Case Study I:
Naïve Bayesian spam filtering

Mike Wiper and Conchi Ausín

Department of Statistics

Universidad Carlos III de Madrid

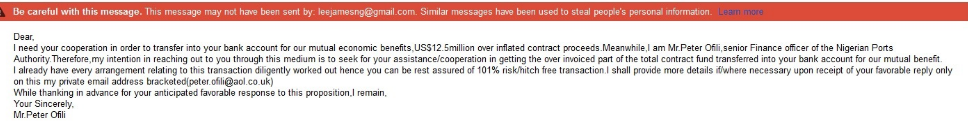
Advanced Statistics and Data Mining Summer School

Objective

We illustrate how to use Bayes theorem to design a simple spam email detector.

$$\Pr(\mathit{spam} \mid \mathit{money}) = \frac{\Pr(\mathit{money} \mid \mathit{spam}) \Pr(\mathit{spam})}{\Pr(\mathit{money})}$$

Spam and ham emails



Be careful with this message. This message may not have been sent by: leejamesng@gmail.com. Similar messages have been used to steal people's personal information. Learn more

Dear,
I need your cooperation in order to transfer into your bank account for our mutual economic benefits,US$12.5million over inflated contract proceeds.Meanwhile,I am Mr.Peter Ofili,senior Finance officer of the Nigerian Ports Authority.Therefore,my intention in reaching out to you through this medium is to seek for your assistance/cooperation in getting the over invoiced part of the total contract fund transferred into your bank account for our mutual benefit.
I already have every arrangement relating to this transaction diligently worked out hence you can be rest assured of 101% risk/hitch free transaction.I shall provide more details if/where necessary upon receipt of your favorable reply only on this my private email address bracketed(peter.ofili@aol.co.uk)
While thanking in advance for your anticipated favorable response to this proposition,I remain,
Your Sincerely,
Mr.Peter Ofili

Example

Assume that we have the following set of email classified as spam or ham.

spam:	“send us your password”
ham:	“send us your review”
ham:	“password review”
spam:	“review us ”
spam:	“send your password”
spam:	“send us your account”

Example

Assume that we have the following set of email classified as spam or ham.

spam:	“send us your password”
ham:	“send us your review”
ham:	“password review”
spam:	“review us ”
spam:	“send your password”
spam:	“send us your account”

We are interested in classifying the following new email as spam or ham:

new email	“review us now”

Using Bayes theorem

spam:	“send us your password”
ham:	“send us your review”
ham:	“password review”
spam:	“review us ”
spam:	“send your password”
spam:	“send us your account”

Using Bayes theorem

spam: “send us your password”

ham: “send us your review”

ham: “password review”

spam: “review us ”

spam: “send your password”

spam: “send us your account”

Prior probabilities are:

$$\Pr(\text{spam}) = \frac{4}{6} \qquad \Pr(\text{ham}) = \frac{2}{6}$$

Using Bayes theorem

spam: “send us your password”

ham: “send us your review”

ham: “password review”

spam: “review us ”

spam: “send your password”

spam: “send us your account”

Prior probabilities are:

$$\Pr(\text{spam}) = \frac{4}{6} \qquad \Pr(\text{ham}) = \frac{2}{6}$$

The posterior probability that an email containing the word “review” is a spam is:

$$\Pr(\text{spam} \mid \text{review}) = \frac{\Pr(\text{review}|\text{spam})\Pr(\text{spam})}{\Pr(\text{review}|\text{spam})\Pr(\text{spam})+\Pr(\text{review}|\text{ham})\Pr(\text{ham})} = \frac{\frac{1}{4}\cdot\frac{4}{6}}{\frac{1}{4}\cdot\frac{4}{6}+\frac{2}{2}\cdot\frac{2}{6}} = \frac{1}{3}$$

For several words...

spam:	“send us your password”
ham:	“send us your review”
ham:	“password review”
spam:	“review us ”
spam:	“send your password”
spam:	“send us your account”

For several words...

spam:	"send us your password"
ham:	"send us your review"
ham:	"password review"
spam:	"review us "
spam:	"send your password"
spam:	"send us your account"

	$\Pr(\cdot \mid \text{spam})$	$\Pr(\cdot \mid \text{ham})$
review	$1/4$	$2/2$
send	$3/4$	$1/2$
us	$3/4$	$1/2$
your	$3/4$	$1/2$
password	$2/4$	$1/2$
account	$1/4$	$0/2$

For several words...

