Expert Systems with Applications 39 (2012) 6557–6561

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Facing the spammers: A very effective approach to avoid junk e-mails

Tiago A. Almeida *, Akebo Yamakami

School of Electrical and Computer Engineering, University of Campinas – UNICAMP, 13083-852 Campinas, SP, Brazil

ARTICLE INFO

Keywords:
Minimum description length
Confidence factors
Spam filter
Text categorization
Machine learning

ABSTRACT

Spam has become an increasingly important problem with a big economic impact in society. Spam filtering poses a special problem in text categorization, in which the defining characteristic is that filters face an active adversary, which constantly attempts to evade filtering. In this paper, we present a novel approach to spam filtering based on the minimum description length principle and confidence factors. The proposed model is fast to construct and incrementally updateable. Furthermore, we have conducted an empirical experiment using three well-known, large and public e-mail databases. The results indicate that the proposed classifier outperforms the state-of-the-art spam filters.

1. Introduction

E-mail is one of the most popular, fastest and cheapest means of communication. It has become a part of everyday life for millions of people, changing the way we work and collaborate. E-mail is not only used to support conversation but also as a task manager, document delivery system and archive. The downside of this success is the constantly growing volume of e-mail spam we receive. The problem of spams can be quantified in economical terms since many hours are wasted everyday by workers. It is not just the time they waste reading the spam but also the time they spend deleting those messages.

According to annual reports, the amount of spam is frightfully increasing. The average of spams sent per day increased from 2.4 billion in 2002[1] to 300 billion in 2010[2] representing more than 90% of all incoming e-mail. On a worldwide basis, the total cost in dealing with spam was estimated to rise from US$ 20.5 billion in 2003, to US$ 198 billion in 2010.

Fortunately, many solutions are being proposed to avoid this "plague" and one of most promising is the use of machine learning techniques for automatically filtering e-mail messages (Cormack, 2008). These methods include approaches that are considered top-performers in text categorization like Rocchio (Joachims, 1997; Schapire, Singer, & Singhal, 1998), Boosting (Carreras & Marquez, 2001), Support Vector Machines (SVM) (Almeida & Yamakami, 2010; Almeida, Yamakami, & Almeida, 2010a; Drucker, Wu, & Vapnik, 1999; Hidalgo, 2002; Kolcz & Alspector, 2001; Ying, Lin, Lee, & Lin, 2010), Collaborative Systems (Lai, Chen, Laih, & Chen, 2009), Concept Drift (Fdez-Riverola, Iglesias, Diaz, Mendez, & Corchado, 2007), Cluster-based Approach (Hsiao & Chang, 2008), Logistic Regression (Goodman & Yih, 2006; Lynam, Cormack, & Cheriton, 2006; Perlich, Provost, & Simonoff, 2003) and Naïve Bayes classifiers (Almeida, Yamakami, & Almeida, 2009, 2010b; Almeida, Almeida, & Yamakami, 2011; Androutsopoulos, Paliouras, & Michelakis, 2004; Guzella & Caminhas, 2009).

A relatively recent method for inductive inference which is still rarely employed in text categorization tasks is the minimum description length principle. It states that the best explanation, given a limited set of observed data, is the one that permits the greatest compression of the data (Barron, Rissanen, & Yu, 1998; Grünwald, 2005; Rissanen, 1978). Other modern technique is the confidence factors (Assis, Yerazunis, Siefkes, & Chhabra, 2006) that was proposed to reduce the noise introduced by features with small counts and de-emphasize those with low class separation power.

In this paper, we present a novel spam filtering approach that is based on the minimum description length principle (Bratko, Cormack, Filipic, Lynam, & Zupan, 2006) and confidence factors (Assis et al., 2006). We have conducted an empirical experiment using three well-known, large, and public databases and the reported results indicate that our approach outperforms currently established spam filters.

A very basic and preliminary version of this work was presented at ACM SAC 2010 (Almeida et al., 2010a). Here, we significantly improve the algorithm and extend its evaluation. First, and the most important, we add the confidence factors to assist the classifier's prediction. Second, we offer different tokenizer and training methods. Additionally, we use more realist e-mail collections and different tasks in our experiments. Finally, we compare the proposed filter with the state-of-the-art spam classifiers.

The remainder of this paper is organized as follows: Section 2 presents the main concepts behind the proposed spam filter.

* Corresponding author. Tel.: +55 (19) 3521 3846; fax: +55 (19) 3521 3866.
E-mail addresses: tiago@dt.fee.unicamp.br (T.A. Almeida), akebo@dt.fee.unicamp.br (A. Yamakami).

[1] See http://www.spamlaws.com/spam-stats.html

[2] See www.ciscosystems.cd/en/US/prod/collateral/cisco_2009_asr.pdf

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.12.049

Section 3 presents details of the proposed approach. Section 4 describes the experimental setup and evaluation methodology. Experimental results are showed in Section 5. Finally, Section 6 offers conclusions and directions for future works.

2. Basic concepts

In this section we present the main concepts of both approaches minimum description length and confidence factors that compose the core of the proposed anti-spam classifier.

Assuming that each message m is composed by a set of terms (or tokens) $m = t_1,\ldots,t_n$, where each term t_k corresponds to a word (“adult”, for example), a set of words (“to be removed”), or a single character (“$”), we can represent each message by a vector $\vec{x} = \langle x_1,\ldots,x_n\rangle$, where $x_1,\ldots,x_n$ are values of the attributes $X_1,\ldots,X_n$ associated with the terms $t_1,\ldots,t_n$. In the simplest case, each term represents a single word and all attributes are Boolean: $X_i = 1$ if the message contains t_i or $X_i = 0$, otherwise.

