Adaptive Spam Filtering Using Only Naive Bayes Text Classifiers

Aris Kosmopoulos
Institute of Informatics and Telecommunications,
N.C.S.R "Demokritos"
Athens, Greece
and
Department of Informatics,
Athens University of Economics and Business,
Athens, Greece
akosmo@iit.demokritos.gr

Georgios Paliouras
Institute of Informatics and Telecommunications,
N.C.S.R "Demokritos"
Athens, Greece
paliourg@iit.demokritos.gr

Ion Androutsopoulos
Department of Informatics,
Athens University of Economics and Business,
Athens, Greece
and
Digital Curation Unit,
Research Centre "Athena",
Athens, Greece
http://www.aueb.gr/users/ion/

ABSTRACT

In the past few years, machine learning and in particular simple Naive Bayes classifiers have proven their value in filtering spam emails. We hereby put Naive Bayes filters to the test, against potentially more elaborate spam filters that will participate in the CEAS 2008 challenge. For this purpose, we use the variants of Naive Bayes that have proven more effective in our earlier studies. Furthermore, we propose a simple active learning method for adapting the filter, under partial online supervision.

1. INTRODUCTION

A variety of approaches to spam filtering have been used in the past. Machine learning classification algorithms have proven to perform very well in this task. Naive Bayes (NB) classifiers in particular have been found to perform reasonably well, despite their simplicity. The simplicity of NB classifiers and their low computational requirements have made them particularly appealling for commercial spam filters. However, commercial filters rely also on a variety of other indicators, in order to improve their spam detection performance. Our purpose here is to put simple Naive Bayes text classifiers to a realistic test against more elaborate filters that will potentially participate in the CEAS 2008 challenge. Thereby, we expect to gain an indication of the potential contribution of NB text classifiers to spam filtering.

In our earlier published work [3] and more recent unpublished experiments, we compared variants of NB on the task of spam filtering, using also a variety of datasets. In those experiments, we assessed not only the performance of the classifiers, but also their computational efficiency, which is important for real-time adaptive filtering. Based on the results of those studies, we have chosen to test two variants of NB in the CEAS 2008 challenge: multinomial NB with transformed term frequency attributes (TF-NB), and multinomial NB with Boolean attributes [3]. Our first choice (TF-NB), which achieved the best performance in most of our previous experiments, uses a transformation of attributes that is based on the paper of Rennie et al. [4] and is further explained in section 3. Our second choice (BOOL-NB) has proven to perform quite well, despite its simplicity, both in our experiments and in related work [2, 6].

In the following three sections, we discuss in turn the feature selection techniques that our filter uses, the two NB variants it employs, and the simple active learning method that we have developed. The paper does not include experimental results, as it was written before the challenge. In the final section, we summarize our approach and indicate a potential path for future development.

CEAS 2008 - Fifth Conference on Email and Anti-Spam, August 21-22, 2008, Mountain View, California USA

2. FEATURE SELECTION

Our filter uses only the textual part of emails. Therefore, we extract features only from the subject and the body of each message, with features corresponding to individual tokens. We also remove HTML tags, and we do not use any stemming or stop-word removal.

As a first feature selection step, we ignore any tokens that appear in less than 5 different training messages. Information Gain scores are then computed for the remaining tokens (in their Boolean form) as in previous work [5, 1], and the 3000 of them with the highest scores are used in the feature vectors of the messages. The choice of the size of the feature vectors (3,000) is based on the results of our previous experiments.

3. NAIVE BAYES VARIANTS USED

Each message j is represented as a vector $\langle x_{1j}, \ldots, x_{mj} \rangle$, where x_{ij} is the value of attribute X_i in message j, and m is the number of tokens we use as attributes (3,000). In BOOL-NB, $x_{ij} = 1$ if the token that corresponds to attribute X_i occurs in the message; otherwise $x_{ij} = 0$. For TF-NB, the value of x_{ij} is initially equal to the frequency (number of occurrences) of the corresponding token in message j. Thereafter, it is transformed following the next three steps:

$$x_{ij} \leftarrow log(x_{ij} + 1) \ \text{(TF transform)}, \tag{1}$$

$$x_{ij} \leftarrow x_{ij} \cdot log(\frac{\sum_k 1}{\sum_k \delta_{ik}}) \ \text{(IDF transform)}, \tag{2}$$

$$x_{ij} \leftarrow \frac{x_{ij}}{\sqrt{\sum_l (d_{lj})^2}} \text{ (length normalization)}, \quad (3)$$

where k ranges over the training messages, δ_{ik} is 1 if the token that corresponds to attribute X_i occurs in message k and 0 otherwise, and d_{lj} is the initial (before the first transformation) value of x_{lj}. These steps are fully explained by Rennie et al. [4]. It is worth mentioning that we do not use the weight normalization that Rennie et al. propose, as it led to worse results in our previous experiments.

