Expert Systems with Applications 37 (2010) 2197–2201

Contents lists available at ScienceDirect

Expert Systems with Applications

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

An ensemble approach applied to classify spam e-mails

Kuo-Ching Ying [a], Shih-Wei Lin [b], Zne-Jung Lee [c,*], Yen-Tim Lin [c]

[a] Department of Industrial Engineering and Management, National Taipei University of Technology, Taipei, Taiwan
[b] Department of Information Management, Chang Gung University, No. 259, Wen-Hwa 1st Road, Tao-Yuan, Taiwan
[c] Department of Information Management, Huafan University, No. 1, Huafan Rd., Shihding Township, Taipei County 22301, Taiwan

ARTICLE INFO

Keywords:
E-mail
Spam
Ensemble
Decision tree
Back-propagation network
Support vector machine

ABSTRACT

Spam e-mails, known as unsolicited e-mail messages, have become an increasing problem for information security. The intrusion of spam e-mails persecute the users and waste the network resources. Traditionally, machine learning and statistical filtering systems are used to filter out spam e-mails. However, there is no unique method can be successfully applied to classify spam e-mails. It is necessary to apply multiple approaches to detect spam and effectively filter out the increasing volumes of spam e-mails. In this paper, an ensemble approach, based on decision tree, support vector machine and back-propagation network, is applied to classify spam e-mails. The proposed approach is based on the characteristics of the spam e-mails. The spam e-mails are categorized into 14 features and then the ensemble approach is performed to classify them. From simulation results, the proposed ensemble approach outperforms other approaches for two test datasets.

1. Introduction

Under the influence of network-globalization, the time and distance of communication are decreased by e-mail. The conveniences of e-mail are more and more concerned by the users. The spam e-mails, unsolicited commercial e-mails (UCE) or unsolicited bulk e-mails (UBE), are sent to receivers' mailbox without their permission, especially the large amount sending and annoying e-mails. The overabundant of spam e-mails not only affects the network bandwidth, but also becomes the hotbeds of malicious programs in information security (Brad, 2002).

The spam filtering systems nowadays are based on the users to classify the e-mails from keywords or word-automatic rules for analyzing the characteristics of e-mails (Androutsopoulos, Koutsias, Chandrinos, Paliouras, & Spyropoulos, 2000; Elisabeth, Judy, & McCreath, 2001; Irena, Josiah, James, & Jason, 2007; Mooney, Bennett, & Roy, 1998; Provost, 2001; Tseng & Wu, 2003; Yue, Abraham, Chi, Hao, & Mo, 2007). However, it is a time consuming and error-prone process for the above methods (Irena et al., 2007; Zorkadis, Karras, & Panayotou, 2005). Bayesian approaches are widely used statistical filtering systems, but they need a lot of training samples to provide a degree of confidence for classification. Machine learning methods are also used with good results in the context of text classification, but the results are still limited when the spam environment is highly variable in the real world (Irena et al., 2007). Recently, the ensemble approach is to combine a set of redundant methods that each method can provide a solution to the same task. The main advantage of ensemble approach is that of improving the generalization ability and guarding against the failure of individual methods (Amanda, 1999; Irena et al., 2007). In this paper, an ensemble approach based on decision tree (DT), support vector machine (SVM) and back-propagation network (BPN) is proposed for spam filtering.

The remainder of this paper is organized as follows. The proposed ensemble approach is based on DT, SVM and BPN. Section 2 first introduces DT, SVM and BPN. Then, Section 3 introduces the proposed ensemble approach for classifying the spam e-mails. Experimental results are compared with those of existing approaches in Section 4. Conclusions are finally drawn in Section 5.

2. The introduction of BPN, SVM and DT

The proposed ensemble approach is based on BPN, SVM, and DT. In this section, the brief descriptions of BPN, SVM and DT are introduced.

2.1. The brief description of BPN

BPN is the most widely used neural network model, and its network behavior is determined on the basis of input-output learning pairs (x_j, d_j) where $j = 1, \ldots, m$ (Hong, May, & Park, 2003; Khaw, Lim, & Lim, 1995; Lezoray & Cardot, 2001; Liao, Fang, & Nuttle, 2004). A three-layer neural network is shown in Fig. 1 for an example. BPN trains a feedfoward multilayer neural network by searching for a set of weight that minimizes the total training error. BPN can be

* Corresponding author. Tel./fax: +886 2 26632102.
E-mail address: johnlee@hfu.edu.tw (Z.-J. Lee).

