A Selective Learning Model For Spam Filtering

Didier Colin
Prism Laboratory
Versailles University
France

Catherine Roucairol
Prism Laboratory
Versailles University
France

Ider Tseveendorj
Prism Laboratory
Versailles University
France

March 9, 2009

Abstract

We investigate how one can use optimization methods to attune spam filters to the specific issues found in the spam filtering area. Among those issues is the need to modelize and manage spammers strategies to delude the filters. To adress this issue, we propose a selective learning scheme designed to maximize learning efficiency. In an offline context, this model uses a simple metaheuristic approach to select a subpart of training data such that the filter induced on that part maximizes its accuracy over the evaluation set. In an online context, we show how one machine learning algorithm can discard incoming messages in order to prevent its knowledge base to be biased by messages which are not good representatives of their class and thus, may lead to a decrease in accuracy if they were to be learned. We show that this approach synergizes well with existing classification models while increasing significantly their efficiency over time. More importantly, we show that this model make existing filters less vulnerable to spammers attempts to delude them.

Introduction

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 and generalization capabilities, classifiers must extract only pertinent informations from the training data — the way that pertinence is modelized depending on the classification model.

But training data may contain more than useless informations. 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 by sending bulk emails that include enough "innocent" words to be classified as legitimate[4]. Some spammers even use optimization techniques to generate spam messages that minimize a "spaminess" score computed by a regular filter. 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 have tested this approach on a Bernoulli naive bayesian filter. 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 online context, where a filter can choose not to learn incoming messages that may decrease its effectiveness over time. We give a very simple algorithm for online selective learning and show that this model can greatly improve a classifier performance while preventing a natural decrease in accuracy over time. While this model involve parameters that may be tricky to deal with, we give some hints as to how one can attune them to specific data flows.

Section 1 presents the base selective learning formalism and algorithm. Section 2 deals with experimental results of this model on the corpus ling_spam[1]. Section 3 presents an online adaptation of this model and some results on a simple test protocol, and discusses its pros and cons. Finally, section 4 gives some conclusions and perspectives as to how the principles of selective learning can be adaptated to the spam filtering problem.

1 Base selective learning model

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. Let $X \in \{0,1\}^n$ be a boolean vector, named *selection vector*, 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. $\mathcal{C}(X) = \{y^i \in \mathcal{C} | X_i = 1, \forall i \in [1,..n]\}$

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 reasonnable size, the solution set is too large for an exhaustive search to be performed. This led us to opt for a metaheuristic approach. In this approach, we use a genetic algorithm to make a population of solutions converge. Solutions are represented by selection vectors. We use $A(f_{\mathcal{C}(X)}, \mathcal{C})$ as a fitness function. Put simply, at each iteration different subcorpora are selected and a filter is trained on each of them. Each filter induced is then evaluated on the global training corpus. Its resulting accuracy is used to determine the quality of the solution.

The genetic learning algorithm is the following :

Algorithm 1: Genetic learning algorithm

Input:
$\mathcal{C}$, a training corpus
f, a classifier
POP_SIZE, an integer
Output:
$\mathcal{C}^* \subset \mathcal{C}$, a training corpus
Data:
X, a boolean vector of dimension n
P, a set of boolean vector

```
begin
    for i from 1 to POP_SIZE do
        X ←random selection vector
        P ← P ∪ X
    end
    while turn < max_turn do
        P ← selection(P)
        P ← reproduction(P)
        P ← mutation(P)
        turn ← turn + 1
    end
    X ←argmax_{X_i ∈ P}{fitness(X_i, f)}
    return C(X)
end
```

Selection is elitist : at each generation, the lower half population in term of fitness score is discarded.

The reproduction process is performed by applying a one-point cross-over on randomly chosen solutions until the new population reach its previous size. The cross-over operation takes two solutions, split them at a random bit. The second half part of each solutions are then interverted, which result in two new child-solutions.

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

The fitness function of a given solution X is com-

puted by training a filter on $\mathcal{C}(\mathcal{X})$ and evaluating it on $\mathcal{C}$.

2 Experiments and results

2.1 Protocol

Preliminary experiments on restrained corpora, with various population size and mutation rate, have shown 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. 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. 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 (which needs a filter to be trained and evaluated in order to be computed), 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.

