Information Sciences 000 (2017) 1–16

Contents lists available at ScienceDirect

Information Sciences

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

Concept drift in e-mail datasets: An empirical study with practical implications

David Ruano-Ordás[a,b,*], Florentino Fdez-Riverola[a,b], José R. Méndez[a,b]

[a] *Department of Computer Science, University of Vigo, ESEI, Campus As Lagoas, Ourense 32004, Spain*
[b] *Centro de Investigaciones Biomédicas (Centro Singular de Investigación de Galicia), Campus Universitario Lagoas-Marcosende, Vigo 36310, Spain*

ARTICLE INFO

Article history:
Received 23 December 2016
Revised 24 October 2017
Accepted 28 October 2017
Available online xxx

Keywords:
Concept drift analysis
Text mining
Spam filtering
E-mail
Classification

ABSTRACT

Internet e-mail service emerged in the late seventies to implement fast message exchanging through computer networks. Network users immediately discovered the value of this service (sometimes for improper purposes such as spamming). As e-mail became indispensable to increase personal productivity, the volume of spam deliveries was constantly growing. With the passage of time, a great number of proposals and tools have emerged to fight against spam. However, the vast majority of them do not properly take into consideration the inner attributes of spam and ham messages such as the noise or the presence of concept drift. In this work, we provide a detailed empirical study of concept drift in the e-mail domain taking into consideration two key aspects: existing types of concept drift and the real class of messages (spam and ham). As a result, our study reveals different weaknesses of multiple e-mail filtering alternatives and other relevant works in this domain and identifies new strategies to develop more accurate filters. Finally, the experimentation carried out in this work has motivated the development of a concept drift analyser tool for the e-mail domain that can be freely downloaded from https://github.com/sing-group/conceptDriftAnalyser.git.

1. Introduction and motivation

On May 3, 1978 the first ever spam e-mail was delivered through ARPAnet. The marketer, Gary Thuerk, was able to send a message to 400 of the 2600 ARPAnet [38] users. This milestone showed the power of Internet as an advertising platform and posed a new challenge: the limits between marketing and spam activities. Since then, competent companies, authorities and research institutions have developed different strategies to fight against spam including such laws as the well-known CAM SPAM Act in the USA [28] and/or diverse software tools such as the earlier Naïve Bayesian filter iFile [39].

Despite all efforts to minimise the impact of spam, the volume of bulk deliveries has been constantly growing [36]. For this reason, wide varieties of techniques have been used to build fully functional anti-spam filters including: (*i*) collaborative schemes; (*ii*) domain authorization methods; (*iii*) machine-learning (ML) approaches; and (*iv*) frameworks to efficiently combine the use of multiple techniques. Included in the first group of alternatives, Razor [37], Pyzor [2] and Distributed Checksum Clearinghouse (DCC) [42] allow sharing information concerning spam messages by using different communica-

[*] Corresponding author at: ESEI: Escuela Superior de Ingeniería Informática. Edificio Politécnico. Campus Universitario As Lagoas s/n, Ourense 32004, Spain.

E-mail addresses: drordas@uvigo.es (D. Ruano-Ordás), riverola@uvigo.es (F. Fdez-Riverola), moncho.mendez@uvigo.es (J.R. Méndez).

https://doi.org/10.1016/j.ins.2017.10.049
0020-0255/© 2017 Elsevier Inc. All rights reserved.

tion schemes. The scientific community has also introduced innovative collaboration schemes such as that proposed by Lay et al. [26]. Moreover, Sender Policy Framework (SPF) [50] and DomainKeys Identified Mail (DKIM) [3] have been designed to provide authorization controls to limit mail deliveries.

Along the same lines, spam filtering has been addressed using different ML approaches including Naïve Bayes [40], Support Vector Machines (SVM) [4], Boosting [8], Artificial Immune Systems (AIS) [16], Case-Based Reasoning (CBR) [10,11], Memory Based Systems [45] or Rough Sets [35]. In connection with these approaches, the work of Guzella and Caminhas [17] provides a detailed summary of ML alternatives used to fight against spam. Finally, some combination frameworks have been recently released with high acclaim including SpamAssassin [5] and Wirebrush4SPAM [33,43,44] (the later has been designed to outperform the throughput achieved by the former). This brief recap of spam filtering methods can be complemented with the survey work published by Blanzieri and Bryl [7].

