A Neural Net Approach to Data Mining: Classification of Users to Aid Information Management

Josephine Griffith, Paul O' Dea, Colm O' Riordan

Department of Information Technology, National University of Ireland,

Galway, Ireland

Abstract. Techniques from the domain of Artificial Intelligence are used increasingly to combat the problem of information overload on the Internet. The vast majority of such techniques and related systems attempt to overcome the problems of information overload by automating the analysis of the content of online documents. In many web-sites (document repositories and e-commerce sites) system logs, documenting user behaviour, are available and can be used as a valuable resource in information management. This information represents a valuable resource to aid in the organisation of information and the presentation of such information to users. In many such systems this information can be represented using a set of tuples indicating which pages/items were visited by a user. Using this information can provide many advantages. A classification of tuples can be used to aid information management and we outline briefly systems which have used the classification algorithm we propose. This paper presents an approach to solving classification problems by combining feature selection and neural networks. The main idea is to use techniques from the field of information theory to select a set of *important* attributes that can be used to classify tuples. A neural network is trained using these attributes; the neural network is then used to classify tuples. In this paper, we discuss data mining, review common approaches and outline our algorithm. We also present preliminary results obtained against a well-known data collection.

Keywords. Data mining, neural networks, classification, web-mining, information management

1 Introduction

The problem of information overload has to come to the fore in recent years with the increasingly large amounts of online data available to users. Many different techniques have been developed over the years.

The vast majority of such techniques and related systems attempt to overcome the problems of information overload by automating the analysis of the content of online documents.

In many web-sites (for example, document repositories and e-commerce sites) system logs, documenting user behaviour, are available. This information represents a valuable resource to aid in the organisation of information and the

P. S. Szczepaniak et al. (eds.), *Intelligent Exploration of the Web*

presentation of such information to users. In many such systems this information can be represented using a set of tuples indicating which pages/items were visited by a user. Using this information can provide many advantages.

This information is usually too large in volume to render any manual approaches feasible. Hence automated intelligent techniques are required. This problem has many parallels with the field of data mining, in particular the problem of classification.

This paper presents an approach to solving classification problems by combining feature selection and neural networks. The main idea is to use techniques from the field of information theory to select a set of *important* attributes that can be used to classify tuples. A neural network is trained using these attributes; the neural network is then used to classify tuples. In this paper, we discuss data mining, review common approaches and outline our algorithm. We also present preliminary results obtained against a well-known data collection.

Section 2 contains an overview of approaches to information management-information retrieval and collaborative filtering. In section 3, we discuss data mining before presenting our proposed approach in section 4. Results obtained with our data mining algorithm are presented in section 5 and we briefly sketch systems which have used the algorithm in information management.

2 Information Management

Techniques from the domain of Artificial Intelligence are used increasingly to aid in counteracting the growing problem of information overload. The vast majority of such techniques and related systems attempt to overcome the problems of information overload by automating the analysis of the content of online documents. Documents are retrieved for a user based on a similarity measure between the representations of user's information need and documents. More recently, techniques have been developed that do not use content directly (e.g., Google search engine, Clever system [4], collaborative filtering systems) but use user behaviour, link structure etc., as extra evidence of the relevance of a document to an information need.

We will review both the content-based retrieval approach (and related fields) and the newer mechanisms which do not solely rely on content representations before discussing in more detail the application of our data mining approach to information management.

2.1 Content Retrieval

Information retrieval (IR) is a well established field in information science, which addresses the problems associated with retrieval of documents from a collection in response to user queries. Information filtering (IF) is a more recent specialisation within information science, having come to the fore due to increasing volumes of online transient data.

The type of comparison effected between the user's information need and the document set is determined to a degree by the representation chosen. Approaches include:

- string matching possibly augmented with proximity and Boolean operators.
- vector-space model [7] where documents and queries are represented as vectors of dimension m, where m is the total number of terms used to identify content. Each of these terms has an associated weight representing its relative importance (based on frequency within a document and across the document collection).
- Latent Semantic Indexing (LSI) [2] attempts to overcome the problems associated with word-based methods, especially the vector space approach, by organising textual information into a semantic/conceptual structure more suitable to information retrieval. The phrase "latent semantic" refers to the inherent underlying associations between words used to express a particular concept.
- Connectionist Approaches to IR: In most connectionist approaches to information retrieval, each node is used to represent an individual keyword. The search mechanism usually used in these systems is the *spreading activation search* (SAS). In this search strategy, activity is propagated throughout the system and nodes with a high level of activity are returned as the result of the search.

