Reactive Web Policing Based on Self-Organizing Maps

Alvin T.S. Chan, Alan Shiu, Jiannong Cao and Hong-Va Leong
Electronic Commerce and Internet Computing Lab
Department of Computing, The Hong Kong Polytechnic University
Hung Hom, Hong Kong, SAR China.
Corresponding Email: cstschan@comp.polyu.edu.hk

***Abstract*- Almost without any doubt, the Internet and Web in particular, have brought about radical changes in information retrieval with unparallel benefits to organizations and individuals to gain access to vast amount of articles and documents. The focus on restricting inappropriate materials at their source is not well suited to the nature and open architecture of Internet, where information source maybe in a different legal jurisdiction than the recipient. This paper proposes a reactive approach based on SOM (Self-Organising Maps) neural network to deal with dynamic blocking and filtering of Internet contents. We describe the design and implementation of a web policing proxy (WebPolice) based on a Kohonen's neural network model that attempts to classify the web contents dynamically using *competitive learning*. The parameter setting of the network has been experimented to obtain the optimal classification rate and performance for the model.**

***Index Terms* - Web policing, PICS, self organizing map, SOM**

I. INTRODUCTION

Almost without any doubt, the Internet and Web in particular, have brought about radical changes in information retrieval with unparallel benefits to organizations and individuals to gain access to vast amount of articles and documents. The ease of access of these information and materials introduce a dimension of social problems that are often overlooked by society and organisations whose focus is primarily on how to better make use of Web to deliver interactive and active contents to end users. The open system architecture of Internet and Web access promote the free flow of web contents and information across a wide area network that spans across countries and continents. The control of web contents delivered to end users are not appropriately enforced by the network but instead depends on end systems to introduce appropriate filtering technologies to control and manage traffics transiting across the network system. While firewalls may be employed to perform traffic filtering at IP and TCP protocol level, it does not scale-well to content filtering and use as the basis to restrict distribution of certain kinds of restricted materials. In particular, as access of Internet from home becomes more pervasive, content filtering mechanisms are required to better control the access of materials by individuals at home. Parents may require such mechanisms to control what types of contents their children can access depending on their maturity and age factors.

The focus on restricting inappropriate materials at their source is not well suited to the nature and open architecture of Internet, where information source maybe in a different legal jurisdiction from the recipient. As such, recipient-centred control of Internet contents may be employed, rather than sender-centred control. The PICS[1](the Platform for Internet Content Selection) is a set of technical specifications proposed by W3C(World Wide Web Consortium) that facilitate the recipient-centred control of Internet content and the inter-operation of different bodies over the issues regarding Internet content control. PICS enables content filtering based on the PICS labels, which represent "meta-data" that provides descriptive information about the Web contents. The demerit of PICS is that some Internet content providers may not wish to label their contents or register with other third party that can provide content classification of their web contents, simply because it may result in less visits to their web sites. Unless explicitly enforced by government bodies or laws, it may be difficult, if not impossible, to rely on Internet content providers to cooperatively adopt this technology and labelling standards for all their contents posted on the web [1].

To address this problem, in this paper, we propose a reactive approach based on SOM (Self-Organising Maps) neural network that is employed to deal with dynamic blocking and filtering of Internet contents. The primary objective of this project is to design and implement a web policing proxy (WebPolice) based on a Kohonen's neural network model [2] that attempts to classify the web contents dynamically using *competitive learning*. The parameter setting of the network has been

0-7803-7101-1/ 01 / $10.00 © 2001 IEEE.

experimented to obtain the optimal classification rate and performance for the model. The following section describes the architecture of WebPolice as a agent that serves as an intermediately proxy between web browser client and web server. Section 3 describes the design and implementation of WebPolice and its performance evaluation. Section 4 highlights the future works and conclusion for the paper.

II. WebPolice Architecture

An important design requirement of WebPolice is to architect a web accessing model that seamlessly integrates existing technologies without the need to re-engineer or change the basic web infrastructure. This important requirement leads us to adopt a proxy intercept approach to effectively intercept traffic flowing between a web server and a web browser client. This approach allows us to avoid any modification to the web server or browser and to focus our design and development effort on the WebPolice proxy. The WebPolice proxy is responsible to intercept transiting HTTP request/reply and to perform real-time matching and classification of web contents over the path between the web server and client browser.