From Bayes's theorem, the probability of message j with vector $\vec{x} = \langle x_{1j}, \ldots, x_{mj} \rangle$ to belong in category c is:

$$P(c|\vec{x}) = \frac{P(c) \cdot P(\vec{x}|c)}{P(\vec{x})} \quad (4)$$

In order to classify message j in the ham (c_h) or spam (c_s) category, we use the following formula:

$$score_j = \frac{P(c_s) \cdot P(\vec{x}|c_s)}{P(c_s) \cdot P(\vec{x}|c_s) + P(c_h) \cdot P(\vec{x}|c_h)}, \quad (5)$$

which indicates how sure our filter is that message j is spam. For values close to 1, we are very confident that it is spam; for values near 0, we are very confident it is ham. Therefore, we can introduce a threshold on $score_j$ in order to make the final decision about the message's category. The value of the threshold controls the tradeoff between false positives and false negatives. In the CEAS challenge, we set the threshold to 0.5, i.e., we clasify message j as spam if $score_j > 0.5$. Our filter, however, also returns $score_j$, making it easy to experiment with other threshold values.

The probability $P(c)$ is estimated by dividing the number of training messages of category c by the the total number of training messages. The probability $P(\vec{x}|c)$ is estimated as $\prod_{i=1}^{m} P(t_i|c)^{x_{ij}}$, where x_{ij} is computed as described above for each form of NB, and t_i is the token that corresponds to attribute X_i. For BOOL-NB, $P(t|c)$ is estimated as:

$$p(t|c) = \frac{1 + M_{t,c}}{2 + M_c}, \quad (6)$$

where $M_{t,c}$ is the number of training messages in category c that contain token t, and M_c is the total number of training messages of category c. For TF-NB, x_{ij} is estimated as:

$$p(t|c) = \frac{1 + N_{t,c}}{m + N_c}, \quad (7)$$

where $N_{t,c}$ is the number of occurrences of token t in the training messages of category c, and $N_c = \sum_{i=1}^{m} N_{t_i,c}$.

The two filters that we submitted to the competition were already trained on 500 messages that we picked randomly from the SpamAssassin corpus; the latter is provided with the TREC 2006 Spam Evaluation Kit. We maintained in the 500 messages the spam to ham ratio of the SpamAssassin corpus. We also created a different sample (dubbed Active Learning Set) of 500 messages from the SpamAssassin corpus, with the same spam to ham ratio, in order to use it in the active learning procedure that is described below.

4. ACTIVE LEARNING

During the active learning task of the contest, category labels for the incoming messages are only available to the filter upon request. Each filter is allowed a fixed number Tr of requests per run; it is also given the total number Cl of incoming messages it will have to classify during the run. For each incoming message, the filter has to decide if it will request the message's true category to be revealed, so that the message can be used for training, or not. We define a ratio K equal to:

$$K = \frac{Tr}{Cl}. \quad (8)$$

This is the ratio of incoming messages that we want to be used for training.

The main idea behind our method is that the probability of an incoming message being useful for training is proportional to the uncertainty of our filter about the message's category. It should be stressed here that all incoming messages could potentially be selected by our active learning method as training examples (i.e., have their true categories revealed), some with larger probability than others. This additional randomness addresses to some extent potential mistakes when assessing the usefulness of the incoming messages as training examples.