2.2 Collaborative Filtering

Collaborative Filtering is not based on analysis of the content of the document set but on the premise that "people with similar interests in the past will have the same interests and preferences in the future" [9]. Collaborative filtering systems and recommender systems attempt to exploit this information to predict users' interests.

Given a set of users, a set of items, and a set of ratings, systems attempt to recommend items to users based on prior ratings. Collaborative filtering systems essentially automate the "word of mouth" process. The problem space can be viewed as a matrix consisting of the ratings of each user for the items in the document set, i.e., the matrix consists of a set of ratings $u_{i,j}$, corresponding to the rating by user i for an item j. Using this matrix, the aim of collaborative filtering is to predict the ratings of a particular user, i, for one or more items in the document set.

The steps involved in the prediction of these ratings for a given user i are:

1. Select a set of users with similar interests/preferences to user i, i.e., users who have similar ratings for items as user i.
2. Predict recommendations for user i from the set selected in step 1, i.e., if these users rated an item j highly, this item will be recommended to user i.

The correlation between two users can be calculated in many ways. Common approaches that have been adopted include: Pearson correlation (used in the original GroupLens system [6]), constrained Pearson correlation (used in the Ringo system [9]), Spearman rank correlation, vector similarity, entropy-based uncertainty measure, and the mean-square difference algorithm.

The neighbourhood chosen for a user can include the closest n neighbours for some value of n or take all neighbours above some predefined threshold.

2.3 Retrieval and Filtering Based on Implicit Forms of Collaborative Filtering

Techniques that do not explicitly maintain a users' rating matrix, but instead use information that has been provided implicitly by users have come to the fore in the recent past, most noticeably in the area of web search engines.

One of the earliest systems to make recommendations based on implicit user recommendations was the Phoaks system [10]. The system provides recommendations for web-pages and web-sites, in a fixed set of domains, based solely on recommendations implicitly expressed by users of related newsgroups. Each reference to a given URL (Uniform Resource Locater) was taken as a positive recommendation (with some rules built in to avoid counting urls in replies or quoted messages etc.) by a user which was used to build a list of recommendations in different domains.

The Clever [4] system also uses implicit ratings made by people to provide recommendations to web-pages based on a query. Given a query, a set of web-pages are returned using standard content-based mechanisms. This acts as a base set which is improved by automatic analysis of the link structure of these and related pages. The initial pages are explored by following links to a predefined depth. The system then attempts to recognise, using this larger set of pages, two types of pages - authority pages (best sources) and hubs (collections of links). The subset of the web represented by the pages and the links between pages can be viewed as a directed graph. Weights (X_p, authority page and Y_p, hub weight) are assigned to each page. All pages are assigned uniform weighs initially. Viewing the graph as an adjacency matrix A, values for hub and authority pages can be updated according to $X_p \leftarrow (A^{\mathrm{T}}A)X_p$ and $Yp \leftarrow (AA^{\mathrm{T}})Y_p$. This results in values for X_p and Y_p indicating good authority pages to return to the user.

As mentioned heretofore this implicit knowledge can be harnessed to aid in the deployment of suitable information management strategies; however, oftentimes this information is too large in volume to render any non-automated approaches possible. Hence automated intelligent techniques are required. This problem has many parallels with the field of data mining, in particular the problem of classification.

If we consider users' visits to a web-page as a positive recommendation for that page (easily extended to make more refined estimates of strength of recommendation based on frequency of visits etc.), we can develop a matrix of recommendations (user x page) which can be used as the basis for providing recommendations. We describe a technique in this paper which attempts to classify users into disjoint classes; a user's membership of a particular class representing that certain information viewed by members should be passed to the user etc. For example, we can view the pages visited by a new user as a tuple comprising, at least, Boolean values. We can recommend (or not) certain pages based on an ongoing re-classification of the user based on previously mined rules.

