Using time to classify spam

Vadim Jefte Zendejas Samano and Matsuura Kanta
Matsuura Laboratory, The University of Tokyo

Abstract

In this article, the introduction of a new spam filter based on the time information of emails is proposed. The proposed filter has two main advantages: it is fully independent of the language of the email and is one of the most efficient filters that exist today; together these provide us with a good tool for implementation in low-power devices and a complementary tool for more complex filters.

1 Introduction

In recent years, the use of different and increasingly more sophisticated spam filtering techniques is a common issue as the technology in spam filtering techniques increases. In fact, many new complex spam filters are in development [16, 3, 7], which have resulted in some efficiency problems. Only a few exceptions of these new approaches are actually used in a real-world environment. This is an important issue to consider, because the tradeoff between the efficiency and the accuracy of the filters is a critical matter for the end-user [12].

Also, in recent years, the possibility of a user that needs to receive multiple emails in different languages is increasing. For this reason, the use of a filter that is fully independent of the language will be necessary in the increasingly globalized world [1].

In this paper, we would like to introduce an efficient and a completely language independent way of classification of spam emails based on the time-stamp in the header of the email.

2 Using Time on Spam Filtering

Concerning the time-based spam filtering, the proposal shows a truly efficient filter. It is based on a single-line read of the entire email body and header, as opposed to the most common filters, such as the context-based types, that must read all the lines of the email. All these in combination with the use of simple probabilistic functions make this filter computational efficient. Also as mentioned, the full independence of any language gives to us an advantage in comparison with other commonly used filters.

The basic idea stems from the observation that many spammers send emails on a regular basis [14].

3 Proposed Implementation

The filter as mentioned before, is based on the time-stamp printed on the email header. Consider an email message, then the line that needs to be read is expressed as follows:

```
  Received:  ...; Wed, 7 Mar 2007
14:05:10 -6000
```

This date-time is printed on the email by the destination email server. It is the last received information of the last hop on the way from the sender to the destination. As we can see it is not the date that the email was sent. So if we consider the email server where the email account resides as a trusted-server, then we can conclude that the information is correct and no one has tampered with it. This is the main assumption in the correctness of the time-stamp analyzed during the filter implementation. Also is assumed that the time stamp will be standardized to the GMT standard time.

3.1 The Evaluation Time Parameters

The choice of the evaluation time parameters is based on the assumptions that the email account that will be analyzed is more than 1 month old.

We obtain two basic and important parameters for the filter; these two parameters are the *day* and *hour* of arriving. We have now the evaluation time parameters necessary to classify and train the filter.

4 Filter Training

The filter training is based on the probability that the email is spam or ham if it arrives on a specific day and hour. As we can see this assumption is also considered a conditional probability of multiple events.

The choice of this method is based on the training process of Naives-Bayesian filtering [8]. In which the authors mention that the Bayes Theorem is a good approach for a learning process of email classification. This training process is based on the Bayes theorem of conditional probability [9]:

$$\Pr(A|B) = \frac{\Pr(B|A)\Pr(A)}{\Pr(B)} \tag{1}$$

4.1 Time and Probability

The use of the Bayesian theorem of conditional probability as the main technique for the filter training, gives us the opportunity to identify different conditional probabilities based on the evaluation parameters mentioned on Sect. 3.1.

To define the corresponding probabilities, first of all we need to define the events of each probability. In this case we will use A as the event of an email being spam or ham. Also we will define B as the event of an email that arrives on a specific day. And, C will be the event that the email arrives on a specific hour. The events B and C are considered to be independent from each other. The probabilities are:

- The conditional probability of mail being spam or ham if it arrives on a specific day. This is represented as $\Pr(A|B)$.
- The conditional probability of mail being spam or ham if it arrives on a specific hour. Represented as $\Pr(A|C)$.
- The conditional probability of mail being spam or ham if it arrives on a specific day and specific hour. This conditional probability of multiple events is based on the Arc Reversal Bayes rule [4]. It is represented as $\Pr(A|B|C) = \Pr(A|C)\Pr(A|B)\Pr(A))$.

4.2 Training Process

The training functionality of the time spam filter is, as mentioned before, based on the conditional probabilities:

The training process has three main inputs. It receives the email file, which must be in a plain-text format with each email message in one file. It also receives the standard judgment from the evaluation system. This standard judgment gives us the correct information if the email is really spam or ham. The third input to the training process will be the previous trained data, where the training process is saved for future email classifications.

