J. Appl. Environ. Biol. Sci., 5(7S)381-386, 2015

ISSN: 2090-4274
Journal of Applied Environmental and Biological Sciences
www.textroad.com

Improving anti-spam filtering, based on Naive Bayesian and neural networks in multi-agent filters

Adeleh Jafar gholibeik[1] , Mohammad amin Edalatirad[2]

[1] Department of Computer Science, Shush Branch, Islamic Azad University, Shush, Iran
[2] Department of Computer Science, Dezful Branch, Islamic Azad University, Dezful, Iran

Received: March 19, 2015
Accepted: May 2, 2015

ABSTRACT

This article will discuss offering pre-processing filter in the stage before processing of filter content. One alternative has been proposed by which a Naive Bayesian classification automatically have been taught to detect spam messages. In this study, a new version has been provided based on Naive Bayesian approach by multi-agent filters. The advantage of simultaneous using of the user interests, Keywords and content-based valuation view is based on issue. Following the above version will be developed and we identify spam using MLP neural network algorithm.
KEYWORDS: spam, multi-agent filter, email, naive bayesian, neural network.

1. INTRODUCTION

The problem of unwanted e-mails known as spam has become a serious problem today, so that 57-80% of the volume of e-mail is spam. Spam creates several problems rather; spam causes traffic and destroys storage space and computing power. Spam causes the users a lot of time to separate and remove wasted mails and also hurt users and causes insecurity, finally spam causes legal problems such as pornography, pyramid schemes and economic fraud as well as Phishing.

Now, according to studies conducted by Pinon and Stuck about different types of tools and techniques of anti-spam, filtering is the most common method to protect against spam [1]. The research shows that the spam filtering is a method based on machine learning that is a very good method among the methods based on learning in terms of efficiency in spam capture and the best value in terms of the lowest false positives. Filtering method is using of Naive Bayesian method which has had a better performance than similar methods such as decision trees and Svm (Support Vector Machine). A study in 1997 found that spam messages almost form 10 percent of received messages on a shared network [2]. It seems that the situation is getting worse. Legislative measures against spam are gradually being concluded, but despite this they have had little and very limited effect. Currently anti-spam filters are software tools that attempt to automatically block spam messages.

2. Multi-Agent filters

Since spam has several different aspects with different characteristics, therefore, methods using several agents were suggested. An important feature of this method is that the user can if there is only one agent in name of A_i , when receives x e-mail, based on value of A_i, the possibility that x is spam or not is shown by $PS_i(x)$. If the possibility of $PS_i(x)$, has a value that cannot be decided on the basis of a single agent, then sends it to the section of agent group identification. In group state A_i , sends x e-mail to other agents in multi-agent system AC and other AC agents specify the possibility that X is spam by $PS_i(x)$ and send it to source agent.

Agent of source A_i makes the final diagnosis based on other agents. In order to classify the ability of any agent, if decision of spam is right, it receives a positive score and otherwise it receives a negative score. Methodology In this method, every agent has the ability to detect, so they are called heterogeneous agents. To learn any agent we use a number of methods such as: Naive Bayesian, decision tree and neural network. Learning of agent is done in two layers: the first layer, based on prior knowledge of agent, without the participation and control of the user. In second layer, according to the user's monitoring, with the knowledge of characteristics of spam and the real e-mail, aids agent to learn more. In fact, every agent used the smart techniques for the diagnosis.

2.1. Naïve Bayesian method

This algorithm is the simplest and most widely used algorithms, and of machine learning methods for text classification which arises from the theory of Bayese decision[3]. Thus this method is to use a training set (which includes a number e-mails that have spam tag or correct mail) to calculate probability of being spam or being correct of any words (features) in the messages. So when a new letter to the user's mailbox by using the probability of being spam or being correct of words

Corresponding Author: Adeleh Jafar gholibeik, Department of Computer Science, Shush Branch, Islamic Azad University, Shush, Iran

in it, the possibility that it is a spam or a proper one would be calculated. By using the Bayese principle, class occurrence probability is expressed using the vector of feature x. To display an e-mail, a feature vector x = {x_1, x_2, … x_n} is used.

n: number of features (words) contained in that letter.

x_i: obtained probability value (via the training set) for class C_i.

In order to use Naive Bayesian principle, it is assumed that the properties of a message independent of each other occur in a letter. However, assuming that feature vector of a mail has features x_n… , x_2, x_1 the following formula (1)is resulted.

$$P(CIX) = \frac{\prod_{i=1}^{n} P(X_i IC) * P(C)}{\prod_{j=1}^{n} P(X_i)} \quad (1)$$

So that o(x_i|c): perhaps feature x_i occurs in class C

P(C): perhaps class C occurs in set of mails.

