Expert Systems with Applications 40 (2013) 4010–4021

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

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

Optimising anti-spam filters with evolutionary algorithms

Iryna Yevseyeva [a,b], Vitor Basto-Fernandes [b,c], David Ruano-Ordás [d], José R. Méndez [d,*]

[a] University of Leiden, Leiden Institute for Advanced Computer Science, 2333 CA Leiden, Netherlands
[b] Informatics Engineering Department, Technology and Management School, Polytechnic Institute of Leiria, 2411-901 Leiria, Portugal
[c] Computer Science and Communications Research Center, Polytechnic Institute of Leiria, 2411-901 Leiria, Portugal
[d] University of Vigo, Campus As Lagoas S/N, 32004 Ourense, Spain

ARTICLE INFO

Keywords:
Anti-spam filters
Multiobjective optimisation
Evolutionary computation
Genetic algorithms

ABSTRACT

This work is devoted to the problem of optimising scores for anti-spam filters, which is essential for the accuracy of any filter based anti-spam system, and is also one of the biggest challenges in this research area. In particular, this optimisation problem is considered from two different points of view: single and multiobjective problem formulations. Some of existing approaches within both formulations are surveyed, and their advantages and disadvantages are discussed. Two most popular evolutionary multiobjective algorithms and one single objective algorithm are adapted to optimisation of the anti-spam filters' scores and compared on publicly available datasets widely used for benchmarking purposes. This comparison is discussed, and the recommendations for the developers and users of optimising anti-spam filters are provided.

1. Introduction

With the increasing proliferation of information and communication technologies and the growing information worldwide exchanges through Internet, making Internet services and resources controllable against malicious usage became vital. The growing of connections to exchange large amounts of data (such as videos, music, etc.) supported by Internet network introduced the need of improving both effectiveness (objectives oriented), and efficiency (optimal usage of resources for achieving specific goals). Recent developments on high-speed computer networks (by using *Fiber-to-the-x* (FTTx) technologies, such as Fiber-to-the-curb, Fibre-to-the-building, Fiber-to-the-house (Idate, 2012)) allowed fast exchanging of large volumes of information. However, the huge amount of spam contents distributed through networks has limited their benefits and currently, a lot of Internet physical resources, technical managers and end users time are wasted for deleting spam messages, closing spam web banners, and downloading unwanted spam information.

Spammers found and developed a wide variety of forms to distribute illegal and fraudulent advertisements. Due to the continuous changing of techniques used to distribute spam, anti-spam filters become obsolete in a short time period and need to be updated on a regular base. This situation created preconditions for the development and wide spreading of professional anti-spam filtering services. Aiming at customer satisfaction and accuracy of emails classification, the modern anti-spam filtering systems are desired to have the following properties: (i) ability of continuous updating of a default anti-spam filter and adding new rules to it with respect to customer preferences and (ii) ability to stay up to date with the latest spam spreading techniques. Behind these services there are teams of experts examining emails and updating anti-spam filters behaviour to detect the newest spam contents. Current filtering frameworks (including *SpamAssassin* (The Apache SpamAssassin Project, 2011) or Wirebrush4SPAM (Pérez-Díaz, Ruano-Ordas, Fdez-Riverola, & Méndez, 2012b) support filter customisation by using a filtering description syntax based on message fields and contents criteria.

Hundreds of enterprises develop and commercialise anti-spam filtering services. Most of these services are based on signup (gathering information about username, pay methods, mail transfer agent server and target domain) and *change mail exchange* (MX) register (Mockapetris,1987) of the target domain (AgentSPAM, 2012). Providing continuous updating of filtering services at affordable cost makes these services very attractive to *small and medium-sized enterprises* (SMEs).

From a technical point of view, the generation of rules to address new trends of spam emails is an easy process. However, discovering the relative importance of (thousands of) rules to assign individual scores for weighting each rule included in a filter, is a complex setup process, performed usually without any guidance or systematic support. This task should be done automatically, taking into account the need of possible reassignment of existing rules scores, when a new rule is added to the system. Currently, this task

* Corresponding author. Address: ESEI: Escuela Superior de Ingeniería Informática, Edificio Politécnico, Campus Universitario As Lagoas s/n, 32004 Ourense, Spain. Tel.: +34 988 387015; fax: +34 988 387001.

E-mail addresses: iryna@liacs.nl (I. Yevseyeva), vitor.fernandes@ipleiria.pt (V. Basto-Fernandes), drordas@uvigo.es (D. Ruano-Ordás), moncho.mendez@uvigo.es (J.R. Méndez).

0957-4174/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2013.01.008

has been addressed by the techniques surveyed in (Basto-Fernandes, Yevseyeva, & Méndez, 2012), such as *evolutionary algorithms* (The Apache SpamAssassin Group, 2009a; The Apache SpamAssassin Group, 2010), *logistic regression* (Dreiseitl & Ohno-Machado, 2002), *neural network* trained with error back propagation by gradient descent (*Perceptron*) (The Apache SpamAssassin Group, 2004) and Grindstone4SPAM (SING Group, 2007; Méndez, Reboiro-Jato, Díaz, Díaz, & Fdez-Riverola, 2012). However there is still the need of solving existing drawbacks such as: (*i*) the absence of automatic customization processes to avoid rules execution when are useless in certain domains (Pérez-Díaz et al., 2012b), (*ii*) the selection of the appropriate rule weights to handle user and business area requirements (SING Group, 2007; Méndez et al., 2012) and finally, (*iii*) the elimination of the irrelevant filtering rules in order to avoid their execution and hence the reduction of the time needed to accomplish the filtering process (Pérez-Díaz et al., 2012b).