After receiving the inputs, the training process will read from the email the evaluation parameters that are the standardized day and hour. With this information, we can update the number of emails that we received on the specific day and hour with the spam or ham classification on the training matrix. After updating the number of emails then we need to recalculate all the probabilities for each day and hour on the training matrix.

5 Mail Classification

The classification process is based on the probabilities calculated during the training process. After having these probabilities, it is necessary to obtain a correct system of classification so the accuracy of the filter can be increased.

First of all, we need to obtain the probabilities of two different events. The first one is the probability of an email being spam if it arrives on a specific day and hour, and is represented by $\Pr(A = \text{spam}|B|C)$. The second probability is the probability of it being ham if it arrives on a specific day and hour; this is represented by $\Pr(A = \text{ham}|B|C)$. To obtain the final classification of the filter we just compare if the $\Pr(A = \text{spam}|B|C)$ is bigger than $\Pr(A = \text{ham}|B|C)$, then we classify the email as spam. In the other case we classify the email as ham.

As the process is based on the law of conditional probabilities, these means that the final probabilities used for the classification are based on all the events that occurred in the scenario, in this case by $\Pr(A|B)$ and by $\Pr(A|C)$. This means that if the spammer sent the email in a random distribution of time, the classification still can be done based on the other events of the email being ham in a specific day and hour. This assumption is based in Sect. 7.1.

During the classification process we need to obtain the spamminess score of the classification. This score indicates the certainty that the classification is correct or incorrect. The classification score will be a number between 0 and 1. If the number is closer to 1, then the filter is relatively certain that the classification has been made correctly.

The process of obtaining the score of the classification is as follows. As we can see, it is obvious that the probabilities compared during the classification process are a good indicator of how sure we are of the classification. For this reason, the score of the classification will be the same as the probability of the classified event. For example, if the $\Pr(A = \text{spam}|B|C)$ is bigger than the $\Pr(A = \text{ham}|B|C)$, then, the score will be the $\Pr(A = \text{spam}|B|C)$. In the other case, the score will be $\Pr(A = \text{ham}|B|C)$.

6 Evaluating the Filter

One of the most important things during the spam filter creation is the evaluation of the filter. With a correct evaluation and correct interpretation of the results we can then compare the filter with other existing ones, and decide if the filter is efficient, accurate and useful for implementation. This scientific evaluation is an essential component of the research.

For the evaluation process of the time-based spam filter, we will use the TREC Spam Filter Evaluation Framework [6]. As mentioned by Goodman et al. (2007) [7], the TREC Spam Filter Evaluation Framework is one of the most trusted and accurate standard evaluation frameworks based on real email streams [5, 13]. The most significant thing is that it defines a standard measure for evaluation, and generates relatively realistic results, so comparisons between different filters can be done easily and reliably.

6.1 Experiment Design and Evaluation Measures

The experiment design is one of the most important aspects to consider in the evaluation process. For this case we define our experiment design based in the Cormack & Lynam (2005, 2007) [5, 6] experiment design proposal. This experiment design is one of the most commonly used and has proved to have a realistic evaluation results. The experiment design is as follows. We chose different scenarios of evaluation considering 100% of the evaluating dataset:

Full. We evaluate the full data set of emails as it is provided.

Ham25. We use a sample of 25% of ham from 100% of the ham emails.

Ham50. We use a sample of 50% of ham from 100% of the ham emails.

Spam25. We use a sample of 25% of spam from 100% of the spam emails.

Spam50. We use a sample of 50% of spam from 100% of the spam emails.

The main reason to use different rates between ham and spam emails is that it helps to prove that no discrimination characteristics exist in the evaluation process. In the case of our filter these will be time discrimination characteristics related with the use of the time characteristic for spam classification.

6.1.1 Estimating False Positives and False Negatives

In the process of evaluating the filter, the evaluation measures that the framework defines [6] are based on the possible outcomes of applying a filter to an email stream[1]. We represent by the letter c the number of False Positives (FP) and the with the letter b the number of False Negatives (FN). a represents the correct classified ham email and letter d the correct classified spam emails. The evaluation measures are represented as follows:

Ham misclassification rate (*hmr*). Is the fraction of ham messages that are misclassified as spam. It is also known as the false positive rate (*FPR*). It is expressed as $hmr = \frac{c}{a+c}$.