$P(x_i)$: perhaps word x_i occurs in message

Output of the class that its probability is more.

2.2. Neural Networks

Neural networks is of the most widely used and practical methods of modelling of complex and great problems (including hundreds variable). Neural networks can be used to classify things (the output of a class), or regression issues (the output of a numeric value). The neural network includes an input layer which each node in this layer is equivalent to one of the predictor variables. Nodes in the middle layer are connected in number of nodes in the hidden layer. Each input node is connected to all nodes in the hidden layer. Nodes in the hidden layer can be connected to other nodes in a hidden layer or can be connected output layer. Output layer contains one or more output variables. Each edge between two nodes has a weight. The weights are used in the calculation of intermediate layers and how to use them in such a way that each node in the middle layers (layers other than the first layer) has some input from various edges, which, as mentioned earlier, each is with a specific weight. Each node in the middle layer multiplies value of each input by its relative edge, and the outcome of these are added together and then applies a predetermined function (activation function) on this outcome and gives the result as the output to nodes of the next layer. Edge weights are unknown parameters that are determined by the training function and training data given to the data system. The number of nodes and the number of hidden layers and how to connect the nodes to each other will specify the architecture (topology) of neural network.

User or software that designs neural networks should specify the number of nodes, number of hidden layers, the activation function and edge weight constraints. Neural networks are powerful estimators that do estimation as well as other techniques and sometimes better. The major disadvantage of this method is to spend a lot of time for parameter selection and training of the network. An example of used neural network is the type of nonlinear feedback with the sigmoid activation function:

$$f(x) = \frac{1}{1+e^{-x}} \quad (2)$$

This activation function produces output in the range of (0, 1). According to the figure below shows a network with an outlet for every word in the dictionary, is formed of a hidden layer and an output neuron. Input and hidden layer include a bias neuron with constant activity 1[4, 5, 6].

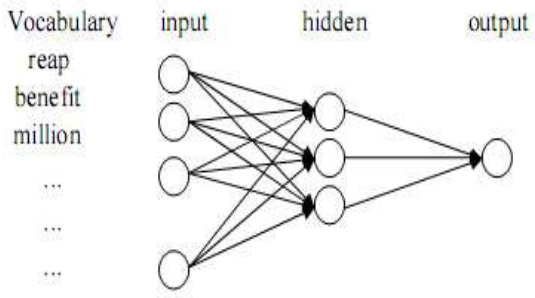


Fig. 1. the structure of neural networks

When an e-mail is introduced to the network, inputs for the words that are found in a mail will be 1. And other inputs 0. If the output value is more than 0/5 the electronic mail is identified as spam. When the desired output network is being trained the correct letters are taken 0/1 and for spam 0/1. The number of correct letters and spam that are introduced to the network should have a little difference, because if the number of instances of a class is more than the other class the method tends to that class and tries to class the letters of that class better. A learning algorithm with descending slope that uses rear-facing publication is used to optimize the weights of the network.

2.3. The proposed method of multi-agent filters in artificial neural networks MLP

Artificial neural networks are those of dynamical systems that with processing on the experimental data transfer knowledge or code behind the data to the network structure. A one neuron networks have some limitations, including the inability to mapping of nonlinear functions. In the meantime, Artificial Neural Networks Multilayer Perceptron (MLP) has ability of approximation and implementation of nonlinear functions. Perceptron network model is as a feed forward network with back-propagation learning law n. In the feed forward network input vector s applied to first layer and its impacts are through intermediate layers to the output layer, in this route, network parameters are considered fixed with no change. In the second route that follows the law BP, the network parameters MLP are regulated and changed. This change is done according to the law of error correction [5, 6].

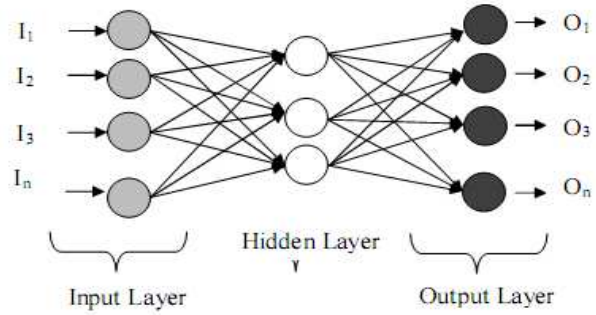


Fig. 2. Multilayer Perceptron network

Amount of error is equal to the difference between the desired response and the actual response of network. After calculating the error amount, in back track and through the network layers would be the distributed in network. Since error distribution occurs against n the opposite direction of weight communication error the term of propagation is said to it. MLP network is formed of an input layer, a number intermediate layers and an output layer.

