

Modeling a User's Domain Knowledge With Neural Networks

Qiyang Chen

Department of Information and Decision Sciences,
Montclair State University

A. F. Norcio

Department of Information Systems,
University of Maryland, Baltimore County

This article presents a neural network approach for user modeling. A set of neural networks is utilized to represent and infer users' task-related characteristics. These networks function as associative memories that can capture the causal relations among users' characteristics for the system adaptation. It is suggested that this approach can be expected to overcome some inherent problems of the conventional stereotyping approaches in terms of pattern recognition and classification of user characteristics.

1. INTRODUCTION

It is well recognized that an interface system can exhibit cooperative behavior if it establishes and maintains a set of assumptions (i.e., system beliefs) about users' task-related characteristics. The construction of this set of assumptions is often referred to as user modeling. The assumptions may vary depending on the application domain. Usually, they are related to users' plans, goals, and domain knowledge as well as their cognitive preferences (Norcio & Stanley, 1989). These assumptions can be categorized in terms of several dimensions. Rich (1979) proposed a three-dimensional taxonomy of user models: The short-term versus long-term dimension concerns the persistence of modeled user characteristics over time. The explicit versus implicit dimension concerns whether the models are built by users themselves or by the system. The canonical versus individual dimension addresses whether a model is specified for a single user or for a group of users. Other dimensions have also been proposed (Norcio & Stanley, 1989). Generally, the dimensional categorization of user models aims at classifying the assumptions about users in terms of the time period when they are valid, the way they are elicited and represented, as well as the degree by which they are specified (Chen & Norcio, 1994).

Requests for reprints should be sent to Qiyang Chen, Montclair State University, Department of Information and Decision Sciences, Upper Montclair, NJ 07043. E-mail: chenq@saturn.montclair.edu.

Most user modeling techniques use the stereotype approach to initialize the system's assumptions. This approach predefines the assumptions into groups (i.e., stereotypes) and organizes them into a generalization hierarchy. Stereotyping concerns the classes of users and makes rapid inferences about users' characteristics. In the generalization hierarchy, a stereotype is defined by a certain number of assumptions. The upper level stereotypes characterize generic information, and the lower level stereotypes carry specific information and inherit the assumptions from their ancestors. Figure 1 illustrates the framework of the stereotyping approach. An individual user profile is actually a substructure of the stereotype hierarchy. The function of the inference mechanism is to extract the substructures ascribed to users without causing any inconsistency.

During a human-computer interaction, user modeling proceeds with stereotype assignment through default reasoning, which allows the model to retain the stereotypical knowledge about a user in the absence of evidence to the contrary. This approach provides a simple way to initiate the modeling process and is successful in some applications. However, this approach has several limitations. These include the following.

First, because the reasoning process is based on default assumptions that may conflict with the new evidence obtained as the interaction progresses, the revision of stereotypical knowledge is necessary to handle the inconsistencies. A common suggestion is to use a dependency-directed backtracking process, referred to as *truth maintenance*, to eliminate the inconsistencies (de Kleer, 1986). This process utilizes the sequential logic to examine one piece of information at a time. It is often inefficient and lacks the ability to detect noisy or inconsistent information that should be ignored (Chen & Norcio, 1994). Therefore, it is possible that the effort of maintaining consistency may bring further conflicts to the subsequent interaction. Thus, model construction may fall into a dilemma that a nonmonotonic process of reconciling conflicts is frequently involved and eventually no decision can be made after a period of interaction (Huang et al., 1991).

Second, the predefined generalization hierarchy limits the system assumptions within each stereotype that can be inherited only by the descendant stereotypes. Therefore, it is difficult to update those assumptions that are no longer significant

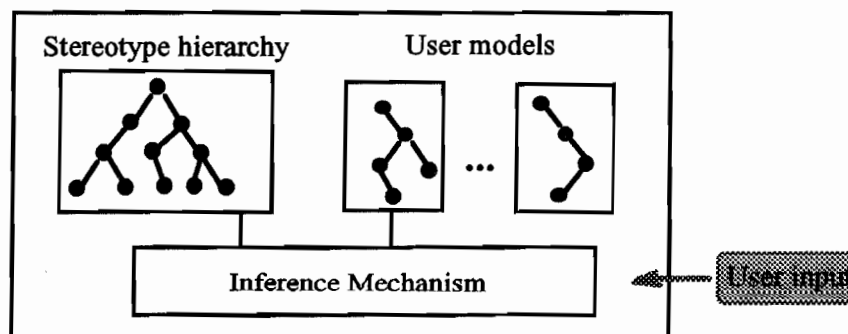


FIGURE 1 A schematic structure of stereotyping.

in the cor
the mode
such a si
might sti
proaches

Finally
systems t
difficult t
rule-base
is often in

This a
utilizes ne

2. PATT

We sugg
the user's
in any m
Norcio, 1
may be n
performa
In other
recogniti
In this ser
behavior
of the int

As a p
associati
enhanc
for imple
generaliz
the infer
seriously
there is ir

Associ
It utilizes
and infer
despite i
mented b
as the sy
universal
are consi
to positiv
extracts e
the hiera
associati

in the context of user modeling. Because a user may fail to fit any set of stereotypes, the modeling process may fail to associate any system decision with that user. In such a situation, however, some assumptions distributed among the stereotypes might still be useful for characterizing that user. In this sense, stereotyping approaches have a limited ability of personalization.