This approach has many parallels with collaborative filtering but, in this paper, it is in effect restated as a data mining problem. In this paper we describe an

approach to data mining which combines ideas from Information theory and neural networks.

3 Data Mining

3.1 Introduction

Data mining techniques are applicable in a wide variety of domains. Currently, the main application area for data mining is e-commerce where it is used to understand and target each customer's individual needs by highlighting both customer preferences and buying patterns from databases containing customer and transaction information. With this information, companies can better target customers with products and promotional offerings.

Data mining techniques are predominantly applied to the problem of finding association and classification rules, as well as to the problems of item-set recognition and sequential pattern recognition. Classification is discussed in this paper.

Classification is the process of finding the common properties among different entities and classifying them into *classes*. The results are often expressed in the form of rules - *classification rules*. By applying the rules, entities represented by tuples can be easily classified into the different classes to which they belong.

Given a set of tuples of attributes $A_1, A_2, \ldots, A_N$, a classification rule may take the form:

$$(A_i = value_i) \wedge \ldots \wedge (A_j = value_j) \rightarrow Class_i .$$

We can restate the problem formally defined by Agrawal *et al.* [1] as follows: let A be a set of attributes $(A_1, A_2, \ldots, A_N)$ and $\mathrm{dom}(A_i)$ refer to the set of possible values for attribute A_i. Let C be a set of classes $c_1, c_2, \ldots, c_m$. We are given a data set, *the training set* whose members are $n+1$-tuples of the form $(a_1, a_2, \ldots, a_n, c_k)$ where $a_i \in dom(A_i), (1 \le i < n)$ and $c_k \in C_i, (1 \le i \le m)$.

For example, a retail outlet may wish to classify customers into classes so as to adopt an advertising strategy to maximise profit. So customers may be placed in disjoint classes based on age, salary, previous purchases etc.; based on these classes, different products might then be advertised differently.

3.2 Techniques and Approaches

Many data mining techniques and approaches have been developed and used. Common approaches are outlined below:

Decision Trees

This technique recursively partitions the data set until each partition contains mostly examples from a particular class [3]. In a decision tree, each internal node represents a split point which tests some property where each possible value of that property corresponds to a branch of the tree, leaf nodes representing classifications. An object of an unknown type may be classified by traversing the tree, testing the object's value for each property at an internal node and taking the appropriate

branch. Eventually a leaf node will be reached which represents the object's classification.

Decision trees are popular because they are easy to understand and results are reasonably accurate. The rules for classifying data are in a form readily understood by humans. However, the performance of the decision tree depends critically upon how the split point is chosen. Often splits between branches are not smooth and the choice of split is made regardless of the effect such a partition will have on future splits. Additionally, in the presence of noise or missing attribute values in the data set, there can be problems with performance.

Neural Networks

Neural networks, a form of subsymbolic computation, are based (simplisitically) on the workings of the brain. A neural network comprises a set of weighted edges and nodes. Learning is achieved by modification of these weights. Most networks contain a number of layers, the first layer being the input layer, the final layer being the output layer. Other internal layers (hidden layers) are often required to ensure sufficient computational power in the network.

A network can be trained to map input values to corresponding output values by learning the features of a provided training set. The network is repeatedly tested and modified to produce the correct output. The generation of output by a neural network is accomplished via firing values from nodes. An input is passed to the input layer which in turn can activate the internal layers, which in turn activates the output layer, finally resulting in an output. Given n links feeding into a node, each link has an input value X_j and a weight W_j. The nodes have an associated threshold, τ. If, according to some activation function, the node has a sufficiently high activation level, the node fires a value onto the next layer. Commonly used activation functions include:

- $$f(a) = \begin{cases} 1 if\ a > \tau \\ 0 otherwise \end{cases}$$

- $$f(a) = \begin{cases} \dfrac{e^a - e^{-a}}{e^a + e^{-a}} \end{cases}$$

where a, the activation of a node, is $\sum_{j=1}^{n} X_j W_i$ and τ is a threshold. The initial input vector is fed into the network; sets of nodes are fired which finally results in an output vector.

