Available online at www.sciencedirect.com
ScienceDirect
Knowledge-Based Systems 20 (2007) 249–254
Knowledge-Based SYSTEMS
www.elsevier.com/locate/knosys

An empirical study of three machine learning methods for spam filtering

Chih-Chin Lai *

Department of Computer Science and Information Engineering, National University of Tainan, Taiwan 700, Taiwan

Received 10 July 2005; accepted 1 May 2006
Available online 5 September 2006

Abstract

The increasing volumes of unsolicited bulk e-mail (also known as spam) are bringing more annoyance for most Internet users. Using a classifier based on a specific machine-learning technique to automatically filter out spam e-mail has drawn many researchers' attention. This paper is a comparative study the performance of three commonly used machine learning methods in spam filtering. On the other hand, we try to integrate two spam filtering methods to obtain better performance. A set of systematic experiments has been conducted with these methods which are applied to different parts of an e-mail. Experiments show that using the header only can achieve satisfactory performance, and the idea of integrating disparate methods is a promising way to fight spam.

Keywords: Spam filtering; Machine learning

1. Introduction

In recent years, e-mails have become a common and important medium of communication for most Internet users. However, spam, also known as unsolicited commercial/bulk e-mail, is a bane of e-mail communication. A study estimated that over 70% of today's business e-mails are spam [1]; therefore, there are many serious problems associated with growing volumes of spam such as filling users' mailboxes, engulfing important personal mail, wasting storage space and communication bandwidth, and consuming users' time to delete all spam mails. Spam mails vary significantly in content and they roughly belong to the following categories: money making scams, fat loss, improve business, sexually explicit, make friends, service provider advertisement, etc., [13]. One example of a spam mail is shown as Fig. 1.

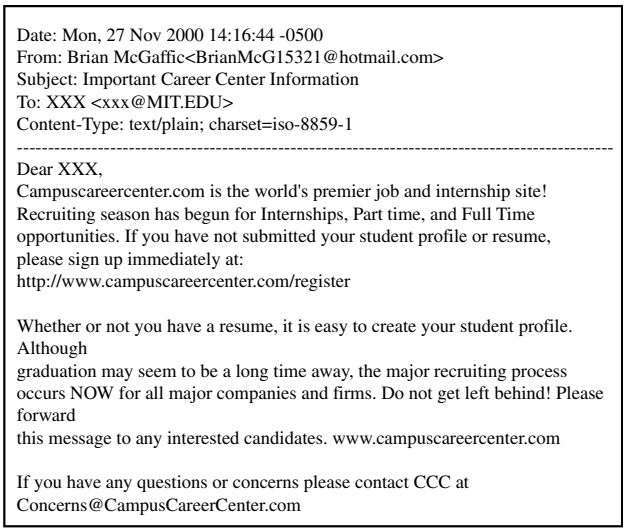
Several solutions have been proposed to overcome the spam problem. Among the proposed methods, much interest has focused on the machine learning techniques in spam filtering. They include rule learning [4,6], Naïve Bayes [2,9], decision trees [5], support vector machines [7,8,16] or combinations of different learners [10]. The common concept of these approaches is that they do not require specifying any rules explicitly to filter out spam mails. Instead, a set of training samples (pre-classified e-mails) is needed. A specific machine-learning technique is then used to "learn" and "produce" the classification model from this data. From the machine learning viewpoint, spam filtering based on the textual content of e-mail can be viewed as a special case of text categorization, with the categories being spam or non-spam [8].

Sahami et al. [9] employed Bayesian classification technique to filter junk e-mails. By making use of the extensible framework of Bayesian modeling, they can not only employ traditional document classification techniques based on the text of e-mail, but they can also easily incorporate domain knowledge to aim at filtering spam e-mails.

Androutsopoulos et al. [2–4] presented a series of papers that extended the Naïve Bayes (*NB*) filter proposed by Sahami et al. [9], by investigating the effect of different number of features and training-set sizes on the filter's performance. Meanwhile, they compared the performance of *NB* to a memory-based approach, and they found both above-mentioned methods clearly outperform a typical keyword-based filter.

* Tel.: +886 62606123; fax: +886 62606125.
E-mail address: cclai@mail.nutn.edu.tw

0950-7051/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.knosys.2006.05.016


Date: Mon, 27 Nov 2000 14:16:44 -0500
From: Brian McGaffic<BrianMcG15321@hotmail.com>
Subject: Important Career Center Information
To: XXX <xxx@MIT.EDU>
Content-Type: text/plain; charset=iso-8859-1

Dear XXX,
Campuscareercenter.com is the world's premier job and internship site! Recruiting season has begun for Internships, Part time, and Full Time opportunities. If you have not submitted your student profile or resume, please sign up immediately at:
http://www.campuscareercenter.com/register

Whether or not you have a resume, it is easy to create your student profile. Although
graduation may seem to be a long time away, the major recruiting process occurs NOW for all major companies and firms. Do not get left behind! Please forward
this message to any interested candidates. www.campuscareercenter.com

If you have any questions or concerns please contact CCC at Concerns@CampusCareerCenter.com

Fig. 1. An example of a spam mail.