Finally, in addition, conventional user modeling systems tend to be rule-based systems that often encounter the problem of knowledge elicitation. It is often difficult to specify the relations among the assumptions and the stereotypes. Also, rule-based systems lack learning ability. The dynamic maintenance of the rule base is often inefficient and error prone (Hayes-Roth, 1991).

This article proposes an alternative approach, *associative user modeling*, that utilizes neural networks as the knowledge representation and inference mechanism.

2. PATTERN RECOGNITION IN USER MODELING

We suggest that the system's beliefs about users should be based on the context of the user's task performance. The fragmented pieces of observations may not result in any meaningful implications if they are not examined associatively (Chen & Norcio, 1994). In addition, the observed information about a user's characteristics may be mixed with noises or inconsistencies. Therefore, all aspects of the user's performance patterns must be examined before any system decision can be made. In other words, the information about a user should be processed by pattern recognition so that the system can establish complete and consistent user profiles. In this sense, user modeling is a process of recognizing a user's patterns (e.g., a user's behavior pattern, knowledge pattern, cognitive pattern, etc.) based on the context of the interaction.

As a pattern recognition process, user modeling requires the features of pattern association and classification, fault tolerance, graceful degradation, and signal enhancement. Neural networks have all these features and therefore can be used for implementing user models. In addition, neural networks can be trained to generalize inferences. In contrast, conventional stereotyping approaches, in which the inference proceeds a step at a time through sequential logic, may become seriously inadequate for processing pattern-formatted information, especially if there is incomplete, noisy, or inconsistent information involved (Pao, 1989).

Associative user modeling views the stereotypical knowledge as a set of patterns. It utilizes neural networks as associative memories for knowledge representation and inference. The networks can associate an input pattern with an output pattern despite incomplete or inconsistent inputs. The system's assumptions are implemented by the network nodes. Each node represents an assumption or an attribute as the system's stereotypical knowledge about users. All assumptions form a universal stereotype. In this approach, all assumptions in the universal stereotype are considered to be relevant to each other in a spectrum that is valued from negative to positive (i.e., from contrary, via irrelevant, to consistent). The modeling process extracts some assumptions to form a stereotype that fits a particular user. Unlike the hierarchical stereotyping approaches that model users at the stereotype level, associative user modeling proceeds at the assumption level. It overcomes the

limitation of the hierarchical stereotypes, which are unable to extract assumptions from different stereotype structures to form a new stereotype. Given the same number of assumptions, associative user modeling produces more user profiles than hierarchical stereotyping approaches. Therefore, it has the better ability to personalize users.

This study focuses on modeling users' domain knowledge. Several network paradigms are used to test the proposed approach. The test-bed application is the domain of programming and database concepts. Modeling such knowledge can facilitate system adaptation. It helps the system provide more effective feedback to users according to their skills and cognitive styles. This adaptation is particularly useful in various applications such as help systems, software engineering, information retrieval systems, and tutoring systems.

3. PATTERN ASSOCIATION AND DEFAULT REASONING

In order to provide the most appropriate response to users, an interface system should be able to infer what a user knows in a task domain. This usually is done by default reasoning that creates a larger number of assumptions as the output, based on a small number of assumptions in the input. Associative user modeling utilizes pattern association to simulate default reasoning used in rule-based systems where sequential logic is applied. Two networks, the bidirection linear associator (BLA; Simpson, 1990) and a back-propagation network (Carpenter, 1989), are implemented and tested for pattern association.

3.1. Bidirection Linear Associator

A BLA is used to capture the causal relations between an arbitrary number of assumptions. Figure 2 shows a structure of the BLA paradigm. Each node represents an assumption. The relations among the assumptions are weighted under certain conditions. The nodes in the network function as both input nodes and output nodes. Once a user's input from the dialogue channel is observed, it forms an input pattern to the BLA. The modeling process is conducted by activating a few nodes as input and propagating the activations throughout the network. Once all nodes reach stable activation levels, the assumptions activated in the output pattern are considered to be the current system beliefs about the user's domain knowledge, which is referred to as the user's knowledge pattern.

A weight matrix is used for representing the causal relations. This matrix explores how the knowledge providers conceptualize the domain and the relevance among the concepts in the domain. This study focuses on reasoning about a user's domain knowledge of database application programming. Twenty concepts are chosen to characterize the general knowledge of this domain. A user model is a collection of assumptions on whether or not a user understands these concepts. Table 1 shows the concepts applied in this study.

There are 20 positive nodes and 20 negative nodes in the BLA. A positive node represents the assumption that a concept is known to users. A negative node represents the assumption that a concept is unknown to users.



No.

