

Exercise Sheet 11

Submit until Tuesday, January 24 at 2:00pm

This exercise sheet is about text classification using Naive Bayes. We recommend that you proceed from the (Python) code provided on the Wiki. It already provides code for reading a set of labeled documents and constructing the corresponding document-term matrix (one row per document!) and label vector (one entry per document). It also provides specifications for Exercises 1 - 3 below as well as the unit tests. In case you don't use the provided code, the test cases must still be implemented!

Like for the previous three exercise sheets, the two main methods (*train* and *predict*) are most elegantly implemented using matrix-vector operations. However, if you prefer and it helps your understanding, you can also do the necessary computations using (plain and boring) for-loops.

Exercise 1 (5 points)

Write a method *train* that computes the probabilities p_c and p_{wc} for a given (training) set of documents. Use smoothing, as explained in the lecture, to make sure that none of the p_{wc} is zero.

Exercise 2 (5 points)

Write a method *predict* that computes the most likely class for each document from a given (testing) set. Use the p_c and p_{wc} learned using *train*, as explained in the lecture.

Consider the implementation advice from the lecture and add the logarithms of the appropriate probabilities, instead of multiplying the probabilities. For a more efficient prediction, you may (but do not have to) already compute the logarithms in the *train* method.

Exercise 3 (5 points)

Complete the code such that it takes the names of two files as arguments: the training data and the test data. The program should learn from the training data, then predict labels for the test data and output precision and recall of the predictions for each class label. Also output, for each class, the value of p_c and the 30 words with the highest p_{wc} value.

Exercise 4 (5 points)

Run your code for the two sets of train and test data on the Wiki (movie genres and ratings). Briefly discuss the results in your *experiences.txt*. In particular: for which classes did the prediction work well and for which not so well and why. Also: how much sense do the top-30 words per class make and how is this related to the precision and recall you observe.

Add your code to a new sub-directory *sheet-11* of your folder in the course SVN, and commit it. As usual, make sure that *compile*, *test*, and *checkstyle* run through without errors on Jenkins. And, of course, never forget the *experiences.txt*, where you inform us about your adventures with this exercise sheet.