Assignment 1: Naive Bayes Classification

Francesca Nannizzi

1.28.14

1 Development, Parts I and II

1.1 Outside Resources

Having not used Python previously, I heavily relied on the Python 2.7.6 documentation to write my code. I also needed help writing bash scripts to format and analyze the training and test sets, so I looked up various things like how to read a file line by line, or how to count the unique words in a file on stackoverflow.com. I can provide the scripts I used if there is any concern.

2 Analysis, Part III

2.1 Scores, Part I

2.1.1 SPAM scores

precision = 358/377 = 0.9496 recall = 358/363 = 0.9862 F-score = (2* 0.9496 * 0.9862)/ (0.9496 + 0.9862) = 0.9676

2.1.2 HAM scores

precision = 981/986 = 0.9949 recall = 981/1000 = 0.981 F-score = (2 * 0.9949 * 0.981)/(0.9949 + 0.981) = 0.9879

2.1.3 NEG scores

precision = 1096/1346 = 0.8143 recall = 1096/1254 = 0.8740 F-score = (2 * 0.8143 * 0.8740)/(0.8143 + 0.8740) = 0.8431

2.1.4 POS scores

precision = 1022/1180 = 0.8661 recall = 1022/1272 = 0.8035 F-score = (2 * 0.8661 * 0.8035)/(0.8661 + 0.8035) = 0.8336

2.2 Scores, Part II

2.2.1 SPAM scores

precision = 334/363 = 0.9201 recall = 334/363 = 0.9201 F-score = (2 * 0.9201 * 0.9201)/(0.9201 + 0.9201) = 0.9201

2.2.2 HAM scores

precision = 971/1000 = 0.971 recall = 971/1000 = 0.971 F-score = (2 * 0.971 * 0.971)/(0.971 + 0.971) = 0.971

2.2.3 NEG scores

precision = 955/1250 = 0.764 recall = 955/1254 = 0.7616 F-score = (2 * 0.764 * 0.7616)/(0.764 + 0.7616) = 0.7628

2.2.4 POS scores

precision = 977/1276 = 0.7657 recall = 977/1272 = 0.7681 F-score = (2 * 0.7657 * 0.7681)/(0.7657 + 0.7681) = 0.7669

2.3 Response

The scores decrease when only 10% of the training data is used for training the classifiers. This is because the models developed by the classifiers are significantly more incomplete, and many words appear that are outside the vocabulary when trying to predict the class of new documents.