In this work, we test the suitability of using different evolutionary computation approaches for automatic scores setting of rules in an anti-spam filter.

The rest of the paper is structured as follows: Section 2 introduces the target problem and surveys the techniques used for optimising scores of anti-spam filters. Section 3 describes the experimental protocol and the experimental results are provided in section 4. Finally, the conclusions and future work are drawn in section 5.

2. Optimisation of anti-spam filters

Recently, the open source SpamAssassin filtering system gained popularity among SMEs users and became a reference in the anti-spam filtering domain. Its popularity is not only due to its public availability to research and development (becoming a disadvantage being available to spammers), but also because of its performance. SpamAssassin introduced to the anti-spam filtering domain two major features (Pérez-Díaz et al., 2012b): (*i*) the possibility of modelling the filter operation as a combination of rules of different types working together and (*ii*) the ability of updating the filter behaviour by introducing new rules into the system. These features have also been widely exploited to develop other advanced anti-spam filtering solutions such as Symantec Brightmail (Symantec Corporation, 2012) or McAffee SpamKiller (McAfee, 2012), addressed mainly to leading big companies and also some SMEs.

As we can see from (Pérez-Díaz et al., 2012b) and (Pérez-Díaz, Ruano-Ordás, Méndez, Gálvez., & Fdez-Riverola, 2012c) SpamAssassin is a plugin middleware and framework for the execution and development of new user defined anti-spam filters and techniques. Each SpamAssassin technique can be combined in a filter depending on user needs. These techniques are implemented in separate plug-ins. Each plug-in is treated as a different entity avoiding dependencies between plug-ins and guaranteeing high modularity to the whole anti-spam system. Moreover, this feature provides a great flexibility to the platform allowing easy creation, manipulation and deployment of new customized anti-spam filtering techniques.

Table 1 introduces a brief description of different types of filtering techniques provided by default in SpamAssassin, extracted from /usr/share/perl5/Mail/SpamAssassin/Plugin directory.

As we can see from Table 1, the SpamAssassin techniques are divided into four different groups: (*i*) responsible for executing an intelligent analysis of message contents, (*ii*) reliable for querying collaborative networks and servers sharing information about spam senders and deliveries, (*iii*) in charge of validating senders legitimacy and finally, (*iv*) regular expressions and parsers for checking email structure and syntax. Using each type of technique on its own is not efficient and therefore, some combinations of techniques of different types are applied. Keeping in mind this idea, a SpamAssassin filter is combination of techniques through rules.

A SpamAssassin filter is mainly composed by a collection of rules and a threshold called *required_score*. Each rule contains a logical test (that works as a trigger condition and uses one of the

Table 1
SpamAssassin filtering techniques description.

Method	Filter Type Technique	Plug-in name	Description
Content-based	Naïve Bayes (NB) (Metsis, Androutsopoulos, & Paliouras, 2006; Androutsopoulos, Koustias, Chandrinos, Paliouras & Spyropoulos, 2000)	Bayes.pm	Calculate the probability of an email being spam by computing NB probability.
	Language Guessing	TextCat.pm	Guesses the language of the received message.
Collaborative	Vipuĺs Razor (Prakash and Ritter, 2007)	Razor2.pm	Distributed, collaborative, spam detection and filtering network.
	Pyzor (Tobin, 2009)	Pyzor.pm	Collaborative, networked system to detect and block spam using digests of messages.
	Distributed Checksum Clearinghouses (Rhyolite Software, 2000)	DCC.pm	Collaborative, networked system to detect and block spam using checksums of messages.
	DNS-based Blackhole List (RBL) (Levine, 2010)	DNSEval.pm	Lists of server Internet Protocol (IP) addresses from Internet Service Providers (ISPs) whose customers are responsible for the spam and from ISPs whose servers are hijacked for spam relay.
	SpamCop (Cisco Systems, 2010)	SpamCop.pm	Free spam reporting service, allowing recipients of Unsolicited Bulk Email (UBE) and Unsolicited Commercial Email (UCE) to report offenders to the ISPs senders.
Domain-authentication	Sender Policy Framework (SPF) (Wong and Schlitt, 2006)	SPF.pm	Is able to detect message spoofing by verifying sender IP addresses.
	DomainKeys Identified Mail (DKIM) (Allman et al., 2007)	DKIM.pm	DKIM implements sender verification scheme using Public Key Infrastructure (PKI) mechanisms.
RFC2822 structure and syntax	Regular Expressions (REGEX)	MIMEEval.pm	Allows regular expression rules to be written against Multipurpose Internet Mail Extensions (MIME) (Freed & Borenstein, 1996c; Freed & Borestein, 1996a; Freed & Borestein, 1996b; Freed & Klensin, 2005a; Freed & Klensin, 2005b; Moore, 1996) headers in the message.
		MIMEHeader.pm	Performs regular expressions tests against MIME headers.
		URIEval.pm	Checks and evaluates message URI (Uniform Resource Identifier) type.
	Content parsers	BodyEval.pm	Checks the correctness of the message body structure.
		HTMLEval.pm	Checks the structure of HyperText Markup Language (HTML) code embedded inside the message.