1
2
3
4
5
6
7
8
9
10

Forty-n systems o participan between c concept, p also unde correspon may also Participan For exam must unde concepts v integrate t

The kn many var min-max a

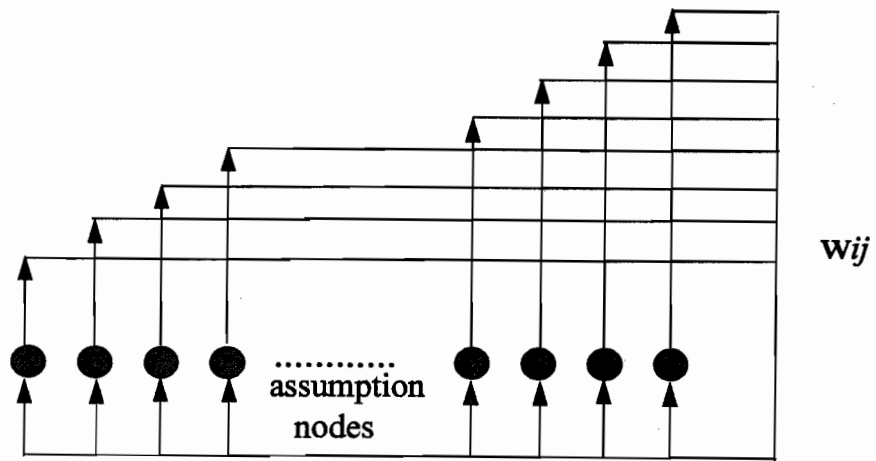


FIGURE 2 The structure of the BLA.

Table 1: Concept Index

No.	Concept	No.	Concept
1	Integer	11	Recursive
2	Real	12	Record
3	Loop	13	Index
4	Array	14	Weak entity
5	Subroutine	15	Data integrity
6	Local variable	16	FD
7	Tree	17	3NF
8	Stack	18	Concurrency
9	Inheritance	19	Locking
10	Interrupt	20	NP-complete

Forty-nine undergraduate students who are majoring in either information systems or computer sciences participated in the data collection procedure. Each participant was asked to create a weight matrix expressing the causal relations between concepts. Assuming that a user understands (or does not understand) a concept, participants were asked to choose other possible concepts the user might also understand (or might not understand), and assign the belief values to the corresponding cells. For example, if it is believed that a user who knows concept x may also know concept y , then fill 1 into the corresponding cell in the matrix. Participants could use any number between -1 and 1 to characterize such beliefs. For example, if the value of w_{ij} is 1, it means that a user who understands concept i must understand concept j . The participants were also instructed not to work on the concepts with which they were not familiar. A simple average function was used to integrate the matrices from the participants.

The knowledge pattern is produced through a propagation algorithm that has many variances. The algorithms can be linear, nonlinear, or fuzzy logic (e.g., min-max approach; Simpson, 1990). This study uses a linear propagation activation

algorithm to associate the input pattern with an output pattern. Two output status can be reached using this algorithm: a deterministic output, or a mode in which outputs cycle among several patterns.

For the second situation, a union operation is applied to the cyclic output patterns to generate the final output (Simpson, 1990). The algorithm can be illustrated as follows:

1. Initialize weight matrix M

$M = (w_{ij})_{n \times n}$, n is the total number of nodes.

2. Apply input vector $V (v_1, v_2, \dots, v_n)$ on M to produce a stimuli vector V' :

$$V' = V \otimes M = f_h \left(\sum_{i=1}^n v_i \times w_{ij} \right),$$

$$\text{where } f_h(v \times w_{ij}) = v_i' = \begin{cases} -1 & \text{if } v_i' \leq -\frac{1}{2} \\ 1 & \text{if } v_i' \geq \frac{1}{2} \\ 0 & \text{others} \end{cases}$$

- a) If the network converges (i.e., $V' = V$), stop.
- b) If the V' is previously created, set $V' = V' \cup V$ (logic union), stop
- c) otherwise set $V' = V$, go to 2.

The output pattern can yield concepts that are not activated in the input. This implies that given a small number of concepts, a larger number of correlated concepts are activated. This simulates the behavior of default reasoning. For example, when input pattern is $(0,0,0,0,1,0,0,0,\dots,0)$, which indicates that a user understands the concept *array*, the associated output pattern is $(1,1,1,1,0,0,0,\dots,0)$, which indicates that the user also understands the concepts *integer*, *real*, and *loop*.

This study designed 110 different input patterns for the test, which sufficiently cover representative input patterns. These input patterns can be classified into three categories: the patterns in which only positive nodes are activated, the patterns in which only negative nodes are activated, and the patterns in which both positive nodes and negative nodes are activated, including the patterns that have inconsistent assumptions (i.e., for the same concept, both positive node and negative node are fired at a time). All of the output patterns satisfied the following conditions:

1. The advanced concepts in the input yield less advanced concepts. The advanced concepts are those that are more difficult to understand than others. The criterion is based on general curriculum design for programming and database courses. For example, the concept of *tree* is more advanced than concept of *array*. The criterion is based on general curriculum design for computer programming and database courses.

2. The inconsistent input does not yield inconsistencies in output pattern. For example, if the input contains *locking*, \neg *locking*, and *index*, the output pattern contains the concepts (i.e., assumptions) only implied by the concept *index*.

3.2. Pat

A feed-forward associative network that enforces input patterns to be associative is shown in Figure 1. The network is negative.

The training process such as the training of the network that are mentioned in the literature.

The network is tested. In the assumption network, the training of the assumption network concepts is for the network functional added, it

3.2. Pattern Association Through the Feed-Forward Network

A feed-forward network trained by back-propagation (BP) is also tested for pattern association in user modeling. The BP network uses a nonlinear activation algorithm that enforces the ability of generalization (Carpenter, 1989). The pairs of input-output patterns from the BLA model are used to train the BP network to generalize the associative information stored in the BLA. A three-layer network as implemented is shown in Figure 3. The positive nodes represent the concepts known to a user; negative nodes represent concepts unknown to a user.

The training data allow the conflicting assumptions to offset each other's influence such that inconsistencies do not appear in the output pattern. Table 2 shows the training and testing results. The recall accuracy is 100%. For the input patterns that are not in the training set, 100% of the testing results satisfy the two conditions mentioned earlier.

The network's ability to generalize inferences on new concepts has also been tested. Introducing a new concept into the network (i.e., this implies a new system assumption is added into the universal stereotype) causes structural change to the network. Thus, the network needs to be retrained. The representation patterns of the training data for the new concepts should reflect the closeness between the new assumptions and the existing assumptions. In other words, functionally closed concepts in input may yield similar concepts in output. The representation patterns for the new concept are organized to teach the network to turn on the nodes that functionally closed concepts might turn on. For example, once a concept *queue* is added, it should yield similar assumptions that might be triggered by the concept

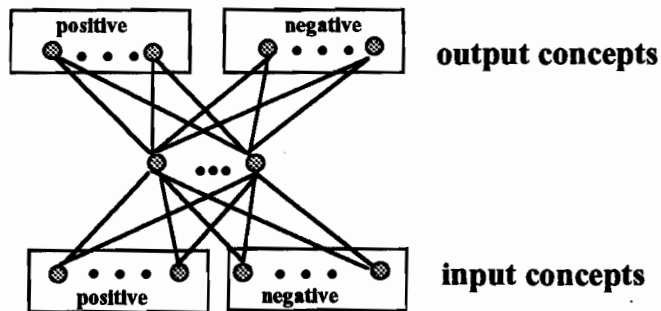


FIGURE 3 The feed-forward network.

Table 2: Training and Testing Results

Learning Rate	0.50
Momentum	0.50
Output Threshold	0.85
Number of Training Cycles	25,000
Number of Training Data	110
Testing/Recall Data	70
Recall Accuracy	100%

stack. In addition, retraining the network should retain the effects of previous training. A partial training is conducted as follows:

- Fix all weights except those that are on the newly added connections (i.e., the fixed connections do not participate in the training process).
- During the training period, present the new training data on both input layer and output layer. The input vector contains the stimulus on the new concept and the output vector contains all concepts implied by the new concept. Table 3 shows an example of a pair of training data.

After partial training, a part of the weight pattern (i.e., weights on the newly added connections) is established. This weight pattern can provide a similar activation in the output for the similar concepts in the input. The test result shows that functionally correlated concepts yield the same concepts in output. This result implies that the network can generalize its reasoning ability to adapt to new system assumptions without being totally retrained. This feature is particularly important for the dynamic modeling process, which often requires updating the structure of system belief space.

The retrained network is also tested by the original testing data set. It yields the same results as before partial training (i.e., conditions 1 and 2 are satisfied).

Compared to the hierarchical stereotyping approach, which uses inheritance as the basic form of generalization, associative user modeling implements the generalization by deriving the similarity of representation. It demonstrates strong ability in both default reasoning and uncertainty management. It also possesses the semantic inheritance and generalization features of the hierarchical stereotyping approach. In addition, partial training retains previous training effects, which facilitates the dynamic adaptation in the modeling process while the new concepts are taken into account.

4. COMPLETENESS OF PATTERN ASSOCIATION

This study suggests that, because the information about users is often correlated, it should be examined by pattern recognition and classification in order to generate relatively complete and consistent system beliefs about users. There are three basic aspects that affect the performance of pattern-recognition-based user modeling:

Table 3: An Example of Training Data

! input/output data for partial training												
	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
&	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
&	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	1.0	1.0	1.0	1.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
&	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
&	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
												! input vector
												! output vector
! queue (new concept) → stack, tree, local var., subrtm, array, loop, real, integer												

- C
- in
- yi
- G
- tra
- ne
- C
- ou
- it
- as

The fir
presents
chical clu
complete
networks

The ma
BP modul
BLA mod
card-sorti
were aske
relatednes
could app
(i.e., the fi
most diffic
was assign

Note th
BLA matr
matrix ref
user who l
yields a pc
related, w

The sin
concepts
patterns fr
categorical
between I
example,
that if a us
and loop a
that inclu
functional
the concep
loop and ar
within the
noticed th

- *Consistency of pattern association*: In a user modeling context, this means that inconsistent input (i.e., the concepts are in conflict with each other) does not yield an inconsistent output pattern.
- *Generalization ability*: Given a new input pattern that is not included in the training set, or changing the network structure to adapt to a new concept, the network can still generate the correct output without being totally retrained.
- *Completeness of pattern association*: Given an input pattern, the associated output pattern reveals correlated information. In the user modeling context, it means that given a few observations, a system can produce a number of assumptions.

The first two aspects have been discussed in the previous section. This section presents a comparison study for the completeness of pattern association. A hierarchical cluster approach is used to test whether the pattern association is relatively complete. A comparison is made between the associated patterns generated by networks and the clusters derived from a distance matrix.

The matrix is created from the same concept set used in developing the BLA and BP modules. The same group of participants who provided data for constructing the BLA model participated in data collection for constructing the distance matrix. The card-sorting approach was used to elicit the data (Wilson, 1989). The participants were asked to sort the concepts into several groups according to their functional relatedness. The participants could create as many groups as they wanted. Concepts could appear in more than one group, and were listed according to level of difficulty (i.e., the first concept in a group is the easiest to understand, and the last one is the most difficult). To convert all sorted groups into a single matrix, each cell of the matrix was assigned the value of frequency that a pair of concepts are grouped.

Note that this distance matrix is different from the BLA matrix in semantics. The BLA matrix only captures the causal relations among concepts, whereas the distance matrix reflects the strength of functional relations among concepts. For example, a user who knows the concept of *NP-complete* must know the concept of *integer*, which yields a positive link in the BLA matrix. However, these two concepts are not closely related, which is represented by a smaller value in the distance matrix.

The single-link, hierarchical clustering approach is used to identify groups of concepts on the basis of proximity data. Figure 4 shows the clusters. The output patterns from the networks (i.e., the BLA and BP modules) reflect the causal and categorical information carried by the clusters very well. A comparison can be seen between Figure 4 and Table 4 that shows a subset of the network output. For example, *real* and *integer* are in the same cluster. The network output also shows that if a user knows *real* then he or she also knows the concept of *integer*. The *array* and *loop* are in the same cluster, and they are also closely linked to another cluster that includes *integer* and *real*. It shows that the network output exhibits both functional relatedness and also causal relations among the concepts. For example, the concept *stack* triggers the concept *tree* and two other clusters (i.e., *integer* and *real*, *loop* and *array*). Furthermore, the concept *NP-complete* not only triggers the concepts within the cluster (e.g., *index*), but also the concepts in the other clusters. It is also noticed that the concepts that are functionally related and classified in the same

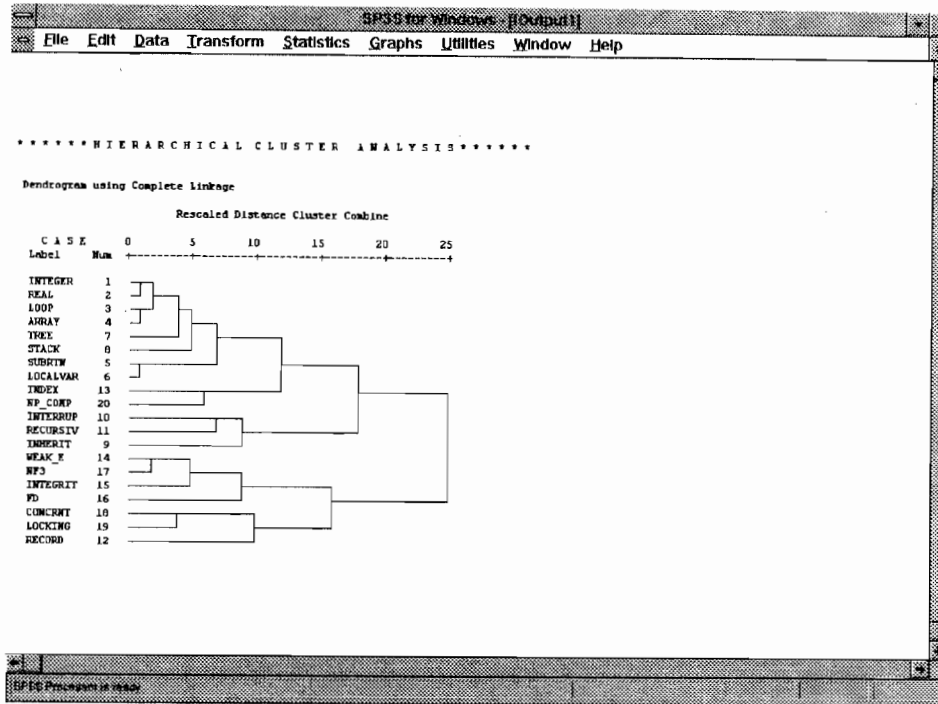


FIGURE 4 Hierarchical clustering for the database programming concepts.

Table 4: A Subset of Network Output and Corresponding Input

Input Concept	Output Concepts
Real	Integer
Array	Loop, real, integer
Tree	Array, loop, real, integer
Stack	Tree, array, loop, real, array
Local_variable	Subroutine, real, integer, loop, array
Subroutine	Real, integer, loop, array
NP-complete	Index, subroutine, stack, tree, integer, real, loop, array, local_variable
Locking	Concurrency, record

categories may not have the same possibilities to trigger each other. For example, the concepts *subroutine* and *local-variable* are in the same cluster. *Local-variable* can trigger *subroutine* as well as other concepts (e.g., *integer*, *real*, *loop*, and *array*), but *subroutine* cannot trigger *local-variable*. This implies that *local-variable* is a more advanced concept than *subroutine*, even though they are functionally related. This implies that network output reveals more information than cluster analysis, because it reveals not only categorical information, but also the causal relations among the concepts (i.e., assumptions).

Generally, the concepts in an input pattern can yield less advanced concepts in the same cluster or category. Certain concepts can also trigger the less advanced

concept
NP-com
ics of th
informa

5. PAT

In the us
user's cu
into a ce
adaptive
users' k
network
closenes
it is part
shows th
the patte
input pa
recalls a
layer res
categori

Twent
node rep
categori
category
knowled
novice, a

The te
to the AI
stored pa
in the ou
pixels rep
rate at 0

concepts in different clusters, as long as they are in the same category. For example, *NP-complete* triggers all concepts in the category of programming. Thus the dynamics of the network simulates the process of default reasoning, which reveals more information than cluster analysis.

5. PATTERN CLASSIFICATION IN USER MODELING

In the user modeling process, an output pattern from a network is considered as a user's current knowledge pattern. It is often necessary to further classify this pattern into a certain category based on which the system can generate its response. An adaptive resonance theory (ART) model (Simpson, 1990) can be used to classify users' knowledge patterns dynamically. It receives the outputs from the other network models and classifies the users' knowledge patterns according to their closeness dynamically. Because the ART model does not need supervised training, it is particularly useful when it lacks expertise about user classification. Figure 5 shows the structure of the ART model. Conceptually, the comparison layer stores the patterns learned from the unsupervised training process. The closeness of the input patterns is controlled by a single scalar called *vigilance*. The comparison layer recalls a pattern that is closest to the input. The categorical nodes in the recognition layer respond to the recalled pattern in a competitive style. Eventually, only one categorical node is activated for the given input.

Twenty input nodes are used to represent the user's knowledge pattern. Each node represents a concept in Table 1. Five output nodes are used to indicate user categories. The unsupervised training process stores the typical patterns for each category. Figure 6 shows an example of the training data that represent the typical knowledge patterns as expert, expert-intermediate, intermediate, intermediate-novice, and novice.

The test results (i.e., the outputs) from the BLA and the BP are directly presented to the ART model. The test result shows that the network successfully matches the stored patterns to input patterns and activates the corresponding categorical node in the output layer. Figure 7 shows examples of such network behavior, where black pixels represent the concepts known to a user. The vigilance is set to 0.8 with learning rate at 0.9. These examples show that even though the test patterns are different

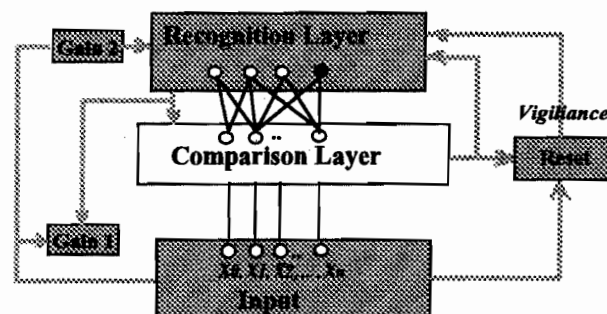


FIGURE 5 The ART structure.

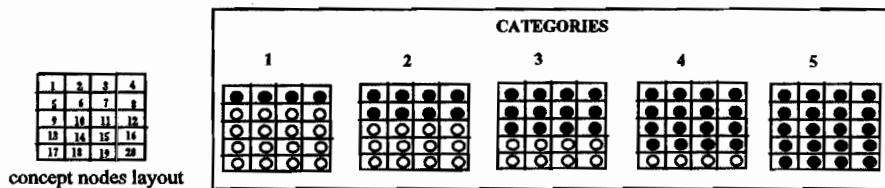


FIGURE 6 An example of the ART training data.

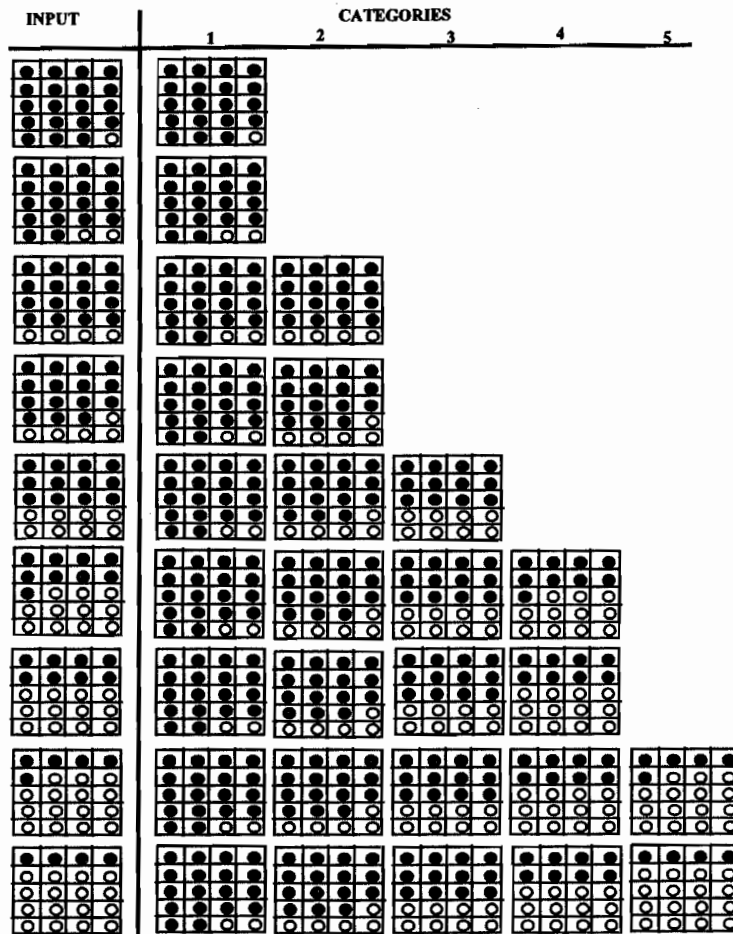


FIGURE 7 An example of the behavior of the ART for user classification.

from the training patterns, the comparison layer (F1) can still invoke a pattern that matches the input with a slight variance (refer to rows 1 and 2, rows 3 and 4, rows 6 and 7, and rows 8 and 9, respectively). This means that the ART model can classify the input patterns into the appropriate category.

User Mo

6. INT. FRA

The net blackbo work ei by exch simulat elicitation the BP n for class profile i user mo

In ad multiple or the v outputs sentatio diate as

7. ADV

The pro neural n

6. INTEGRATION OF THE NETWORK MODULES—A BLACKBOARD FRAMEWORK

The network modules used in this study can be integrated into a framework of a blackboard system in which each model functions as a knowledge source that can work either independently or cooperatively. This framework has been simulated by exchanging the input or output from one network to another. Figure 8 shows the simulation process. The construction of the BLA matrix is the process of knowledge elicitation. The BLA's input and output can be used as input (or training data) to the BP model; the output from BP or BLA can be represented to the ART module for classification. The output from any network is viewed as part of the current user profile in the context of interaction. Thus, this framework is effective for dynamic user modeling.

In addition, this framework allows each module to be utilized separately for multiple purposes. The training process can be conducted for either a single module or the whole system, which makes system adaptation simple and flexible. The outputs from each module provide a clear insight into system knowledge representation and the modeling process, which facilitates the explanation by intermediate assertions in the inference (Chen & Norcio, 1994).

7. ADVANTAGES OF ASSOCIATIVE USER MODELING

The proposed approach, referred as associative user modeling, utilizes a set of neural networks as the knowledge base and reasoning mechanism. It organizes all

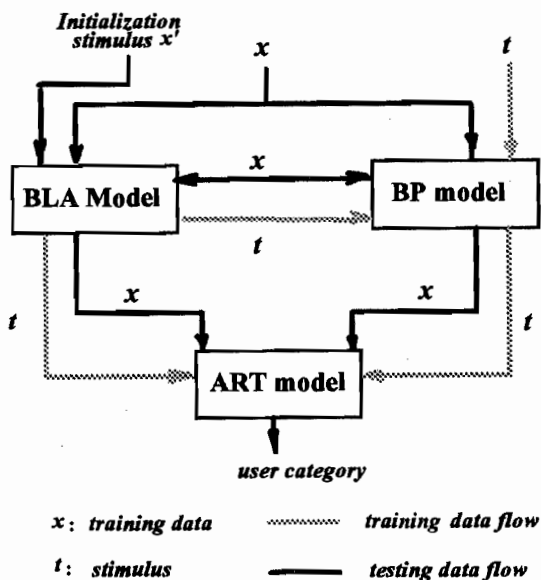


FIGURE 8 The blackboard simulation process.

assumptions about the characteristics of possible users into associative memories in which the relations among the assumptions are weighted under certain conditions. It associates a user's domain knowledge pattern with a completely hypothetical pattern that characterizes that user. Also, such patterns can be further classified into different categories in terms of similarities. Several attractive features of the neural networks can facilitate the performance of user modeling systems: The networks can produce a large body of relevant information based on a few stimuli. They can generalize the response by seeking similarity between new inputs and previous ones. They can learn to extract the prototype of a set of repeated experiences in ways that are similar to the concept-learning characteristics seen in the human cognitive process (McClelland & Rumelhart, 1986). In addition, the networks are fault tolerant. They degrade gracefully with inconsistent or erroneous input. The associative user modeling approach is inspired by these features. It particularly facilitates the user modeling systems in the following aspects:

- *Default reasoning and generalization:* Associative user modeling tends to retrieve what is common to the user's domain knowledge. Given an incomplete input about a user's domain knowledge, the system is able to associate other related information to stereotype the user. Because one piece of domain knowledge is usually based on another one, modeling such knowledge is an appropriate application of default reasoning. The ability to respond to novel inputs that bear some similarity to previously learned examples is one of the major strengths of the neural networks.

- *Insensitivity to inconsistent input (fault tolerance):* The neural network models can handle partial or erroneous cues without ill effects. As previous system beliefs may conflict with current beliefs, the information about a user's domain knowledge should be examined in terms of pattern recognition, so that the complete and consistent system beliefs can be obtained. Neural networks provide an effective means for pattern recognition.

- *Personalization:* In associative user modeling, all system beliefs are fully connected under certain conditions (i.e., weighed connections). The modeling process is not confined by predefined stereotypes (i.e., the sets of system beliefs). This approach allows the system to generate more individual user profiles because the number of different profiles is determined by the number of combinations of system beliefs rather than the number of belief sets. Predefined stereotyping may fail to associate any system belief to a user even though there are some assumptions distributed among the stereotypes that may be applicable for characterizing that user. Unlike the stereotype approach, all underlying concepts about users are associated with each other in a spectrum that is valued from negative to positive (i.e., contrary to consistent). The pattern of fired nodes represents system beliefs that underlie a user's profile. Given the same number of initial system beliefs, this approach can personalize users' characteristics more specifically than hierarchical stereotype approaches.

- *Dynamic modeling process:* The proposed approach eases the system adaptation. During each interaction, a user's responses fire several assumptions, and a stimuli input is formed to activate the modeling process. The output patterns reflect the current system beliefs and can also be dynamically classified into different catego-

ries the
short-t

- K
on kno
time-co
on the
establis
it is eas
formul

- R
mainten
system
cated c
evident
networ
tion. Th
comple
mainten

In ac
convent
tances i

8. CON

This art
associat
views u
tages su
tent inp
to the r
implem
the rule
initiate
informa
for syste

REFER

Carpente
memc
Chen, Q.
Procee
de Kleer,

ries that do not need to be predefined. This feature allows the system to model both short-term and long-term characteristics of users.

- *Knowledge elicitation:* Conventional rule-based systems require extensive work on knowledge elicitation to set up an appropriate rule base, which tends to be a time-consuming and error-prone process. Associative modeling systems only focus on the extraction of causal relations between each pair of concepts rather than establishing an entire rule base. This eases the knowledge elicitation process because it is easier for a human to identify a causal relation between two concepts than to formulate a rule that might involve more concepts.

- *Reducing system overhead:* It is widely recognized that the implementation and maintenance of neural network models are much simpler than that of rule-based systems (Pao, 1989). Furthermore, rule-based systems often involve either complicated conflict resolutions in default reasoning, or the belief value revision in evidential reasoning. By contrast, the consistency can be easily maintained in neural network systems due to their ability to handle inconsistent or incomplete information. The networks can consistently associate an incomplete pattern with a relatively complete pattern. Modeling is more efficient because there is no need for truth maintenance, which is required in rule-based systems.

In addition, the neural network approach exhibits the same advantages of conventional modeling approaches, such as fast stereotyping and property inheritances in terms of pattern association.

8. CONCLUSION

This article presents a study that tested and integrated several neural networks as associative memories in the user modeling process. The neural network approach views user modeling in terms of pattern recognition. It has shown several advantages such as rapid default reasoning and generalization, insensitivity to inconsistent input, personalization, pattern classification, and learning ability. Compared to the rule-based user modeling systems, associative user modeling is easier to implement and maintain. Also, the knowledge elicitation process is simpler than the rule-based construction, because only the causal relations are considered to initiate the modeling process. Further research is aimed at incorporating the task information into associative user models to provide a more comprehensive basis for system adaptation.

REFERENCES

- Carpenter, G. A. (1989). Neural network models for pattern recognition and associative memory. *Neural Networks*, 2, 243-257.
- Chen, Q., & Norcio, A. F. (1994). Stereotyping users and tasks with associative networks. In *Proceedings of IEEE International Conference on Neural Networks* (pp. 1169-1173).
- de Kleer, J. (1986). An assumption-based TMS. *Artificial Intelligence*, 28(2), 127-162.

- Hayes-Roth, F. (1991). Making intelligent systems adaptive. In K. VanLehn (Ed.), *Architecture for intelligence* (pp. 301-321). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.
- Huang, X., Mccalla, G. I., Greer, J. E., & Neufeld, E. (1991). Revising deductive knowledge and stereotype knowledge in a student model. *User Modeling and User-Adapted Interaction*, 1(1), 87-115.
- McClelland, J. L., & Rumelhart, D. E. (1986). *Parallel distributed processing* (Vol. 2). Cambridge, MA: MIT Press.
- Norcio, A. F., & Stanley, J. (1989). Adaptive human-computer interfaces: A literature survey and perspective. *IEEE Transactions on System, Man and Cybernetics*, 19(2), 399-408.
- Pao, Y. (1989). *Adaptive pattern recognition and neural networks*. Reading, MA: Addison-Wesley.
- Rich, E. (1979). User modeling via stereotypes. *Cognitive Science*, 3, 329-354.
- Simpson, P. K. (1990). *Artificial neural systems: Foundation, paradigms, applications and implementations*. New York: Pergamon.
- Wilson, M. (1989). Task models for knowledge elicitation. In D. Diaper (Ed.), *Knowledge elicitation: Principles, techniques and applications* (pp. 101-121). New York: Ellis Horwood.

The
ma
of a
val
gen
mis
cog
dire
and
an
sho
was

1. IN

Task a
Its pu
used f
tion.
comp
Analy
to est
navig
predic
interf
and p

Req
Pleasant