The filter used is a Bernoulli bayesian classifier[5][2]. Features are boolean variables indicating occurence 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 probabilist classification model where the class C of a message is determined by a probality given by the Bayes theorem :

$$P(C = c|Y = y) = \frac{P(Y = y|C = c).P(C = c)}{P(Y = y)}$$

The naive bayesian assumption assume conditional independence of the features, which allow to easily compute the above formula :

$$P(C = c|Y = y) = \frac{P(C = c)\prod_{i=1}^{n} P(y_i|C = c)}{\sum_{k\in\{spam,ham\}} P(C = k)\prod_i P(y_i|C = c)}$$

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

The vocabulary is constituted by features w_i that achieve the highest mutual information score, relative to each considered class.

$$MI(w, C) = \sum_{w\in\{0,1\}} \sum_{C\in\{spam,ham\}} P(w|C = c).\log \frac{P(w|C = c)}{P(w).P(C = c)}$$

The vocabulary size is set to 60 words, as it provided the best result in an exhaustive learning scheme. 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.

We use the total cost ratio (TCR) score introduced by the same authors 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.

2.2 Results

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.

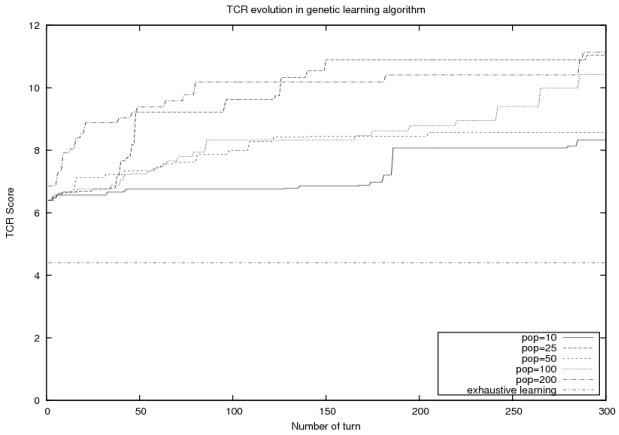

Figure 1: TCR evolution for various population size

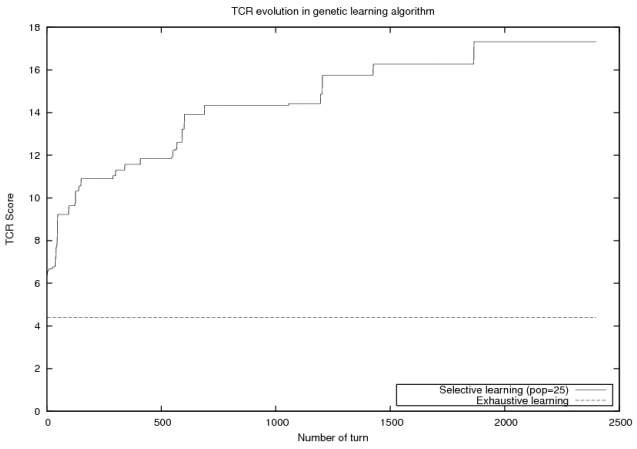

Figure 2: TCR evolution for a population of 25 individuals

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 improvments. Here, we report the resulting performances in term of spam recall and spam precision :

This confirms that a selective learning approach allows a filter to significantly improve its accuracy.

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. This approach have been succesfully employed in some large public softwares that uses heavy data mining techniques like Google Desktop™.

Table 1: Comparison of spam precision and spam recall for exhaustive and selective learning algorithm

	Exhaustive learning	**Selective learning (initial)**	**Selective learning (best)**
Precision	96.82 %	96.85 %	98.72 %
Recall	88.33 %	89.60 %	96.47 %

It is also instructive to note that the best solutions contains only 31% of the training corpus, distributed equally among spam and legitimate messages. This result may look surprising. In fact, it is a common statement that the more a classifier learns, the more

smart it becomes. While this statement is not absolutely false, these experiments shows that, in the context of adversarial classification, it has limits. Reasons why only a few messages are really useful may be found in the fact that human communication is, by nature, repetitive. Other explanations may be found in the classifier nature. It is very possible that vector-based classification models are more sensible to data multiplication than other non-vectorial classification models.

3 Online selective learning

In a classical 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's 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 *adversarial classification*[3]. In this context, spammers tries to delude the filters by sending spam messages which evolve constantly to escape the classification models. As a result, a filter must be able to adapt quickly while retaining the major features of spam messages.

As a first step to adress this issue and anticipate agressive deluding strategies that the spammers may want to use, we propose to adapt the selective learning principle to an online context. The idea is very simple : each incoming message is classified and tested for learning. If the test is positive, the message is learned. If not, it is discarded.

In order to have a first overview of this idea, we have chosen a simple test. After classification, each incoming message is learned by a duplicate of the filter. The filter's accuracy is then compared to its duplicate over the last N messages received. If the duplicate's accuracy is better, then the mail is learned.

