Spam Detection Methods Using Naïve Bayes Filtering

Asher Langton, langton@cs.wisc.edu
Bryan Berns, berns@cae.wisc.edu
Jiong Fan, jfan@wisc.edu

Abstract –

The Naïve-Bayes classifier has been shown to be very reliable in spam detection. In this paper, we describe the implementation of a Naïve-Bayes classifier with a heuristic feature selection and evaluate its efficiency on a variety of test cases, including purposely over-fit datasets.

I. Theory

Bayes Networks & Classifiers.
A Bayesian network is a graph that represents of relational probability distribution. An edge between 'nodes' in the Bayes network indicates a probabilistic, conditional influence between nodes as shown below.

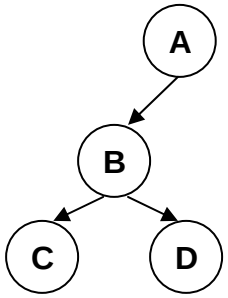

Figure #0: General Bayesian Networks

In general, a Bayesian classifier is the deterministic value of a single node of the Bayesian network given knowledge in the elsewhere in the network. As where this model does very well in many applications, the computational and space complexity of a fully constructed network is not practical. For this reason, we will make one 'naïve' assumption: no specific feature in the network depends on another.

Naïve Bayes Classifier & Bayes' Rule.
The naïve Bayes classifier is a simplification of a general Bayesian network which assumes there are no relationships between features in a tree. Visually depicted below, this is a two-layer network, where the single node at the top is the classifier.

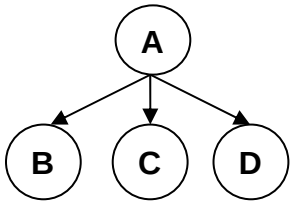

Figure #1: Naïve Bayesian Networks

Using this model, both training and testing is of linear space and computational complexity, allowing for many features. Training this network by computing the probability between the classifier and a given feature is done via Bayes' Rule given as:

$$P(C \mid X) = \frac{P(X \mid C)P(C)}{P(X)} \qquad \text{(Equation \#0)}$$

Once this probability is computed, testing is trivial under the Naïve Bayes assumption. Since all edges in the network are independent of one another, the probability is simply:

$$P(X \mid C) = \prod_i P(X_i \mid C) \qquad \text{(Equation \#1)}$$

Email Features & Classification.
A human being has no trouble deciding an email's legitimacy within a few milliseconds of its opening. The obvious traits are single words or word phrases such as 'free' or 'high credit'. Punctuation also can play a role, for example the existence of a question or exclamation point in an email. For simplicity, we will use single word phrases greater than 3 letters. In terms of the Naïve Bayes model we therefore will create a network as shown below:

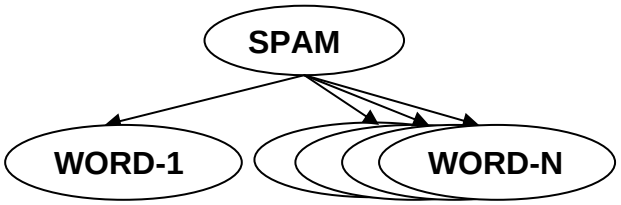

Figure #2: Naïve Bayesian Classifier for Spam Detection

Other locales of text in a message, aside from the body, may also have some relevance to classifying the document. For this reason, we will include both words found in the subject area and those found in the body area. With the variety and inconsistency of other information found in the message header, ignoring all other entries seems to be a valid assumption. Detecting whether the reported sender's domain corresponds to the messages actual origin may also be a desirable feature, but such an implementation is beyond the scope of this study.

II. Implementation & Testing

Program Construction.
A key factor in the efficiency of a Naïve-Bayes classifier is feature selection. In our implementation, we compile two ordered lists from the training set: the words that appear in spam, and the words that appear in legitimate email. Both lists are sorted by the frequency with which the words appear in the training set, without distinguishing upper from lower case. We then take as our feature set those words which appear among the top-*k* most frequent words on *exactly one* of the lists. That is, we want words that are frequently in one type of email or the other, but we have little use for words that are common in both types of email. In practice, the spam contributed words like "net", "market", "remove", "ordering", and "links" to the feature set, while the legitimate mail added such words as "forms", "society", and "linguist".

