IDEA GROUP PUBLISHING
701 E. Chocolate Avenue, Suite 200, Hershey PA 17033, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com
ITP4492

Connectionist Models for Tracing User-task Profiles

Qiyang Chen, John Wang and Qiang Tu
Dept of Info and Decision Sciences, Montclair State University, Upper Montclair, NJ07043, U. S. A., chenq@mail.montclair.edu

ABSTRACT

This paper discusses an approach of modeling users' task related characteristics in interactive systems. Connectionist models are used to trace the on-going interaction and user-task profiles. These models function as associative memories that can capture the causal relationships among users' characteristics for the system adaptation. This approach can be used for stereotyping users' characteristics.

INTRODUCTION

User modeling is very useful in intelligent agents that are growing in popularity as a way to help user accomplish various kinds of tasks. An intelligent agent is a software surrogate for an end user or a process that fulfills a stated need or activity. An intelligent agent uses its build-in and learned knowledge base about users or the tasks being performed by users. The objective of an intelligent agent is to fulfill the intentions of a user, to make the human computer interaction to be more cooperative and efficient.

One of user characteristics is the user's domain knowledge. Knowing users' domain knowledge level will allow a system to adapt users to fulfill their tasks. User knowledge assessment constitutes generic information that can be used for many purposes [1]. The system beliefs on user's expertise for a given subject have become important components of user models in adaptive human-computer interfaces. Such beliefs can be classified into several dimensions such as long-term vs. short term, static vs. dynamic, and given vs. inferred [2].

The approach of modeling a user's cognitive characteristics is to use a set of pre-defined assumptions to initialize the system's beliefs about te user's domain knowlge andthe tasks that the user is performing. The pre-defined assumptions are organized into a *generalization hierarchy*. During a human computer interaction, the system assigns assumptions to a user through default reasoning. This process creates a model of the user's cognitive characteics, which retains the stereotypical knowledge about a user's domain knowledge and cognitive preferences in the absence of evidence to the contrary. This approach provides a simple way to initialize the modeling process and was successful in some applications [1,2]. However, if this process is implemented by a rule based system, the following limitations become very obvious.

In the rule based modeling process, the inference is based on extensive default assumptions that may conflict with the new evidence obtained as the interaction progresses, the revision of the system knowledge base is necessary to handle the inconsistencies. A common suggestion is to use dependency-directed backtracking process to accomplish the truth maintenance, which examines one piece of evidence at a time in a non-monotonic way [3]. This approach is often inefficient and lacks the ability to detect noisy or inconsistent information that should be ignored [4]. Therefore, it is very possible that current effort of maintaining consistency may bring further conflicts in the subsequent interaction. Thus, model construction may fall into a dilemma where a non-monotonic process of reconciling conflicts is frequently involved and eventually no decision can be made after a period of interaction [5].

In addition, the pre-defined hierarchy limits the system beliefs within each assumption set that can only be inherited by the descendant assumption set. Therefore, it is hard to effectively update those system beliefs that are no longer significant in the context of task performance. Also, since a user may fail to fit any set of assumptions, a modeling process will then fails to associate any assumption to that user. In such situation, however, some of the assumptions distributed in the generalization hierarchy might be still useful for characterizing that user. In this sense, the pre-defined generalization hierarchy is limited in the degree of individualizing a user.

PATTERN RECOGNITION

All the aspects of the user's performance patterns have to be examined before any system decision can be made. In other words, user modeling should be a process of pattern recognition, which requires that the modeling system have the capability of fault tolerance, graceful degradation, and signal enhancement. Therefore, the neural network techniques become natural choice for the implementation. Whereas, the rule based approaches, in which the inference proceeds a step at a time through sequential logic, may become seriously inadequate for processing pattern-formatted knowledge especially when there are incomplete, noisy or inconsistent information involved [6].

In our approach, each network node represents an assumption or an attribute as the system's belief about user's domain knowledge. The modeling process activates some of assumptions to dynamically form a unique profile that fits a particular user. Unlike the rule-based approaches that only associate the user with a pre-defined set of assumptions, a neural network based user model includes various assumptions associated to a user dynamically. In other words, associative user modeling proceeds at the assumption level rather than the assumption set level.