Spam misclassification rate (*smr*). Is the fraction of spam messages that are misclassified as ham. It is also known as the false negative rate (*FNR*). It is expressed as $smr = \frac{b}{b+d}$.

Error and Accuracy rate. Represents the overall misclassification fraction and the overall accuracy fraction. It is calculated as follows: $m = 1 - accuracy = \frac{b+c}{a+b+c+d}$.

Ham/spam learning curves. Error rates as a function of number of messages processed.

ROC ham/spam tradeoff score. This is the ROC area above the curve. The ROC area is the probability that the spamminess score of random ham message equals or exceeds the spamminess score of a random spam message.

Time. This measure represents the amount of time in hours that the evaluation takes. The time includes all the process of classification and training. These measures give us a good comparison measure for the efficiency of different filters.

6.2 Evaluation Process

The evaluation process as defined on the Framework for evaluating spam filters can be defined as follows. It consists of two main processes, the training/classification process and the evaluation process. The training/classification consists of a cycle of processes that will classify and evaluate each email in the corpora. The first component is the initialization module that will generate, compile and create the database necessary for the filter to work.

[1] The standard results as mentioned by Cormack & Lynam (2007) [6], are the real classification results of the email corpora. The classification accuracy of the standard results is 100% with a *hmr* and a *smr* of 0%.

The second step is the classification module as described in Sect. 5. The results of the classification will be the inputs together with the standard judgment to the training module as described in Sect. 4.2. These two modules will be in a cycle for each mail in the corpora.

With the result file of the classification, the framework can evaluate the filter behavior. The evaluation process consists of analyzing the data, obtaining the evaluation measures and generating the graphs of the evaluation process.

6.3 Evaluation Results

The dataset that is used in the base evaluation (testing environment) of the time-based spam filter is the one defined by Cormack & Lynam (2005, 2007) [6, 5]. It is based on the Spamassassin [13] email corpus and other private and public corpus. It is composed of a data set of 92,189 real emails. It is called synthetic corpora, consisting of a rare public corpus of good email, combined with a modified set of recent spam. The full dataset contains 42.74% of ham (39,399 emails) and 57.26% of spam (52,790 emails).

The interpretation of the results on Table 1 gives us an idea on how accurate the time-based filter is. As we can see, the average accuracy of the filter is 90.03% and has an average error rate of 9.97%. This means that for each hundred emails, we will have 90 mails classified correctly and 10 misclassified. With analysis of the multiple results in the different rates of spam and ham chosen in the experiment design, we can see that the classification is similar in different environments; this means that the filter can behave similarly in different conditions.

The time-based spam filter, in average, behaves similarly in the real environment to the testing environment. Thus, it proves that our proposal can be used in a real application and the results obtained will be similar to the ones obtained in the experimental phase.

6.4 Comparison Between Other Filters and Time-based Filter

One of the important things during the evaluation of a filter is the comparison with existing filters. This comparison can give us an exact and more accurate point of view in the filter accuracy.

One of the advantages of choosing the TREC Spam Filter Evaluation Framework is that it offers a realistic way of comparison between different filters.

In this comparison scenario, comparing the time-based spam filter results with filters that behave similar is important. The spam filters chosen are good candidates for comparison with the time-based spam filter because they are accurate, efficient, are commonly used in real applications and some are even language independent. The filters that we are evaluating are proposed by Cormack & Lynam (2007) [6] as good filters for comparison. For a content-based filter we mainly chose Bayesian classifiers but also used other probabilistic classifiers such as Markovian discrimination. The selected filters are: CRM114, Dspam, Bogofilter, Dbacl, Spambayes, Spamprobe, and Popfile. For the language independence characteristic we choose Social Network Analysis (SNA) and Spamassassin. The SNA was not evaluated by us, the results mentioned here where obtained by Boykin & Roychowdbury (2005) [2].

We can see in Table 2 the average classification results generated by the evaluation tool. We notice that the results of the time-based spam filter are similar in accuracy as with the other filters. But the main advantage of time-based spam filter over the others is the efficiency of the filter. This in combination with the language-independence makes the time-based spam filter an excellent candidate for use in low-power processing devices. [12]

We can also conclude based on Kong et al. (2005) [10] assumptions, that the time-based spam filter is a good technique as an auxiliary tool, in combination with other more sophisticated filters to fight spam.