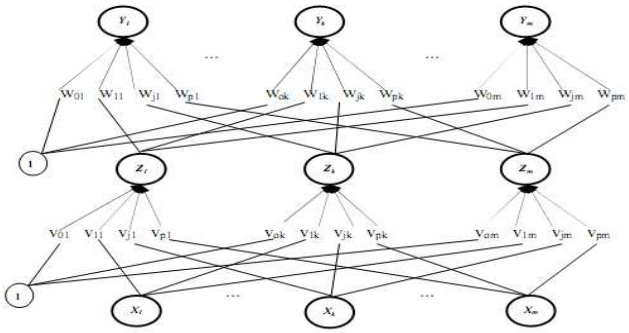


Fig. 3. Shows the structure of an MLP network

2.3.1. MLP neural network learning algorithm

List of terms used in the algorithm[7]:

Input vector: $X=(x_1 \ldots\ldots x_i, \ldots\ldots x_n)$

Output vector desired : $t=(t_1 \ldots\ldots t_i, \ldots\ldots t_n)$

δ_k: a percentage of error to correct weight W_{jk} that in fact is used to distribute error from output layer Y_K to previous layers of Z_i (hidden layer).

δ_j: a percentage of error to correct weight V_{ij} . this error is transferred from middle layer Z_j to input layer X_i.

α: learning rate ;X_i: input unit if i; Z_j: hidden unit of j

input network of Z_j is shown by z_in_j.

$$Z\_in_j = v_{oj} + \sum_{i=1}^{m} x_i v_{oj} \quad (3)$$

Output signal Z_j is shown by z_j

$$z_j = f(z\_in_j) \quad (4)$$

Y_k: output unit of K

Input network Y_k is shown by y_in_k

$$y\_in_k = w_{0k} + \sum_{j=1}^{p} z_j w_{jk} \quad (5)$$

Output signal Y_k is specified by y_k

$$y_k = f(y\_in_k) \tag{6}$$

2.3.2. training algorithm

Network training is performed by propagation in three stages:

1. Feed forward propagation of input learning patterns
2. Feedback propagation to the obtained error
3. Set Weights

Training algorithm runs as follows:

Step 0: Determine the initial weights

Stage 1: until the stop conditions are wrong, do steps 2 to 9.

Step 2: For each pair of corresponding input and output do steps 3 to 8.

2.3.3. feed forward networks

Step 3: Every Input unit (X_i , i=1,….,n) receives input signal X_i and sends it to all of the higher layers units (hidden units).

Step 4: every hidden layer (Z_j , j = 1, …, p) calculates the total of its input signals .

$$Z\_in_j = v_{oj} + \sum_{i=1}^{m} x_i v_{oj} \tag{7}$$

Trigger function is used to calculate the output signal and send it to the other layers units.

$$z_j = f(z\_in_j) \tag{8}$$

- step 5: every output signal (Y_k , k = 1, ….,m) calculates total of its input weighted signals.

$$y\_in_k = w_{0k} + \sum_{j=1}^{p} z_j w_{jk} \tag{9}$$

And uses its active function to calculate its output signal.

$$y_k = f(y\_in_k) \tag{10}$$

2.3.4. Error propagation error

step 6: every output unit Y_k receives the output of a desired pattern corresponding to the input training pattern that calculates δ_k according to obtained and desired output.

$$\delta_k = (t_k - y_k) f'(y\_in_k) \tag{11}$$

$$\begin{cases} \Delta w_{0k} = \alpha \delta_k \\ \Delta w_{jk} = \alpha \delta_k z_j \end{cases} \tag{12}$$

And sends δk to underlying layers.

Step 7: every hidden layer Z_j calculates total of its input delta.

$$\delta\_in_j = \sum_{k=1}^{m} \delta_k w_{jk} \tag{13}$$

And multiplies result by trigger function derivative to obtain error information term:

$$\delta_j = \delta\_in_j f'(z\_in_j) \tag{14}$$

And from δ_j we measure the values ΔV_{ij} and ΔV_{0j} that are later needed to correct V_{0j} and v_{ij} are measured.

$$S \to L$$

2.3.4. correction weights and biases

Step 8: every output unit Y_k corrects its bias and weights (j=0,…p).

$$w_{jk}(new) = w_{jk}(old) + \Delta w_{jk} \tag{15}$$

Every hidden unit Z_j corrects its bias and weights as follows:

$$v_{ij}(new) = v_{ij}(old) + \Delta v_{ij} \tag{16}$$

Step 9: finally we check the stop conditions.

