A Collaborative Filtering System of Information on the Internet

DongSeop Lee[1], HyungIl Choi[2]

Soongsil University, 1-1 Sando-5Dong, DongJak-Gu, Seoul, Korea
[1] leeds@vision.soongsil.ac.kr,
[2] hic@computing.soongsil.ac

Abstract. In this paper we describe a collaborative filtering system for automatically recommending high-quality information to users with similar interests on arbitrarily narrow information domains. It asks a user to rate a *gauge set* of items. It then evaluates the user's ratings and suggests a *recommendation set* of items. We interpret the process of evaluation as an inference mechanism that maps a gauge set to a recommendation set. We accomplish the mapping with FAM (Fuzzy Associative Memory). We implemented the suggested system in a Web server and tested its performance in the domain of retrieval of technical papers, especially in the field of information technologies. The experimental results show that it may provide reliable recommendations.

1 Introduction

In this paper we describe a collaborative filtering system for automatically recommending high-quality information to users with similar interests on arbitrarily narrow information domains. Our system follows the same operational principle as the *Eigentaste* system [4]. It asks a user to rate a *gauge set* of items. It then evaluates the user's ratings and suggests a *recommendation set* of items. We interpret the process of evaluation as an inference mechanism that maps a gauge set to a recommendation set. We accomplish the mapping with FAM (Fuzzy Associative Memory).

FAM provides a framework that maps one family of fuzzy sets to another family of fuzzy sets [3]. This mapping can be viewed as a set of fuzzy rules that associate input fuzzy sets (gauge sets) with output fuzzy sets (recommendation sets). FAM also provides a Hebbian-style learning method that establishes the degree of association between an input and output [2]. This learning method is very simple and takes very little computation time.

Another aspect of collaborative filtering is how to form groups of users with similar tastes, so that the known preference of a group of users may be exploited to predict the unknown preference of a new user. This is a typical problem of clustering [6]. However, our approach does not require this type of explicit clustering, since the clustering is embedded in connection weights of FAM. In fact, FAM generates fuzzy rules that classify data into groups of classes. This grouping is supervised at the stage

P.M.A. Sloot et al. (Eds.): ICCS 2002, LNCS 2331, pp. 1090–1099, 2002.

of learning the connection weights. The details will be discussed in the next section[1].

Our collaborative filtering system consists of two main parts, a learning part and inferring part. The learning part operates off-line. It analyzes training data made up of input and output pairs in order to form fuzzy sets, where input data correspond to a set of rates on a gauge set of items and output data correspond to a set of rates on a recommendation set of items. Our system asks users to rate their preference on a continuous rating scale. To rate items, we may use a horizontal "rating bar" as in [4], where a user is supposed to click a mouse. The learning module then generates a correlation matrix that shows the degree of association between input and output fuzzy sets. The details will be discussed in section 3. The inferring part operates online. It presents a gauge set to a new user. It then processes the rates on the gauge set and draws a recommendation with the fuzzy rules built up in the learning part. Our system makes a conclusion in the form of induced preference rates on a recommendation set. The details will be discussed in section 2.

