NEURAL NETWORKS-BASED MULTI-INTEREST INFORMATION FILTERING

Dai Xuewu[1], Vic Grout[2], Tang Haokun[3] and Li Jianguo[1]

[1]Southwest Normal University, Chongqing, China
daixuewu@swnu.edu.cn
[2]University of Wales, NEWI, Wrexham, UK
v.grout@newi.ac.uk
[3]Chongqing University of Posts & Telecommunications, Chongqing, China
tanghaokun@vip.sina.com

This paper was supported by the Youth Foundation of Southwest China Normal University (220-413050)

ABSTRACT

With an increasing amount of information available, users find it difficult to obtain the most relevant Web pages from the large number returned by search engines. This is particularly true if users have more than one interest and require information spanning several of them. In this paper, we present an improved adaptive neural fuzzy network providing an information filtering system for the Web to sift the results provided by external search engines. We discuss how to model a user's multi-interests and filter information according to 'IF-THEN' rules and how to optimize and adjust the parameters stored in the network as a user's multi-interests. Preliminary experiments show that our prototype system improves the performance of current search engines for the user who has multi-interests. The distinguishing features are that of a user model embedded in a neural fuzzy network to process the multi-interests and the replacement of the traditional cosine measure method by a parameterized non-linear map allowing multi-interests to be processed. The results achieved support the choice of the neuro-fuzzy network for multi-interest information filtering and show that the information retrieval system can benefit from the technology available in Soft Computing for better retrieval.

KEYWORDS

Information Filtering, Neural Networks, User Modeling, ANFIS

1. INTRODUCTION AND RELATED WORK

With the exponential growth in the number of web pages, information retrieval becomes more difficult for users. The current information retrieval system, based on search engines and keywords, does not sufficiently take into account users' different interests. Users get the same results if they submit the same query words, with the valuable and worthless difficult to distinguish. Also, it is hard for users to describe precisely their interests in a few words. In most cases, a user's interests are fuzzy, blended and cross multiple categories. Thus, intelligent information filtering and user modeling [1] derived from AI and machine learning [2] are needed to improve information retrieval systems. The challenge is to identify an appropriate description of a user's complex multiple interests and to develop an adaptive information filtering system, which together are used to provide the personalized information service.

Several approaches in information retrieval and filtering have been developed to produce better search results or reduce the information overload. In *IfWeb* [3], the agents navigate the related pages, then classify and show pages according to user interests. Users provide explicit feedback on pages in terms of ratings or preferences. However, in the real world, it may be unlikely that a user has only one independent interest. Cheung et al [4] apply collaborative filtering,

considering the opinions (in the form of ratings) of other "like-minded" users to filter out the irrelevant pages. It requires a high performance server with large storage to process the large quantity of user interest information. *ProFusion* [5] is a meta-search engine. That is, it passes the user's request to more than one engine, and then rates the results returned from these different engines. More sophisticated techniques include the use of ontology-based similarity measures [6].

In this paper we propose a hybrid information system, combining user modeling and neural fuzzy networks, for ranking and selecting HTML pages from the result pages provided by external search engines and we attempt to resolve the problem of describing user multi-interests to improve the precision of current search facilities. The proposed system recommends the relevant pages to users according to the multi-interests. To do this, the system makes use of a User Modeling system [2] to acquire, store and restore the user's interests and non-interests. An advanced neural fuzzy network is applied to provide adaptive information filtering. The distinguishing features of the presented system are not only embedding a user model in the neural fuzzy network to process the user's mixed interests, but also the replacement of the traditional cosine measure method by a parameterized nonlinear map, so that it becomes straighforward to process the multi-interests. Derived from the neuro-fuzzy network *ANFIS* (*Adaptive Neuro-Fuzzy Inference System*) [7], our proposed technique consists of two main concepts: structure identification (comprising a resource model and a user model) and parameter identification (comprising a filtering algorithm and a learning algorithm). The system has been developed on a Matlab and MS VC based platform. From our preliminary experiments, the proposed system appears to reduce the amount of irrelevant information presented to the user.

The rest of this paper is structured as the follows. We first present briefly the general structure of AUMIF, our proposed system. The next section details the structure identification and how the resource model and user model have been constructed. Then we explain the parameter identification and how to achieve adaptability. Finally, we present our preliminary experimental results and give a conclusion.