3. Implementation and experimental results of filter multi-agent method in neural network

3.1. Simulation

In order to obtain an optimum structure to understand the behavior it is necessary to investigate neural network error for the number of different repetitions, a variety of trigger functions and different layers. The network with the least square error of Square root will be the desired network.

3.2. Optimized architecture

As mentioned previously we considered input number of the network according to properties as equal to the number of words in the desired dictionary and the number of output 2. The number of middle layers was obtained using trial and error method. That is, the number of 1 to 3 hidden layer and 2 to 20 neurons for each layer have been tested that at this point two trigger functions are considered. The following table, for example shows root mean square error of a number of done tests. The best architecture has the lowest root mean square error. At this point, frequencies is also of particular importance because each structure shows different behavior for the number of different repetitions. Figure (4) shows root mean square error (RMSE) of any structure for the number of repetitions.

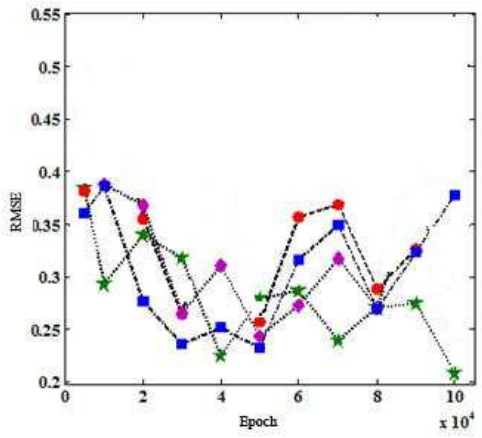
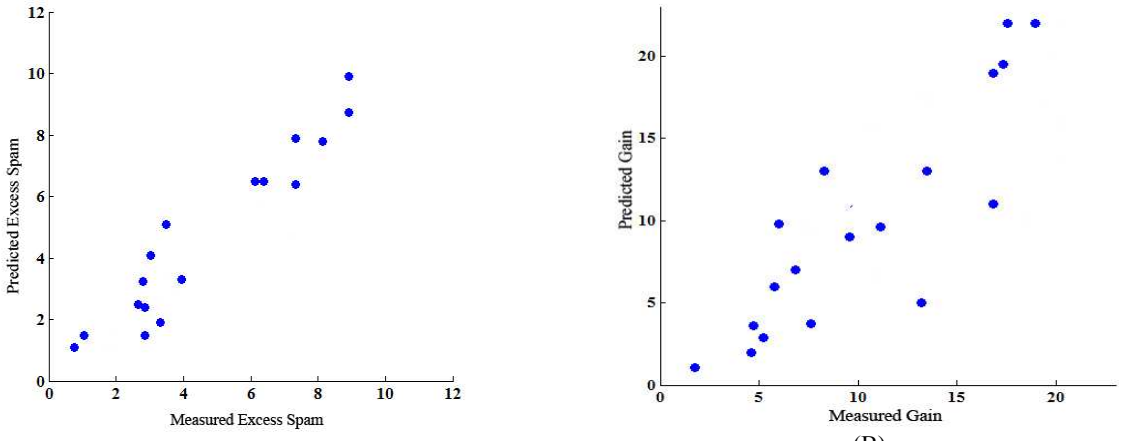


Fig. 4. Error of (RMSE) of every structure per number of different repeat

3.3. Forecast of Network

After defining the network architecture in previous section, in order to assess and compare the proposed methods with real data, we applied the test series to the network. Figure (5) (A) shows the comparison (B) of real and proposed interest. As you can see, the horizontal axis is allocated to network response and vertical axis to the predicted value. Distribution of points around the first bisector indicates how network behaves. Proximity of points to the bisector means network closer response to expected value.