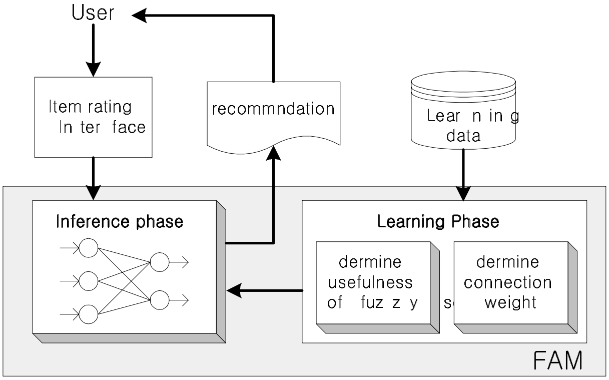

Fig. 1. System organization

2 Inferring Model With FAM

FAM can be viewed as the fusion of associative memory and fuzzy logic. It associates a family of fuzzy sets with another family of fuzzy sets. Figure 2 shows a basic structure of FAM. The antecedent term A_i in fuzzy association (A_i, B_j) denotes an input associant, the consequent term B_j denotes an output associant, and the synaptic weight w_{ij} denotes the degree of association between input and output associants. In our collaborative filtering system, the input associant corresponds to a fuzzy set of a gauge items, the output associant corresponds to a fuzzy set of recommendation items, and the synaptic weight corresponds to the degree of correlation between these two sets. As an example, one may consider fuzzy association ("*high preference of computer*", "*high preference of internet*") with the association degree of 0.8. There are several ways of interpreting the synaptic weight

w_{ij}. One popular interpretation considers it as a fuzzy Hebbian-style correlation coefficient in which the weight is encoded as the minimum of input and output associant values [3]. The details of this interpretation will be discussed in the third section.

If we now somehow encoded the set of synaptic weights, then FAM carries out forward recalling through max-min composition relation [3]. Suppose that a fuzzy set A_i is defined on the domain of an item x_i and $a_i = \mu_{A_i}(x_i)$ is the fit value of x_i to a membership function μ_{A_i}, and a fuzzy set B_j is defined on the domain of an item y_j and $b_j = \mu_{B_j}(y_j)$ is the fit value of y_j to a membership function μ_{Bj}. Then FAM exhibits forward recalling as in (1)

$$A \circ W = B \tag{1}$$

where $\circ$ denotes a max-min composition operator. That is, the recalled component b_j is computed by taking an inner product of fit values a_i's with the *j*th column of W.

$$b_j = \max_i\{\min(a_i, w_{ij})\} \tag{2}$$

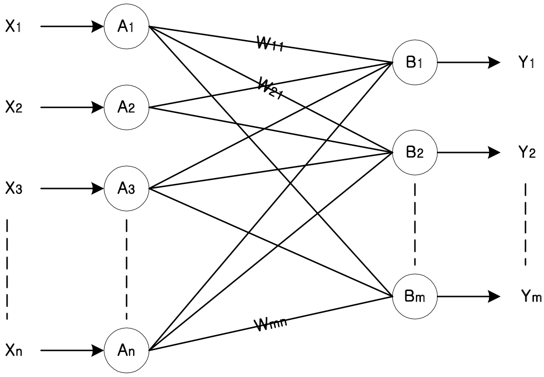

Fig. 2. Fuzzy associative memory

We suggest a fuzzy inference system that employs FAM for implementing fuzzy rules. These rules embody the task of grouping users with similar tastes in the form of association between gauge items and recommendation items. That is, we interpret a gauge items as an antecedent part of a fuzzy rule, a recommendation items as a consequent part, and a synaptic weight as the degree of reliability of the rule. When an antecedent part of a rule has several fuzzy terms, the corresponding input associant forms a conjunction of such fuzzy terms. Figure 3 shows the structure of our inferring model that consists of four layers. For the purpose of illustration, we assume that there are *n* gauge items x_i (*i*=1, ..., *n*) and m recommendation items y_j (*j*=1, ..., *m*), each furnishes p_i fuzzy sets, and each is represented by a fuzzy set B_j.

The input layer of Figure 3 just accepts input values that correspond to rates that a user assigns to a set of gauge items. Thus, the number of nodes in the input layer becomes *n*. The fuzzification layer contains membership functions of input items. Since there are *n* input items and each input item x_i produces p_i fuzzy sets, the total number of nodes in this layer becomes $\sum_{i=1}^{n} p_i$. The output of this layer then becomes the fit values of input rates to associated membership functions. The antecedent layer contains antecedent parts of fuzzy rules, which have the form of logical AND of individual fuzzy terms. This layer requires $N = \prod_{i=1}^{n} p_i$ nodes, since we allow every possible combination of fuzzy sets drawn one from each group of p_i fuzzy sets. Thus, each node in the antecedent layer has *n* incoming links. Each incoming link has a weight that represents the degree of usefulness of an associated fuzzy set. If links from some node of the fuzzification layer have a high value of weight, it means that the fuzzy set contained in the node is very useful in inferring a desired recommendation.