2. GENERAL ARCHITECTURE

Our proposed system, entitled AUMIF (Adaptive User Modeling and Information Filtering), consists of two key models: the resource model and the user model, as well as two integrated algorithms to improve adaptability: a filtering algorithm and a leaning algorithm.

The general architecture is shown in Figure 1. It consists of the following modules:

- The *Resource Model*, a *Vector Space Model* (*VSM*) [8], applied to describe the contents of the web pages . In the Resource Model, the web pages returned by the External Search Engines are mapped onto vectors, so that it becomes easy to measure the relevance between the web pages and the user's interests.
- The *User Model*, embedded in the fuzzy neural network, storing and representing the interests and the information needs of a particular user. The user's interests are stored as a group of parameters in the fuzzy neural network.
- The *Filtering component*, also embedded in the fuzzy neural network, which selects the relevant pages for the user, according to the user's interests - stored in the User Model.
- The *Leaning component*, capable of dynamically building the user model, and optimizing the parameters of the fuzzy neural network through interaction between user and computer. It consists of two parts, known as *online learning* and *offline learning* respectively. The offline learning is a form of batch learning, transforming the web pages provided by a user, as a description of their interests,

into an interest vector, and initializing or optimizing the user interest parameters in the user model. The online learning is used to adapt the weight of different interests.

- The *External Search Engines,* which search the Web and return the pages.
- The *User Interface*, which manages the interaction with the user, including the acquisition of user key words, example documents and feedback as well as presenting the filtered pages to the user.

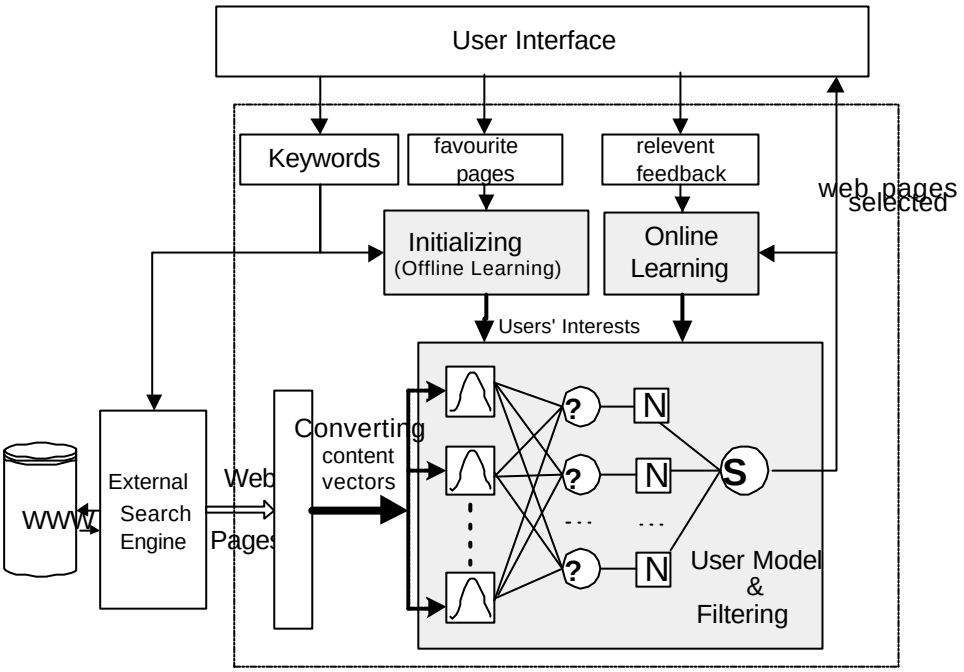

Figure 1. The General Architecture

2.1 The Interaction Process

To begin with, the user model must be initialized through the inputting of interests by the user. Then, the interests parameters stored in the network are initialized and the keywords submitted to the external search engine. After the results are returned, the system works through the following steps to select the relevant pages from those returned:

1. Convert the web pages into content vectors, p_j.
2. Filter the information by measuring the similarity of each web page to user interests or ranking the match between a web page and user interests. This key process makes use of an improved *ANFIS* network. The input is the *content vector*, p_j, the node parameters are the sets of *user interests* $\{u_i\}$ and the output is the similarity of each p_j to $\{u_i\}$.
3. Evaluate the pages, comparing the *similarity* R_j to a threshold $?$. If $R_j > ?$, the page is submitted to the user. Otherwise, it is discarded.
4. Optimize the parameters of the network. In the optimizing process, the system works using two algorithms based on user feedback. The first, the online learning algorithm, is used to update the weights of different interests. The second, the offline learning algorithm, is used to update user interests.

