A Selective Learning Model for Spam Filtering

Didier Colin *

* UFR Sciences Exactes et Naturelles (Université de Reims Champagne Ardenne)
Campus Moulin de la Housse, 51100 Reims
colin.didier@gmail.com

ABSTRACT. We investigate how one can use optimization methods to attune spam filters to some specific issues found in the spam filtering area. Among those issues is the need to modelize and manage spammers strategies to delude filters. To adress it, we propose a selective learning scheme designed to maximize learning efficiency. We show that this approach synergizes well with existing classification models while being easy to implement.

KEYWORDS: spam filtering, optimization, classification, Meta-heuristic.
1. Introduction

Content-based filtering have nowadays reached high accuracies, to the point where the best content-based classifiers are less error-prone than a human who must classify a set of text documents in a limited time. A quite large consensus in the machine learning community is that text classification have today reached a maturity stage from which little improvements are predicted.

Spam filtering can be viewed as a text-classification problem where a human user would in fact have limited time to make a classification decision. Indeed, a user usually wants to minimize the time spent on deleting illegitimate messages. From this point of view, spam filtering is no different than other classification problems.

However, spam brings new issues in the light of which a classification approach is incomplete.

– *Classes definition*: there is no universal definition for spam and legitimate messages. The discriminating classes are subject to the user perception.

– *Adversarial classification*: spams are created by an adversary who is aware of the spam filter and seeks to make it produce false positives.

– *Need for autonomy*: a spam filter needs to be adaptative. Therefore, as spam evolves, it must learn new documents in order to keep its effectiveness. Ideally, a filter should be able to automatically learn messages that present new features, thus minimizing the need for a human intervention and maintaining its accuracy despite spam evolutions.

– *False positive issue*: nearly every classification problem is subject to false positives. But in the case of spam filtering, there is no acceptable amount, as an email incorrectly rejected may be of crucial importance for the user.

All these issues sketch out typical optimization problematics: such as maximizing utility/adaptativity or minimizing human intervention/false-positives. Therefore, we advocate for an optimization approach of the spam filtering problem where filters are seen as structures to optimize. In this work, we propose an optimization model based on the selective learning principle. We formalize the learning process of
an approach on two data sets.

Section 1 presents the selective induction model and discusses its relevancy in regards to the spam filtering problem. Section 2 and 3 respectively present two algorithms for offline and online selective learning. Section 4 details the experimental results of these algorithm on the ling_spam and spamassassin corpus. Finally, we discuss the pros and cons of this approach in regards to the future of spam classification.

2. The selective induction model

A straightforward way to optimize filters performances is to maximize their learning efficiency. Indeed, the way a classifier is induced from a given training corpus greatly affects its future behaviour. In order to reach maximum accuracy, classifiers must extract only pertinent information from the training data — the way that pertinence is modeled depending on the classification model.

But training data may contain more than useless information. Indeed, some may hold destructive knowledge, i.e. knowledge that will decrease a filter performances. Examples of potentially destructive knowledge are messages incorrectly labelled or tricky mails. In a typical classification problem, this phenomenon would theoretically be marginal. But in the spam filtering area, it is more likely to appear as spammers try to delude filters, often by sending bulk emails that include enough "innocent" words to be classified as legitimate [3] [5]. These strategies result in more and more destructive knowledge which can defeat a filter on the long-run.

Thus, we emit the idea that a classifier can improve its accuracy if it chooses not to learn some data. This principle, called selective learning, leads to the issue of identifying destructive knowledge in order to avoid it.

We propose a selective learning scheme where a genetic algorithm is used to select a subpart of the training corpus such that training on this part maximizes the classifier precision on the evaluation corpus. We...
will show that the selective learning algorithm quickly generate better solutions than an exhaustive one.

Furthermore, we show that this approach can be extended to an on-line context, where a filter can choose not to learn incoming messages that may decrease its effectiveness over time. While this model involves parameters that may be tricky to deal with, we give some hints as to how one can attune them to specific data flows.

3. The selective learning problem

3.1. Formalism

Let $\mathcal{C}$ be a corpus containing $n$ messages. We note $f_{\mathcal{C}}$ a classifier induced from the corpus $\mathcal{C}$ using an exhaustive learning method. In other words, $f_{\mathcal{C}}$ is a classifier obtained when the entire corpus $\mathcal{C}$ has been learned.

