Homework 2

Naive Bayes and Logistic Regression for Text Classification

In this homework you will implement and evaluate Naive Bayes and Logistic Regression for text classification. It is acceptable to look at WEKA's Java code. However, you cannot copy code from WEKA.

You can use either C/C++, Java or Python to implement your algorithms. Your C/C++ implementations should compile on Linux gcc/g++ compiler.

0 Points Download the spam/ham (ham is not spam) dataset available on the class web page. The data set is divided into two sets: training set and test set. The dataset was used in the Metsis et al. paper [1]. Each set has two directories: spam and ham. All files in the spam folders are spam messages and all files in the ham folder are legitimate (non spam) messages.

25 points Implement the multinomial Naive Bayes algorithm for text classification described here: `http://nlp.stanford.edu/IR-book/pdf/13bayes.pdf` (see Figure 13.2). Note that the algorithm uses add-one laplace smoothing. Make sure that you do all the calculations in log-scale to avoid underflow. Use your algorithm to learn from the training set and report accuracy on the test set.

25 points Implement the MCAP Logistic Regression algorithm with L2 regularization that we discussed in class (see Mitchell's new book chapter). Try different values of λ. Use your algorithm to learn from the training set and report accuracy on the test set for different values of λ. Use gradient ascent for learning the weights. Do not run gradient ascent until convergence; you should put a suitable hard limit on the number of iterations.

25 points Improve your Naive Bayes and Logistic Regression algorithms by throwing away (i.e., filtering out) stop words such as "the" "of" and "for" from all the documents. A list of stop words can be found here: `http://www.ranks.nl/resources/stopwords.html`. Report accuracy for both Naive Bayes and Logistic Regression for this filtered set. Does the accuracy improve? Explain why the accuracy improves or why it does not?

Extra Credit, 10 points: Implement a smoothing or feature selection method for improving the accuracy of your classifier. Report the accuracy with this method. Why did your method work or why it did not?

What to Turn in

- Your code and a Readme file for compiling and executing your code.
- A detailed write up (**worth 25 points**) that reports the accuracy obtained on the test set, parameters used (e.g., values of λ, hard limit on the number of iterations, etc.) and answers with suitable explanations to the questions posed above. We should be able to replicate your results based on your writeup.

References

[1] V. Metsis, I. Androutsopoulos and G. Paliouras, "Spam Filtering with Naive Bayes - Which Naive Bayes?". Proceedings of the 3rd Conference on Email and Anti-Spam (CEAS 2006), Mountain View, CA, USA, 2006.