Pattern Recognition of User's Domain Knowledge

Qiyang Chen
Department of Computer and Information Sciences
Towson State University, U. S. A.

A. F. Norcio
Department of Information Systems
University of Maryland, Baltimore County, U. S. A.

ABSTRACT

This paper presents a neural network approach for user modeling. A set of neural networks is utilized to represent and infer users' task-related characteristics. 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 about users' task-related characteristics. The construction of this set of assumptions is often referred to as user modeling. Most user modeling techniques use the stereotype approach to initialize the system's assumptions. This approach pre-defines the assumptions into groups and organizes them into a generalization hierarchy [1]. In a generalization hierarchy, a stereotype is defined by a certain number of assumptions. A user model is actually a substructure of such hierarchy. The function of the inference mechanism is to extract the substructures ascribed to users without causing any inconsistency. 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:

(1) Since 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 dependency-directed backtracking process, referred to as truth maintenance, to eliminate the inconsistencies [2]. This process utilizes the sequential logic to examine one piece of information at a time. Therefore, it is possible that the effort of maintaining consistency may bring in further conflicts in subsequent interaction [3].

(2) The pre-defined generalization hierarchy limits the system assumptions within each stereotype that can be only inherited by the descendant stereotypes. Since a user may fail to fit any set of stereotypes, modeling process may fail to associate any system decision to that user. In such situation, however, some assumptions distributed among the stereotypes might still be useful for characterizing that user [4].

(3) In addition, conventional user modeling systems tend to be the rule-based systems that often encounter the problem of knowledge elicitation. It is often difficult to specify the relationships among the assumptions and the stereotypes. Also, rule-based systems lack the learning ability. The dynamic maintenance of the rule base is often inefficient and error-prone [5].

This paper presents an alternative approach, associative user modeling, which utilizes the neural networks as the knowledge representation and reasoning mechanism.

2. PATTERN RECOGNITION IN USER MODELING

We suggest that the information about a user should be processed through the way of pattern recognition so that the system can establish the 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 [4]. Neural networks have all these features and therefore can be used for implementing user models.

Associative user modeling views the stereotypical knowledge as a set of patterns. It associates an input pattern with an output patterns despite incomplete or inconsistent inputs. The system's assumptions are
implemented by the network units. Each unit represents an assumption as the system’s stereotypical knowledge about users. In this approach, all assumptions 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 is unable to extract assumptions from different stereotype structures to form a new stereotype.

Several network paradigms are used to test the proposed approach by modeling users’ generic knowledge of programming and database concepts.

3. PATTERN ASSOCIATION

Associative user modeling utilizes the pattern association to simulate default reasoning in rule-based systems where the sequential logic is applied. Two networks, Bidirection Linear Associator and Back-Propagation network, are tested.

A Bidirection linear associator (BLA) is used to capture the causal relationships between an arbitrary number of assumptions. Figure 1 shows a structure the BLA paradigm. The relationships among the assumptions are weighted under certain conditions. Once a user’s input from the dialogue channel is observed, it forms an input pattern to BLA. The modeling process is conducted by propagating the activations throughout the network, to associate the input pattern with an output pattern. The assumptions represented by 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.

![Figure 1. The structure of BLA](image)

A weight matrix is necessary for representing the causal relationships. It is constructed by the card sorting method [6]. This method utilizes concepts of a domain to explore how the knowledge providers conceptualize the domain and the relevance among these concepts. This study focuses on reasoning about a user’s generic domain knowledge of database application programming. Twenty concepts are tested in this study. The related system’s beliefs are assumptions on whether or not a user understand these concepts. Forty-nine undergraduate students who are majoring either information systems or computer sciences participated in the data collection procedure. Each subject is asked to create a weight matrix expressing the causal relationships between concepts. Assuming that a user understands (or does not understand) a concept, subjects are asked to choose other possible concepts the user might also understand (or does 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. Subjects may use any number between -1 and 1 to characterize such beliefs. A simple average function is used to integrate the matrices from the subjects.