We then count the number of spam and legitimate messages in which these words occur, and compute the probabilities of a feature being present in a given email, based on the email's classification as either spam or non-spam.

To classify a message, we simply note the presence or absence of each word in the feature list, and then determine, using Bayes's Rule and the Naïve-Bayes assumption, whether the message is more likely spam or a legitimate email.

Testing.
Two groups of spam and legitimate mail were used in the testing phase. One group was composed of mainly linguistics and computer science data, which was used as the test set for spam research in a similar study. We will refer to this set as 'Linguistic'. Assorted with the legitimate mail was spam. The entire set, which has 4 subsets, contains 1689 non-spam and 337 spam messages total.

The other was composed of spam and legitimate mail gathered at the Computer Aided Engineering building at the University of Wisconsin – Madison over the last few years. The specific names of users who donated the mail were taken out to avoid over fitting the data. This data also contained a larger variety of mail topics ranging from sports, sailing, driving, cars, and media. We will reference this test set as 'General'. This test set contains 8228 non-spam and 4626 spam messages.

In testing, when a group is tested against itself, the groups are actually randomly chosen subsets of the main group. Training and then testing on the exact same data would not yield a reliable metric for gathering informative results.

Test #0: Control Test

Using no training data, a test set of 50% spam, 50% non-spam was tested. As expected there was 50% error as all spam was evaluated as non-spam.