	$\Pr(\cdot \mid \text{spam})$	$\Pr(\cdot \mid \text{ham})$
review	$1/4$	$2/2$
send	$3/4$	$1/2$
us	$3/4$	$1/2$
your	$3/4$	$1/2$
password	$2/4$	$1/2$
account	$1/4$	$0/2$

- Assuming that the words in each message are independent events:

$$\begin{aligned}\Pr(\text{review us now} \mid \text{spam}) &= \Pr(\{1,0,1,0,0,0\} \mid \text{spam}) \\ &= \frac{1}{4}\left(1-\frac{3}{4}\right)\frac{3}{4}\left(1-\frac{3}{4}\right)\left(1-\frac{2}{4}\right)\left(1-\frac{1}{4}\right) = 0.0044\end{aligned}$$

$$\begin{aligned}\Pr(\text{review us now } \mid \text{ham}) &= \Pr(\{1,0,1,0,0,0\} \mid \text{ham}) \\ &= \frac{2}{2}\left(1-\frac{1}{2}\right)\frac{1}{2}\left(1-\frac{1}{2}\right)\left(1-\frac{1}{2}\right)\left(1-\frac{0}{4}\right) = 0.0625\end{aligned}$$

Using Bayes theorem

Then, the posterior probability that the new email "review us now" is a spam is:

$$\begin{aligned}\Pr(\text{spam} \mid \text{review us now}) &= \Pr(\text{spam} \mid \{1,0,1,0,0,0\}) \\ &= \frac{\Pr(\{1,0,1,0,0,0\}|\text{spam})\Pr(\text{spam})}{\Pr(\{1,0,1,0,0,0\}|\text{spam})\Pr(\text{spam})+\Pr(\{1,0,1,0,0,0\}|\text{ham})\Pr(\text{ham})} \\ &= \frac{0.0044\cdot\frac{4}{6}}{0.0044\cdot\frac{4}{6}+0.0625\cdot\frac{2}{6}} = 0.123\end{aligned}$$

Consequently, the new email will be classified as ham.

- Note that the independence assumption between words will not in general be satisfied, but it can be useful for a naive approach.

Example: SMS spam data

Consider the file “sms.csv” which contains a study of SMS records classified as “spam” or “ham”.

```
rm(list=ls())
sms <- read.csv("sms.csv",sep=","))
names(sms)
head(sms)
```

We want to use a naive Bayes classifier to build a spam filter based on the words in the message.

Prepare the Corpus

A corpus is a collection of documents.

```
library(tm)
```

```
corpus <- Corpus(VectorSource(sms$text))
inspect(corpus[1:3])
```

Here, `VectorSource` tells the `Corpus` function that each document is an entry in the vector.

Clean the Corpus

Different texts may contain “Hello!”, “Hello,” “hello”, etc. We would like to consider all of these the same. We clean up the corpus with the `tm_map` function.

- Translate all letters to lower case:

```
clean_corpus <- tm_map(corpus, tolower)
```

```
inspect(clean_corpus[1:3])
```

- Remove numbers:

```
clean_corpus <- tm_map(clean_corpus, removeNumbers)
```

- Remove punctuation:

```
clean_corpus <- tm_map(clean_corpus, removePunctuation)
```

Clean the Corpus

- Remove common non-content words, like to, and, the,.. These are called *stop words*. The function stopwords reports a list of about 175 such words.

```
stopwords("en")[1:10]
clean_corpus <- tm_map(clean_corpus, removeWords,
stopwords("en"))
```

- Remove the excess white space:

```
clean_corpus <- tm_map(clean_corpus, stripWhitespace)
```

Word clouds

We create word clouds to visualize the differences between the two message types, ham or spam.

First, obtain the indices of spam and ham messages:

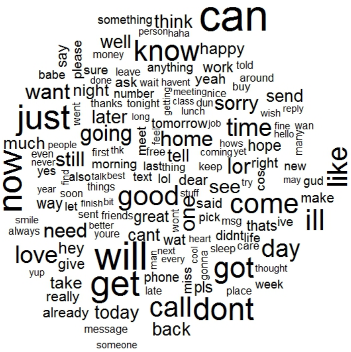
```
spam_indices <- which(sms$type == "spam")
spam_indices[1:3]
```

```
ham_indices <- which(sms$type == "ham")
ham_indices[1:3]
```

Word clouds

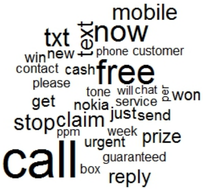
```
library(wordcloud)
```

```
wordcloud(clean_corpus[ham_indices], min.freq=40, scale=c(3,.5))
```



Word clouds

```
wordcloud(clean_corpus[spam_indices], min.freq=40)
```



Building a spam filter

Divide corpus into training and test data. Use 75% training and 25% test:

```
sms_train <- sms[1:4169,]
sms_test <- sms[4170:5559,]
```