The knowledge pattern is produced through a propagation algorithm which has many variances[7]. This study uses a linear propagation 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.

The output pattern includes a certain number of concepts not appeared in the input. This implies that given a small number of concept, a larger number of correlated concepts are activated. This simulates the process of default reasoning.

The pairs of input pattern and output pattern can also be used to train a feed-forward network to generalize. This study tested 110 different input patterns, including the input patterns that has inconsistent concepts (i.e., for same concept, both known and unknown units are fired at a time). 100% of the output patterns satisfied the following conditions:

(a) the advanced concepts in the input yield less advanced concepts, and
(b) the inconsistent input does not produce inconsistencies in output pattern.

A feedforward network trained by Back-Propagation (BP) is also tested for pattern association. The BP network uses nonlinear activation algorithm that enforces the ability of generalization[7]. A three-layer
network is implemented shown in Figure 2. The positive nodes represent the concepts known to a user; negative nodes represent concepts unknown to a user.

There are two types of data in the training set: The exemplars from the testing results of the BLA network, and the exemplars for conflict reconciliation. The second type of data allows the conflicting concepts to offset each other’s influence such that inconsistencies do not appear in the output pattern. For example, if the input contains locking, → locking and index, the output pattern contains the concepts only implied by concept index.

The training and testing results show 100% of recall accuracy. For the input patterns that are not in training set, 100% of the testing results satisfy conditions (a) and (b) mentioned above.

![Diagram](image)

**Figure 2. The feedforward network**

The network’s ability to generalize inferences on new concepts has also been tested. Introducing a new concept into the network causes the structural change to the network. Thus, the network need to be retained. 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, the closed concepts in input should yield similar concepts in output. The representation patterns for the new concept are organized to teach the network to turn on the units that 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 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, the output vector contains all concepts implied by the new concept.

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 the functional correlated concepts yield similar concepts in output. This result implies that the network can generalize its reasoning ability to adapt new system assumptions without being totally retrained. This feature is particularly important for the dynamic modeling process, which often requires to update the structure of system belief space.

Compared to the hierarchical stereotyping approach, which uses inheritance as the basic form of generalization, associative user modeling implements the inheritance feature by the similarity of derived representation. It demonstrates strong ability in both default reasoning and uncertainty management. It also possesses the semantic inheritance and generalization features of 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 user modeling should be a process of pattern recognition and classification. Since the information about users is often correlated, it should be examined in patterns 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:

- **Consistency of pattern association**: In a user modeling context, this means that inconsistent concepts in an input does not yield an inconsistent concepts in 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 enough correlated information. In the user modeling context, it means that given a small
number of observations, a system can produce a large number of assumptions.

The first two aspects have been illustrated above. 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.

A distance matrix is created from the same concept set used in developing the BLA and BP modules. The same group of subjects participated in data collection for constructing the matrix. The subjects are asked to group the concepts according to their functional relatedness. The subjects can create as many groups as they want. As the result of combining the data groups from subjects, each cell of the distance matrix is assigned a value that represents the frequency of co-occurrence of a pair of concepts.

The hierarchical clustering approach is used to identify groups of concepts. Figure 3 shows the clusters. The output patterns from the networks reflect the causal and categorical information carried by the clusters very well. A comparison can be seen between Figure 3 and Table 1 that shows a subset of the network output. For example, 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 not only exhibits the functional relatedness but also the causal relationship among the concepts. For example, the network allows concept `stack` to trigger 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 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 also implies that network output reveals more information than cluster analysis where only the categorical information is shown. The concepts in an input pattern yield less advanced concepts in the same cluster. Some concepts can also trigger the less advanced concepts in different clusters, as long as they are in the same category.