0957-4174/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2009.07.080

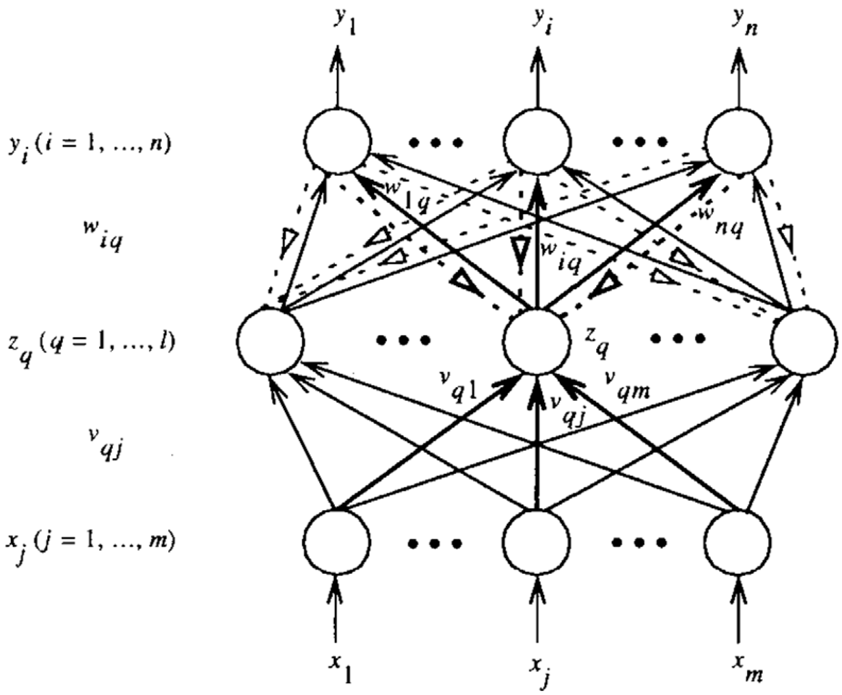

Fig. 1. A three-layer neural network.

divided into two stages, the message forward propagation and the error backward propagation (Wang & Huang, 2007). In the forward propagation stage, it is to forward propagate the input pattern to produce the current output. The net input to each neuron in the hidden and output layers is computed as a linear combination of its inputs. The inputs to the neuron are, in fact, the outputs of the neurons connected to it in the previous layer. To compute the net input of the neuron, each input connected to the neuron is multiplied by its corresponding weight, and then summed. For a neuron q in the hidden layer, the net input $net_q = \sum_{j=1}^{m} v_{qj}x_j$ and its output $z_q = a(\sum_{j=1}^{m} v_{qj}x_j)$, where $a(\sum_{j=1}^{m} v_{qj}x_j)$ is called the activation function and v_{qj} is the weight between input x_j and neuron q. For a neuron i in the output layer $net_i = \sum_{q=1}^{l} w_{qj}z_q = \sum_{q=1}^{l} w_{iq}a\left(\sum_{j=1}^{m} v_{qj}x_j\right)$, its output is $y_i = a(net_i) = a\left(\sum_{q=1}^{l} w_{iq}a\left(\sum_{j=1}^{m} v_{qj}x_j\right)\right)$. In the backward propagation stage, it is to backward propagate the error to update the weights in a gradient descent fashion and the weights are updated to reflect the propagated error. For the weights in the hidden-to-output connections are updated as $\Delta w_{iq} = \eta(d_i - y_i)a'(net_i)z_q$, where η is called the learning constant. For the weights in the input-to-hidden connections are updated as $\Delta v_{qj} = \eta\sum_{i=1}^{n}((d_i - y_i)a'(net_i)w_{iq})a'(net_q)x_j$. The error can be computed by backward propagate error from its following layer and this is why this learning algorithm is called the backward propagation (Sexton, McMurtrey, & Cleavenger, 2006; Wang, Wang, Huang, Nguyen, & Krishnakumar, 2008; Yan, Jiang, Zheng, Peng, & Li, 2006).

2.2. The brief description of SVM

SVM introduced by Vapnik and co-workers is a learning system that uses a hypothesis space of linear function in a high dimensional feature space (Vapnik, 1998). SVM has been successfully applied in a wide variety of fields (Burges, 1998; Cristiamini & Shawe-Taylor, 2000; Drucker, Donghui, & Vapnik, 1999; Lee, 2008; Pontil & Verri, 1998). Let $(x_1, y_1), \ldots, (x_m, y_m) \epsilon X \times \{\pm 1\}$ be a set of training data, X represents some nonempty sets from which the input x_i is taken, referred to as the domain, and y_i is called the label (Mülle, Mika, Rätsch, Tsuda, & Schölkopf, 2001; SchÖlkopf & Smola, 2002). Suppose a set of input $x_1, \ldots, x_m \epsilon H$ is given and H is a dot product space. Consider the hyper-plane in H can be written as

$$\{x \in H | \langle x_i \cdot w \rangle + b = 0\}, \text{ where } w \in H, \quad b \in R, \ i = 1, \ldots, m. \tag{1}$$

where w is a vector orthogonal to the hyper-plane. A separating hyper-plane can be derived from Eq. (2).

$$\underset{w \in H, b \in R}{Min} \ \tau(w) = \frac{1}{2} w^T w \tag{2}$$

Subject to:

$$y_i(\langle x_i \cdot w \rangle + b) \ \geqslant \ 1, \text{ for all } i = 1, \ldots, m. \tag{3}$$

Combining Lagrange's polynomial and soft margin with Eq. (3) produce the following equation:

$$\underset{\alpha \in R^n}{Max} \ W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j \langle x_i \cdot x_j \rangle \tag{4}$$

Subject to:

$$\begin{aligned} &0 \le \alpha_i \le C, \quad i = 1, \ldots, m \\ &\text{and } \sum_{i=1}^{m} \alpha_i y_i = 0 \end{aligned} \tag{5}$$

where $\alpha_i \geqslant 0$ denotes the Lagrange multiplier and C is a parameter of the cost of penalty. When considering expanding the constraints of Eq. (5), the determinant function of the hyper-plane is written as follows:

$$f(x) = \text{sgn}\left(\sum_{i=1}^{m} y_i \alpha_i \langle x, x_i \rangle + b\right) = 0 \tag{6}$$

In practice, SVM will map the data into a higher dimensional space for classification if the data cannot be classified explicitly in the current dimensional space. The feature space vectors x_i, x_j are constructed in terms of the kernel k, evaluated on input patterns x_i, x_j where $k(x_i, x_j) = \langle x_i, x_j \rangle$. By constructing the feature space, $k(x_i, x_j) = \langle x_i, x_j \rangle$ can be substituted into Eq. (4) and then it can be