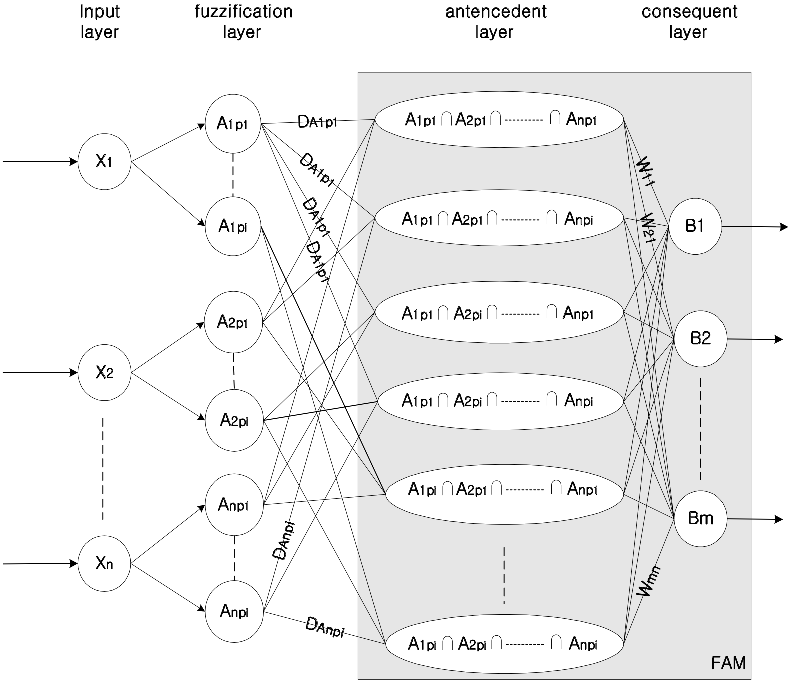
Input layer
fuzzification layer
antencedent layer
consequent layer
X1
X2
Xn
A1p1
A1pi
A2p1
A2pi
Anp1
Anpi
DA1p1
DA1p1
DA1p1
DA1p1
DAnpi
DAnpi
A1p1 ∩ A2p1 ∩ --------- ∩ Anp1
A1p1 ∩ A2p1 ∩ --------- ∩ Anpi
A1p1 ∩ A2pi ∩ --------- ∩ Anp1
A1p1 ∩ A2pi ∩ --------- ∩ Anpi
A1pi ∩ A2p1 ∩ --------- ∩ Anp1
A1pi ∩ A2pi ∩ --------- ∩ Anpi
W11
W21
Wmn
B1
B2
Bm
FAM

Fig. 3. Inference model with FAM

The details of how to determine the weights will be discussed in section 3. Each node of this layer just compares incoming weighted values and takes the minimum of them, since the truth value of an antecedent part of a rule is determined by taking a logical AND of individual truth values of participating fuzzy terms.

The consequent layer contains consequent parts of fuzzy rules. This layer contains *m* membership functions, each of which is to determine the preference of an individual recommendation item. We allow full connections between the antecedent layer and the consequent layer. But, each connection may have a different value of weight, which represents the degree of credibility of each connection. We basically follow the max-min compositional rule of inference. Thus, when *N* antecedent nodes $A_1,\cdots,A_N$ are connected to the *j*th consequent node B_j with weights w_{ij} 's, the output of the *j*th consequent node is a deffuzified value determined as in (3).

$$y^*{}_j = \mu^{-1}{}_{B_j}(\max_{1\le i\le N}\{\min(w_{ij}, output(A_i))\}) \qquad (3)$$

where $\mu_{B_j}(y)$ is a membership function contained in the *j*th consequent node, and $output(A_i)$ is an output of the *i*th antecedent node. The output of each consequent node is a final crisp conclusion in the form of the degree of recommendation on the corresponding item. Here, we use an inverse mapping for deffuzification, which maps the max-min composed membership to the argument of a corresponding function.