Définition 3.1 (Selection vector) A selection vector $X$ is a boolean vector $X \in \{0, 1\}^n$, where each component $X_i$ indicates whether the $i$-th message in the corpus $\mathcal{C}$ should be learned or not. We define $\mathcal{C}(X)$ the corpus resulting from the selection of each message $y_i$ in $\mathcal{C}$ such that $X_i = 1$, i.e. $\mathcal{C}(X) = \{y_i \in \mathcal{C} | X_i = 1, \forall i \in [1, ..n]\}$. $\mathcal{C}(X)$ is named the subcorpus selected from $\mathcal{C}$ by $X$.

Définition 3.2 (Selective Learning Problem) The selective learning problem (SLP) formalizes as finding $X$ such that the accuracy of a filter induced from $\mathcal{C}(X)$ on the corpus $\mathcal{C}$ is maximum. Thus, the SLP is an optimization problem where the objective function is $z = \max A(f_{\mathcal{C}(X)}, \mathcal{C})$, where $A(f, \mathcal{C})$ is the ratio of messages in the corpus $\mathcal{C}$ that are correctly classified by $f$.

For a corpus of reasonable size, the solution set is too large for an exhaustive search to be performed. A typical training corpus contains thousands to billions of messages. Each solution must be evaluated by training a new classifier on the selected subcorpus and evaluating it on the evaluation corpus. A typical classifier learns in polynomial time.
Therefore, an exhaustive enumeration is clearly out of reach. Thus the need for an optimization approach.

This led us to opt for a metaheuristic approach. This is a convenient way to undertake a problem for which there is no problem-specific algorithm. While metaheuristics may be computationally expensive, they allow for a first tackle of the problem which may lead to more insights and specific procedures.

3.2. Algorithm

The genetic learning algorithm operates on boolean strings representing selection vectors. Each bit in the string is associated with a message in the learning corpus. If the bit is set to 1, then the message is selected for learning.

The fitness function is the objective function of the SLP: $f = \max A(f_{C(X)}, C)$ where $A(f, C)$ is the weighted accuracy of filter $f$ on corpus $C$:

$$A(f, C) = \alpha A^+(f, C) + \beta A^-(f, C),$$

where $A^+$ and $A^-$ are the accuracy of the filter on spam and legitimate messages, respectively, with $\alpha = 0.1$ and $\beta = 0.9$.

In order to compute the fitness of a given solution $X$, the algorithm must train a new classifier on the corpus selected by $X$ and make it classify the entire training corpus. A classifier typically learns in polynomial time, and classifies a corpus in linear time. Thus, the fitness computation is expensive. To limit this problem, we will opt for an efficient filter and quick genetic operators.

– Selection is elitist. At each generation, the lower half population in term of fitness score is discarded. This is a rather extreme choice in regard to the issues outlined in the previous section. Indeed, an elitist selection have a tendency to bring the search into local extrema. But this approach is fast and tends to converge quickly.

– Reproduction is performed by applying a one-point crossover on randomly chosen solutions until the new population reaches its previous
size. The SLP do not have constraints on how the solutions must be modified and recombined, except for the fixed length of their genomes. Moreover, as in most classification problems, we have no structural knowledge on the training data which would lead to specific constraints or principles on how to generate new solutions. Therefore, there is no \textit{a priori} reason to opt for a more sophisticated crossover. Additionally, the one-point crossover is the fastest crossover technique available.

- Mutation is performed by inverting a randomly chosen bit from a solution, for a number of solutions equal to a predefined mutation rate.

4. The online selective learning model

4.1. Problem description

In a typical classification problem, the initial training phase is very important as new data are likely to obey to a static ontology, which may or not be known. Thus, the classifier performances are likely to stay constant over time. On the contrary, in a spam filtering problem, the classifier faces an opponent who actively works against the filters. This is the so-called \textit{adversarial classification} [4].

Theoretically, machine learning models provide the required adaptivity to spam mutation. A content-based filter can adapt its vocabulary to new messages, such that if spammers begin to use new words, those are progressively included in the filter vocabulary. However, in fact, there are many methods spammers can use to effectively pass around filters; random text generation, vocabulary optimization, viruses, etc.

