INF5830, 2017, Obligatory assignment 2

Deadline: Oct. 6, 18.00
To be delivered in devilry

Exercise 1 – Data set and first classifier (10%)

We will consider document classification. As our data set, we will use the Movie Reviews Corpus that comes with NLTK. Initially we will use tools for classification from NLTK. You should work your way through the first part of chapter 6 in the NLTK book up to and including the part on Document Classification (Ch. 6, sec. 1.0-1.3 in the internet edition.)

To do it ourselves, we will be a bit more careful than the book. Before extracting the features or anything else, we will take away 400 of the 2000 documents for final test data and keep 1600 documents for development data. Pick the test data at random. One possibility is to shuffle the data. To be able to use the same split across several experiments, you can use a seed for the random generator and use the same seed across the experiments. To illustrate

```
    a = [i for i in range(10)]
    random.shuffle(a)
```

will give different results each time we try it, while

```
    a = [i for i in range(10)]
    random.seed(14)
    random.shuffle(a)
```

will yield the same result each time it is run. You can use your date of birth as random seed, e.g. 1509 for the fifteenth of September.

Then split the development data into 1400 sentences for training and 200 for evaluation. Following the recipe in NLTK, build a Naive Bayes classifier and test it on the development test set. In contrast to NLTK, you should only use the training set, not the hold data set, for determining the 2000 most frequent words. Calculate accuracy, recall, precision and F-score.

Deliveries: The code you made for the experiment. The numbers with an explanation of how they were calculated.

Exercise 2 – 10-fold cross-validation, variation and estimation (30%)

a) We are using the same development set of 1600 sentences, but now we will instead conduct a 10-fold cross-validation experiment. Do that, and record the accuracies for each of the 10 experiments.

Deliveries: The code you made for the experiment. The numbers for the accuracies.

b) The ten accuracies is a set of ten numbers. What is the mean, variance and standard deviation of these numbers?

Deliveries: The numbers with explanations of how you found them.

c) Strictly speaking, we are constructing 10 different classifiers in (b). But as any pair of classifiers share 8/9 of the training material, we can nearly consider them as one and the same classifier. If we do that, the 10 different test sets can be considered 10 random samples from a larger set of items, where each item belongs to one of two classes: the ones that are classified correctly by the classifier, and the ones that are classified incorrectly. What kind of (theoretical) distribution can we assume the samples to follow? What is the theoretical standard deviation for this distribution, and how does this correspond to the observations in (b)?

Deliveries: Answers to the questions with explanations.

d) Given the same simplifying assumption as in (c) – that we consider the 10 different classifiers as the same classifier – the mean of the 10 evaluations we found in (b) can be considered an estimate for the accuracy of this classifier. Since this is derived from a sample (of 1600 individuals), it is only an estimate of the true accuracy. Now estimate a 95% confidence interval for the true accuracy.

Deliveries: Answers to the questions with explanations.

Exercise 3 – Text classification (20%)

The NLTK Naive Bayes classifier is a Bernoulli classifier. For text classification, one often uses the multinomial classifier. NLTK does not support this. We are therefore moving on to scikit learn (http://scikit-learn.org/stable/index.html), which supports a whole array of different classifiers. To get familiar with the toolkit, you should first read the general introduction: http://scikit-learn.org/stable/tutorial/basic/tutorial.html.
As we explained in lecture 3, scikit uses representations different from NLTK.

The goal is to build a MultinomialNB classifier for the movie reviews. You can to a large degree follow the recipe from
http://scikit-learn.org/stable/tutorial/text_analytics/working_with_text_data.html.
It will make the job easier if you first work through the scikit example before moving on to the movie reviews. For me, the command

```
from sklearn.datasets import fetch_20newsgroups
```

worked without downloading anything.