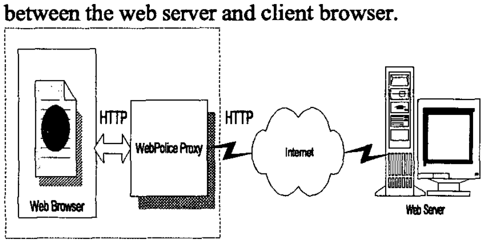


Figure 1: WebPolice Proxy Architecture

Figure 1 shows a request-response chain that involves the WebPolice proxy as an intermediary. In this scenario, the user agent makes its request to the proxy instead of the origin server. The proxy then makes the request to the origin server on behalf of the client. Once the request has been processed, the server replies to the proxy. At this point, the proxy is required to examine and perform real-time classification of the web contents, and to ensure that the contents contain only textual materials that are unrestrictive. If the materials are classified as valid, the proxy relays the web contents to the client, thus completing the request.

III. WebPolice Design

To classify the web contents and filter of Web pages retrieved and its associated contents whenever no PICS metadata about the web page are provided, an intelligent classifier using Kohonen's self-organising neural network is investigated and designed. Kohonen's network is selected to be the artificial neural network topology for this project because of its relatively shorter learning phase and its stochastic approximation property[2]. In fact, applications of Kohonen's network have been studied and used extensively in data processing, data compression, linguistic and text classification problems[3].

A. Learning Vector Quantization

The Learning Vector Quantization (LVQ) supervised learning extension of the Kohonen network method is employed as the basis for categorising input vectors derived from the textual web contents. During the learning process, several output neurons are assigned to each class and referred to as codebook vectors. For input, x, the neuron with the closest weight vector is declared to be the winner[2]. The following are assumptions made while applying the model:

- The classification of Web content is based solely on keywords extracted from the HTML content of the retrieved web page.
- The training and testing samples are categorised into two categories, Valid or Invalid.

The LVQ algorithm works as follows:

Initialize all weights to random values in the range [0,1];
repeat
  Adjust the learning rate ç(t);
  for each input pattern I_k in the training set, **do**
    find node j whose weight vector W_j is closest to i_k;
    for l = 1,……,n, **do**
      Update the weight $W_{j,l}$ as follows:
      if the class label of node j equals the desired class of $I_{k,}$
        then ä$W_{j,l}$ = ç(t)($I_{j,l}$ + $W_{j,l}$)
        else ä$W_{j,l}$ = ç(t)($I_{j,l}$ - $W_{j,l}$)
    end-for
  end-for
until network converges or computational bounds are exceeded.

The network architecture of the LVQ Network is shown in Figure 2[3] which identifies the input and the parameters that may affect its structure. The input layer has *d* nodes. Each node in the input layer is fully connected to each output node in the output layer. There are *N* categories and each category has C number of neurons(cookbook vectors). In our design, the dimension of the vector space is determined by the number of keywords in keywords store, which is 207(d=207) and N is equal to 2.

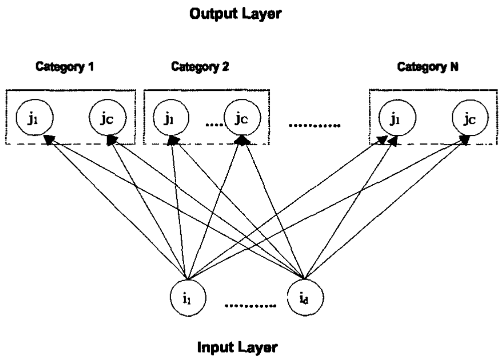


Figure 2: Network Architecture of LVQ

Distance and similarity are reciprocal concepts. There are a number of metrics to measure similarity of two patterns, namely *Correlation, Direction Cosines, Euclidean Distance*. In this project, the Euclidean Distance is used, where the minimum distance represents the closest match.

