H=Hypothesis. But what does that really mean? Imagine you have to explain this to someone who doesn't understand machine learning or probability at all. How would you do it in a paragraph or two without using any jargons? Use an example from real life to ground the explanation. [4 points]

Let's say, hypothetically, that you and I are in a room together, and between the two of us is a partition – like a curtain or a folding screen – such that we can't see what the other person is doing, but we can still talk to each other without a problem. Now let's say that I have a small bag with two coins inside. One coin is a fair coin with a 50-50 chance of getting either heads or tails. The other coin is a trick coin with heads on both sides, so no matter what, the outcome is going to be heads. Now let's say that I pull out a coin at random, flip it, and then call out that I got a head. What is the probability of that one coin being the trick coin, given that I've gotten a head? Remember, you can't actually see the coin that I've pulled out; you can only hear the result that I've called out.

We can calculate that probability by doing some math with some of the other probabilities that we can figure out given the scenario. It turns out that the probability of the coin that I pulled out being the trick coin given the I've gotten a head (which is the probability that we're trying to figured out), multiplied by the probability of getting heads irrespective of whatever coin I pull out (which is ¾ ,or 0.75, because of the 1 head from the fair coin plus the 2 heads from the trick coin, all divided by the 4 possible sides of any coin that can come up), is actually equal to the probability of getting a head given that I've pulled out the trick coin (which is 1, because head is the only outcome for the trick coin), multiplied by the probability of getting the fair coin (which is ½, or 0.5, because it is one of two possible coins). Altogether, the equation would look like this: (Unknown Probability) * ¾ = 1 * ½. We can divide both sides of this equation by ¾ to isolate the value that we're looking for in order to get (1 * ½) /(¾), which is equal to 2/3 or roughly 0.67. Therefore, the final answer – that is, the probability of that one coin being the trick coin, given that I've called out getting a head – will be 2/3 or roughly 0.67.

Q2. Download a YouTube spam collection dataset available from this link (http://archive.ics.uci.edu/ml/machine-learning-databases/00380/YouTube-Spam-Collection-v1.zip).

This is a public set of comments collected for spam research. It has five datasets composed by 1,956 real messages extracted from five videos. These 5 videos are popular pop songs that were among the 10 most viewed on the collection period.
All the five dataset has the following attributes:

COMMENT_ID: Unique id representing the comment
AUTHOR: Author id,
DATE: Date the comment is posted,
CONTENT: The comment,
TAG: For spam 1, otherwise 0

For this exercise use any 4 of these 5 datasets to build a spam filter and use that filter to check the accuracy on the remaining dataset. Make sure to report the details of your training and the model. [6 points]

Of the five datasets, the following four were used to generate the training set for the Naïve Bayes model:

1. Youtube01-Psy.csv
2. Youtube02-KatyPerry.csv
3. Youtube03-LMFAO.csv
4. Youtube04-Eminem.csv

The last of the five datasets, Youtube05-Shakira.csv, was used to generate the testing set.

The original .csv files used the feature name CLASS instead of TAG to hold information regarding whether or not the comment is spam. However, the meaning of the values still remains the same: 1 = the comment is spam, 0 = the comment is not spam.

The comments for the YouTube videos where kept as strings of text, which were then cleaned up in the following manner for frequency evaluation via document-term matrix:

1. All characters were changed to lower case
2. All numbers were removed
3. Stop words were removed (no restriction was placed on the language since comments can be posted in any language)
4. Punctuation was removed
5. White space was removed
6. Text was formatted as plain text document

Moreover, any text with frequency less than 5 was filtered out of both the training set and the testing set. This was to reduce the processing time to a reasonable amount.

With a slight modification to the CLASS variable in order to get the model-generating function to work, a Naïve Bayes model was generated using the training set, and the predictive capability of the model was then evaluated using the testing set. The output of the testing phase is as follows:

```
> confusionMatrix(prediction_nb_youtube, test_nb_1dataset$CLASS)
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 193  28
         1   3 146

               Accuracy : 0.9162
                 95% CI : (0.8832, 0.9424)
    No Information Rate : 0.5297
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8305
 Mcnemar's Test P-Value : 1.629e-05

            Sensitivity : 0.9847
            Specificity : 0.8391
         Pos Pred Value : 0.8733
         Neg Pred Value : 0.9799
             Prevalence : 0.5297
         Detection Rate : 0.5216
   Detection Prevalence : 0.5973
      Balanced Accuracy : 0.9119

       'Positive' Class : 0
```

According to the results of this model test, the accuracy of this particular spam-detecting Naïve Bayes model is 0.9162, thus making the model very good for filtering out spam YouTube comments. Worth noting here is that a Laplace smoothing factor of 1 was applied to the Naïve Bayes model to account for any possibility of a 0 showing up in the prediction-reference table.

R Script

The .R file for this assignment ("590 - Class 5 - Naive Bayes - Assignment.R") was submitted as an attachment along with a copy of this report.

References

1.) Mathematics, examples, and R code were based off of notes taken for the following video podcasts by Dr. Chirag Shah:
   a. 5.1 Naive Bayes – lecture
   b. 5.2 Naive Bayes – practice
2.) An example of Bayes theorem was taken from the following YouTube video featured at Khan Academy: https://www.khanacademy.org/math/ap-statistics/probability-ap/stats-conditional-probability/v/bayes-theorem-visualized
3.) Information on joining data frames vertically in R was taken from the following webpage: https://www.statmethods.net/management/merging.html
4.) Information on the e1071 package for R was taken from the following webpage: https://cran.r-project.org/web/packages/e1071/e1071.pdf
5.) Information on the caret package for R was taken from the following webpage: https://cran.r-project.org/web/packages/caret/caret.pdf
6.) Information subsetting a data frame in R was taken from the following webpages:
   a. https://www.statmethods.net/management/subset.html
   b. https://stackoverflow.com/questions/5234117/how-to-drop-columns-by-name-in-a-data-frame
7.) Information on resolving the issue of NULL levels when using the naiveBayes() function to generate a model was taken from the following webpage: https://stackoverflow.com/questions/12835515/classification-with-naivebayes-e1071-does-not-work-levels-returns-null
8.) Information on generating a spam filter in R was taken from the following webpage: https://rpubs.com/mzc/mlwr_nb_sms_spam
9.) Information on an "'i, j' invalid" error when running the DocumentTextMatrix() function in R was taken from the following webpage: https://support.rstudio.com/hc/en-us/community/posts/115008004587-Error-in-simple-triplet-matrix-i-j-v-nrow-length-terms-ncol-length-corpus-i-j-invalid
10.) Information on the VCorpus() function in R was taken from the following webpage: https://www.rdocumentation.org/packages/tm/versions/0.7-3/topics/VCorpus
11.) Information on the "tm" package for text mining in R was taken from the following webpage: https://cran.r-project.org/web/packages/tm/vignettes/tm.pdf
12.) Information on stop words was taken from the following webpage: https://en.wikipedia.org/wiki/Stop_words
13.) Information on document-term matrix was taken from the following webpage: https://en.wikipedia.org/wiki/Document-term_matrix
14.) Information on additive smoothing was taken from the following webpage: https://en.wikipedia.org/wiki/Additive_smoothing