And the clean corpus:

```
corpus_train <- clean_corpus[1:4169]
corpus_test <- clean_corpus[4170:5559]
```

Compute the frequency of terms

Using DocumentTermMatrix, we create a sparse matrix data structure in which the rows of the matrix refer to document and the columns refer to words.

```
sms_dtm <- DocumentTermMatrix(clean_corpus)
inspect(sms_dtm[1:4, 30:35])
```

Divide the matrix into training and test rows.

```
sms_dtm_train <- sms_dtm[1:4169,]
sms_dtm_test <- sms_dtm[4170:5559,]
```

Identify frequently used words

Don't muddy the classifier with words that may only occur a few times.

To identify words appearing at least 5 times:

```
five_times_words <- findFreqTerms(sms_dtm_train, 5)
length(five_times_words)
five_times_words[1:5]
```

Create document-term matrices using frequent words:

```
sms_dtm_train <- DocumentTermMatrix(corpus_train,
control=list(dictionary = five_times_words))
```

```
sms_dtm_test <- DocumentTermMatrix(corpus_test,
control=list(dictionary = five_times_words))
```

Convert count information to “Yes”, “No”

Naive Bayes classification needs present or absent info on each word in a message. We have counts of occurrences. To convert the document-term matrices:

```
convert_count <- function(x){
y <- ifelse(x > 0, 1,0)
y <- factor(y, levels=c(0,1), labels=c("No", "Yes"))
y
}
```

Convert document-term matrices

```
sms_dtm_train <- apply(sms_dtm_train, 2, convert_count)
sms_dtm_train[1:4, 30:35]
```

```
sms_dtm_test <- apply(sms_dtm_test, 2, convert_count)
sms_dtm_test[1:4, 30:35]
```

Create a Naive Bayes classifier

We will use a Naive Bayes classifier provided in the package e1071.

```
library(e1071)
```

We create the classifier using the training data.

```
classifier <- naiveBayes(sms_dtm_train, sms_train$type)
class(classifier)
```

Evaluate the performance on the test data

Given the classifier object, we can use the `predict` function to test the model on new data.

```
predictions <- predict(classifier, newdata=sms_dtm_test)
```

Classifications of messages in the test set are based on the probabilities generated with the training set.

Check the predictions against reality

We have predictions and we have a factor of real spam-ham classifications. Generate a table.

```
table(predictions, sms_test$type)
```

	True ham	True spam
Predicted as ham	1202	31
Predicted as spam	5	152

Spam filter performance:

- It correctly classifies 99% of the ham;
- It correctly classifies 83% of the spam;
- This is good balance.

Problems with the Naive Bayes classifier

	$\Pr(\cdot \mid \text{spam})$	$\Pr(\cdot \mid \text{ham})$
review	$1/4$	$2/2$
send	$3/4$	$1/2$
us	$3/4$	$1/2$
your	$3/4$	$1/2$
password	$2/4$	$1/2$
account	$1/4$	$0/2$

Thus,

$$\Pr(\text{review your account } \mid \text{ham}) = \Pr(\{1,0,0,1,0,1\} \mid \text{ham}) = 0$$

$$\Pr(\text{ham} \mid \text{review your account}) = \frac{\Pr(\{1,0,0,1,0,1\}|\text{ham})\Pr(\text{ham})}{\Pr(\{1,0,0,1,0,1\}|\text{spam})\Pr(\text{spam})+\Pr(\{1,0,0,1,0,1\}|\text{ham})\Pr(\text{ham})}$$
$$= \frac{0 \cdot \frac{2}{6}}{0.0014 \cdot \frac{4}{6} + 0 \cdot \frac{2}{6}} = 0$$

Therefore, any new email with the word “account” will be classified as spam.

Bayesian Naive Bayes classifier

- Using a Bayesian approach, the unknown probabilities are random variables,

$$\theta_s = \Pr(\text{spam}) \qquad 1 - \theta_s = \Pr(\text{ham}).$$

- A prior distribution is assumed on θ_s describing our prior beliefs. For example, we may assume a uniform $\theta_s \sim U(0,1)$ such that, $f(\theta_s) = 1$.