We will use the same data set as in exercise 1. We can access these data through NLTK as before. One difference to exercise 1 is that we there use the tokenized texts with the command

```
movie_reviews.words(fileid)
```

Following the recipe from the scikit Working with Text page, we can instead use the raw documents which we can get from NLTK by

```
movie_reviews.raw(fileid)
```

scikit will then do the tokenization for us as part of

```
count_vect.fit_transform
```

Consider http://scikit-learn.org/stable/modules/feature_extraction.html, in particular the section 4.2.3 Text feature extraction to understand what is going on.

a) Use the same splits as in exercise 2. Run 10-fold cross-validation using MultinomialNB classifier without tfidf and record the the accuracies for each of the ten experiments.

Deliveries: The code you made for the experiment. A 10*2 table showing the accuracies for each split for the Bernoulli classifier from exercise 2 together with the accuracies for the MultinomialNB classifier.

b) Run a similar 10-fold cross-validation experiment with the same split, this time with tfidf. Record the mean of the accuracies. Also, record the mean of the accuracies from point (a) and compare the two numbers to each other and to the mean accuracy from exercise 2b.

Deliveries: The accuracies together with the code you used.

Exercise 4 – BernoulliNB in scikit learn(20%)

We will now see how the experiments from exercise 1 and exercise 2 can be carried out in scikit learn. As in exercise 1 and 2, the features will be Boolean and based on the 2000 most frequent words. We can therefore not use the setup from exercise 3 without changes. To find the 2000 most frequent words and extract the features, we can do this exactly as in exercise 1 and 2. Then the documents will be represented by feature dictionaries. It remains to transform these dictionaries to numpy arrays of the form scikit accepts. For this, we can use scikit's DictVectorizer, see sec tion 4.2.1 in http://scikit-learn.org/stable/modules/feature_extraction.html. (Alternatively, you can extract the features directly as arrays without using dictionaries. It is a little more work, but the experiments will run faster.)

a) Repeat experiment (2a) in scikit as explained and record the mean accuracy.

(Do you get the same result as in (2a)? Probably not exactly the same result. One reason for this is that NLTK and scikit has different defaults for smoothing. If you use BernoulliNB(alpha=0.5) in scikit learn (Lidstone smoothing, "add 0.5"), you should come close to the same result as NLTK.)

b) With this setup, we may also try other classifiers, in particular Logistic Regression. Exchange BernoullNB with LogisticRegression and repeat the experiment. You import LogisticRegression by

```
from sklearn.linear_model import LogisticRegression
```

c) The default from NLTK is to use the 2000 most frequent words. We will explore the effect of the size of the feature set. Repeat the 10-fold cross-validation experiment with the 1000, 2000, 5000, 10000 and 20000 most frequent words as features, both with BernoulliNB and with LogisticRegression.

Deliveries: A 5*2 table showing the mean accuracies for 10-fold cross-validation for 1000, 2000, 5000, 1000, and 20000 features for the two different classifiers.

Exercise 5 – Evaluation and matched pairs(20%)

Say you have constructed two classifiers, classifier A and classifier B and you test them on 1600 items. On this test set, classifier A classifies 1280 items correctly. It has an estimated accuracy of 0.8. While classifier B classifies 1300 items correctly. It has an estimated accuracy of 0.8125. Can you conclude that classifier B is better than classifier A if this is all you know?

a) To compare the two, we will first use the two-sided t-test. What is the p-value you get when you compare the two with the two-sample t-test?

$$t = \frac{\bar{x}_B - \bar{x}_A}{\sqrt{\frac{s_A^2}{N} + \frac{s_B^2}{N}}}$$

Would you conclude that parser B is better than parser A?

Deliveries: Answer the question and explain how you got the numbers.

b) Since the two classifiers where tested on the same test set, you have some more information available. You can compare the two classifiers item by item, and the joint results may be summarized as follows.

- There are 1260 items where both classifiers succeed.
- There are 20 sentences where A succeeds and B fails.
- There are 40 sentences where B succeeds and A fails.
- There are 280 sentences where both classifiers fail.

Can you now conclude that classifier B is better than classifier A?

Deliveries: Answer the question and explain how you found the answer.

The END