Learning from Data – Week 3
Assignment 3: Support Vector Machines and Feature Analysis

General remarks

This assignment is meant to get your acquainted with Support Vector Machines. You will also have to explore feature contribution quite a bit further, and do some feature analysis. Additionally, you're encouraged to experiment a little further with report writing, by making your own choices (how do you want to report on the experiments? How about feature description? And feature analysis?). But don't forget the feedback of the lecture!

For the practical parts, you are asked to train and test a few models using SVM, and produce what you think the best settings are for the data we're using. You are also asked to experiment with different features (see the slides of this week) and understand what role different features are playing.

Please note: for this assignment you will work in **groups of 3**. In the report, please include a section at the end where you clarify who did what.

What you have to hand in, **as a group**:

- Code with your SVM model (see Exercise 3.1). You have to assume that we will run it like this:
  python `LFD_assignment3.py -i <trainset> -ts <testset>`
  where trainset and testset have the same format. You are given the trainset, but we're holding out the test set again. Please take all feedback from the previous assignment into account.
- Research report (based on the template) that describes your experiment and also includes or directly incorporates answers/discussion to all questions in this document. Again, take all feedback from the previous report into account (also see the slides). The code is worth 25% of your grade and the report the other 75%.

Deadline: 27 September 2020, 10:59.

Data

For this assignment, we will be using the review data we used for Assignment 1 and Assignment 2, and which you find can still find on Nestor.

Exercise 3.1 – Support Vector Machines

You will be working on the *binary sentiment classification* with the dataset from Assignment 2, which is uploaded on Nestor again for this assignment. We are again withholding the test set, and we will evaluate your model on it.

3.1.1 Default settings

Run a support vector machine with a linear kernel (`cls = svm.SVC(kernel='linear', C=1.0)` on the binary sentiment classification. Use default settings and report results using either cross-validation or a portion of the dataset as development set. Just make sure you document what you have done and what features you have used (so far).

3.1.2 Setting C

Try to change the value of the C parameter, and see if you can get better results (either with cross-validation or on a development set, just use the same settings as above). Please, in the report include details of what you observe by changing C, and also explain what the C parameter is used for. Were the results to be expected?

3.1.3 Using a non-linear kernel

Do the same as above, but use a radial basis function (rbf) kernel. You will also have to specify a *gamma* parameter, in addition to C (e.g. `cls = svm.SVC(kernel='rbf', gamma=0.7, C=1.0`) You can check information on the *gamma* (and C) parameter here: `https://scikit-learn.org/stable/auto_examples/svm/plot_rbf_parameters.html`.
Experiment with changing these values, and report what you observe. It has been claimed that rbf kernels are not great for text classification, and are normally outperformed by linear kernels — do you observe the same? Do the results hold for different feature sets?

You are free to use grid search to find the best parameters, but bear in mind that it might be computationally challenging.

3.1.4 Implementation differences

In scikit there are two SVM implementations: `svm.SVC` and `svm.LinearSVC`. By checking the documentation online, and by testing both implementations for your experiments, please write up a short paragraph in your report where you explain what the differences between the two implementations are, and any usage recommendation you might have (when to use one implementation or the other).

Remember that it's also possible to convert the labels to numerical values and use an SVM regressor (svm.SVR()). The predictions are now numerical values, but we still like to evaluate using accuracy and F-score, so please convert the predictions back to categories. How can you best do this? In general, do the results change a lot by trying this? Was it helpful? Please report what you observe.

3.1.5 Best SVM model

Based on your observations and results on cross-validated data or on your development set, make a final decision on your features, kernel and parameters, and report it in the document. Send in your Python file with best SVM model, like you did for Assignment 2 (see above for submission instructions).

You are expected to experiment with adding features, too. Possibly n-grams might help? Or part-of-speech information? You have to build larger, more informed systems, so this is a good place to start experimenting. We will run it on held-out data, and send back the results to you. Please, make sure that your script takes arguments, as you did for Assignment 2 (see above for specifics). Again, the best performing group(s) get a bonus on their grade!

3.1.6 Feature contribution

As mentioned in the previous section (Section 3.1.5), you are encouraged to experiment with additional features. For this portion of the exercise, you are also asked to more closely *explore* which features actually contribute the most. You can use the attribute `svm.coef_` to do this. Try to explore how to use it, and what information it gives you (weight? direction?), and comment on the results you obtain with your model. Were the results as expected? Do they point to something? (Please note that this attribute only works when using a linear kernel.)

Exercise 3.2 – Report

Using the provided template, and incorporating the general recommendations on Assignment 1, write a report in the form of a research paper that describes the experiments you've run. Try to answer the questions asked in the assignment in a natural way throughout the paper. Also, be aware of your previous feedback and the general feedback that was given in the lecture!