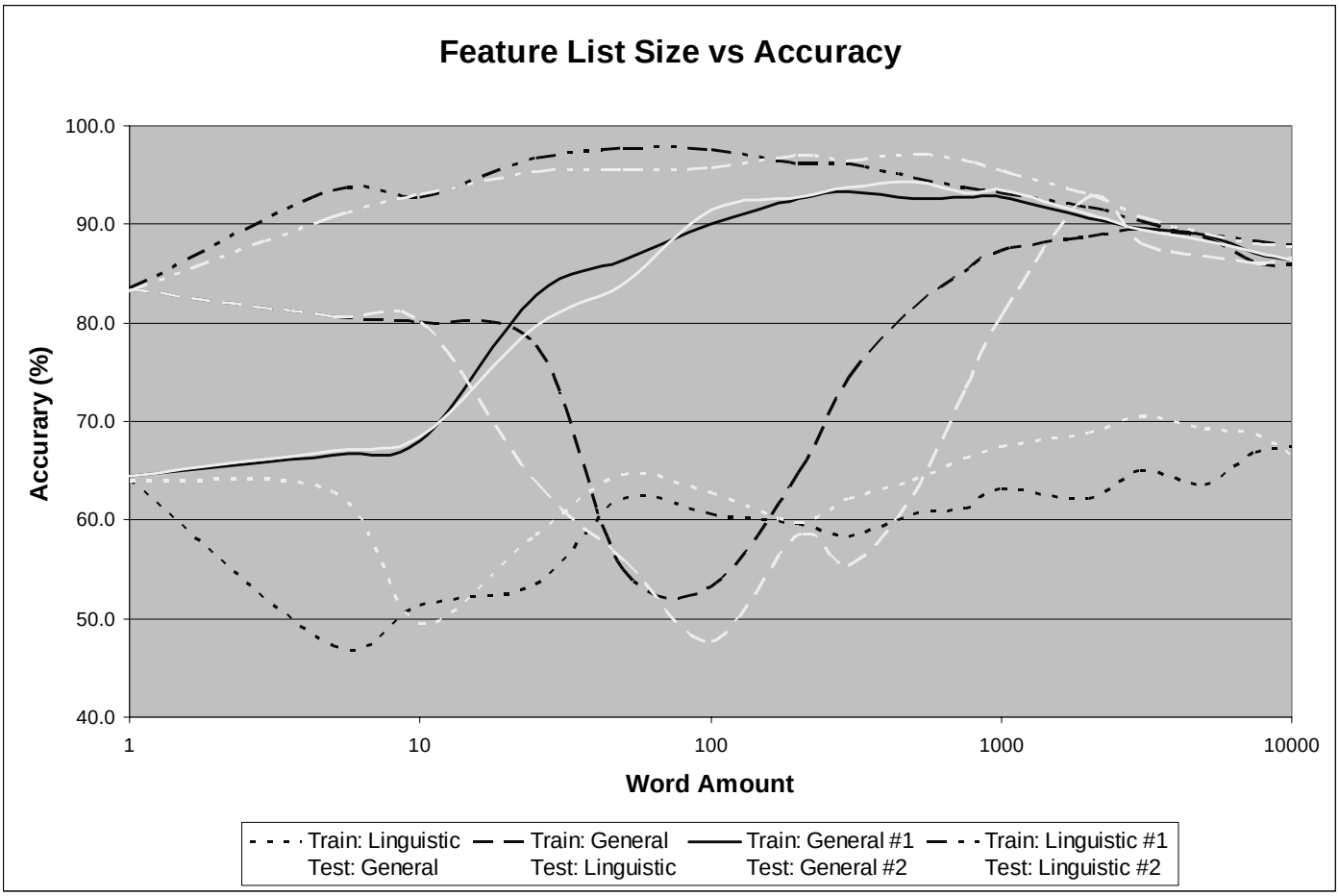

Figure #3: Test results data for Test #1 (darker lines) and Test #3 (lighter lines).

Test #1: Top-k Words versus Accuracy:

In this test, Naïve Bayes model is trained and pruned to the intersection of the most frequent k-number of features, which are whole words or character strings in our test.

The results of 64 runs of this test are shown in Figure #3 (darker line sets) as logarithmic word amount versus accuracy. From these results, we can easily see the affect of over fit data from the linguistic section. However, when the linguistic set was used at training data to test the 'General' mail category, it failed miserably. Combining these results, we can conclude that Linguistic set did very well at telling what was legitimate mail by use of over-fitting. In fact, over-fitting worked so well that only a few words where needed to separate the non-spam from the spam. Let us also note the relatively good performance of the General category on the Linguistic at higher numbers.

Test #2: Lemmatization and Stop-Word Test:

From a previous study, the 'Linguistic' data was duplicated and modified to take out certain 'stop' words and change words into their base form (lemmatized). We ran these data sets using our program using the top 300 single-word features. The results are shown below.

Feature Set	**Accuracy**
Control Run	93.3
Stop Words Removed	97.1
Lemmatized	97.8
No Stop Words & Lemmatized	97.5

Table #0: Test results data for Test #3 lines).

Test# 3: Top-k Words versus Accuracy Using One and Two Word Features.

Variant of Test #1 but instead using features with one or two word feature sets. As expected the values are very similar but do marginally better than the one word feature dataset. Results are shown in Figure #3 as the lighter lines, respective to Test #1.

II. Conclusions & Further Implementation.

In the general, the Naïve Bayes classifier did well in classifying spam and non-spam. If we examine the high-accuracy phenomena that occurs between the same-type training and testing data in Test #1 (i.e. 'linguistic' versus 'linguistic'), it is evident that the even a small dataset is seemingly good at sorting spam and non-spam. For this reason, a dynamically trained SPAM filer may be a very applicable concept. Since a users' mail is most likely be contain similar words, it is reliable to filter mail based more on that criteria.

As far as general performance is concerned, the accuracy may be improved using a different formula for ranking data for the 'top-k' features. In our implementation, the one used ran a slight risk of eliminating features that may have a beneficial contribution. A more formal ranking formula, such as Info-Gain, may perform better in choosing features that give the most information.

Regardless of test methods, sometimes there is a thin line between spam and non-spam. We may sign up for a newsletter from website 'A' about special promotions we are actually interested in. Website 'A' may sell its list to Website 'B' offering the same promotions. Although some may consider these both spam mail, others may want to receive both or just one. Certain spam is so relative to a person's interests, a certain level of 'active learning' might be needed for future spam detectors, and, unfortunately Naïve-Bayes might not cut it.

References.

Androutsopoulos, J. Koutsias, K. V. Chandrinos, G. Paliouras, and C. D. Spyropoulos. An Evaluation of Naive Bayesian Anti-Spam Filtering. In Proc. of the workshop on Machine Learning in the NewInformation Age, 2000.

Jefferson Provost. Naive-Bayes vs. Rule-Learning in Classification of Email.

S. Russell & P. Norvig. Artificial Intelligence: A Modern Approach.Prentice Hall, second edition, 2002.

Mehran Sahami, Susan Dumais, David Heckerman, and Eric Horvitz. A bayesian approach to filtering junk e-mail. In AAAI-98 Workshop on Learning for Text Categorization, 1998.