In the tested application an assumption is about if a user understand or does not understand a concept used in file system. All assumptions are considered to be associated to each other in a spectrum that is valued from negative to positive (i. e., from contrary, via irrelevant, to consistent). There are no pre-defined assumption sets. Thus, it overcomes the limitation of the rule-based *generalization hierarchy*, which is unable to extract assumptions from different assumption sets to form a new profile about the current user and task. Therefore, this approach has a better ability to individualize a user.

FUZZY COGNITIVE MAP

A fuzzy cognitive map (FCM) [7] is used as a classification paradigm to capture the causal relationships between an arbitrary number of assumptions on the tasks. The associations among the

Figure 1 the structure of FCM model

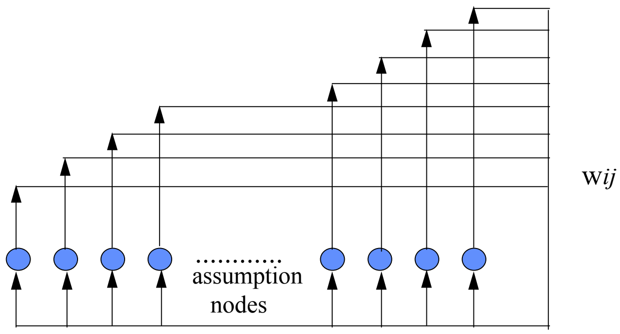
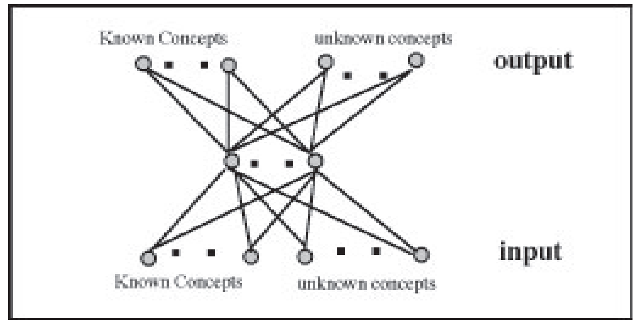


Figure 2. Feed forward network



assumptions are weighted under certain conditions. Figure 1 shows a structure of such paradigm. Once a user's input from the dialogue channel is observed, it forms an input to FCM model. The modeling process is conducted by propagating the activation level throughout the network, to associate this input with an output pattern. This output pattern is considered as the current system beliefs about that user. This process simulates the behavior of default reasoning.

A weight matrix used in FCM model for propagating activation is constructed by the card sorting method [8]. To demonstrate the idea, we use Unix environment as a design system interface. Eighteen Unix commends (refer to Table 1) are used in the test. These commands can be categorized into two groups: file operation and programming. Twenty Unix users participated in the data collection procedure. Each subject is asked to create a weight matrix expressing the relationships among the relevant concepts. Given an assumption that a user used a command in his/her task, subjects are asked to choose other possible commands the user might (or might not) use and assign the belief values to the corresponding cells. For example, if it is believed that a user who used command *x* may also use concept *y*, then fill "1" into the corresponding cell (*x,y*). Subjects may use any number between -1 and 1 to characterize such beliefs. An average function is used to consolidate the matrices from the subjects.

Table 1. The UNIX commands used in task modeling

File operations

cd	cb
mkdir	lint
rmdir	cc
ls	as
pwd	cpp
chmod	pc
cp	pi
mv	pix
rm	px

It is obvious that some of commands are more functionally correlated. Some commands are more advanced that others. For example people who usually learn and understand the concept of "rm " first before they know "px".

BACK-PROPAGATION MODEL

To test FCM's ability of pattern association, a linear propagation activation algorithm [7] is used. Two hundred different input patterns are tested, including the input patterns that include inconsistent concepts (e.g. for same concept, both known and unknown nodes are activated). 100% of the output patterns satisfied the conditions: (a) the *advanced concepts* in the input yield *less advanced concepts* and (b) the inconsistent input does not produce inconsistent output.