5. Repeat the previous steps as necessary (until the user terminates the session).

In this way the system is adaptive. That is, the user model can be updated according to user feedback and preferences.

2.2 Inputting the Search Query

Two methods are used to express the user's multiple complex interests quickly and precisely. Figure 2 shows the user interface used in our system and its operation is explained below.

Submitting keywords with weights

The user types in several keywords and sets an initial weight value for each to describe the importance of that keyword. *1* represents the highest weighting, *0* indifference and *-1* indicates that the key word is *not* to be included in the web page. (The default value is 1, matching the behaviour of conventional search methods.) This specifies the initial query.

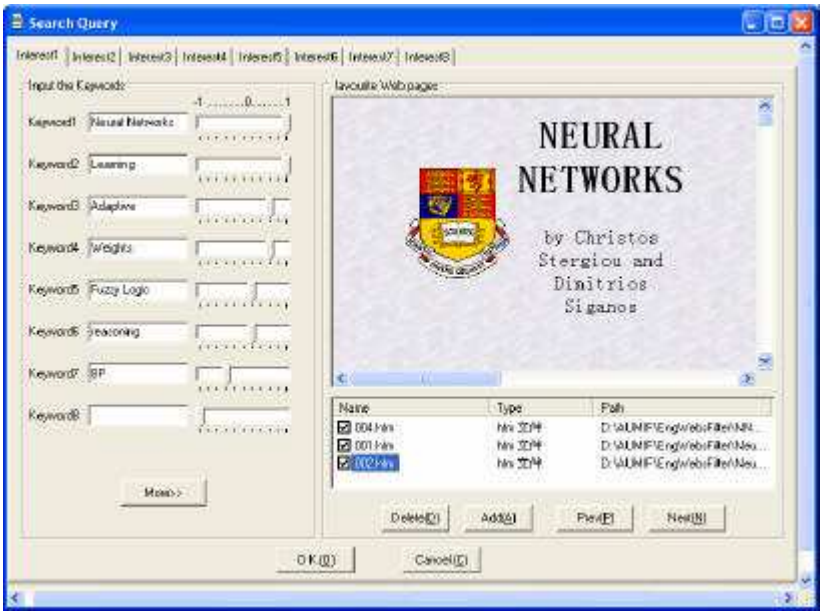

Figure. 2 The User Interface for Input

Submitting favourite pages to express complex interests

Interests may be hard to express merely as a combination of simple key words. In order to help the user to express mixed interests precisely, the system accepts favourite pages and performs analysis to determine the query keywords and the interest vector. The favourite pages can be added and browsed through the interface.

3. STRUCTURE IDENTIFICATION

System modeling comprises two stages: structure identification and parameter identification. The intent of structure identification is to *formalise* the structure of the system. There are two objectives in this first stage: selecting the system variables and determining the reasoning rules.

3.1 Selecting Input/Output Variables

To select the variables it is firstly necessary to define how many should be used in a problem (the *selection* of the variables) and, secondly, to determine the distribution of these variables (the *partitioning* of the variables) into the input-output space. This involves specifying which data should be selected to describe the user interests and the content of each page. We use the

content of web pages and the user information request as the system's input variables. The system calculates the similarity of each page to user interests as output.

A *VSM* is applied in our system with each page's content represented by a vector of terms. The *content vector* p_j refers to the content of a page in the vector space. Therefore, in order to compare the similarity of a page and the user's interests, we also use a vector, u_i, called the *interest vector*, to store each of the user's interests. Both the content vector and the interest vector are input variables based on the same vector space where the number of dimensions is dependent upon the number of terms. Since one user could have several interests, there are a set of interest vectors $\{u_i\}$ in our system.

There is only one output variable: the evaluated similarity, as a numeric score, denoted R_{pred}, on the continuous interval *[-1,1]*. As with neural networks, this is essentially a prediction measure. It should be very near to the real similarity. If $R_{pred} > 0$, the web page is relevant to user's interests; if $R_{pred} < 0$, the web page is irrelevant. In the extreme, $R_{pred} = 1$ indicates complete relevance and $R_{pred} = -1$ total irrelevance.

3.2 Finding the Reasoning Rules

Using fuzzy logic theory [7], it is simple enough to build the reasoning rules for a given problem, primarily because human knowledge on the particular field can be easily converted into a group of 'IF-THEN' rules. To design a set of IF-THEN rules, the number of rules, (and then for each) the rule premise, the rule consequence and its parameters must be decided.