We assume that the probability of a message j to be useful for training follows a normal distribution over $score_j$ with mean $\mu = 0.5$ (where the classifier's uncertainty is highest) and standard deviation σ.[1] In other words, scores near $\mu = 0.5$ are more likely to correspond to useful training messages. We use K to set the standard deviation of the distribution. Initially, we use a sample of TM (500) messages from the Active Learning Set (section 3), in order to select the value of σ so that T of the TM messages have scores within one standard deviation from $\mu = 0.5$, where T is estimated as:

$$T = round(TM \cdot K). \quad (9)$$

This way, the interval from $\mu - \sigma$ to $\mu + \sigma$ contains the scores of the sample's messages that we would have wanted to have been selected for training, i.e., the T messages with scores closest to $\mu = 0.5$. Note that setting σ so that the scores of the T messages fall within $\mu - 3\cdot\sigma$ to $\mu + 3\cdot\sigma$ would guarantee that the probabilistic selection that we use (described below) would almost always pick messages whose scores fall within the interval that contained the scores of the T messages, provided that the messages of the contest follow the normal distribution we assume. Instead, by setting σ so that the scores of the T messages fall from $\mu - \sigma$ to $\mu + \sigma$, we allow the probabilistic selection to use (with lower probability) as training examples messages whose scores are outside the interval that contained the T messages of the sample.

Provided that the total number of selected training examples has not already reached Tr, we select an incoming message with $score_j$ as a training example with the following probability:

$$P(train|score_j) = \begin{cases} cdf(score_j; \mu, \sigma), \text{ if } score_j \leq 0.5 \\ cdf(1 - score_j; \mu, \sigma), \text{ otherwise} \end{cases} \quad (10)$$

where $cdf(x; \mu, \sigma)$ is the cumulative distribution function of the normal distribution, $\mu = 0.5$, and σ is estimated as above. Formula 10 assigns the same selection probability to messages whose $score_j$ is at the same distance from $\mu = 0.5$, regardless of whether $score_j$ is smaller or larger than 0.5; furthermore, the probability increases as $score_j$ approaches

[1] Note that $score_j \in [0, 1]$, whereas the normal distribution that we assume assigns non-zero probability to values of $score_j$ in $(-\infty, 0)$ and $(1, +\infty)$. For simplicity, we overlook this mismatch.

μ, i.e., as the classifier's uncertainty increases. Since formula 10 leads to probability values in $(0, 0.5)$, we normalize the resulting values by multiplying them by 2. Again it is worth noting that this probabilistic choice allows even messages with low uncertainty to be selected for training.

Because the initial sample may differ significantly from the messages of the contest, we modify the distribution whenever an incoming message arrives during the contest. Specifically, the new message is added to the existing TM ones, TM is incremented, and T, σ are reestimated as above.

5. CONCLUSION AND FUTURE WORK

In this short paper we described briefly two email spam filters that employ different forms of the Naive Bayes classifier and focus on the text of the messages. The main criteria for the choice of the two Naive Bayes forms were their good performance in a series of experiments with different data sets and their computational efficiency. The ultimate goal of this effort is to measure the value added by non-textual features and more elaborate classifiers, by comparing our simple text classifiers with other participants in the contest.

Our immediate plans are to study the results of our filters and their competitors and draw potentially interesting conclusions about the spam filtering process. Additionally, we are working on providing our overall filter as a plug-in for a well known free email client, in order to allow measuring the filter's effectiveness under real circumstances, by real users.

6. REFERENCES

[1] I. Androutsopoulos, J. Koutsias, K. Chandrinos, and C. Spyropoulos. An experimental comparison of Naive Bayesian and keyword-based anti-spam filtering with encrypted personal e-mail messages. In *23rd ACM SIGIR Conference*, pages 160–167, Athens, Greece, 2000.

[2] A. McCallum and K. Nigam. A comparison of event models for Naive Bayes text classification. In *AAAI Workshop on Learning for Text Categorization*, pages 41–48, Madison, Wisconsin, 1998.

[3] V. Metsis, I. Androutsopoulos, and G. Paliouras. Spam Filtering with Naive Bayes – Which Naive Bayes? In *Proceedings of 3rd Conference on E-mail and Anti-Spam (CEAS 2006)*, Mountain View, CA, USA, 2006.

[4] J. D. M. Rennie, L. Shih, and D. R. Karger. Tackling the Poor Assumptions of Naive Bayes text Classifiers. In *20th International Conference on Machine Learning*, Washington DC, 2003.

[5] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz. A Bayesian approach to filtering junk e-mail. In *Learning for Text Categorization – Papers from the AAAI Workshop*, pages 550–62, Madison, Wisconsin, 1998.

[6] K.-M. Schneider. A comparison of event models for Naive Bayes anti-spam e-mail filtering. In *10th Conference of the European Chapter of the ACL*, pages 307–314, Budapest, Hungary, 2003.