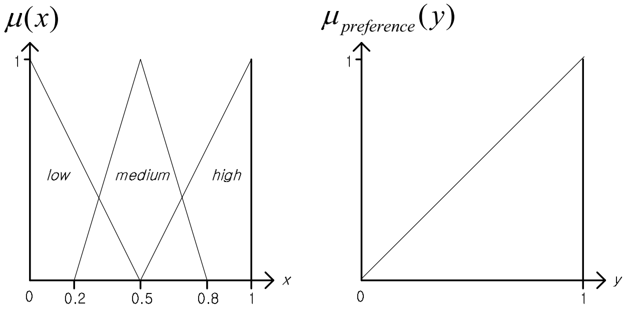

Fig. 4. Basic structures of membership functions

3 Learning Model Based on Observation

Our inferring model can work properly only when membership functions as well as synaptic weights are determined in advance. In this section, we propose a learning method that derives the necessary information from input-output training data that correspond to rates on gauge-recommendation items obtained form users. The first issue is how to determine the number of fuzzy sets for each item and corresponding membership functions. There must be some systematic way to divide the range of each input and output item into subranges and associate each subrange with a proper membership function. For this purpose, one may consider a tuning approach that exploits the distributions of data [2]. Here we take a simple approach and supplement it with the usefulness measure to be described later. Figure 4 shows the basic structures of our membership functions.

A user is supposed to rate his preference on an item by a degree between 0 and 1. A large value denotes a high preference and a small value denotes a low preference. For a gauge items, we divide the entire range into three parts and assign them with *low*, *medium*, *high* fuzzy sets. Their membership functions are defined as in (4).

For a recommendation items, we take an entire range for a fuzzy set "*preference*" and define its membership function to be monotonically increasing, $\mu_{preference}(y) = y$. This is to reduce the number of nodes in the consequent layer of our inferring model. But, one may introduce several fuzzy sets also for a recommendation items, if one wants to be more specific.

$$\mu_{low}(x) = \begin{cases} 1-2x, & 0 \le x < 0.5 \\ 0, & 0.5 \le x \le 1 \end{cases}$$
$$\mu_{medium}(x) = \begin{cases} (10x-2)/3, & 0.2 \le x < 0.5 \\ (-10x+8)/3, & 0.5 \le x \le 0.8 \end{cases} \tag{4}$$
$$\mu_{high}(x) = \begin{cases} 0, & 0 \le x < 0.5 \\ 2x-1, & 0.5 \le x \le 1 \end{cases}$$

To determine the usefulness of an input fuzzy set, we evaluate how the fuzzy set discriminates recommendation items. If an input fuzzy set shows such a characteristic that the training data that fall on the fuzzy set have their corresponding rates on recommendation items evenly distributed, we claim that the fuzzy set is not useful in terms of recommending some specific items. However, if their rates on recommendation items show a spiky type of distribution, we claim that the fuzzy set is useful in inferring recommendation. This criterion is equivalent to emphasizing a distinctive fuzzy set.

$$D_{A_l} = 1 + \sum_{j=1}^{M} P_{A_l}(j) \cdot \ln(P_{A_l}(j))$$
$$P_{A_l}(j) = \frac{\sum_{i=1}^{K} \mu_{A_l}(x_i) \cdot r_{B_j}(x_i)}{\sum_{j=1}^{M} \sum_{i=1}^{K} \mu_{A_l}(x_i) \cdot r_{B_j}(x_i)} \tag{5}$$

To evaluate the discriminating power, we use the measure of entropy. The entropy is a statistical measure of uncertainty, and it is good at revealing the dispersion of data. We compute degrees of usefulness of input fuzzy sets as in (5). In (5), x_i denotes *i*th input-output training data, $\mu_{A_1}(x_i)$ denotes the fit value of x_i to an input fuzzy set A_l, $r_{B_j}(x_i)$ denotes the rate of x_i on a recommendation item B_j, *M* is the number of recommendation items, and *K* is the total number of training data. Thus,

$P_{A_l}(j)$ denotes the probability density of rates on the *j*th recommendation item, which are associated with an input fuzzy set A_l.