The following variables and rules are designed and applied in our system:

Input variables: content vectors, $p_j = < w_{j,1}, w_{j,2}, ... w_{j,t}, ... w_{j,|T|} >$, where j is the identifier code of the j^{th} page, T is the set of terms in VSM, $|T|$ is the number of the elements in the set T, t is a term in the set T, $w_{j,t}$ is the value of the t^{th} term in the content vector - referred to as the *weight* of term t in the j^{th} page.

Output variable: R_{pred}, an evaluation of the similarity of p_j to u_i.

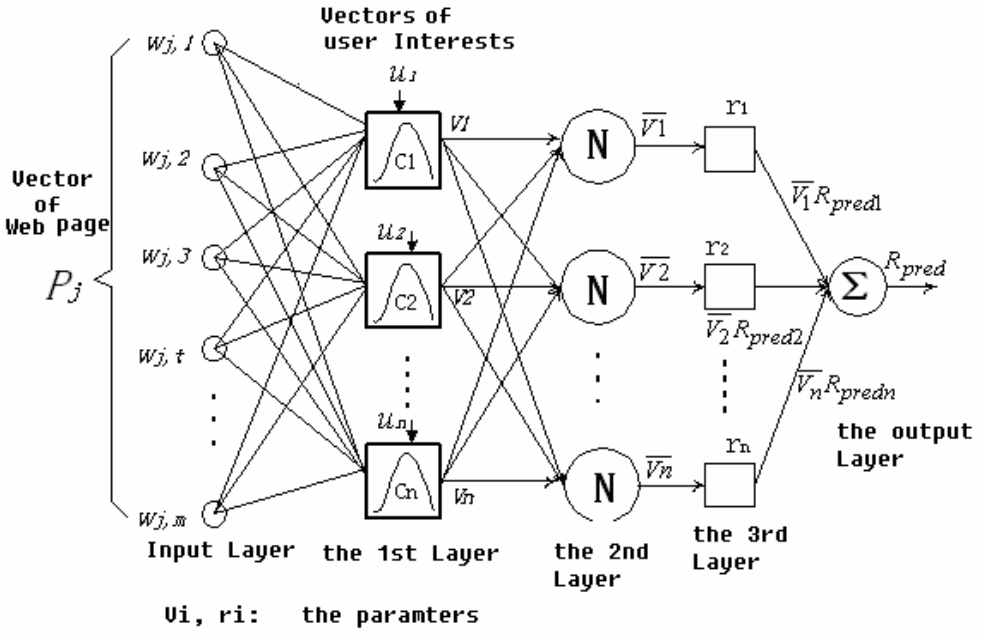

Figure. 3 The Topology of the User Modeling and Information Filtering Network

Parameter sets: ? the rule premise parameters u_i. Assuming a user has n interests, the i^{th} interest is specified by $u_i =< w_{i,1}, w_{i,2}, ...w_{i,t},... w_{i,|T|} >$ *(i? [1,n])*. u_i are non-linear parameters. ? the rule consequence parameters r_i *(1= i= n)* : specifying the weight of the output of the i^{th} rule. r_i are linear parameters.

Set of IF-THEN rules (n rules in total): In vector form, and considering a Sugeno [9] type of fuzzy system to compute IF-THEN rules, these can be further written as

? IF p_j is similar to u_1 THEN $R_{pred1} = f(r_1, p_j)$

? IF p_j is similar to u_2 THEN $R_{pred2} = f(r_2, p_j)$

… … … …

(n) IF p_j is similar to u_n THEN $R_{predn} = f(r_n, p_j)$

where f is a non-linear function.

These calculations are shown in diagram form in Figure 3. This is the core of the user modeling and information filtering process.

3.3 The Information-Filtering Algorithm

Before filtering, the network is initialized through the user inputting the key words and providing favourite pages, the offline-learning algorithm being used to generate the user's interest vectors. After this, the network can work to filter pages.

In the first hidden layer, every node represents one of the user's interests as parameter u_i. The output is the IF-THEN rule premise results V_i, giving the similarity of the content vector Pj to the interest vector u_i. $V_i = Sim(P_j, u_i)$. Where the cosine of the angle between these two vectors can be used as the member function to measure the similarity.

On obtaining the rule premise result V_i, $\overline{V_i} = V_i \Big/ \sum_{i=1}^{n} V_i$ is calculated.