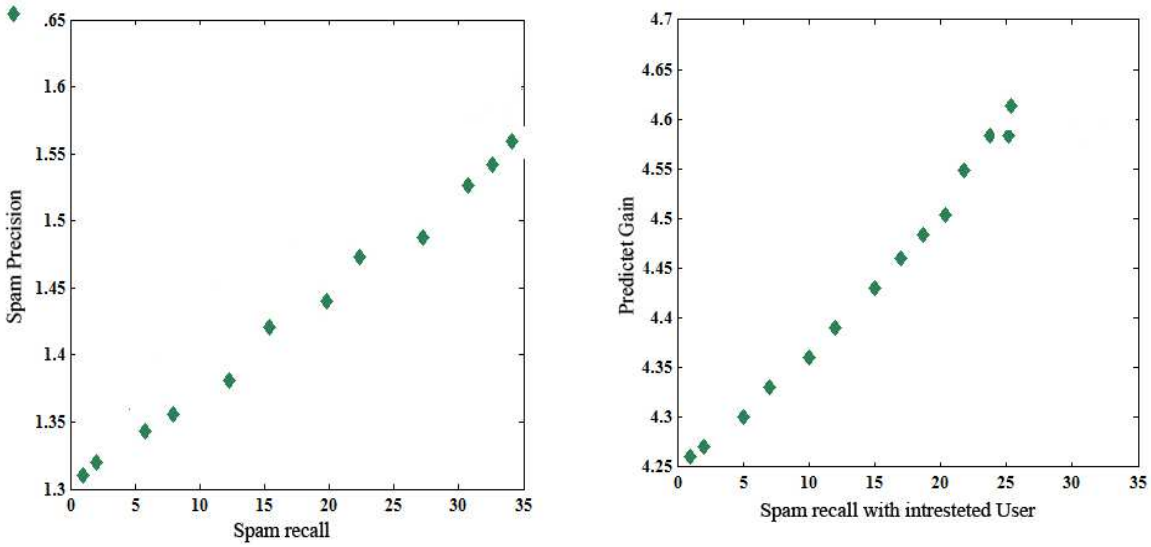


(A) (B)

Fig. 5. Comparison(A) , Real interest and recommended (B)

3.4 Evaluation of Network

To be able to show the trained neural network, the output of Figure (6) shows the user's interests. Testing is by subject, content and combination that indicates better performance.



(A) (B)

Fig. 6. Combining of methods and the user's interests (A) the user's interests interest (B)

Table 1. data of mass emails published in public way

5,000 private emails	Detect spam based on subject	Detect spam based on sender	Detect spam based on content	Detect spam based on user interests
Subject + user interests	61/09%	52/86%	72/60%	88/50%
Subject + transmitter	61/09%	52/60%	71/82%	88/50%
Transmitter + user's interests	61/09%	52/60%	72/60%	98/18%
Subject + transmitter + user's interests	61/09%	52/60%	8/60%	99/49%

4. Conclusion

As was observed, adding a pre-processing stage for processing e-mail content to users using user categories to categories based on the content and theme and the sender and the combination of three methods based on user interests, determining the positive and negative rate of false would improve accuracy of statistical filters in the processing stage and also would cause more speed and better function of these filters in processing stage. These calculations would cause that in an electronic message server users' interests are considered in a category as the threshold and for this reason e-mails that users of this group have not shown interest to see them are considered as spam that this will result in no need to send the mail for processing at a later stage.

REFERENCES

1. Siponen, M., and Stucke, C, Effective antispam strategies in companies: An international study. In Proceedings of HICSS '06,2006, vol 6, PP 245-252.

2. Androutsopoulos, I., Karkaletsis, V., Sakkis, G., Spyropoulos, C., and Stamatopoulos, P., Paliouras, G. Learning to filter spam e-mail: A comparison of a naive bayesian and a memorybased approach. In H. Zaragoza, P. Gallinari, and M. Rajman, editors, Proceedings of the Workshop on Machine Learning and Textual Information Access, 4th European Conference on Principles and Practice of Knowledge Discovery in Databases,2000, pp 1- 13.

3. Golbeck, J., and Hendler, J. Reputation network analysis for email filtering. In Proceedings of the First Conference on Email and Anti-Spam,2004, VOL 4825.

4. Bhandari, N.M., and Kumar P., Rtifical Neural Network - A tool for Load Rating of Masonry Arch Bridges,2006, Advances in Bridge Engineering.

5. Menhaj M.B., Fundamentals of neural networks. Computational intelligence, International MultiConference of Engineers and Computer Scientis, 2008,vol1.

6. Tang X., Hybrid Hidden Markov Model and Artifical Neural Network for Automatic Speech Recogonation,2009 Pacific-Asia Conference on Circuits, Communications and Systems, pp 682-685.

7. Seyed Mostafa Kia.,Neural networks in Matlab.Tehran,2009, Kian Publication of Rayaneh Sabz.