7 Pros and Cons of Time-based Spam Filtering

One of the most important pros of the time-based spam filter is the efficiency of the filter. As mentioned before in Sect. 3, the efficiency of the filter is based on the assumption that the training/classification process need only read 1 line in the header of the email file and that the training is based on the learning premise of a fixed size learning matrix. Also this efficiency is based on the use of simple probabilistic functions for the learning process. With this information we can say that the efficiency of the time-based spam filter is constant, and has a time complexity of $O(1)$. Among the filters that we have tested, the time-based spam filter is far more efficient in the training/classification process; being in average from 40.32% to 91.48% more efficient then the other filters.

[2] Represents the efficiency of the filter compared with the time-based spam filter.

Table 1: Time-based spam filter classification accuracy results of the dataset for the testing environment.

	Full	**Ham25**	**Ham50**	**Spam25**	**Spam50**	*Average*
hmr (FPR)	1.05% (414/39399)	1.92% (187/9751)	1.82% (356/19586)	1.01% (398/39399)	1.18% (465/39399)	*1.40%*
smr (FNR)	17.29% (9127/52790)	12.18% (6430/52790)	14.57% (7692/52790)	26.75% (3525/13179)	24.12% (6339/26283)	*18.98%*
error	10.35% (9541/92189)	10.58% (6617/62541)	11.12% (8048/72376)	7.46% (3923/52578)	10.36% (6804/65682)	*9.97%*
accuracy	89.65% (82648/92189)	89.42% (55924/62541)	88.88% (64328/72376)	92.54% (48655/52578)	89.64% (58878/65682)	*90.03%*
1-ROCA	0.1245	0.1324	0.1147	0.1569	0.1488	*0.1355*
time (hours)	4:00:44	2:39:48	3:14:44	2:16:30	2:50:03	*3:00:22*

Table 2: Average classification results of spam filters in the testing environment. LI represents the language independent filters and CB represents the content-based filters.

	Filter	**hmr**	**smr**	**error**	**accuracy**	**1-ROCA**	**time**	**efficiency**^{2}
LI	**Time-based**	1.40%	18.98%	9.97%	90.03%	0.1355	3:00:22	100%
	SNA	66%	44%	52.34%	47.66%	-	-	-
CB	**Spamassassin**	0.19%	4.05%	1.94%	98.06%	0.3752	31:41:17	9.49%
	CRM114	1.48%	11.77%	6.36%	93.64%	0.3285	6:40:21	45.05%
	Dspam	2.09%	7.15%	4.30%	95.07%	1.0482	35:16:39	8.52%
	Bogofilter	0.02%	12.97%	6.11%	93.89%	0.0782	5:02:12	59.68%
	Dbacl	0.98%	13.59%	7.11%	92.89%	0.3919	22:45:24	13.21%
	Spambayes	3.65%	12.45%	7.48%	92.52%	0.1687	8:41:42	34.57%
	Spamprobe	1.44%	7.19%	4.15%	95.85%	0.1178	13:41:40	21.95%

As mentioned by Sipior et al. (2004) [12] and Zhang et al. (2004) [15], the efficiency of the filter is an important factor. The time-based spam filter can be an important contributor to spam fighting in mobile devices, where the efficiency and simplicity of the filter is considered one of the main points during the filter construction.

Other of the important advantages of the time-based spam filter, is that it is fully-independent of the language of the email. Almost all of the commonly content-based spam filters are not fully independent of language, occasionally making the classification less accurate when the user receives emails in different languages. Sometimes the language of the email can be a tool used by spammers to confuse filters, but in the case of the time-based spam filter the language of the email is not an important consideration.

7.1 Countermeasures

One of the main goals of spammers is to defeat anti-spam tools. And as any other spam filter, there are some countermeasures that might foil our proposal. The most obvious attack against our proposal is the one in which the spammer can tamper with the received time-stamp printed by the destination server. This attack requires that the spammer have access to the server in order to manipulate the time-stamp on the email. But as we normally consider the destination server as a trusted one, then this kind of attack in real life will be difficult to succeed.

Another imaginable attack against the time-based spam filter is the one related with the time distribution of the email. If spammers can modify in one way or another the sending basis, then the accuracy of the filter will decrease.

When spammers send spam email in a linear distribution of time, then the time-based spam filter still can learn the difference existing between the time distribution and the linear distribution for the classification. Using a chi-square test we prove that the differences between the linear distribution and the time distribution are still significant [11]. The chi-square equals 31170.099 with 743 degrees of freedom. The two-tailed P value is less than 0.0001. By conventional criteria, this difference is considered to be extremely statistically significant.