(Note: the above description relates to a simple feed-forward network; other more complicated (and computationally expressive) architectures are possible).

To train the network, any errors are *back-propogated* throughout the network. An example of a system which uses a neural network approach is NeuroRule [5] where the number of input nodes corresponds to the dimensionality of the input tuples and the number of output nodes is equal to the number of classes to be classified. The first stage, network training, is terminated when a local minimum is

reached. Secondly network pruning is carried out, and finally extraction of the classification rules from the pruned network.

The major criticisms of a neural network approach include the fact that because neural networks learn the classification rules by multiple passes over the training data set, the learning time, or the training time needed for a neural network to obtain a high classification rate, is usually long [1]. In addition, there is difficulty in understanding the rules generated by neural networks as they are buried in the network architecture and the weights assigned to the links between the nodes. Also, there is difficulty in incorporating any available domain knowledge.

4 Algorithm

4.1 Information Theory

Information theory is widely used in computer science and telecommunications, including such applications as determining the information-carrying capacity of communications channels, developing data compression algorithms, and developing noise-resistant communication strategies. Information theory provides a mathematical basis for measuring the information content of a message. We may think of a message as an instance in a universe of possible messages; the act of transmitting a message is the same as selecting one of these messages. From this point of view, it is reasonable to define the information content of a message as depending upon both the size of this universe and the frequency with which each possible message occurs.

Shannon formally defined the amount of information in a message as a function of the probability of occurrence of each possible message [8]. Given a universe of messages, $M = \{m_1, m_2,...,...m_n\}$ and a probability, $p(m_i)$, for the occurrence of each message, the information content of a message in M is given by:

$$I(M) = \sum_{i=1}^{n} - p(m_i) \log_2 (p(m_i)) .$$

The information in a message is measured in bits. This definition formalises many of our intuitions about the information content of messages. Assume a set of training instances, C. If we make property P, with n values, the root of the current tree, this will partition C into subsets, $\{C_1, C_2,...,C_n\}$. The expected information needed to complete the tree after making P the root is:

$$E(P) = \sum_{i=1}^{n} \frac{|C_i|}{|C|} I(C_i) .$$

The gain from property P is computed by subtracting the expected information to complete the tree from the total information content of the tree:

$$\text{gain}(P) = I(C) - E(P).$$

As discussed earlier, neural networks provide a powerful mechanism for classification obtaining high precision whilst being robust in the presence of noise. Unfortunately the time to train a network can be prohibitive and the classification rules can be buried in the neural network architecture. The more traditional approaches based on information theory perform well but tend to deal poorly with

noise. The main advantages associated with this approach is that the rules are in a more understandable format and the technique is far more computationally efficient. The motivation behind the algorithm proposed in this paper is to exploit the advantages of the neural network approach while maintaining a degree of computational efficiency by utilising ideas present in information theory. The algorithm comprises two phases:

1. firstly, using ideas from information theory, select *important* attributes, and
2. secondly, apply a neural network to allow classification of tuples based on the attributes selected as being *important*.

More formally, given a set of n tuples $t_1 \ldots t_n$ with attributes $a_1 \ldots a_n$, the information content of each of the n attributes is calculated according to their ability to classify the tuples $t_1 \ldots t_n$. The ability of an attribute (A_i) to classify the tuples is calculated by measuring the expected information (E_i) via:

$$E_i = \sum_{k=1}^{N_i} \frac{n_{ik}}{n} . I(S_{ik})$$

where n is the total number of tuples, N_i is the total number of values attribute A_i can take and $I(S_{ik})$ is the entropy of the subset S_{ik}.

In the second phase, we attempt to classify the tuples using a subset of the attributes. This subset of attributes $s_1 \ldots s_n$ comprises those with an information content above a threshold. We then consider the set as a whole thereby avoiding, to a degree, the attribute independence assumption prevalent in traditional approaches. A feed-forward network (as described earlier), using back-propagation as a learning algorithm, is used to train the network to classify the tuples $t_1 \ldots t_n$ based on the attributes $s_1 \ldots s_n$.