Euclidean Distance between a pair of N-by-1 vectors x_i and x_j:

$$d_{ij} = || x_i - x_j ||$$
$$= [\sum_{n=1}^{N} (x_{in} - x_{jn})^2]^{\frac{1}{2}}$$

where
$x_I = [x_{i1}, x_{i2}, \ldots\ldots, x_{iN}]^T$,
superscript T denotes matrix transposition,
x_i defines a point in an N-dimensional space called Euclidean space

B. System Architecture for Neural Classifier

The system architecture of the module of neural network classifier shown in figure 3 illustrates different components that the module composes.

- Preprocessor
  This is a preprocessing stage that transforms the input training samples that have been selected in advance to its corresponding vector representations for further processing.
- Kohonen Neural Network
  The Kohonen neural network learns from the training samples and produces a generalized mapping between the input patterns and the outputs. Figure 24 demonstrates the final result of neurons in the output layer of Kohonen neural network.
- Postprocessor
  The postprocessor gets the output signal from the Kohonen network and estimates which category a given sample belongs to.

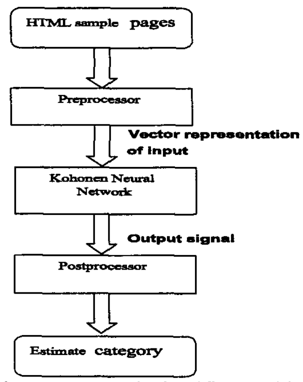


Figure 3: Neural Classifier Architecture

C. Learning Process

The learning mechanism of the classifier is illustrated in Figure 4. During this phase, the selected HTML training samples are pre-processed and matched against the keywords in keywords store to produce the corresponding vector representations for learning in later stage.

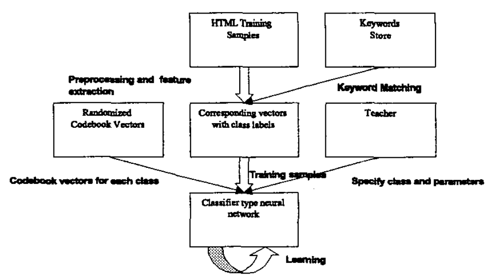


Figure 4: Learning Process

Accompanied to the output are the class labels for each of the vectors. The pre-processed samples in vector form are then applied to train the network with other parameter values specified by the teacher. The teacher is the trainer that supervises (oversees) the learning phase of the network.

During the phase of learning and the iterations in applying the training set, the codebook vectors for each category are adjusted and finally converge in a state that generalize the patterns of all training samples. The resulted codebook vectors are then used to test against the testing samples in the classification process.

IV. Experimental Results and Evaluation

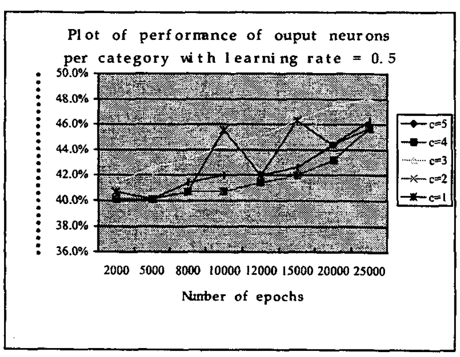
During the phase of performance evaluation, different parameters are adjusted and the effects of different parameters are correspondingly measured for further analysis. The effects of the following parameters are studied:

- ♦ Initial Learning Rate, $\varsigma(0)$
- ♦ Number of cycles, m
- ♦ Codebook vectors per category, c
- ♦ Percentage of correctly classified, t_p

To facilitate the evaluation of the classifier's performance in an efficient manner, the testing phase is performed in an offline mode. This is achieved by:

- Setting up a Microsoft Personal Web Server in a local personal computer with Internet Explorer 4.0 and Windows 98 installed.
- Downloading all testing samples in a directory under the Personal Web Server and configuring the web browsers to browse them locally.

The following graphs depict the results of the experiments conducted over the experimental platform with varying parameters.