![Hierarchical Cluster Analysis](image)

**Figure 3.** A cluster analysis

<table>
<thead>
<tr>
<th>Input concept</th>
<th>output concepts</th>
</tr>
</thead>
<tbody>
<tr>
<td>real</td>
<td>integer</td>
</tr>
<tr>
<td>array</td>
<td>loop, real, integer</td>
</tr>
<tr>
<td>tree</td>
<td>array, loop, real, integer</td>
</tr>
<tr>
<td>stack</td>
<td>tree, array, loop, real, array</td>
</tr>
<tr>
<td>local_variable</td>
<td>subroutine, real, integer, loop, array</td>
</tr>
<tr>
<td>subroutine</td>
<td>real, integer, loop, array</td>
</tr>
<tr>
<td>NP-complete</td>
<td>index, subroutine, stack, tree, integer, real, loop, array, local_variable</td>
</tr>
<tr>
<td>locking</td>
<td>concurrency, record</td>
</tr>
</tbody>
</table>

Table 1. A subset of network output

5. PATTERN CLASSIFICATION

Given a user’s knowledge pattern, it is often necessary to classify it into certain category based on which the system can generate its response. An Adaptive Resonance Theory (ART) model is tested for classifying outputs to obtain the categorical information about users. It receives the output from the other network models and classifies the users’ knowledge patterns according to their closeness. Figure 4 shows the structure of ART model. The comparison layer stores
Twenty input units represent the user's knowledge pattern. Five output units are used to indicate user categories as expert, expert-intermediate, intermediate, intermediate-novice, and novice respectively. The unsupervised training process stores the typical patterns for each category. The vigilance is set to 0.8 with learning rate 0.9. The test result shows that the network successfully associates the input patterns to the closest stored-patterns and activates the corresponding categorical unit in output layer. The test results show that even though the test patterns are different from the training patterns, the comparison layer (F1) can still invoke the closest pattern to match the input with a slight variance.

6. INTEGRATION OF THE NETWORK MODELS - 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. For example, the BLA's input and output can be used as input (or training data) to BP model; the output from BP or BLA can be represented to ART module for classification. The output from any network is viewed as the part of current user profile in the context of interaction. Thus, this framework provides an effective way for dynamic user modeling.

7. ADVANTAGES OF ASSOCIATIVE USER MODELING

The proposed approach utilizes a set of neural networks as the knowledge base and reasoning mechanism. The networks can produce a large body of relevant information based on a few stimuli. They can learn to extract the prototype of a set of repeated experiences in a way that are similar to the concept-learning characteristics seen in human cognitive process [10]. 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**: Given an incomplete input about a user's domain knowledge, the system is able to associate other related information to stereotype the user. This simulates the default reasoning.

- **Insensitivity to inconsistent input (fault tolerance)**: The neural network models can handle partial or erroneous cues without ill effects. Since previous system beliefs may conflict with current beliefs, the information about a user's domain knowledge should be examined in terms of pattern recognition.

- **Personalization**: The modeling process is not confined by predefined stereotypes. 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 assumption rather than the number of assumption sets.

- **Dynamic modeling process**: The proposed approach eases the system adaptation. The output patterns reflect the current system beliefs and can also be dynamically classified into different categories that do not need to be predefined.

- **Knowledge elicitation**: Associative modeling systems only focus on the extraction of causal relationships between each pair of concepts. It is easier for human to identify a causal relationship 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[7]. Furthermore, rule-based systems often involve either complicated conflict resolutions in default reasoning, or the belief value revision in evidential reasoning. In contrast, the consistency can be easily maintained in neural network systems due to their ability to handle inconsistent or incomplete information. Modeling is more efficient because there is no need for truth maintenance, which is required in rule base systems.
8. CONCLUSION

This paper presents a study that tested and integrated several neural networks as associative memories in user modeling process. 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 be implemented and maintained. Also, the knowledge elicitation process is simpler than the rule-base construction, because only the causal relationships 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