Drucker et al. [7] used support vector machine (*SVM*) for classifying e-mails according to their contents and compared its performance with Ripper, Rocchio, and boosting decision trees. They concluded that boosting trees and *SVM* had acceptable test performance in terms of accuracy and speed. However, the training time of boosting trees is inordinately long. Woitaszek et al. [17] utilized a simple *SVM* and a personalized dictionary to identify commercial electronic mail. The SVM-based mail classification system was implemented as an add-in for *Microsoft Outlook XP*, allowing desktop users to quickly identify unsolicited e-mail.

Although it is a popular topic in machine learning, very few approaches using instance-based nearest neighbor techniques are presented for spam filtering. Trudgian and Yang [14] examined the performance of the *kd*-tree nearest neighbor algorithm for word based spam mail classification and compared it to other common methods.

Several attempts have been made to evaluate the performance of machine-learning methods on spam filtering task; however, these studies focused on features which extracted from message body only. Here we study different parts of an e-mail that can be exploited to improve the categorization capability, by giving experimental comparisons of three respective machine learning algorithms. These techniques are Naïve Bayes (*NB*), *k*-nearest neighbor (*k-NN*), and support vector machines (*SVMs*). We considered the following four combinations of an e-mail message: all (*A*), header (*H*), subject (*S*) and body (*B*). The above-mentioned three methods with these features are compared to help evaluate the relative merits of these algorithms. In addition to using a single method for spam filtering, we adopted an integrated approach which considered two different methods to anti-spam filtering and evaluated its performance.

The rest of this paper is organized as follows. Section 2 gives a brief review of three machine learning algorithms and details of the integrated approach. Section 3 provides the considered features and experimental results designed to evaluate the performance of different experimental settings are presented in Section 4. The conclusions and directions for future works are summarized in Section 5.

2. Machine learning methods and proposed combined apparoch

2.1. Naïve Bayes

The Naïve Bayes (*NB*) classifier is a probability-based approach. The basic concept of it is to find whether an e-mail is spam or not by looking at which words are found in the message and which words are absent from it. This approach begins by studying the content of a large collection of e-mails which have already been classified as spam or legitimate. Then when a new e-mail comes into some user's mailbox, the information gleaned from the "training set" is used to compute the probability that the e-mail is spam or not given the words appearing in the e-mail.

Given a feature vector $\vec{\mathbf{x}} = \{x_1, x_2, \ldots, x_n\}$ of an e-mail, where are the values of attributes $X_1, \ldots, X_n$, and n is the number of attributes in the corpus. Here, each attribute can be viewed as a particular word occurring or not. Let c denote the category to be predicted, i.e., $c \in \{\text{spam, legitimate}\}$, by Bayes law the probability that $\vec{\mathbf{x}}$ belongs to c is as given in

$$P(c|\vec{\mathbf{x}}) = \frac{P(c) \cdot P(\vec{\mathbf{x}}|c)}{P(\vec{\mathbf{x}})}, \tag{1}$$

where $P(\vec{\mathbf{x}})$ denotes the a-priori probability of a randomly picked e-mail has vector $\vec{\mathbf{x}}$ as its representation, $P(c)$ is also the a prior probability of class c (that is, the probability that a randomly picked e-mail is from that class), and $P(\vec{\mathbf{x}}|c)$ denotes the probability of a randomly picked e-mail with class c has $\vec{\mathbf{x}}$ as its representation. Androutsopoulos et al. [2] notes that the probability $P(\vec{\mathbf{x}}|c)$ is almost impossible to calculate because the fact that the number of possible vectors $\vec{\mathbf{x}}$ is too high. In order to alleviate this problem, it is common to make the assumption that the components of the vector $\vec{\mathbf{x}}$ are independent in the class. Thus, $P(\vec{\mathbf{x}}|c)$ can be decomposed to

$$P(\vec{\mathbf{x}}|c) = \prod_{i=1}^{n} P(x_i|c) \tag{2}$$

So, using the *NB* classifier for spam filtering can be computed as

$$C_{NB} = \arg\ \max_{c \in \{spam, legitmiate\}} P(c) \prod_i P(x_i|c) \tag{3}$$

2.2. K-nearest neighbor

The most basic instance-based method is the *k*-nearest neighbor (*k-NN*) algorithm. It is a very simple method to classify documents and to show very good performance