Machine Learning MT 2012: Week 3, Practical 1

Lecturer: Phil Blunsom
Demonstrator: Vasile Palade

Introduction

In this introductory practical you will complete the implementation of a Naive Bayes model for document classification.

Python framework code is provided such that the Naive Bayes model can be implemented by filling in the missing functions. A binary document classification dataset is supplied which consists of text reviews of movies divided into negative and positive classes depending on the prevailing sentiment of the reviewers towards the movies.

Code

The code for this project is written to run under Python 2.6. Googling Python will reveal a wealth of API information and tutorials. You should also note that the python interactive interpreter is invaluable for experimenting with the language. In particular the `dir()` and `help()` functions are very useful for learning about the interfaces of the APIs. For example `help(open)` displays a short help page on the built-in function `open`.

At the following URL you will find an archive of files for this practical which, when downloaded and unzipped, will provide a directory `ml_practical_1`:

`http://www.cs.ox.ac.uk/teaching/materials12-13/machinelearning/practical1.tar.gz`

This directory contains a partially implemented Python source file and a directory containing movie reviews. The `naive_bayes.py` source file contains calls to a `todo()` placeholder function which throws an exception. You should carefully read through the provided code to understand how it implements the text classification model described in the lecture notes. To complete this practical you will then need to provide the code that should replace these `todos`.

naive_bayes.py This file contains an implementation of a binary Naive Bayes classifier for the review data. It contains two functions, `experiment` and `main`. `experiment` reads the review documents, randomly partitions the documents into training and test sets, collects the word counts needed to train the Naive Bayes classifier, and then tests the classifier on the test documents. The `main` function processes the command-line arguments and calls `experiment`, possibly multiple times.

stopwords.py This file contains a list of stop-words which can be used to filter the word features for the Naive Bayes classifier.

Data

Review data: This data consists of the text of movie review divide into positive and negative sets. The creation of this data is described in the research paper:

> Bo Pang and Lillian Lee, *A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts*, Proceedings of ACL 2004.
> `http://www.cs.cornell.edu/people/pabo/movie-review-data/`

Task

The file `naive_bayes.py` contains two `todo()` statements that you need to replace with code in order to complete the implementation of the classifier. You should implement the MAP calculations for the probabilities `p(C)` and `p(word|C)` with Dirichlet prior parameters set from the command-line (`args['alpha']` and `args['beta']` respectively). As mentioned in the code, it is a good idea to calculate the required probabilities in log space (python function `log()`) in order to avoid numerical underflow caused by multiplying lots of small numbers together.

Once you have a working implementation, run the model repeatedly (use the `-i` argument) for multiple different values for alpha and beta (`-a` and `-b`). Graph the results by hand and identify optimum values for the prior parameters.

Assessment

This practical will not be assessed. It is intended to reinforce the elements of the Naive Bayes algorithm, document classification, and setting model parameters.