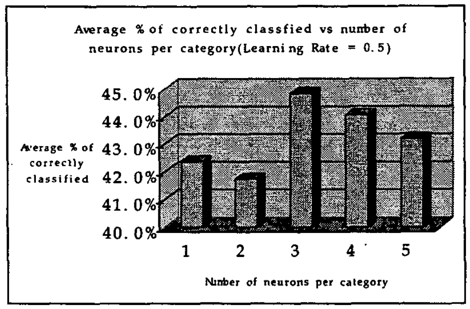
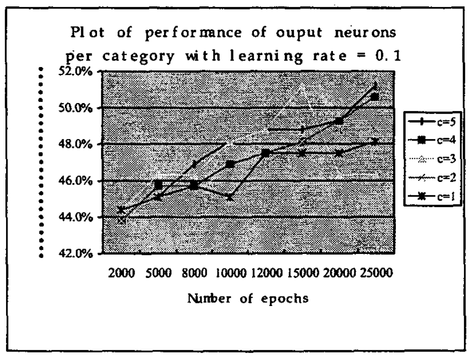


Figure 5: Performance with learning rate = 0.5



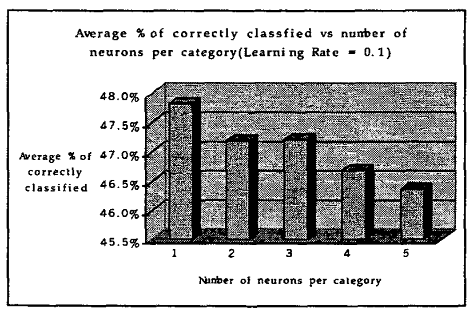
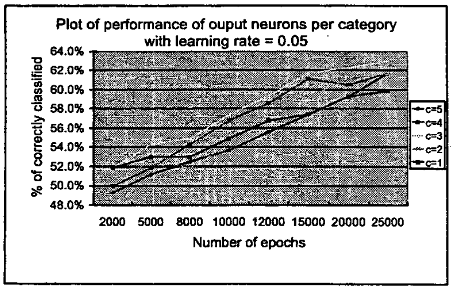


Figure 6: Performance with learning rate = 0.1



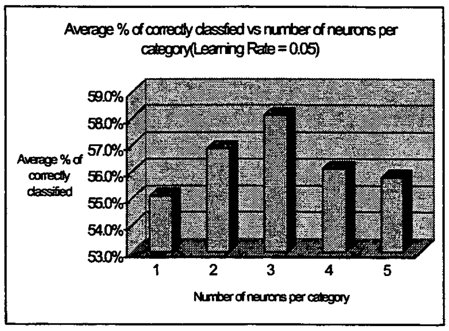


Figure 7: Performance with learning rate = 0.05

Results of experiments indicate that the optimised number of neurons per category in the output layer is 3 to achieve a better classification rate. It implies that c=3 is sufficient to represent all the different samples in each category. If the initial learning rate ς(0) is too large or too small, the classification rate may suffer. For learning rate=0.5, the classification rate is low as the steps of convergence may not be smooth enough and the result is greatly affected by the randomisation of the codebook vectors at the initial stage. However, if the initial learning rate is too low, the classification rate is still not optimised because the network may not be able learn and generalize from the samples.

There is also a general trend that the classification rate increases with the number of epochs(cycles) of the training sets applying to the network. This reflects the convergence of the network. However, the classification rate flats off or even decreases under some learning rate. This may reflect the effects of over-training on the network.

V. Concluding Remarks

In this paper, we have presented the WebPolice proxy and its architecture that is designed to perform real-time reactive policing of web contents based on artificial neural network model. In particular, we have employed Kohonen's neural network model as the basis to introduce unsupervised learning process and to use the trained model to perform real-time textual classification of web contents. While the applications of Kohonen's network have been studied and used extensively in data processing, data compression, linguistic and text classification problems, we believe that the use of such network for Web policing is novel and presents unique challenges. Initial experimental results indicate a respectable classification rate of above 64%, while being able to process the classification in a real-time mode. In the future, we aim to continue to tune the parameters for the network and to carry out extensive experiments to find the optimal configurations of the network.

References:

[1] Paul Resnick, James Miller, "PICS: Internet Access Controls Without Censorship", Communications of the ACM, 1996, vol. 39(10), pp.87-93.

[2] Teuvo Kohonen, Springer Series in Information Sciences : Self-Organizing Maps, Springer, 1995;23-160.

[3] Kishan Mehrotra, Chilukuri K. Mohan, Sanjay Ranka, Elements of Artificial Neural Networks, Massachusetts Institute of Technology, 1997; 164-174.