As a first step to address this issue and anticipate aggressive deluding strategies that the spammers may want to use, we propose to adapt the selective learning principle to an online context. The idea is that each incoming message is classified and tested for learning. If the test is positive, the message is learned. If not, it is discarded. Thus, the online selective learning formalizes as a decision problem.

Définition 4.1 (Online selective learning problem) \textit{Given a trained filter }$f_C$\textit{ and an incoming message }$x_i$, \textit{the online selective learning pro-}
blem formulates as answering the question : is \( f_{C \cup X} \) a better classifier than \( f_C \) ?

In an online context, where a filter deals with a message flow rather than a corpus, it is in theory impossible to compare two classifiers. However, as we do not have any information on the next messages in the flow, we must choose a test which result will be as close to this information as possible.

4.2. Algorithm

A possibility that we explore in this work is to look at the most recent messages. After classification, each incoming message is learned by a duplicate of the filter. The filter accuracy is then compared to its duplicate over the last \( N \) messages received. If the accuracy of the duplicate is better, then the mail is learned.

Définition 4.2 (Selective learning window) Let \( F = (x_1, \ldots, x_m) \) be an ordered set of messages, called mail flow, and \( i, n \) some integers. We define as the selective learning window at time \( i \) the corpus \( C = \{x_j | i - n \leq j \leq i \} \).

5. Experimental results

5.1. Settings

We use a Bernoulli bayesian classifier[6][2]. Features are boolean variables indicating occurrence of words in a document. Therefore, messages are represented as boolean vectors which dimension is the size chosen for the vocabulary. A naive bayesian classifier is a probabilistic classification model where the class \( C \) of a message is determined by a probability given by the Bayes theorem :

\[
P(C|Y) = \frac{P(Y|C) P(C)}{P(Y)}
\]
The naive bayesian assumption assume conditional independence of the features, which allows to easily compute the above formula:

\[
P(C|X) = \frac{P(C) \prod_{i=1}^{n} P(Y_i|C)}{\sum_{k \in \{spam,ham\}} P(C = k) \prod_i P(Y_i|C)}
\]

If \( P(C = spam|X) \) exceeds a given threshold, then the message represented by the vector \( X \) is classified as spam. In our case, the threshold is equal to 0.9.

The vocabulary size is set to 60 words, as it provided the best result in an exhaustive learning scheme on the lingspam corpus. Experiments have been conducted on the ling_spam corpus, which have been made public by Androutsopoulos et al. [1] and have been widely used in the anti-spam filtering community. A sample test has also been realized on the spamassassin corpus.

The reasons why we choose a naive bayesian classifier are that it is a simple and efficient approach to spam filtering, which has been widely used by commercial filters while being easy to implement. Furthermore, a bayesian filter does not need a large training set to be efficient.

5.1.1. Evaluation metrics

We use the total cost ratio (TCR) score introduced Androutsopoulos et al. [1] to evaluate the quality of the induced classifier. TCR is simply the ratio of the weighted (subjective) error rate of the induced classifier over the weighted error rate of a filter that do nothing but accept all incoming messages.

Such a filter will have a low weighted error rate because it does not generate false-positives, which are the more penalizing errors in a subjective view. Therefore, the TCR score gives a more pertinent measure of the subjective quality of a filter. The weight associated with each class is 0.9 for the legitimate messages and 0.1 for the spam.
6. Offline selective learning

6.1. Experiment protocol for offline selective learning

Preliminary experiments have been made by splitting the ling_spam corpus in ten folds with equal spam proportion. A selective learning scheme has been tested on each fold in order to have a first grasp of the performance of a selective learning approach on restrained corpora and help us tuning the parameters. During these preliminary experiments, we used different population sizes in the 10 to 500 range, and various mutation rates ranging from 5% to 75%.

Results showed that the best solutions found often contained only 5 to 15% of the legitimate messages in the training corpus and that 30 to 70% of the spam messages were selected. In most cases, we were able to induce perfect classifiers.