Despite the high performance (and throughput) achieved by some of above-mentioned spam filtering tools and techniques, a solution able to achieve the greatest level of accuracy (close to 100%) is still required. Moreover, it is a fact that most of the current proposals have not properly considered some intrinsic characteristics of the problem domain together with the various difficulties identified in previous works [10,34]. Specifically, these studies have revealed key issues that should be considered in the anti-spam domain: (*i*) the presence of noise in spam messages; (*ii*) the disjointness between ham and spam concepts; (*iii*) multi-language questions; (*iv*) the existence of concept drift; and (*v*) the complexity of spam filtering software.

Furthermore, and with the goal of hindering the detection of illegal messages, spammers have developed a wide variety of techniques to obfuscate words, such as: (*i*) the inclusion of punctuation marks to replace some characters of tokens (V¡agra); (*ii*) the use of fake HTML tags inside words that are not rendered by HTML viewers (v < xx > iagra); or (*iii*) varying the spacing of the characters that compose the words ("b u y v i a g r a"). In addition, the disjointness of spam and ham concepts should be also kept in consideration. In fact, ham (or spam) can be defined as a set of subject matters (or topics) that are (or not) of interest to a given e-mail account holder. These subject matters included in each concept (i.e. spam or ham) are the same regardless of the specific language of the message (multi-language scenario). Moreover, due to unpredictable or hard to identify variations (such as in user life/needs, seasons, fashion or the general interest in particular goods), these sets (spam and ham definitions) may change over time. These alterations on spam and ham concepts are known as concept drift, and their proper management by existing filtering techniques becomes cumbersome, especially due to their increasing complexity and the need to deal with the recognition of useful words in obfuscated spam messages.

In this context, it is obvious that the inherent particularities of these problems should be correctly handled by next generation filtering software in order to bring results close to a theoretically perfect solution. Specifically, we found that concept drift is one of the hardest challenges that exist nowadays. However, to the best of our knowledge, the concept drift problem has not been thoroughly analysed in the particular case of the spam-filtering domain. In this work, we present a comprehensible empirical study of concept drift to discover its origin, types, and undesired effects in the context of e-mail classification. This work contributes: (*i*) a method to discover different forms of concept drift in e-mail that could be applied to other target problems, (*ii*) a reference implementation for it, (*iii*) the results of analysing a large dataset using it (which revealed the existence of all forms of concept drift), (*iv*) a drawback detected in a SDAI methodology designed to evaluate the effects of concept drift in spam filters [34] and (*v*) the evidence that current research on handling concept drift is not enough to deal with the particular forms of concept drift found in e-mail.

While this section has introduced the main motivation for this study, the remaining sections are structured as follows: Section 2 provides a survey of related work on concept drift. Section 3 presents a detailed description of experiments carried out and the obtained results. Section 4 discusses in detail the main outcomes derived from our experimental analysis and their practical implications. Finally, Section 5 summarises essential conclusions and identifies future research lines derived from this work.

2. State of the art regarding concept drift

Concept drift occurs primarily in online supervised learning scenarios in which the relation between the input data and the target variable changes over time. This phenomenon has a great presence in most areas of real life such as: (*i*) monitoring and control of systems [31]; (*ii*) decision making [18,47]; (*iii*) artificial intelligence (AI) systems and robotics [19]; or (*iv*) personal assistance and information [24]. Specifically, the problematics associated with the occurrence of concept drift were made evident in results achieved by early ML applications [46]. In particular, this work discussed how different changing concepts had a negative influence on the effectiveness of an incremental learning system (STAGGER). To cope with this overall situation, many authors have introduced different approaches to tackle concept drift in supervised classification [1,24,49], time series prediction [18,31] and clustering scenarios [23,27,47].

Regarding supervised environments, the authors in [39] proposed a learning and forgetting framework (FLORA) based on the following key ideas: (*i*) the model knowledge is represented through attribute-value logic rules in the form "$colour = white \wedge temperature = low$" for positive, negative and general instances, (*ii*) the model only represents the knowledge about a fixed set of current trusted examples (window); (*iii*) storing past model versions and reusing them when a previous context reappears; and (*iv*) Window Adjustment Heuristic (WAH) heuristic is able to reduce window size when concept drift is detected (accuracy lost detection). Additionally, the authors in [1] present the well-known IB3 algorithm, an instance-based approach that improves the performance of Nearest Neighbour (NN) procedure. The main differences regard-