A time distribution attack will decrease the accuracy of the filter; but not in a drastic way as proved using different evaluation scenarios and with the time distribution comparison in ham emails.

The use of spyware software in order to know the ham email timing distribution, or in order to modify the received time-stamp of the email; can be a successful tool for the spammer to confuse the filter. If the spammer can modify the time-stamp, and cause all ham and spam email to have the same time-stamp, then the filter cannot make the classification.

8 Conclusions

Everyday, users receive thousands of spam emails. The ongoing battle against spam continues and there is an increasing need of an efficient, accurate and language independent tool for spam classification.

The use of the time as a tool for email classification is a new concept and an under-estimated one. However, the conducted experiments demonstrate that the time characteristic of the email is an extremely efficient and accurate tool for developing a spam filter.

Two of the main problems of the actually in use spam filters are the efficiency and the language-independency. The time-based spam filter proposed in this article is in average from 40.32% to 91.48% more efficient than the ones more commonly used nowadays in the world. Furthermore, being that the time characteristic of the email is the only variable to consider, the time-based spam filter is completely independent from the language.

As the results show, the spam filter proposed in this article accomplishes the required levels of accuracy and exceeds the standard levels of efficiency.

Finally, we consider that the use of time-based spam filter as an auxiliary tool for more complex spam filters presents a possible scenario for further research in the area.

References

[1] R. Baeza-Yates and B. Ribeiro-Neto. *Modern information retrieval.* Addison-Wesley Harlow, England, 1999.

[2] P. O. Boykin and V. P. Roychowdbury. Leveraging social networks to fight spam. *IEEE Computer*, 38(4):61–68, 2005.

[3] A. Bratko, G. V. Cormack, B. Filipic, T. R. Lynam, and B. Zupan. Spam filtering using statistical data compression models. *Journal of Machine Learning Research*, 7:2673–2698, 2006.

[4] D. M. Buede, J. A. Tatman, and T. A. Bresnick. Introduction to bayesian networks:a tutorial for the 66th mors symposium. *66th MORS Symposium, California :Naval Postgraduate School Monterey*, 1998.

[5] G. V. Cormack and T. R. Lynam. Spam corpus creation for trec. *Proceedings of Second Conference on Email and Anti-Spam CEAS*, 2005, July 2005.

[6] G. V. Cormack and T. R. Lynam. Online supervised spam filter evaluation. *ACM Trans. Inf. Syst.*, 25(3):11, 2007. ISSN 1046-8188.

[7] J. Goodman, G. V. Cormack, and D. Heckerman. Spam and the ongoing battle for the inbox. *Commun. ACM*, 50(2):24–33, 2007. ISSN 0001-0782.

[8] P. Graham. A plan for spam. *http://www.paulgraham.com/spam.html*, 22, August 2002.

[9] J. Joyce. Bayes' theorem. *Stanford Encyclopedia of Philosophy*, 2003. http://plato.stanford.edu/entries/bayes-theorem/.

[10] J. Kong, P. O. Boykin, B. Rezaei, N. Sarshar, and V. P. Roychowdbury. Let your cyberalter ego share information and manage spam. *Arxiv preprint physics/0504026*, 2005.

[11] J. Neter, M. Kutner, C. Nachtsheim, and W. Wasserman. *Applied linear statistical models.* Irwin, 1996.

[12] J. Sipior, B. Ward, and P. Bonner. Should spam be on the menu? *Communications of the ACM*, 47(6):59–63, 2004.

[13] Spamassassin.org. Spam mails archive. *http://spamassassin.org/publiccorpus/*, 2003.

[14] S. Whittaker and C. Sidner. Email overload: exploring personal information management of email. In *CHI '96: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 276–283, New York, NY, USA, 1996. ACM Press. ISBN 0-89791-777-4.

[15] L. Zhang, J. Zhu, and T. Yao. An evaluation of statistical spam filtering techniques. *ACM Transactions on Asian Language Information Processing (TALIP)*, 3(4):243–269, 2004.

[16] Y. Zhou, M. Mulekar, and P. Nerellapalli. Adaptive spam filtering using dynamic feature space. *Proceedings of the 17th IEEE International Conference on Tools with Artificial Intelligence*, pages 302–309, 2005.