- Given the observed data, we update to the posterior distribution using the Bayes theorem:

$$f(\theta_s \mid \text{data}) = \frac{f(\text{data} \mid \theta_s) f(\theta_s)}{f(\text{data})} \propto f(\text{data} \mid \theta_s) f(\theta_s)$$

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{evidence}} \propto \text{likelihood} \times \text{prior}$$

Bayesian Naive Bayes classifier

- In the example, we had 4 spam emails and 2 ham emails. Then, the likelihood is:

$$f(\text{data} \mid \theta_s) \propto \theta_s^4 (1-\theta_s)^2$$

- Then, the posterior distribution is:

$$f(\theta_s \mid \text{data}) \propto \theta_s^{5-1} (1-\theta_s)^{3-1}$$

  Thus, the posterior distribution follows a beta distribution:

$$\theta_s \mid \text{data} \sim Beta(1+4, 1+2)$$

- A Bayesian point estimate can be obtained using the mean of the posterior distribution:

$$\Pr(\text{spam}) = \frac{1+4}{2+6} = \frac{5}{8} \qquad \Pr(\text{ham}) = \frac{1+2}{2+6} = \frac{3}{8}$$

Bayesian Naive Bayes classifier

spam: “send us your password”

ham: “send us your review”

ham: “password review”

spam: “review us ”

spam: “send your password”

spam: “send us your account”

Similarly, we may assume uniform prior distributions for the probabilities:

$$\theta_{rs} = \Pr(\text{review} \mid \text{spam}) \qquad \theta_{rh} = \Pr(\text{review} \mid \text{ham})$$

And obtain the posterior distributions:

$$\theta_{rs} \mid \text{data} \sim Beta(1+1, 1+3) \qquad \theta_{rh} \mid \text{data} \sim Beta(1+2, 1+0)$$

Therefore, Bayesian point estimates are:

$$\Pr(\text{review} \mid \text{spam}) = \frac{1+1}{2+4} = \frac{2}{6} \qquad \Pr(\text{review} \mid \text{ham}) = \frac{1+2}{2+2} = \frac{3}{4}$$

Bayesian Naive Bayes classifier

spam:	“send us your password”
ham:	“send us your review”
ham:	“password review”
spam:	“review us ”
spam:	“send your password”
spam:	“send us your account”

Consequently, the Bayesian probability that an email with the word “review” is a spam is:

$$\begin{aligned}\Pr\left(\text{spam} \mid \text{review}\right) &= \frac{\Pr(\text{review}|\text{spam})\Pr(\text{spam})}{\Pr(\text{review}|\text{spam})\Pr(\text{spam})+\Pr(\text{review}|\text{ham})\Pr(\text{ham})} \\ &= \frac{\frac{1+1}{2+4}\frac{1+4}{2+6}}{\frac{1+1}{2+4}\frac{1+4}{2+6}+\frac{1+2}{2+2}\frac{1+2}{2+6}} = 0.425\end{aligned}$$

which is somewhat larger than the estimated probability with the classical approach (1/3).

Bayesian Naive Bayes classifier

For several words, the Bayesian estimates are:

	$\Pr(\cdot \mid \text{spam})$	$\Pr(\cdot \mid \text{ham})$
review	$2/6$	$3/4$
send	$4/6$	$2/4$
us	$4/6$	$2/4$
your	$4/6$	$2/4$
password	$3/6$	$2/4$
account	$2/6$	$1/4$

Then,

$$\Pr(\text{spam} \mid \text{review your account}) = \frac{\frac{2}{6}\frac{2}{6}\frac{2}{6}\frac{4}{6}\frac{3}{6}\frac{2}{6}\frac{5}{8}}{\frac{2}{6}\frac{2}{6}\frac{2}{6}\frac{4}{6}\frac{3}{6}\frac{2}{6}\frac{5}{8} + \frac{3}{4}\frac{2}{4}\frac{2}{4}\frac{2}{4}\frac{2}{4}\frac{1}{4}\frac{3}{8}} = 0.369$$

Therefore, the new email will be classified as ham, unlike the classical approach.

Bayesian Naive Bayes classifier

The Bayesian Naive Bayes with uniform priors is equivalent to the frequently called “Laplacian smoothing”.

In the sms example, it can be incorporated using:

```
B.clas <- naiveBayes(sms_dtm_train, sms_train$type,laplace = 1)
class(B.clas)
B.preds <- predict(B.clas, newdata=sms_dtm_test)
table(B.preds, sms_test$type)
```

	True ham	True spam
Predicted as ham	1204	31
Predicted as spam	3	152

The Bayesian Spam filter performance is slightly better than the classical approach.

Summary

- We have shown how to use the Bayes theorem to construct a spam email detector.
- First, we have constructed a clean Corpus based on a collection of texts.
- Then, we have obtain a word could for each group of texts.
- Using a (training) data subset, we have constructed a document-term matrix using the most frequent words.
- We have constructed the naive Bayes classiffier based on empirical frequencies.
- Finally, we have shown how to extend the naive Bayes classiffier using Bayesian estimates which avoid zero probabilities in the the classifier.