Next $R_{predi} = r_i \times \overline{V_i}$ is calculated in the third hidden layer.

Finally, in the output layer, the similarity of P_j to $\{ u_i \}$ is evaluated:

$$R_{pred} = \sum_{i=1}^{n} R_{predi} = \sum_{i=1}^{n} r_i \times \overline{V_i} = \frac{\sum r_i \cdot V_i}{\sum V_i}$$

4. PARAMETER IDENTIFICATION

From the network structure in Figure 3, two improved learning algorithms are proposed to enhance the adaptability of the fuzzy neural network. The first is the online learning algorithm that identifies the linear rule consequence parameters r_i. Suppose that the rule premise parameters u_i are fixed. By updating r_i, the algorithm attempts to make the total error as small as possible. The error is the difference between the network's evaluation of similarity, R_{pred}, and the user's feedback ranking value . Since our system runs on the client, we adapt the Windrow-Hoff algorithm [10] to update the linear parameters r_i .

The second algorithm is the offline learning algorithm, used to identify the non-linear parameters u_i. According to field knowledge, when the interaction times between user and the system reaches a certain number, there are enough relevant and irrelevant pages as well as these pages' similarities to the user's interests. It is then possible to adjust the user's interest u_i.

Suppose S_{u_i} is the page set of the most relevant pages to user interest u_i. This is the page set that give the greatest value of $Sim(Pj, u_i)$.

$$S_{u_i} = \{ p_j \mid p_j = \arg\max_j (Sim(p_j, u_i)) \}$$

Then $u_{i\ new}$, the new value of u_i, is updated as follows:

$$u_{i\ new} = u_{i\ org} + \frac{1}{|S_{ui}|} \cdot \sum_{p_j \in S_{ui}} p_j \cdot R_{usr\ j}$$ where $S_{usr\ j}$ is the user's feedback ranking value.

5. THE PRELIMINARY EVALUATION

A preliminary evaluation of our system has been undertaken through filtering the result pages provided by the *Google*, then measuring the *precision* (the proportion of relevant pages out of those returned). High precision means most of the recommended pages are relevant, with the irrelevant discarded to avoid information overload. However, the list order is also critical to reduce user effort. The ideal is descending order, with the pages listed from high relevance value to low. The most relevant page should be in first place. The precision measurement does not take order into account. Therefore, we also measure the difference between the order of the pages before and after filtering by calculating the Spearman rank correlation coefficient [11].

Table 1. The Precision of Different Research Methods

	Search Method	**Precision**
Before Filtering (Google Searching directly)	Interest 1: ("Neural Networks")	0%
	Interest 2: ("User Modeling")	0.01%
	Interest 3: ("Information Filtering")	0.01%
	Combining Interests 1 & 2	25.1%
	Combining Interests 2 & 3	18%
	Combining Interests 3 & 1	21.3%
	Combining Interests 1, 2 & 3	33.9%
After Filtering	3 interests described by 11 keywords without favourite pages	35.5%
	3 interests described by 11 keywords with 5 favourite pages	38.1%

During the tests, the user submits three interests in (an average of) 11 keywords and several favourite pages. The pages the user requires are interdisciplinary pages across the three interests. After getting the 261 different returned pages from the *Google*, as the test pages set, O, these are analysed manually to find the relevance values and, from these, the set of relevant pages R, 89 in all.

The results of precision testing are shown in Table 1. The precision is calculated by the formula:

$$\Pr ecision = \frac{|S \cap R|}{|S|} \times 100\%$$

where S is the set of pages recommended to the user.

Table 1 shows that the precision is improved to 35.5% and 38.1% respectively by our information filtering system. Since the user requires pages across the three interests, the precision of searching with a single interest is very low at nearly 0% and the precision of searching with the three interests combined rises to 34.1%. After filtering, this increases to 35.5% without the input of favourite pages and to 38.1% with. It shows that our system can indeed improve the precision. However, the improvement is limited if the interests are entered without favourite pages. Describing interests using favourite pages can give a significant improvement in precision.

In addition to precision, document rankings and orderings are also measured. Prior to applying our system to filter the test set O, according to the pages' relevance values given by the user, we resort R with a list of ordinal numbers, $X = [x_1, x_2, \ldots x_i, \ldots x_n]$ ($n=89$). After filtering, the selected pages recommended by our system are assigned a list of ordinal numbers, $Y=[y_1, y_2, \ldots y_j, \ldots y_n]$ ($n=89$). Using the Spearman rank correlation coefficient, the difference, between the original *Google* ordering and the system output ordering, is measured. This is calculated as

$$RankCorrelation = 1 - 6 \cdot \sum_{i=1}^{N} \frac{d_i^2}{N(N^2-1)}$$

where $N=89$ and $d_i = (x_i - y_i)$

The results of these comparisons are shown in Table 2.