Thus, initial solutions are randomly generated such that the legitimate and spam messages selected represent respectively 10 and 50% of the legitimate and spam emails in the training corpus. We also include in the initial population a particular solution which is the exhaustive selection of the corpus.

This allows the algorithm to quickly converge to good solutions. In order to compensate for the loss of genetic diversity resulting from this choice, and allow the research to quit local extrema, we introduce a growing mutation rate. The rate is initially equal to 5%, and is incremented by 1 at each iteration, with a cap at 75%.

While these settings may make the research converge too quickly to local extrema, it ensures that better solutions than the exhaustive one are found quickly. Given the complexity of the fitness evaluation, it is unrealistic to let the algorithm run for too long on large corporas or in a real time context, thus the need for a rapid useable output.
6.2. Results of the offline selective learning scheme

Below is the TCR score obtained with various population sizes, depending on the number of iterations. The TCR score obtained by a standard exhaustive learning is also reported.

![TCR evolution in genetic learning algorithm](image)

Figure 1 – TCR evolution for various population size, on the lingspam corpus.

Results show that the algorithm quickly converges. A peculiar point of interest is that the algorithm does not require to run a minimum number of iterations to find better solutions than the exhaustive version. This means that a selective learning process is always preferable to an exhaustive one, provided that initial solutions are well constructed.

In the context of our experiment, the optimal population size seems to be 25, since it converges faster than higher sizes while attaining the second best TCR score over 300 turns, by a very small margin. We have continued the experiments with this value to see if the filter could be further improved.

With these settings, the best solution is found after 1900 iterations. We have let the algorithm run for nearly 6000 iterations without improvements.
Figure 2 – TCR evolution for a population of 25 individuals, on the lingspam corpus.

Tableau 1 – Comparison of spam precision and spam recall for exhaustive and selective learning algorithm, on the lingspam corpus.

<table>
<thead>
<tr>
<th></th>
<th>Exhaustive learning</th>
<th>Selective learning (initial)</th>
<th>Selective learning (best)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spam Precision</td>
<td>96.82 %</td>
<td>96.85 %</td>
<td>98.72 %</td>
</tr>
<tr>
<td>Spam Recall</td>
<td>88.33 %</td>
<td>89.60 %</td>
<td>96.47 %</td>
</tr>
</tbody>
</table>
Finally, the algorithm was run on the spamassassin corpus, with mitigate results. Indeed, while on the lingspam corpus, the algorithm find better solutions than an exhaustive approach at the first iteration, this is not the case in spamassassin where the exhaustive approach remains the best solution for more than ten generations. Reasons why the selective approach is less efficient for spamassassin may be found in the fact that it is a hard heterogeneous corpus, which rassembles the most diverse kind of spam and legitimate communications as possible, in order for commercial filters trained from it to be as universally useful as possible. The lingspam corpus, on the contrary is extracted from a unique and coherent mailstream, making it more subject to redundant communications.

Another point of interest in the case of the spamassassin corpus is that the standard naive bayesian filter, with an exhaustive learning, achieves bad results, with a TCR score below the 1 threshold. This is essentially due to our “naive” implementation, which does not include a smart lemmatizer, and makes no use of hybrid approaches as is the case in most commercial filters. This means that we have a filter which has enough flaws to be less accurate than a “no rejection” policy. While
the selective learning approach allows to slightly increase its accuracy, it does not allow to overcome these flaws.

These results, while confirming that a selective learning approach allows a filter to significantly improve its accuracy, shows that it do little for the induction of filters on heterogeneous corpus, but can make the induction of a filter on a corpus extracted from a single mailstream a viable alternative.

The only drawback to this method is its heavy computational complexity. In fact, the best solution is found in approximately 30 hours on an intel pentium 4 processor at 3.20GHz. While this is obviously a concern, it doesn’t seem critical for two reasons.

First, we have shown that even initial solutions provided by a selective learning scheme are better than exhaustive ones. Second, this complexity may be acceptable in some context and even reduced in some ways. For instance, filters commercialized with ready-to-use training model could afford to spend more time in training before being released. Multi-core or multi-processors machines may use a parallel approach to decrease the training time. One may even consider to use a processor idle periods to improve the current training model of a filter installed on a user machine.