2.1. Minimum description length principle – MDL

The purpose of statistical modeling is to discover regularities in observed data. The success in finding such regularities can be measured by the length with which the data can be described. This is the rationale behind the minimum description length (MDL) principle (Rissanen, 1978). The fundamental idea is that any regularity in a given set of data can be used to compress the data.

The MDL principle is a formalization of Occam's Razor in which the best hypothesis for a given set of data is the one that yields compact representations. The traditional MDL principle states that the preferred model results in the shortest description of the model and the data, given this model. In other words, the model that best compresses the data is selected. This model selection criterion naturally balances the complexity of the model and the degree to which this model fits the data.

Let $\mathcal{Z}$ be a finite or countable set and let P be a probability distribution on $\mathcal{Z}$. Then there exists a prefix code C for $\mathcal{Z}$ such that for all $z \in \mathcal{Z}$ the code length $L_C(z) = \lceil -\log_2 P(z)\rceil$. C is called the code corresponding to P. Similarly, let C be a prefix code for $\mathcal{Z}$. Then there exists a (possibly defective) probability distribution P such that for all $z \in \mathcal{Z}, -\log_2 P'(z) = L_{C'}(z)$. P' is called the probability distribution corresponding to C'. Thus, large probability according to P means small code length according to the code corresponding to P and vice versa (Barron et al., 1998; Grünwald, 2005; Rissanen, 1978).

The goal of statistical inference may be cast as trying to find regularity in the data. Regularity may be identified with ability to compress. MDL combines these two insights by viewing learning as data compression: it tells us that, for a given set of hypotheses $\mathcal{H}$ and data set $\mathcal{D}$, we should try to find the hypothesis or combination of hypotheses in $\mathcal{H}$ that compresses $\mathcal{D}$ most (Barron et al., 1998; Grünwald, 2005; Rissanen, 1978).

This idea can be applied to all sorts of inductive inference problems, but it turns out to be most fruitful in problems of model selection and, more generally, dealing with overfitting. According to Grünwald (2005), an important property of MDL methods is that they provide automatically and inherently protect against overfitting and can be used to estimate both the parameters and the structure of a model. In contrast, to avoid overfitting when estimating the structure of a model, traditional methods such as maximum likelihood must be modified and extended with additional, typically adhoc principles.

In essence, compression algorithms can be applied to text categorization by building one compression model from the training documents of each class and using these models to evaluate the target document.

2.2. Confidence factors – CF

The confidence factors were first proposed by Assis et al. (2006) with the motivation of reducing the noise introduced by features with small counts and/or low significance. This is an attempt to mimic what we do when inspecting a message to tell if it is spam or not. We intuitively consider only a few tokens, those which carry strong indications, according to what we have learned and remember, and discard the ones that may occur approximately equally in both classes.

The confidence factors ($0 \leqslant CF < 1$) for a term (t_i) is calculated taking into account the weight, the maximum and the minimum frequency of the feature over the classes, using the empirical formula (Assis et al., 2006):

$$CF(t_i) = \frac{\left(\frac{(H_{max}-H_{min})^2+(H_{max}\times H_{min})-\frac{K_1}{SH}}{SH^2}\right)^{K_2}}{1+\left(\frac{K_3}{SH}\right)},$$

where:

- H_{max} corresponds to the number of documents with the feature on the class with maximum local probability.
- H_{min} is the number of documents with the feature on the class with minimum local probability.
- SH corresponds to the sum of H_{max} and H_{min}.
- K_1, K_2, K_3 are empirical constants. They adjust the decay speed of the confidence factor as the difference in counts reduces and the influence of the weights.

H_{max} and H_{min} are normalized to the maximum number of learnings of the two classes involved.

3. MDL-CF spam filter

Given a set of pre-classified training messages $\mathcal{M}$, the task is to assign a target e-mail m with an unknown label to one of the classes $c \in \{spam, ham\}$. So, the method measures the increase of the description length of the data set as a result of the addition of the target document. Finally, it chooses the class for which the description length increase is minimal (Almeida et al., 2010a).

In this work, we consider each class (model) c as a sequence of terms (tokens) extracted from the messages and inserted into the training set. Each term t from message m has a code length L_t based on the sequence of terms presented in the messages of the training set of c. The length of m when assigned to the class c corresponds to the sum of all code lengths associated with each term of $m, Lm = \sum_{i=1}^{|m|} L_{t_i}$. We calculate $L_{t_i} = \lceil -\log_2 P_{t_i}\rceil$, where P is a probability distribution related with the terms of class. Let $n_c(t_i)$ the number of times that t_i appears in messages of class c, then the probability that any term belongs to c is given by the maximum likelihood estimation:

$$P_{t_i} = \frac{n_c(t_i) + \frac{1}{|\Delta|}}{n_c + 1},$$

where n_c corresponds to the sum of $n_c(t_i)$ for all terms which appear in messages that belongs to c and $|\Delta|$ is the vocabulary size. In this work, we assume that $|\Delta| = 2^{32}$, that is, each term in an uncompress mode is a symbol with 32 bits. This estimation reserves a “portion” of probability to words that the classifier has never seen before.

Briefly, the proposed MDL-CF spam filter classifies a message by following these steps: