

A cognitive approach for agent-based personalized recommendation

Chunyan Miao ^{a,*}, Qiang Yang ^b, Haijing Fang ^c, Angela Goh ^a

^a School of Computer Engineering, Nanyang Technological University, Singapore

^b Hong Kong University of Science and Technology, Kowloon, Hong Kong

^c School of Computing Science, Simon Fraser University, Burnaby, BC, Canada

Received 7 March 2005; accepted 3 June 2006

Available online 26 October 2006

Abstract

There is an increasing need for various e-service, e-commerce and e-business sites to provide personalized recommendations to on-line customers. This paper proposes a new type of personalized recommendation agents called fuzzy cognitive agents. Fuzzy cognitive agents are designed to give personalized suggestions based on the user's current personal preferences, other user's common preferences, and expert's domain knowledge. Fuzzy cognitive agents are able to represent knowledge via extended fuzzy cognitive maps, to learn users' preferences from most recent cases and to help customers make inferences and decisions through numeric computation instead of symbolic and logic deduction. A case study is included to illustrate how personalized recommendations are made by fuzzy cognitive agents in e-commerce sites. The case study demonstrates that the fuzzy cognitive agent is both flexible and effective in supporting e-commerce applications.

© 2007 Published by Elsevier B.V.

Keywords: Intelligent agent; Fuzzy cognitive map; Case-based reasoning; Personalized recommendation

1. Introduction

In recent years, people have seen a huge increase of e-service, e-commerce and e-business applications operating over the Internet [1]. Recommendation systems are increasingly being used by application providers for making suggestions to their customers [2]. However, most of the traditional recommendation systems mainly focus on extracting and recommending the common preferences based on user's historical data [3,4]. Although general users' common preference may be of relevant consideration, an individual user also has his/her own personal preferences. He/she may also reply on the domain expert's knowledge to some extent to make decisions. Moreover, very often, while using traditional recommendation systems, it is not easy for the users to distinguish whether the items contained in a page are actual recommendations or simply the contents of the page which are displayed

indiscriminately to all users. Hence, traditional recommendation systems do not give the customers the impression of being treated individually. Jeff Bezos, CEO of Amazon.com drew a conclusion that "If I have 3 million customers on the Web, I should have 3 million stores on the web" [3]. Personalized recommendation agents are emerging to overcome the impersonal nature of integrated recommendations by using technology to assist customers to do decision-makings in treating each customer individually [5].

Agent technology is one of the most promising technologies for facilitating personalized recommendations. Software agents are being used in an increasingly wide variety of software applications – ranging from comparatively small systems such as personalized e-mail filters to large complex mission critical systems such as air-traffic control [6]. The notable characters associated with software agents such as autonomous, pro-active, goal-oriented, intelligent, social, etc. make software agents well suited for playing the role of personalized recommenders to individual users of various e-service, e-commerce and e-business sites.

* Corresponding author. Tel.: +65 67906197; fax: +65 67926559.
E-mail address: ascymiao@ntu.edu.sg (C. Miao).

The key characteristic of e-commerce/business applications is that they will inevitably move more and more into a customer-centric paradigm in order to increase competitiveness [7]. Therefore, nowadays, there are increasing demands for applying the capabilities of agents to e-commerce/business applications [8]. However the potential of agent's usage in customer-centric paradigm such as personalized recommendations is highly unrealized in the current e-commerce/business applications [9]. An analysis by the Gartner Group reveals that a large proportion of users find difficulty in determining what they really want while visiting electronic commerce sites [10]. This is because the constraints of physical space no longer dictate the organization of information in electronic shopping environments [11]. Online vendors are able to offer a very large number of products to consumers. On one hand, easy access to large amounts of product information allows consumers to select products that better match their personal preferences. On the other hand, human beings have limited cognitive capacity of information-processing. A solution to the above information-overload problem is that users tend to rely heavily upon an electronic agent's product recommendation in order to reduce the amount of effort required to make a purchase decision [5]. However, although a number of electronic commerce sites have already provided shopping agents to assistant users, the help that they offer is very limited [8,9]. What customers expect and require are more knowledgeable shopping agents who are able to understand their exact concerns, represent both their knowledge and domain expert's knowledge, and help them to recommend satisfactory products, similar to what happen in real life shops.

In order to meet the customer's expectation, software agents must have the ability to represent knowledge, learn and reason which remain as research challenges in the area of agent world [12]. First, software agents have to represent user's desires or intentions that are usually imprecisely or vaguely expressed using human language. For instance, a user may give a vague goal such as: "*Locate a flight to Hong Kong on a carrier with a high safely record, is not expensive price and has quality services.*" Secondly, the environment that the agent acts in may contain a lot of uncertain and fuzzy information as a large number of the features that characterize the real world objects are described by imprecise linguistic expressions. Finally, besides the capability of understanding the users and perception of environment, agent must also be able to learn from previous experiences and to infer based on its knowledge.

In this paper, we propose a new type of personalized recommendation agents called fuzzy cognitive agents. Fuzzy cognitive agents are designed to give personalized suggestions to the on-line customers based on the current user's personal preferences, other user's common preferences, and the expert's knowledge. Our focus is on integrating the two types of preferences, which represents our novel contribution to recommendation systems.

The paper is organized into 5 sections. Following this introduction, Section 2 describes the proposed fuzzy cognitive agents. A case study and some experimental results are included in Section 3 to illustrate how personalized recommendations are made by fuzzy cognitive agents in e-commerce sites. Section 4 reviews some related research work. Finally the conclusion and the future work are given in Section 5.

2. Fuzzy cognitive agent

A personalized recommendation agent is conceptualized as a software agent that (a) attempts to understand and represent a human decision maker's preferences with respect to a particular domain or product category, (b) makes recommendations by its learning and inference ability in the form of a sorted list of alternative provided to the human in a decision task based on its understanding of that individual's preferences [5]. In this section, we present a particular type of personalized recommendation agents called fuzzy cognitive agent to meet the requirements and challenges we discussed in Section 1. Fuzzy cognitive agents are able to present the following behaviors:

- Communicate with users.
- Perceive the environment.
- Represent knowledge including current user's personal preferences, other users' common preferences and expert's knowledge, etc.
- Learn from general users' most recent behavior records.
- Make inference based on its represented knowledge.
- Make personalized recommendations to individual users based on current user's personal preferences, general users' common preferences and expert's knowledge.

To enable fuzzy cognitive agents to represent (express) both the user's preferences and the experts' knowledge with learning and inference capabilities, following, we propose an agent knowledge model based on the FCM theory and its extension [13,14].

2.1. The agent's knowledge model

The knowledge model of a fuzzy cognitive agent can be viewed as an extended fuzzy cognitive map. The model comprises two types of objects: *concept* and *weight*. The factors that fuzzy cognitive agents need to perceive within a given environment are represented by concepts. These concepts are connected by weights which indicate there are causal-effect relationships between the concepts. Therefore, the concepts existing in an agent environment can be organized into signed and weighted directed maps.

An example directed graph of the agent knowledge model is shown in Fig. 1. Concepts are indicated by nodes, and weights are presented by directed, signed and weighted edges. The weight edge points from the "cause concept" to the "effect concept". The value of the weight describes the

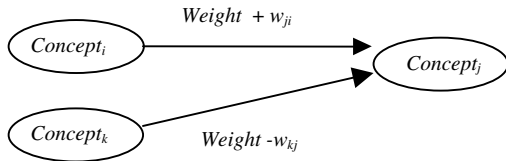


Fig. 1. The directed graph of agent knowledge model.

strength of the cause–effect. A positive sign expresses an increase cause–effect while a negative sign illustrates a decrease cause–effect between concepts. For instance, in Fig. 1, the higher the value of the Concept_{*i*} is, the higher the value of the Concept_{*j*} will be. In contrast, the higher the value of the Concept_{*k*}, the lower the value of the Concept_{*j*} will be.

Definition 1. An agent knowledge model of fuzzy cognitive agents M_{FCA} , consisting of concepts and weights, is defined as a 2-element tuple: $M_{FCA} = \{C, W\}$:

- $C = \{c_i | c_i \in [-1, 1]; i = 1, 2, \dots, n\}$ represents a set of concepts.
- $W = \{w_{ij} | w_{ij} \in [-1, 1]; i = 1, 2, \dots, n; j = 1, 2, \dots, n\}$ represents a set of weights.

As defined in our agent knowledge model, both the concept and the weight have a fuzzy value in $[-1, 1]$. There are mainly three elements for modeling causal relationships for a real world problem, the cause, the effect, and the cause–effect relationship. Cognitive maps [15] model the causal relationships between concepts without describing how strong the relationship is. It only has a concept value set in binary form $\{0, 1\}$. Fuzzy cognitive maps improve the cognitive map by describing the strength of the cause–effect relationship using a fuzzy weighted value in $[-1, 1]$ [16]. However, a fuzzy cognitive map still leaves the concept values in a binary form within $\{0, 1\}$. The binary value of a concept can indicate whether the concept is active or not. However, it is impossible for the binary value to indicate the degree of how active the concept is. To enable the fuzzy cognitive agents to express the complex status of the concepts in the real world, we extend the FCM to allow the concept values to be a member value in a fuzzy set ranged $[-1, 1]$.

Moreover, in order to represent the imprecisely or vaguely expressed concepts discussed in Section 1, each concept has a *fuzzy mapping* function, and a *decision* function. The fuzzy mapping functions maps the original expression for a concept in the real world into a fuzzy concept value defined in the agent knowledge model. The decision function decides the status (value) of the concept upon receiving the effects from other concepts. The decision function of a concept takes all impacts together into account, and works out a new value of the concept.

Assuming the value of Concept_{*i*} is denoted by x_i , $i = 1, 2, \dots, n$; the causal effect from Concept_{*j*} to Concept_{*i*} is y_{ij} , $j = 1, 2, \dots, n$; the decision function of factor i , d_i decides the new value of Concept_{*i*} as

$$x_i = d_i(y_{i1}, y_{i2}, \dots, y_{in}) \tag{1}$$

This function has to be provided by the designers in order to fully specify the FCM itself.

2.2. The agent knowledge representation

The agent knowledge model based on the extended fuzzy cognitive maps has rich capability in knowledge representation. A case study is introduced that will be used to illustrate how fuzzy cognitive agents can act as personal assistants for giving personalized recommendations to individual on-line users in a used car electronic market over the Internet.

This case study involves a car-purchasing domain from Canada’s largest database of new and pre-owned cars which can be found at www.carclick.com. Using a crawler agent, a set of online listings is traced. It is assumed that buyers have a high degree of satisfaction corresponding to the car if a record in the listing is sold in a short period. Otherwise, if a record has been there for a long time, say, one month, we assume the car is not attractive to buyers. According to the domain expert’s knowledge, buyers are mainly concerned about following five attributes (i.e., concepts) of second-hand cars: *price*, *make*, *model*, *mileage* and *age*. The agent knowledge model can then be constructed to include the above concepts and causal connections (causal–effect relationships) between the concepts. The causal connections are derived by combining the users personal preferences and the expert’s knowledge.

Fig. 2 shows the initial agent knowledge model set up for the case study based on the user’s personal preference and the expert’s knowledge. As we can see, the price, mileage and age have an inverse on the satisfaction degree of the buyer. For instance, the higher the price, the less the buyer will be satisfied. In contrast, the model and manufacturer are directly related to the satisfaction degree. For instance, the newer the model, the more the buyer will be satisfied.

Both the user’s personal preference and the expert’s knowledge are well integrated and represented by the agent knowledge model. With all the causal relationships and weights specified by the agent knowledge model, fuzzy

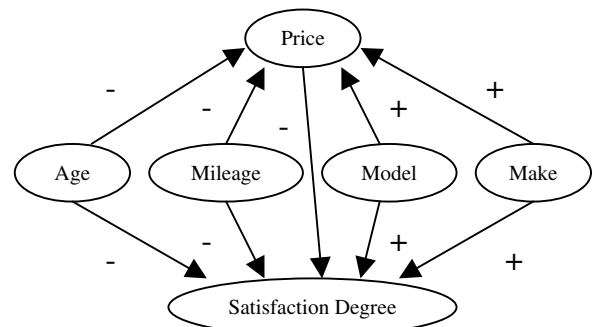


Fig. 2. An example of agent knowledge model based on the case study.

cognitive agent will be able to do the inference based on its represented knowledge.

2.3. The agent inference

As described, the directed graph of the agent knowledge model can be viewed as an extended FCM. Based on FCM theory, it is known that the state value of a given concept node can be obtained from the prior state value of all the causal concept nodes that affect the given concept node. The inference can be carried out as follows:

Assuming that there are n concept nodes in an agent knowledge model, a $n * n$ Weight Matrix (W) can be built:

$$W = \begin{bmatrix} w_{11} & w_{12} & w_{13} & \dots & w_{1n} \\ w_{21} & w_{22} & w_{23} & \dots & w_{2n} \\ w_{31} & w_{32} & w_{33} & \dots & w_{3n} \\ \dots & \dots & \dots & \dots & \dots \\ w_{n1} & w_{n2} & w_{n3} & \dots & w_{nn} \end{bmatrix}$$

A $1 * n$ Concept Matrix (C) can be also built:

$$C = [x_1, x_2, x_3, \dots, x_n]$$

By multiplying Matrix (C) and Matrix (W), a new output $1 * n$ Matrix (O) is obtained:

$$O = C * W = [y_1, y_2, y_3, \dots, y_n]$$

where

$$y_i = \sum_j w_{ij} x_j \quad (2)$$

y_i is the sum of the products of the state value of all the causal concepts of $Concept_i$ and the weight values between the two concept nodes. Obviously, y_i represents a value that is related to the new status of $Concept_i$, as a result of the effects of its causal concepts. Taking y_i as the input of the decision making function of the $Concept_i$, the state value of the $Concept_i$ can be further computed:

$$x_i = d_i(y_i) = d_i \left(\sum_j w_{ij} x_j \right) \quad (3)$$

where x_i is the new state value of the $Concept_i$, w_{ij} is taken from the weight matrix W and d_i is the decision making function of $Concept_i$.

Therefore, each step of the inference becomes a matrix multiplication, followed by the concept decision making function. Recall the case study introduced in Section 2.2 (Fig. 2), the fuzzy cognitive agent can thus infer the user's level of satisfaction based on the user's personal preference and expert's knowledge, and recommend those cars with high value of user's level of satisfaction.

2.4. The agent learning

As described, it is relatively easy for fuzzy cognitive agents to represent the knowledge using extended fuzzy

cognitive maps and make reference based on its knowledge. In order to recommend the previous users common preferences, the fuzzy cognitive agents also need to learn and elicit them from users past behaviors. However, fuzzy cognitive map theory itself does not have a learning mechanism to support the agent learning. In the following, we describe a novel approach for fuzzy cognitive maps learning based on the user's previous behaviors. The theoretical foundation of our approach includes case based reasoning [17], self-organization map [18], and neural network learning [19].

We regard each user's past behavior as a case. A case base consists of all the past cases i.e., the users' past behavior records. The initial structure of fuzzy cognitive map is set up based on the domain expert's knowledge. To learn the concept state values, an algorithm based on the self-organized map learning mechanism is used here to learn the member function for transferring the case base into fuzzy case base:

For each concept node in a fuzzy cognitive map, we have a set of data $X = (x_1, x_2, \dots, x_n)$ related to its state values from the case base, assuming that the initial mean values m_1, m_2, \dots, m_k are assigned arbitrarily, where

$$\min(x_1, x_2, \dots, x_n) < m_i < \max(x_1, x_2, \dots, x_n)$$

The data are then grouped around the initial means according to

$$|x_j - m_c| = \min_i \{|x_j - m_i|\} \quad 1 \leq i \leq k \text{ and } 1 \leq j \leq n \quad (4)$$

where m_c is the mean with which the datum x_j associates.

Data groups and the mean values are optimized by the following iterative process:

Let $x_j(t)$ be an input and $m_c(t)$ the value of m_c at iteration t ($t = 0, 1, 2, \dots$), then

$$m_c(t+1) = m_c(t) + \alpha(t) + [x_j(t) - m_c(t)] \quad (5)$$

if x_j belongs to the grouping of m_c , and

$$m_c(t+1) = m_c(t) \quad (6)$$

if x_j does not belong to the grouping of m_c

Note that $\alpha(t)$ [$0 < \alpha(t) < 1$] is a monotonically decreasing scalar learning rate. The iteration stops at a certain number of cycles decided by the user or when the condition $|m_c(t+1) - m_c(t)| \leq \delta$ is satisfied, where δ is an error limit assigned by the user. The variances of membership functions can be determined by Eq. (7) below:

$$\sigma_i = \frac{1}{R} \sqrt{\frac{1}{p_i} \sum_{j=1}^{p_i} (x_j - m_i)^2} \quad (1 \leq i \leq k) \quad (7)$$

where σ_i = variance of membership function i , m_i = mean of membership function i , x_j = observed data sample, k = total number of membership function nodes, p_i = total number of data samples in i th membership function group, and R = overlap parameter.

Then the member function can be further obtained:

$$\exp_i = -\left(\frac{x_i - m_i}{\sigma_{ij}}\right)^2; \quad \text{and} \quad f_i = e^{\exp_i} \quad (8)$$

In this expression, x_i is the input and m_i and σ_i are the mean and variance of the membership function respectively.

The learned member function for each concept node in the fuzzy cognitive map acts as the fuzzy mapping function to transfer the real customer behavior attribute value to the concept state value of the fuzzy cognitive map.

In the next step, we show how the weights of fuzzy cognitive maps can be learned from the fuzzy cases. By mapping the nodes to the neurons of neural network, fuzzy cognitive maps can be viewed as a fuzzy neural network. Therefore, the changes of weights in fuzzy cognitive map can be simulated as changes of synapses in a fuzzy neural network.

Assuming that there are n concepts nodes in a FCM, as described in Section 2.3, the weights of FCM constructs a $n \times n$ Weight Matrix (W). The state value of the concepts in the FCM is defined as a Concept State Vector (C)

$$C = [x_1, x_2, x_3, \dots, x_n]$$

Each causal–effect relationship in FCM is regarded as an input to the concept from its causal concept. The input to the concepts is defined as a Concept Inputs Vector (I)

$$I = [y_1, y_2, y_3, \dots, y_n]$$

The output value of the concepts in the FCM is defined as a Concept Output Vector (O)

$$O = [o_1, o_2, o_3, \dots, o_n]$$

Then y_i sums all the input of the Concept $_i$

$$y_i = \sum_j w_{ij}x_j \quad (9)$$

And the output of Concept $_i$ at next step is computed as follows:

$$o_i = \frac{1}{1 + e^{(-y_i)}} \quad (10)$$

For a giving training set of concept value $[x_1, x_2, x_3, \dots, x_n]$, using the delta error learning algorithm, the estimated weights of the FCM can be obtained by minimizing the overall error

$$E = \frac{1}{2} \sum_i \sum_t (x_i(t) - o_i(t))^2 \quad (11)$$

where $o_i(t)$ is the computed value of the concept i at time t , $i = 1, 2, \dots, n$, t is a sequential discrete time interval. The weight matrix is initialized according to expert's knowledge. The weight changing rule is derived from gradient descent method.

As illustrated, through the above method, both the concept value and the weights of the FCM can be learned from the user's previous behaviors records. The learned FCM represents the common preferences of the historical users. When t is close to the current user's online time, the learned

FCM represents the most recent common preferences of the users.

2.5. The agent recommendation

Recent empirical research shows that due to the limited information-processing capability of the human mind, users tend to rely heavily upon the personalized agent's product recommendation to make a purchase decision [20]. Therefore, it is crucial for e-commerce/business/service sites to provide knowledgeable recommendation agents to online users.

The workspace of a fuzzy cognitive agent consists of a case base that keeps all the users' past behavior records and a knowledge base that maintains the agent's knowledge represented by extended FCMs. A fuzzy cognitive agent serves online users as a personal assistant in the following ways: At the beginning, a fuzzy cognitive agent only carries the expert's knowledge that is represented by an initialized extended FCM. Then the agent will learn the weights from the historical users' behavior records. The learned weights represent general users' common preferences. They are kept in the knowledge base. The agent is then able to do inference with the learned FCMs to generate recommendation lists. In order to elicit users' most recent common preferences, the agent keeps learning from users' most recent behaviors, and updating its knowledge frequently.

When a fuzzy cognitive agent servers an online user, the agent will learn his/her personal preferences from his/her own past behavior records. The agent also obtains its user's own current preferences through personalized interaction. Recall the introduced case study in Section 2.2, for the following concepts {*mileage*, *age*, *price*}, user A may specify their importance to him/her as {*not so important*, *important*, *very important*} while user B may specify their importance differently as {*very important*, *very important*, *not so important*}. The agent will then adjust the affected weights of learned FCM based on its user's current preferences. By applying the following algorithm, a fuzzy cognitive agent is able to learn, infer and recommend its user with two top lists: one is based on unadjusted weights that integrate general users' common preferences and expert's knowledge; the other one is based on adjusted weights that combine the individual user's preferences, expert's knowledge and general users' common preferences.

Algorithm. Combining online individual user's preferences, experts' knowledge and general users' common preferences to generate the recommendation lists.

Input: Initialized n -concept node FCM0 based on experts' knowledge; Individual user's personal preferences $P = \{p_{ij} \mid i = 1, \dots, n; j = 1, \dots, n\}$.

Output: Recommendation list RL1 and RL2.

Method:

Step 1: Retrieve the learned weights $L = \{w_{ij} \mid i = 1, \dots, n; j = 1, \dots, n\}$ of FCM0;

Step 2: For $i = 1$ to n

```

For j = 1 to n
  w0ij = wij; # where w0ij is the initialized weight
  based on experts' knowledge
End for j
End for i
Step 3: Do inference based on w0ij to compute RL1;
Step 4: Learn the personalized weights  $K = \{wk_{ij} |$ 
 $i = 1, \dots, n; j = 1, \dots, n\}$  using the user's past
behavior records based on current weights  $L$ .
Step 5: For i = 1 to n
  For j = 1 to n
    If pij is null
      w0ij = wkij
    Else
      wpij = fuzzify(pij);
      If |wpij - wkij| < e # where e is a pre-defined
      allowable error
        w0ij = wkij
      Else
        w0ij = wpij
      End if
    End if
  End for j
End for i
Step 6: Do inference based on w0ij to compute RL2
Step 7: Return RL1, RL2

```

The user will then make purchase decision based on the two top lists recommended by its agent. We see that fuzzy cognitive agents are able to give personalized recommendations that combine individual user's personal preferences, general user's common preferences and the expert's knowledge. We believe the availability of fuzzy cognitive agents in an electronic shopping environment will result in a substantial reduction in the amount of consumer's pre-purchase information search for doing decision makings.

3. Case study and the experimental result

A prototype of the fuzzy cognitive agents has been implemented using JADE (Java Agent Design Environment). Fig. 3 shows the design of the fuzzy cognitive agents in UML (Unified Modelling Language) [21]. UML is a graphical representation language originally developed to standardise the design of object-oriented systems. Recently, UML has been extended for modelling and analysing the agent systems [22].

From the class diagram of the fuzzy cognitive agents shown in Fig. 3, we can see that the fuzzy cognitive agent possesses a goal as well as knowledge to act towards its goal autonomously. It is able to perceive its environment and communicate with each other. Moreover, based on its knowledge, the agent can learn, make inference and determine the action to be taken. The knowledge class implements the agent knowledge model described in Section 2.1, and the learning and reasoning class realizes the

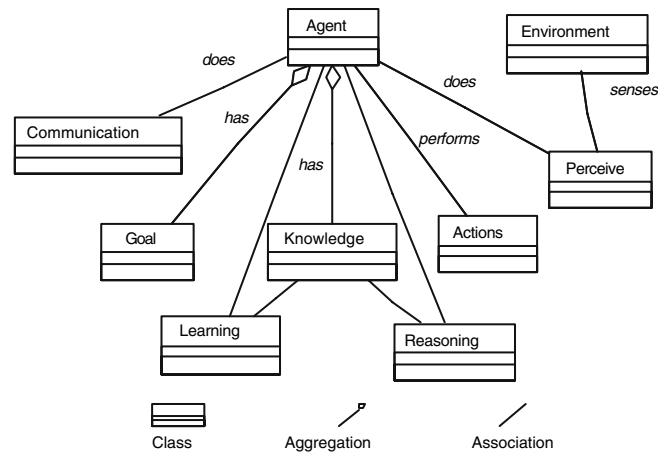


Fig. 3. The design of fuzzy cognitive agents.

learning and inference algorithms given in Section 2.3 and Section 2.4, respectively.

Following the same case study introduced in Section 2.2, a web crawler was used to keep collecting the most recent online data. Fig. 4 shows a snapshot of this data forms the case base of the past behavior of buyers.

In the definition of the agent knowledge model, the value of each concept ranges from -1 to 1 . In order to normalize the numerical attribute data, the member function learning algorithm described in Section 2.4 is implemented to map real world values to the fuzzy concept state value. The case base can then be further transferred to a fuzzy case base which is used for the weight learning. By mapping the nodes of the agent knowledge model to the neurons of the neural network, the agent knowledge model for our case study can be transferred to a two-layer fuzzy neural network shown in Fig. 5.

Therefore, the weight in the agent knowledge model can be learned by simulating the synapses learning in the fuzzy neural network. The experimental result shows that the computed satisfaction degrees for recommendations are well matched to the actual satisfaction degree. The comparison of the computed values and the actual values is show in Fig. 6.

Encouraged by the above experimental result, we further map the weights (causal relationships between the concepts) in the agent knowledge model to the synapses of the fuzzy neural network. Note that in the above experiment, only nodes in the agent knowledge model have been mapped to the neurons of the fuzzy neural network. All the synapses are considered as unknown relationships between the neurons. In fact, from the agent knowledge model, it is known that there is a causal relationship between two concepts, and the relationship is a positive or negative cause-effect. Hence, some synapses of the neural network can be initialized during the training process. For instance, it is obvious that there is no causal relationship between the *make* and the *mileage*. Therefore, the synapse which points from the *make* to the *mileage* can be initialized to zero.

Age	Make	Model	Mileage	Price	Satisfaction degree
1998	Honda Accord	Ex-v6	87000	23000	High
1998	Honda Accord	Lx	76800	17950	Middle
1990	Honda Accord	Ex-r	116000	4750	High
1992	Honda Accord	Ex	160000	6700	High
1994	Honda Accord	Ex-r	108000	9870	High
1998	Honda Accord	Lx	107000	16500	Middle
1998	Honda Accord	Lx	85000	16950	Middle
...					

Fig. 4. A snapshot of the training data.

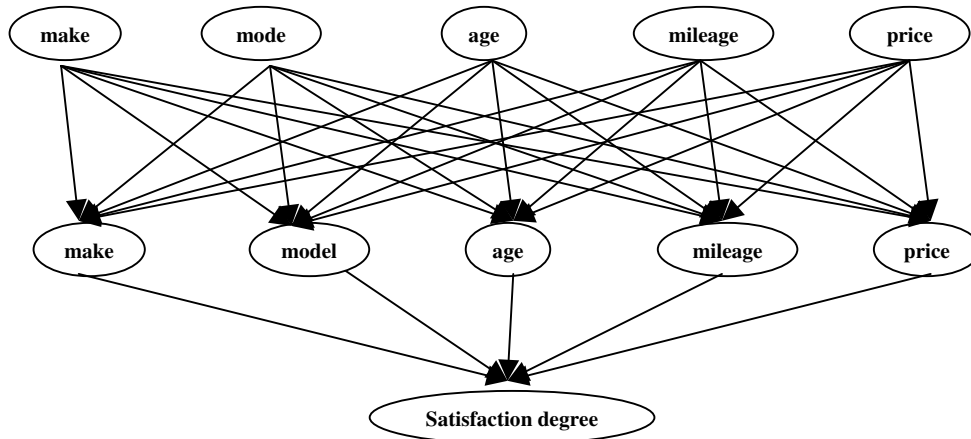


Fig. 5. Mapping the agent knowledge model to fuzzy neural network.

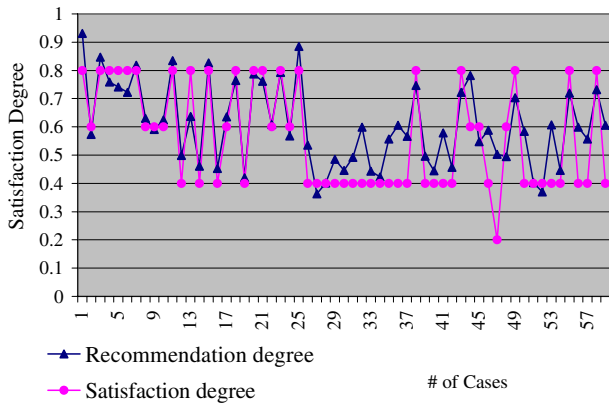


Fig. 6. Comparison of the computed values vs. actual values (1).

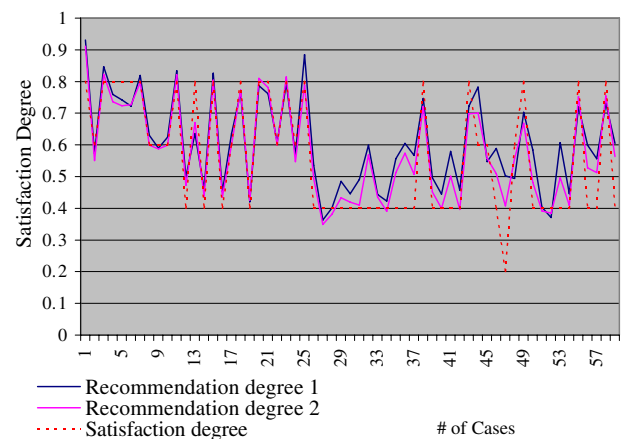


Fig. 7. Comparison of the computed values vs. actual values (2).

The result of 2nd experiment compared with the 1st experiment is shown in Figs. 7 and 8. Fig. 8 summarizes MSE (Mean Squared Error) and Accuracy for both of the experiments. The number of corrected cases in Fig. 8 is counted using closest point method. As we can see, the computed recommended values of the 2nd experiment are closer to the actual values than that of the 1st experiment. Moreover, the learning speed is enhanced in doing the 2nd experiment.

4. Related research

Most of the early recommender systems are based on database queries [3]. To date, there are still many of com-

Measurement	1 st Experiment	2 nd Experiment
# of Cases	59	59
# of Corrected Cases	43	47
MSE	0.01189	0.00594
Accuracy	72.881%	79.661%

Fig. 8. Error measure and accuracy of the experiments.

mercial recommender systems that only use database query techniques. An even earlier approach for recommender systems is the well-known nearest neighbor collaborative filtering algorithms [23,24]. Nearest neighbor algorithms are based on computing the distance between consumers based

on their preference history. However, the search for good neighbors in a large database is very slow [4]. In Bayesian networks, each node and edge of the model represents some consumer information. These can be trained off-line using historical data [25]. Association rules express the relationships between variables such as one product is often purchased along with another product [4,26]. The number of possible association rules grows exponentially with the number of products in a rule. Most of the existing methods as described above have a clear goal: recommending information based on other people's previous experiences.

In many cases, while considering other users' common preferences, an individual user also has his/her own specific preferences, and relies on the domain expert's knowledge to some extent. The final decision of the user is based on the combination of the above knowledge.

Fuzzy decision trees combine symbolic decision trees with approximate reasoning offered by fuzzy representation [27]. But the trees grows very fast if many factors are taken into account. Yet they are difficult to represent causal relationships between factors. Users often make decisions based on the impact factors and the causal relationships between them.

In this paper, we propose a new approach that combines the on-line user's personal preferences, general user's common preference from users' most recent experiences, and expert's knowledge for personalized recommendations. The extended FCM that represents current user's preference can be easily constructed instantly by fuzzy cognitive agents while interacting with the individual on-line users. The users common preference together with expert's knowledge can also be represented by the fuzzy cognitive agents via extended FCM which its weights are learned from general user's most recent experience. In today's rapid changing environment, users may also change their preferences from time to time. Fuzzy cognitive agents have the ability of keep learning the users' common preference from their most recent behavior records and update its knowledge autonomously and frequently. The inference algorithm of the extended FCM can be regarded as a similarity function for recommending those information that closely matches both the current user's personal preferences and the general users' common preferences combined with the expert's knowledge.

5. Conclusion and future work

This paper proposes a new type of personalized recommendation agent, fuzzy cognitive agent, to give personalized suggestions based on the current user's preferences, general user's common preferences, and the expert's knowledge. Fuzzy cognitive agents are able to represent knowledge via extended fuzzy cognitive maps, to learn user's common preferences from most recent cases and to help customers to make inference/decisions through numeric

computation instead of symbolic and logic deduction. The case study shows that fuzzy cognitive agents can be applied into various e-commerce/business/service applications.

Our future work consists of two aspects: (1) The evaluation of the recommendation methods compared with other methods. (2) The design and implementation of the fuzzy cognitive agent factory as an agent service centre that generates fuzzy cognitive agents for electronic commerce sites.