Indeed, compared to other corpora, such as the spamassassin or TREC corpus, the lingspam corpus is quite small. However, those are public corpora which are useful to train filters which are then distributed to the final users ready to be used. As such, these filters can afford long training sessions, making a selective learning approach a valid solution.

However, a selective learning approach really shines on small training data sets, allowing to build a functional classifier from a restrained number of samples. This feature makes the induction of a classifier from an individual mailbox a viable alternative to filters built from large public corpora. This is a desirable goal because the classes to discriminate in a spam filtering problem are mostly user-dependent. Thus, a filter trained on an individual mailbox should be able to better catch the specificities of the spam and legitimate mail streams associated with its user.
7. Online selective learning

7.1. Experiment protocol for online selective learning

First experiments have been made both on a regular and noisy version of the corpus ling_spam, which was obtained by inverting the label of randomly chosen messages with a noise rate fixed at 5%. This allows for the classification task to be a little more difficult and to test the online selective learning model in a context where it should obviously perform better than an exhaustive learning scheme. The corpus is learned iteratively by a Bernoulli bayesian filter which starts with no knowledge. The model has been tested with various values for $N$.

7.2. Results of the offline selective learning scheme

![Figure 4 – TCR evolution in a online learning context, with a regular corpus, for various values of $N$](image)

Results show that, in a noisy message flow, the online selective learning model both performs better than an exhaustive approach and induces a filter which maintain a slightly increasing accuracy over time, while the exhaustive model slowly loses its effectiveness. In the case of a regular flow, both approaches have similar performances with the selective model performing slightly better only with $N = 500$. This
result tends to demonstrate that a "conservative" approach is generally preferable in the case of "easy" corpora.

In the case of a noisy message flow, it is worth noting that the choice of $N$ has a great impact on the behaviour of the filter. If $N$ is too high, then the filter becomes more and more conservative, resulting in performances which are close to an exhaustive learning. On the contrary, if $N$ is too low, then the filter do not have sufficient information to make pertinent decisions on the long-run, resulting in non-optimal performances.

Another point of interest is that the TCR score drops below 1 with $N = 500$ or an exhaustive learning while it is above for the other values. A TCR score below 1 means that the filter effectiveness is lower than a filter that do nothing but accept all incoming messages. This observation suggests that, given proper parameters and the fact that the selective approach can provide an increasing accuracy over time, an online selective learning model could be used to construct a spam filter "from scratch", with no need for an initial training corpus.

These results also reveals the importance of a very well known issue in the spam filtering area, which is the need for a human correction for filters to maintain their effectiveness. In fact, a standard exhaustive
approach tend to learn, and thus, repeat, its mistakes, unless a human user modifies the incorrect labels, by marking a message as spam or removing a legitimate one incorrectly sent to the spam directory.

Therefore, it seems that a selective learning process can reduce the need for human intervention. As many users do not take time to correct their mailbox filter, or simply do not care, filters tend to be less and less effective over time, allowing for spam campaigns to reach a satisfying number of mailboxes for the spammers. By using a selective approach, one could make filters able to automatically maintain or improve their accuracy, resulting in less viable spam campaigns on a global scale.

8. Conclusions and perspectives

The selective learning scheme have proven to be a robust approach to optimize filters performances, simple to implement and easy to use with any existing anti-spam technology. We have shown that the selective learning principle adresses some of the specific issues of the spam filtering area, namely automatization and adaptativity to spammers strategies. It is our intuition that, while classification techniques have reached a satisfying maturity, by working around these models to introduce simple optimization routines, one could greatly improve these techniques in regard to the spam problem specificities.

Another field of interest is how one could tune the parameter $N$ in the online selective learning process and how it could be dynamically adapted to spammers strategies. There are many ways to adress this problem. In the case of classification models based on vectorial representations, it may be possible to analyze the trajectory of the incoming messages in the description space, and to adapt the value of $N$ based on its regularity.

On a related note, it may be worth asking ourselves if the messages which are not learned contain information which could be used for other purposes such as the recognition of useless or destructive knowledge, or the tuning of the learning method parameters.
In our future works, we plan to deeply investigate the selective learning principles, its various applications to the spam problem, and its synergies with existing classification techniques.

Références