5 Results of Data Mining

5.1 The German Credit Data Set

In order to facilitate testing of the developed approach, experiments were conducted using the German credit data set[1]. The German credit data set contains information on 1000 loan applicants. Each applicant is described by a set of 20 different attributes. Of these 20 attributes, seventeen attributes are discrete while three are continuous. To facilitate feature selection and neural network training in the later phase, the values of the three continuous attributes were discretised. Each of the three attributes was discretised by dividing its range into subintervals.

A classification assigned to each of the applicants determines whether an applicant is a good or bad credit risk. Thus the problem is to classify each pattern as either *good* or *bad*. In the data set, there are a total of 700 cases of good applicants and 300 cases of bad applicants. Furthermore, the data set is divided into

[1] This data set is available publicly from the University of California-Irvine machine learning repository via anonymous ftp to *ics.uci.edu*

a training set and a test set. The training set consists of 666 tuples and the test set contains 334 tuples.

Using the feature selection algorithm outlined, of the original 20 attributes describing each pattern in the German dataset, 7 were selected. In Table 1, we list the information gains G, normalised gains G' and their averages for the German credit problem. Those attributes deemed *selectable* are: *status, duration, credit history, credit amount, savings, housing* and *foreign worker*.

Table 1. Attribute gains of the German credit data set

No.	Attribute	G	G'
1.	status	0.08166752	0.04533132
2.	duration	0.01565728	0.01175013
3.	credit history	0.03506461	0.02039325
4.	purpose	0.02510743	0.00945620
5.	credit amount	0.01835606	0.02097592
6.	savings	0.04112237	0.02461317
7.	employment duration	0.01262678	0.00581796
8.	installment rate	0.00467093	0.00258875
9.	personal status	0.00621573	0.00404868
10.	debtors	0.00481950	0.00893288
11.	residence	0.00117720	0.00064023
12.	property	0.01892740	0.00971905
13.	age	0.01454505	0.00788522
14.	installment plans	0.00604603	0.00699498
15.	housing	0.01267492	0.01136562
16.	existing credits	0.00131140	0.00119170
17.	job	0.00468588	0.00326961
18.	liable people	0.00030049	0.00049862
19.	telephone	0.00002599	0.00002691
20.	foreign worker	0.00523591	0.02339419
	AVERAGE	0.01551192	0.01094472

5.2 Learning with Feature Selection

In the second phase, the number of units in the input layer of the neural network was determined. The *thermometer* coding scheme was employed to get the binary representations of the attribute values for inputs to the neural network. Hence for attribute *status*, a value of *less-200DM* was coded as {001}, a value of *over-200DM* was coded as {011}, and a value of *no-account* was coded as {111}. Zero status (*0DM*) was coded by all zero values for the three inputs. The second attribute *duration* was similarly coded. For example, a duration value less than 20 months was coded as {0001}, a duration value in the interval [20,40] was coded as {0011}, etc. The coding scheme for the other attributes are given in Table 2.

Table 2. Binarisation of the attribute values

Attribute	Input Number
status	$I_1 - I_3$
duration	$I_4 - I_7$
credit history	$I_8 - I_{12}$
credit amount	$I_{13} - I_{16}$
savings	$I_{17} - I_{21}$
housing	$I_{22} - I_{24}$
foreign worker	I_{25}

With this coding scheme, we have a total of 25 binary inputs. Two nodes were used at the output layer. The target output of the network was {1,0} if the tuple belonged to class *good*, and {0,1} if the tuple belonged to class *bad*. The number of hidden nodes in the network was initially set as three. Thus, there were a total of 81 links in the network. The weights for these links were given initial values that were randomly generated in the interval [-1,1].

Figure 1 illustrates the convergence behaviour during a typical training phase for a network constructed with only selected attributes.

