Homework 2

Machine Learning

Naive Bayes, Logistic Regression and Perceptron for Text Classification

In this homework you will implement and evaluate Naive Bayes, Perceptron and Logistic Regression for text classification.

You can use either C/C++, Java or Python to implement your algorithms. Your C/C++ implementations should compile on Linux gcc/g++ compiler. If we are unable to compile your code on the university linux machines, you will get a zero.

0 Points Download the spam/ham (ham is not spam) datasets available on the class web page. The datasets were used in the Metsis et al. paper [1]. There are three datasets. You have to perform the experiments described below on all three datasets. Each data set is divided into two (sub)sets: training set and test set. Each of them 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. Divide the given training set into two sets using a 70/30 split (namely the first split has 70% of the examples and the second split has the remaining 30%). Learn parameters using the 70% split, treat the 30% data as validation data and use it to select a value for λ. Then, use the chosen value of λ to learn parameters from the full training set and report accuracy on the test set. Use gradient ascent for learning the weights (you have to set the learning rate appropriately. Otherwise, your algorithm may diverge or take a long time to converge). Do not run gradient ascent until convergence; you should put a suitable hard limit on the number of iterations.

25 points Implement the perceptron algorithm (use the perceptron training rule and not the gradient descent rule). Notice that unlike logistic regression which is a batch algorithm, the perceptron algorithm is an incremental or stochastic algorithm. Treat number of iterations in the perceptron algorithm as a hyper-parameter and use the 70-30 split method described earlier to choose a suitable value for this hyper-parameter. Then, use the chosen value of hyper-parameter, train on the full training dataset and report accuracy on the test set.

Extra Credit, 15 points: Use the automatic parameter tying method described here (`http://www.hlt.utdallas.edu/~vgogate/papers/aaai18.pdf`) on the three algorithms. Again, treat k (the number of unique parameters) as a hyper-parameter and select it using the 70-30 split. Does parameter tying improve accuracy of Naive Bayes and/or Logistic Regression.

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.