Our inference model requires a predetermined correlation matrix that represents the degrees of associations between input and output fuzzy sets. We take a Hebbian-style learning approach to build up the correlation matrix. The Hebbian learning is an unsupervised learning model whose basic idea is that "the synaptic weight is increased if both an input and output are activated." In this way, the phenomena of habit and learning through repetition are often explained. Many artificial neural networks which take a Hebbian learning approach increase network weights according to the product of excitation levels of an input and output. In our fuzzy associative memory, input and output values are fit values to membership functions. Thus, we replace a product operation with a minimum operation and an addition operations with maximum operation. That is, when $a_i(n)$ is an input associant for the *n*th learning datum and $b_j(n)$ is an output associant for the *n*th learning datum, the change of weight is carried out as in (6).

$$w_{ii}(n) = (1-\eta)\cdot w_{ii}(n-1) + \eta\cdot(a_i(n) \quad AND \quad b_j(n)) \tag{6}$$

In (6), η is a positive learning rate that is not greater than 1. This learning rate controls the average size of weight changes. Our inference model allows full connection between the antecedent layer and the consequent layer. This full connection may present too many rules, causing too much computation time. The connection weight may be considered as the confidence measure of a related fuzzy rule, since our learning approach is based on Hebbian learning. We may then consider the rules with small connection weights to be unimportant in inferring a useful conclusion. Thus in practical situations, we may prune some of the connections whose weights are less than some predetermined threshold and reduce the number of rules.

4 Experimental Results and Conclusion

To confirm the effectiveness of our suggested model, we chose the domain of retrieval of technical papers, especially in the field of information technologies. We implemented our learning and inferring module using C++ in a Web server. We also developed Web interface module using *Java* that allows internet users to rate papers. The interface module presents papers and collects ratings as users click on a rating bar. In the learning phase, all the papers in both the gauge set and recommendation set are presented to users one by one. After a user rates each paper, another is presented. The collected ratings are then used to build up membership functions and connection weights of our FAM. In the testing phase, our interface module presents to a user the papers in the gauge set and asks for ratings on the papers. After all papers in the gauge set are rated, our system recommends to the user the papers in the recommendation set with induced preferences. Our system also collects user's ratings on each recommended paper, so that it may compare recommended ratings against user's ratings.

To evaluate the performance of our system, we performed three types of experiments: we first examined the usefulness measure of a fuzzy set in terms of its effect on accuracy of induced preferences. The second experiment was to examine the effect of the number of connections of FAM on the accuracy of induced preferences. We also compared the performance of our system against those of other systems. In our experiments, the learning rate η in (6) was set to 1.0 and the initial weights were set to 0. 100 users were involved in the learning phase, and also 100 different users were involved in the testing phase. The gauge set contains 10 different papers and the recommendation set contains 20 different papers, so the numbers of nodes in input layer and consequent layer are 10 and 20, respectively.

The table I lists the degrees of usefulness of input fuzzy sets. These values reflect how input fuzzy sets discriminate recommendation items. For illustration, A_{21} has the high value of 0.87, since the training data that fall on this fuzzy set have the most of their recommendation rates only in 4 items. In contrast, A_{62} has the small value of 0.17, since the training data that fall on this fuzzy set have their recommendation rates spreaded among 17 items.