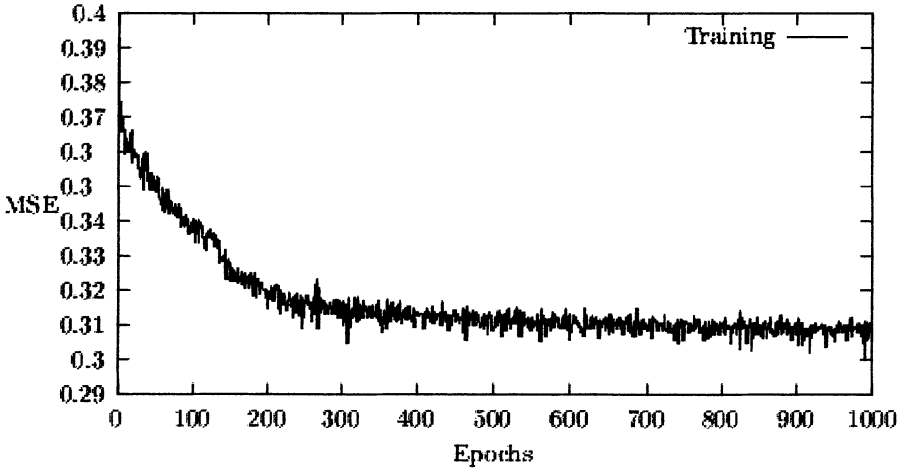

Fig. 1. Back-propagation convergence curve for neural network constructed using only selected attributes

The network was then trained until a local minimum point of the mean squared error function had been reached. In Figure 1, we can observe that the error curve reaches a local minimum point of the mean squared error function in the interval [0.30,0.32]. Through experiment, a network constructed using only selected attributes generally tends to reach a local minimum point of the mean squared error function in the interval [300,600] epochs.

The end result of the above training phase was a fully connected trained network which achieved an accuracy of 78.53% on the training data where classification accuracy is defined as:

$$accuracy = \frac{number\ of\ tuples\ correctly\ classified}{total\ number\ of\ tuples}.$$

The degree of error in this result can be attributed to two characteristics of the training data. In the first case, due to the presence of noise, it is not possible to achieve 100% accuracy on training data for the German credit problem. There are always problems with real-world data. Noise in the data encompasses irrelevant, missing, incorrect, and contradictory data. In general, noise in the data weakens the predictive capability of the features. The other possible factor is due to the imbalance in the training set.

The end result of training is that we now have a network which will act as a classifier for applicants of unknown class. In general, it will be the performance of the classifier on the test set, which does not participate in the training phase, that will be the most salient. The network obtained from the training phase described above achieved a classification accuracy of 74.25% on the test data.

5.3 Description of Results

In order to test the usefulness of the proposed approach, we present empirical analysis comparing the accuracy of networks constructed with and without feature selection over the chosen test collection.

Twenty neural networks were constructed with the full set of twenty attributes and another twenty networks were constructed using only the seven selected attributes. Of the twenty networks constructed in each case, five networks had 1 hidden unit, another five had 2 hidden units etc. Table 3 and Table 4 summarise the classification accuracies achieved by these networks on both the training data and the test data of the German credit problem.

Table 3. Results from the German credit problem with 7 selected attributes used as input

Units	Links	Acc. on train set (%)		Acc. on test set (%)	
		Ave.	Std. Dev.	Ave.	Std. Dev.
1	27	77.83	0.23	75.85	0.35
2	54	77.58	1.02	74.45	0.46
3	81	78.88	1.36	74.45	1.65
4	108	80.38	1.09	73.15	0.46

Table 4. Results from the german credit problem with all 20 attributes used as input

Units	Links	Acc. on train set (%)		Acc. on test set (%)	
		Ave.	Std. Dev.	Ave.	Std. Dev.
1	74	85.99	0.31	72.66	1.13
2	148	86.49	1.48	72.36	2.21
3	222	88.19	2.34	72.36	0.17
4	296	92.69	0.75	71.46	1.75

From these tables, we can observe that removing redundant attributes for the German credit problem increased slightly the predictive accuracy of a neural network. For the German credit problem, the accuracy on the training data actually decreased when 13 attributes were removed from the input data. However, the predictive accuracy was marginally higher with the exclusion of the redundant attributes.