Table. 2 Spearman Rank Correlation Coefficient test results

Search Method	*RnkCo*
Combine Interest 1 & 2 (before filtering)	-1.00
Combing Interest 1, 2, 3 (before filtering)	0.4038
3 interests described by 11 keywords with 5 favourite pages (after filtering)	0.4698

In the first column, the three different search methods are given, whilst the second row shows the three calculated Spearman rank correlation coefficients between X and Y. Before filtering, if only two interests are applied to the external search engine, the ordering looks out of order. This means the most relevant pages found by the engine are placed last, not first. After filtering by our system, the order of the pages presented to the user is closer to the ideal order listed by relevance value. This should be very helpful for users to find what they most want, more quickly.

So we can see that, due to the filtering from the system, the search results are more precise and the order of the presented pages makes it easier for the user to find the pages they most want. These statistical results suggest that our system can be adaptable and useful in improving the precision of the search and reducing the information overload.

6. CONCLUSIONS

In this paper, we have proposed a fuzzy neural network-based approach to building an adaptive personalized Information Filtering system with a simple interface for specifying user requests, capable of selecting and ordering pages returned from the external search engines.

One advantage of this approach is that, using the fuzzy logic principles and IF-THEN rules, it is straightforward to build a reasoning system to process a user's fuzzy information needs, which could be a combination of multi interests. The neural network makes the system adaptive and capable of being personalized. In this work, the preliminary test results show that our information filtering system can improve the precision of search engines and reduce the information overload.

A more extensive test programme is needed to evaluate the performance of the system in user modeling and information filtering. The aims of further tests should include:

1. Comparing and selecting the appropriate similarity function in the first hidden layer. There are two likely candidates for functions. The first, the traditional cosine measure, and the second, the Radial Basis function.

2. Deciding the threshold value of? . This will be a balance between the precision and the level of recall. Is there a better way to select the value of ? other than by experimentation? That is, is a theoretical derivation possible?

This should offer a constructive advance to the field and technologies of user modelling and information filtering. There is a considerable amount of useful work to do with regard to the development of the personal information service.

REFERENCES

[1] Tingshao Zhu, Russ Greiner & Gerald Haeubl (2003), "Learning a Model of a Web User's Interests", *9th International conference on User Modeling, UM'03,2003*

[2] Geoffrey I.Webb & Michanel J.Pazzani etc. (2001) "Machine Learning for User Modeling", *User Modeling and User-Addapted Interaction,* pp19-29.

[3] Aniscar, F.A. & Tasso, C. (1997), "ifWeb: a Prototype of User-Model-Base Intelligent Agent for Document Filtering and Navigation in the World Wide Web", *Proc. of the workshop Adaptive Systems and User Modelling on the World Wide Web, 6th International Conference on User Modelling UM97,*

[4] Cheung K.W. & Tian Lily F. (2004), "Learning user Similarity and Rating Style for Collaborative Recommendation", *Information Retrieval,* Vol. 7, No. 3, pp 395-410

[5] Intelliseek. (2004) *Profusion Web Site*, available from <http://www.profusion.com>, accessed 5 June 2004

[6] Bradley K, Rafter R & Smyth B (2000) "Case-based user profiling for content personalization", *Proc. of the International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, Trento, Italy.

[7] Jyh-Shing Roger Jang & Chuen-Tsai Sun etc (2000) *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*, Prentice-Hall.

[8] Raghavan, V. V. & Wong, S. K. M (1986), "A critical analysis of vector space model for information retrieval", *Journal of the American Society for Information Science*, Vol.37, No.5, pp 279-287

[9] Sugeno, M. & Kang, G.T. (1988), "Structure identification of fuzzy model", *fuzzy sets and Systems*, Vol 28, No. 1, pp15-33

[10] Widrow, B.& Lehr, M.A. (1990), "30 years of adaptive neural networks: Perceptron, madline, and backpropagation", Proceedings of the IEEE, Vol. 78, No. 9, pp 1415-1442

[11] Available from<http://mathworld.wolfram.com/SpearmanRankCorrelationCoefficient.html>, accessed 29 Dec 2004