In order to generalize the results from FCM model, a feed forward network trained by back-propagation method is used in this study. This network is used to generalize the associative knowledge stored in FCM's matrix. The trainning data comes from FCM's testing results. A three-layer network is implemented shown in Figure 2. A *current command* node represents the assumption that a user has used this command in the current dialog. An *unknown concept* node represents the assumption that a user does not understand a concept in the field. Table 2 shows the training and testing results.

Table 2. Training and testing results

Learning rate	0.50
Number of training cycles	25,000
Number of training data	200
testing/recall data	70
recall accuracy	99.5%

Similar to the test results for FCM model, for the testing data that is not included in the training set, 100% of the testing results satisfy the conditions: (a) the *advanced concepts* in the input yield *less advanced concepts*, and (b) the inconsistent input does not produce inconsistent output. This means that the truth maintenance is simply enforced by the networks' ability of pattern recognition. This network has been also tested for pattern generalization. Introducing a new assumption node into the network (i.e. this implies a new system beliefs coming into the modeling process) causes the structural change to the network. In order to retain the effects of previous training, a partial training is conduced as follows:

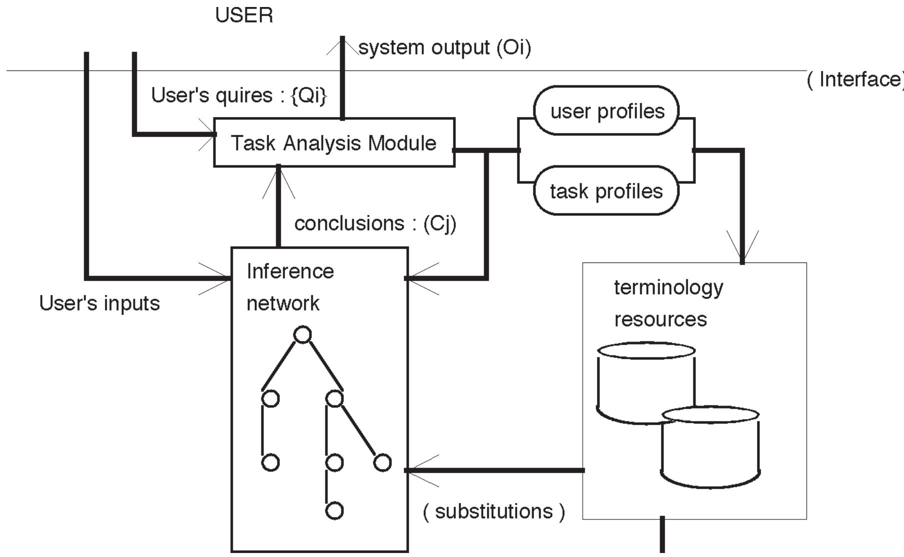
1. Freeze all weights except those that are on the newly added connections. (i.e., to ensure that the frozen connections cannot participate the training process).
2. During the training period, present the new training samples on both input layer and output layer. The input pattern contains the stimulus on the new concept, the output pattern contains all concepts implied by the new concept.

The representation pattern for a new concept (i.e., the training data) reflects the closeness between the new concept and the existing concepts. The test result shows that the functional correlated concepts yield similar concepts in output. In other words, a new concept makes the network to turn on the nodes that the correlated concepts might turn on. For example, once a concept "queue" is added, it yields the similar concepts in the output as the concept "stack" yields. This result implies that the network can generalize its reasoning ability to adapt new system assumptions without being totally retrained. This feature is particular important for the dynamic modeling process in which it is often necessary to update the structure of system belief space.