Algorithm 2: Online selective learning

Input:
W_i, the i-th message on the mail flow
f, a classifier
λ, a real such that $0 \leq \lambda \leq 1$
N, an integer
Output:
accept, a boolean
Data:
f' a classifier
$\mathcal{C}$, a corpus

```
begin
    f' ← copy(f)
    if f(W) ≥ λ
    then
        learn(f', W, spam)
    else
        learn(f', W, ham)
    end
    C ← {W_j, i − N ≤ j ≤ i}
    if A(f,C) ≥ A(f', C)
    then
        return false
    else
        return true
    end
end
```

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 allow 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.

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 ap-

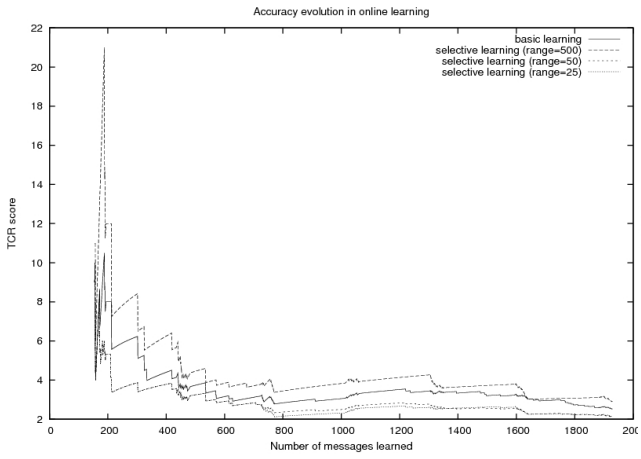

Figure 3: TCR evolution in a online learning context, with a regular corpus, for various values of N

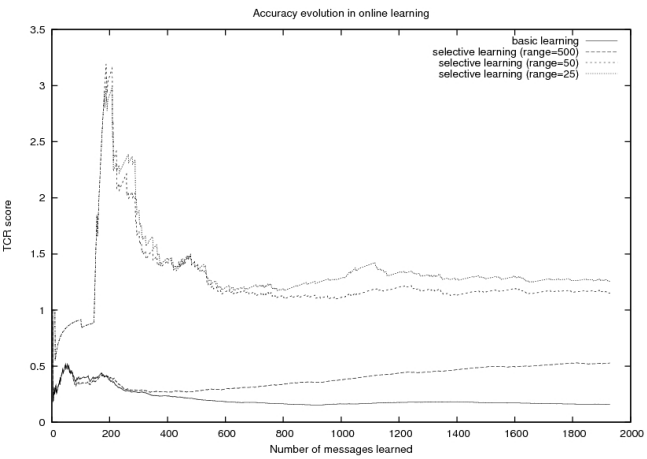

Figure 4: TCR evolution in a online learning context, with a noisy corpus (5%) for various values of N

proaches 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 filter's behaviour. If N is too high, then the filter's become 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 informations 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's 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 uncorrect 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.

4 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 strate-

gies. It is our intuition that, while classification techniques have reached a satisfying maturity, by working aroung these models to introduce simple optimization routines, one could greatly improve these techniques in regard to the spam problem specificities. The offline selective learning has yet to be extensively tested with other optimization methods heuristics and other popular classifiers such as SVMs or neural networks.

Another field of interest is how one could tune the parameter N in the online selective learning process and how it could be dynamically adaptated 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 analyse 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 informations 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.

References

[1] Ion Androutsopoulos, John Koutsias, Konstantinos Chandrinos, George Paliouras, and Constantine D. Spyropoulos. An evaluation of naive bayesian anti-spam filtering. *Computing Research Repository*, cs.CL/0006013, 2000.

[2] Thomas Bayes. An essay towards solving a problem in the doctrine of chances. *Philosophical Transactions of the Royal Society of London*, 53:370–418, 1763.

[3] Nilesh Dalvi, Pedro Domingos, Mausam, Sumit Sanghai, and Deepak Verma. Adversarial classification. In *Proceedings of International Conference on Knowledge Discovery and Data Mining 2004*, pages 99–108, 2004.

[4] John Graham-Cumming. What spammers are doing to get around bayesian filtering & what we can expect for the future. In *Large Installation Systems Administration*, 2004.

[5] Mehran Sahami, S. Dumains, David Heckerman, and E. Horvitz. *A bayesian approach to filtering junk E-mail*. Learning for Text Categorisation: Papers from the 1998 Association for the Advancement of Artificial Intelligence, 1998.