Damien Matias - Victor Lagrenaudie

machine learning report

Discriminative and Generative Models

# Task

From a dataset composed of 5000 lines, we must be able to tell if a line comes from a real message or from a spam.

First we have to define a training set and a test set. Arbitrarily, we divide the dataset in two, so the training set is composed of 2500 lines, as the testing set.

From this training set we created a Dictionnary. Within it, we add each word which was not “ham” either “spam”, and their occurrence. Then , to have better results we filtered the dictionary by removing words of 1 character and words not containing alphabetic characters only. Finally, we returned the dictionary with the 3000 most used words, based on occurrence.

After that, we filled a matrix of Y based on the first word of the line. If it’s ‘ham’ (a real message), we put for instance a 0, if it’s ‘spam’, we added a 1.

Then, we create a future matrix for training set. This matrix as shape equal to the number of line in training set multiplied by the number of words in ou dictionary. In our case it’s a 2500\*3000. We initialized it with a 0 value in each cell, but we well soon fill it with other value. For each line, we checked if a word of the dictionary was part of this line. If it’s true, we filled the matrix such as this way [lineID,wordID] = words.count(word). In other words, we combined the occurrence of a word with its line number and the ID of this word.

We get the Y\_test labels with the same process as the one used for Y\_train. We also get the matrix for test with the technique used the paragraphe above.

Now, we are able to create and train our two models, SVM and Naive bayes classifier, with out train\_matrix and y\_train.

Then, for each of these models, we use the predict() function with our test\_matrix .

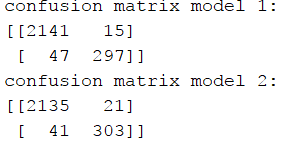
For the last part, we need to see if our models worked. To do that, we can compare the result of what we got from the predict() function and our Y\_train matrix (which represents the good answers for training set).

# Results :

To begin, we wanted to know the score of our models, which compare the results we get from our predict() functions and our Y\_train.



As we see here, we have the exact same score for both models, and both are really close to 1, which is good. To have a deeper view on what model predict what value, we printed the confusion matrix for each model :



The confusion matrix look very close to each other. For each of them the number of False Negative or False Positive are very low.

As last value we show here the F score for models:



# Conclusion

To conclude both of the models worked very well for this exercise . We have a very good prediction using these algorithms.