ADAPTIVE RESONANCE THEORY (ART) MODEL

In a user modeling system it is also often necessary to classify users' characteristics for system adaptation. An Adaptive Resonance Theory model [6] is used to further classify the outputs from above two network models. This network allows unsupervised training that classifies the input patterns based on the similarity. All positive assumptions (i.e., the assumption that a user understands a concept) are implemented by input nodes. Five output nodes are used to indicate user categories as expert, expert-intermediate, intermediate, intermediate-novice, and novice respectively. After unsupervised training process, the network stores the most typical pattern for each category. The test result shows that the network successfully associates the test patterns to the closest stored-patterns and activates the corresponding categorical node in output layer. This means that if two input patterns are not identical, as long as they are close enough, they are classified into one category.

Figure 4. An integrated framework of an intelligent agent



A BLACKBOARD FRAMEWORK

The blackboard framework is tested. This framework allows each model to function either independently or cooperatively (refer to Figure 3). This system framework has been simulated by passing the input patterns or output patterns from one network to another. For example, the input patterns and output patterns from back-propagation model can be used as input (or training data) to Back-propagation model; the output patterns from Back-propagation model or FCM model can be presented to Adaptive Resonance Theory model for classification. An output pattern from any network is viewed as the current user profile in the dialogue context. Thus, this framework provides an effective way for dynamic user modeling.

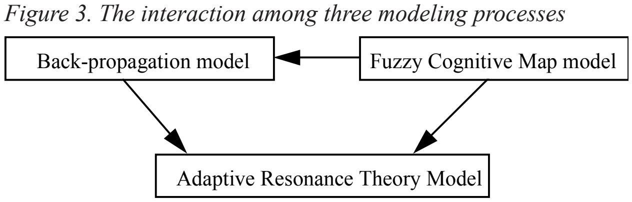
Figure 4 shows a framework of an intelligent agent in a design system, which establishes both user profiles and task profiles. These profiles provide the stereotypes for the system to adapt the users' behavior in the design process.

CONCLUSION

This study tested and integrated several user modeling approaches for intelligent agents in interactive systems. It has shown several advantages such as fast default reasoning and generalization, insensitivity to inconsistent input, personalization, and learning ability. In addition, comparing with the rule based systems, it is easier to implement and maintain the proposed system. Also, the knowledge elicitation process is easier than the rule-base construction, because only the causal relationships among two assumptions are considered at a time which do not enforce any consistency constraints to initiate the modeling process.

The further study is aimed at incorporating the task model into user models so that an agent can have a more comprehensive picture about not only who the user is but also *what he/she is going to do*. A multi-level network framework may be necessary. The modeling process should be not only handling a user's domain knowledge, but also the assumptions about the user's goals, plans and detailed actions in the task performance.

Figure 3. The interaction among three modeling processes



REFERENCES

1. Chen, Q., J. Wang, Q, Tu, "Building intelligent agents in the interface of design systems," Proceedings of 3rd. International Conferenceon Engineering Design and Automation, pp. 711-716, 1999
2. Chen, Q. and A. F. Norcio, "Modeling a user's domain knowledge with neural networks", *International Journal of Human-Computer Interaction*, Vol. 9, No. 1, pp. 25-40, 1997.
3. Drashansky, T.,. "Networked Agents for Scientific Computing," *Communications of ACM,* Vol. 42, No.3 1999.
4. de Kleer, J., "An Assumption-Based TMS," *Artificial Intelligence*, Vol. 28 No. 2, pp. 127-162 (1986).
5. Chen, Q. and Norcio, A. F., "An Associative Approach in Dynamic User Modeling," *Proceedings of 5th Intentional Conference of Human Computer Interaction,* pp 909-914 (1993)
6. Huang, X., Mccalla, G. I., Greer, J. E. and Neufeld, E., "Revising Deductive Knowledge and Stereotype Knowledge in a Student Model," *User Modeling and User-Adapted Interaction*, Vol. 1, No. 1, (1991)
7. Pao, Y., *Adaptive Pattern Recognition and Neural Networks*, Addision-Wesley Publishing Co., (1989).
8. Kosko, B., *Neural Networks and Fuzzy Systems*. Prentice Hall, 1992
9. Ted Selker, "Coach: A Teaching Agent That Learns," Communications of the ACM, Vol. 37, No. 7, 1994.

Related Content