It is also worth noting that the average number of function/gradient evaluations is typically less when only seven attributes are used. A network constructed with only selected attributes as input, tends to reach a minimum of its error function before that of a neural network constructed with all attributes. In addition, the removal of irrelevant data through feature selection results in a simpler network construction with fewer links. A simpler network architecture reduces the computational costs, while a large network with many parameters may over-fit the training data and give a poor predictive accuracy on new data not in the training set. Finally, a simpler network results in a faster training time. Thus a network constructed using only those attributes which provide the most information for classification, can be said to be more computationally efficient than a network which models all attributes.

We can also observe from both tables, that the accuracy of the constructed networks on the test data decreases as the number of hidden nodes in the network increase. This emphasises the importance of limiting the number of hidden units in a network.

6 Application to Information Management

Given the recent need for information management solutions, attention has turned to alternative means to traditional content filtering. These have included collaborative filtering using both explicit and implicit recommendations.

The problem of harnessing the valuable knowledge embedded in implicit ratings has many parallels with problems existing in the field of data mining. We mine for classification rules, which we use to classify users in order to make recommendations. In this regards it is quite similar to collaborative filtering. We may also extend our mining to use any available domain knowledge.

We have implemented the algorithms as part of a web-based retrieval system, where browsing habits of a set of users over an expanding set of pages (items) are mapped to a two-dimensional matrix. Given a user accessing the system, any pages that have been added to the site which have been viewed by other users can be

recommended to the user rather than requiring the user to traverse a larger list of items.

One further application of our algorithm is in a web-based education system where we use the above classification scheme to classify users together in order to recommend material (web-pages).

7 Conclusion

In this paper we presented an approach to the classification problem which combines feature selection (based on information theoretic approaches to identifying useful attributes) with neural networks. We outlined the motivations behind adopting such an approach-namely the high accuracy to be obtained using neural networks, its robustness to noise and its increased computational tractability given the reduced number of attributes over which the network is trained. We also presented results to show the validity of such an approach.

We have also presented an overview of approaches within in the field on information management. We posit that in order to deal with the problems of information overload that mechanisms that exploit knowledge available from implicit user recommendations can improve the quality of information management systems and we discuss briefly how our data mining algorithm has been used in this context.

References

1. Aggrawal, C.C., Yu, P.S. (1999): Data Mining Techniques for Associations, Clustering and Classification. *Proceedings of the 3rd Pacific-Asia Conference on Methodologies for Knowledge Discovery and Data Mining (PAKDD-99).*
2. Dumais, S. (1991): Improving the Retrieval of Information from External Sources. *Behavior Research Methods Instruments and Computers.* Vol. 2, No. 23, pp. 229 – 236.
3. Mehta, M., Shafer, J., Agrawal, R. (1996): SPRINT: A Scalable Parallel Classifier for Data Mining. *Proceedings of the 22nd VLDB Conference.*
4. Kleinberg, J. (1997): Authoritative Sources in a Hyperlinked Environment. IBM Research Report RJ 10076, May, 1997.
5. Lu, H., Setioni, R., Liu, H. (1995): NeuroRule: A Connectionist Approach to Data Mining. *Proceedings of the 21st VLDB Conference.*
6. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Reidl, J. (1994): GroupLens : An Open Architecture for Collaborative Filtering of NetNews. *Proceedings of ACM 1994 Conference on CSCW.* pp. 175 – 186.
7. Salton, G.A., McGill, M.J. (1983): *Introduction to Modern Information Retrieval.* McGraw Hill International, 1983.
8. Shannon, C. (1948): A Mathematical Theory of Communication. *Technical Report, Bell Systems.*
9. Shardanand, U., Maes, P. (1995): Social Information Filtering: Algorithms for Automating ``Word of Mouth". *Computer-Human Interfaces (CHI '95).*
10. Terveen, L., Hill, W., Amento, B., McDonald, D. (1997): PHOAKS: A system for sharing Recommendations. *Communications of the ACM.* Vol. 40, No. 3, pp. 59-65.