Homework 2
CS 436/580L: Introduction to Machine Learning

Instructor: Arti Ramesh

Instructions

1. You can use either C/C++, Java or Python to implement your algorithms.
2. **Your implementations should compile on remote.cs.binghamton.edu.**
3. Make sure remote.cs.binghamton.edu has the packages that you require before starting to implement.
4. This homework requires you to **implement** the algorithms. Using existing packages for the algorithms is not allowed.
5. Your homework should contain the following components:
   (a) README.txt file with detailed instructions on how to compile and run the code.
   (b) Code source files
   (c) Type-written document containing the results on the datasets and answers to point estimation questions.
6. Refer to submission instructions on myCourses on naming conventions. If you fail to adhere with the submission instructions points **will** be deducted.

1 Naive Bayes for Text Classification

In this question, you will implement and evaluate Naive Bayes for text classification.

0 Points Download the spam/ham (ham is not spam) dataset available on myCourses. 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.

40 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. Ignore punctuation and special characters and normalize words by converting them to lower case, converting plural words to singular (i.e., "Here" and "here" are the same word, "pens" and "pen" are the same word). Normalize words by stemming them using an online stemmer such as `http://www.nltk.org/howto/stem.html`. 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.

10 points Improve your Naive Bayes 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/stopwords`. Report accuracy for Naive Bayes for this filtered set. Does the accuracy improve? Explain why the accuracy improves or why it does not?

2 Logistic Regression

40 points Implement the MCAP Logistic Regression algorithm with L2 regularization that we discussed in class (see Mitchell's new book chapter). Try five different values of λ (constant that determines the strength of the regularization term). Use your algorithm to learn from the training set and report accuracy on the test set for different values of λ. Implement 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.

10 points Improve your 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 Logistic Regression for this filtered set. Does the accuracy improve? Explain why the accuracy improves or why it does not?

3 Extra Credit

Extra Credit, 20 points: Implement a 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
- README file for compiling and executing your code.

- A detailed write up that contains:
    1. The accuracy on the test set for both algorithms.
    2. Accuracy on the test set for the variations with stop word removal and feature selection.

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.