References

- [1] R. Kalakota, M. Robinson, *E-Business: roadmap for success*, Addison Wesley Longman, Inc, 2000.
- [2] J.B. Schafer, J.A. Konstan, J. Riedl, Recommender Systems in E-Commerce, in: *ACM Conference on Electronic Commerce (EC-99)*, 1999, pp. 158–166.
- [3] J.B. Schafer, J.A. Konstan, J. Riedl, E-commerce recommendation applications, *Data Mining and Knowledge Discovery* 5 (2001) 115–152.
- [4] B. Sarwar, G. Karypis, J. Konstan, J. Riedl, Analysis of recommendation algorithms for e-commerce, in: *Proceedings of ACM E-Commerce*, 2000.
- [5] G. Häubl, K. Murray, Recommending or Persuading? The Impact of a Shopping Agent's Algorithm on User Behavior, in: *Third ACM Conference on Electronic Commerce*, Tampa, Florida, USA, 14–17 October, 2001.
- [6] N.R. Jennings, K. Sycana, M. Wooldridge, A roadmap of agent research and development, *Autonomous Agents and Multi-Agent Systems* 1 (1) (1998) 7–38.
- [7] M.P. Papazoglou, Agent oriented technology in support of E-business, *Communication of the ACM* 44 (4) (2000).
- [8] R. Guttman, A. Moukas, P. Maes, Agent-mediated electronic commerce: a survey, *Knowledge Engineering Review* 13 (1998).
- [9] C.Y. Miao, Angela Goh, Y. Miao, Z.H. Yang, Agent that models reasons and makes decisions, *Knowledge Based Systems* 15 (Issue 3) (2002).
- [10] J. Wilderman, *The Es Have It: E-Business, E-Commerce, E-tailing and the Web*, Gartner Group Research Report, 2000.
- [11] J. Eric, L. Gerald, N. Mandel, *Computer Based Choice Environments: Four Approaches to Designing Marketplaces of the Artificial*, working paper, Columbia University.
- [12] H. Nwana, D. Ndumu, A perspective on software agents research, *Knowledge Engineering Review* 14 (2) (1999) 1–18.
- [13] B. Kosko, Fuzzy cognitive maps, *International Journal Man-machine Studies* 24 (1986) 65–75.
- [14] C.Y. Miao, Angela Goh, Y. Miao, Z.H. Yang, A dynamic inference model for intelligent agent, *International Journal of Software Engineering and Knowledge Engineering* 11 (5) (2001) 509–528.
- [15] R. Axelrod, *Structure of Decision: The Cognitive Maps of Political Elites*, Princeton University Press, Princeton, New Jersey, 1976.
- [16] Y. Miao, Z.Q. Liu, On causal inference in fuzzy cognitive map, *IEEE Transactions on Fuzzy Systems* 8 (1) (2000) 107–120.
- [17] M. Lenz, H.D. Burkhard, B. Bartsch-Sporl, S. Wess, *Case-Based Reasoning Technology: From Foundations to Applications*, Lecture Notes in Artificial Intelligence, 1400, Springer-Verlag, 1998.
- [18] T. Kohonen, The self-organization map, *Proceedings of the IEEE* 78 (1990) 1464–1480.
- [19] M. Anthony, P.L. Bartlett, *Neural Network Learning: Theoretical Foundations*, Cambridge University Press, Cambridge, New York, 1999.
- [20] H. Gerald, V. Trifts, Consumer decision making in online shopping environment, *Marketing Science* 19 (1) (2000) 4–21.
- [21] B. Bauer, J.P. Muller, J. Odell, Agent UML: A Formalism for Specifying Multiagent Software Systems, in: *Proceedings of the First*

- International Workshop on Agent-Oriented Software Engineering (AOSE-2000), Springer-Verlag, Berlin, Germany.
- [22] J. Schumann, J. Whittle, Automatic Synthesis of Agent Designs in UML, in: *Proceedings of Workshop on Agent-based Systems*, Springer, 2000.
- [23] J. Herlocker, J. Konstan, A. Borchers, J. Riedl, An algorithmic framework for performing collaborative filtering, in: *Conference on Research and Development in Information Retrieval*, 1999.
- [24] N. Good, J. Schafer, J. Konstan, A. Borchers, B. Sarwar, J. Herlocker, J. Riedl, Combining collaborative filtering with personal agents for better recommendations, in: *Proceedings of the Sixteenth National Conference on Artificial Intelligence*, 1999.
- [25] Y.-H. Chien, E.I. George, A bayesian model for collaborative filtering, in: *Proceedings of the Seventh International Workshop on Artificial Intelligence and Statistics*, Morgan Kaufmann, San Francisco, 1999.
- [26] N. Megiddo, R. Srikant, Discovering predictive association rules, *KDD*, 1998.
- [27] C.Z. Janikow, Fuzzy decision trees: issues and methods, *IEEE Transactions on Systems, Man, and Cybernetics* 28 (1) (1998) 1–14.