Table I. Usefulness of each input fuzzy set

Fuzzy set	Usefulness	Fuzzy set	usefulness	Fuzzy set	Usefulness
A_{01}	0.78	A_{02}	0.85	A_{03}	0.32
A_{11}	0.76	A_{12}	0.44	A_{13}	0.15
A_{21}	0.87	A_{22}	0.92	A_{23}	0.86
A_{31}	0.05	A_{32}	0.46	A_{33}	0.65
A_{41}	0.84	A_{42}	0.65	A_{43}	0.82
A_{51}	0.65	A_{52}	0.14	A_{53}	0.09
A_{61}	0.86	A_{62}	0.17	A_{63}	0.60
A_{71}	0.92	A_{72}	0.82	A_{73}	0.14
A_{81}	0.57	A_{82}	0.65	A_{83}	0.79
A_{91}	0.87	A_{92}	0.88	A_{93}	0.88

$$\begin{aligned} OMAE &= \frac{1}{M} \times \sum_{j=0}^{M} MAE(j) \\ MAE(j) &= \frac{1}{N} \times \sum_{i=0}^{N} \left| r_{ij} - p_{ij} \right| \end{aligned} \qquad (7)$$

To evaluate the accuracy of the performance, we use the Overall Mean Absolute Error metric that is often used in the literature. In (7), p_{ij} is the prediction for how user i will rate item j and r_{ij} is the actual rate given by user i for item j, M is the number of items user i has rated, and N is the total number of users involved in the test.

In (7), *MAE(j)* represents the mean error for the *j*th recommendation item over the whole users, and *OMAE* represents the mean error over the whole items and users. Table II shows the performance of our system with and without the usefulness measure of input fuzzy set. As can be noted, the performance is improved by 0.07 on average with the help of the usefulness measure.

Table II. *MAE* of each recommendation item

Recommendation Item	MAE without usefulness measure	MAE with Usefulness measure	Recommendation item	MAE without usefulness measure	MAE with usefulness measure
Paper01	0.234	0.215	Paper11	0.154	0.142
Paper02	0.164	0.156	Paper12	0.189	0.185
Paper03	0.209	0.201	Paper13	0.220	0.215
Paper04	0.186	0.178	Paper14	0.237	0.231
Paper05	0.160	0.149	Paper15	0.205	0.196
Paper06	0.207	0.203	Paper16	0.156	0.154
Paper07	0.242	0.234	Paper17	0.151	0.147
Paper08	0.219	0.216	Paper18	0.214	0.206
Paper09	0.182	0.179	Paper19	0.178	0.171
Paper10	0.170	0.168	Paper20	0.203	0.194

We examined the effect of the number of synaptic connections on the performance of the system. We pruned out synaptic connections whose weights are less than some threshold *Th*, so that the number of connections is reduced and the system is simplified. For illustration, we have 25% reduced connections for *Th*=0.1, and 35% reduced connections for *Th*=0.2. We also compared the performance of our systems with those of other two systems: POP [5] and Eigentaste [4]. Table III shows the results. As can be noted, our system with *Th*=0.1 surpasses the others. It is interesting to note that the performance of our system does not drop drastically with increasing the threshold *Th*.

Table III. Comparison of performance

System	OMAE
Our system with th = 0.0	0.187
Our system with th = 0.1	0.192
Our system with th = 0.2	0.245
POP[9]	0.302
Eigentaste[6]	0.284

Acknowledgement

This work was supported by the Korea Science and Engineering Foundation (KOSEF) through the Advanced Information Technology Research Center(AITrc).

Reference

1. Dae-Sik Jang, Hyung-Il Choi, Fuzzy Inference system based on fuzzy associative memory, journal of Intelligent and fuzzy systems, Vol.5, 271-284. 1997.
2. Hideyuki T, Isao H, NN-Driven fuzzy reasoning, Int. J. Approximate Reasoning, 191-212. 1991.
3. Kosko B, Neural Networks and Fuzzy Systems," Prentice-Hall International. 1994.
4. Zimmermann HJ, Fuzzy Set Theory and Its Applications, KALA. 1987.
5. Ken Goldberg and Theresa Roeder and Dhruv Gupta and Chris Perkins, Eigentaste: A constant time Collaborative Filtering Algorithm, University of California, Berkeley, Electronics Research Laboratory Technical Report M00/41. 2000.
6. Jonathan Herlocker, Joseph Konstan, Al Borchers, and John Riedl. An algorithmic framework for performing collaborative filtering, In Proceedings of the SIGIR. ACM, August 1999.