Predicting Resource Usages with Incomplete Information

Jung-Jin Lee and Robert McCartney
Department of Computer Science and Engineering
University of Connecticut
Storrs, CT 06269-3155
[jjl|robert]@cse.uconn.edu

Abstract

This work explores the benefits of using user models for plan recognition problems in a real-world application. Self-interested agents are designed for the prediction of resource usage in the UNIX domain using a stochastic approach to automatically acquire regularities of user behavior. Both sequential information from the command sequence and relational information such as system's responses and arguments to the commands are considered to typify a user's behavior and intentions. Issues of ambiguity, distraction and interleaved execution of user behavior are examined and taken into account to improve the probability estimation in hidden Markov models.

1. Introduction

Either cooperative or self-interested agents try to improve their ability to recognize the likely actions of other agents in multiagent systems. The recognition of other agents' intentions is an important task. It is particularly so when an agent is expected to produce some useful information in an interactive computing environment, where acquiring knowledge of the current world serves as a basis for immediate or future actions of the system.

As part of acquiring knowledge of the current world, we focuse on acquiring knowledge of an observed agent such as its behavior or preferences. Self-interested agents are designed for the prediction of resource usage such as printer or file system in the UNIX domain; specifically, assessing the likelihood of upcoming demands by users on limited resources and detecting potential problems by observing human-computer interactions. The agent's discernability of what to observe will contribute to patternize user behavior of using resources by dealing with both uncertainty of the state of the world and the complexity of user behavior. It uses a stochastic approach to automatically acquire regularities of user behavior from history data of his/her interactions. A user's preferences are learned and built into a mathematical model then the individual mathematical model, based on a hidden Markov model, is used as a probabilistic model of recognizing/predicting the user's behavior.

2 Plan recognition using user model

The general area of inferring the goals and intentions of people through observations is commonly known as *plan recognition*. Inducing the plan of observed actions can be useful for predicting the agent's future behavior, interpreting its past behavior, or taking actions to influence the plan. The plan recognizer in this work is a *reactive* agent: it assesses the various hypotheses, selects the best, and takes some actions based on what is *currently* recognized using individual user models (UMs). The complete plan recognition problem in this role is extremely difficult; however, for some problems, *partial* recognition may be sufficient to predict the next behavior of a user, the usage of resources, or a short-term goal conflict [1].

Our domain of interest is human-computer interaction in a large, ongoing, and dynamic environment such as UNIX and WWW. Some difficult features in these domains include the nonstrict temporal orderings of actions, the interleaving of multiple tasks, the large space of possible plans, some sequence of actions are shared, suspended, and resumed to lead to multiple goals, and conditional plans where the condition is neither explicit nor directly observable. The UNIX domain is used as a testbed for this work.

As to the benefits of such predictions, the agent can use the prediction to take control actions both to help users and to better manage resources. For users, the agent can suggest, upon the predictions of their behaviors, users to send a file to printer2, since printer1 is jammed and printer3 has many jobs in the queue. This is a kind of *information push*, that is, the software agents are constantly trying to push information and services toward the user[3] rather than users take initiative to pull some information. For overall system performance, based on the measure of predicted use of printers, the agent can take the action of changing cartridges or warming up the printers or pull some information from other agents. We look at three prediction problems: predicting next behavior considering only a previous action, predicting the possibility of using resources with the partial sequence of actions observed, and predicting which resource is more likely to be used among competing ones.

3 Learning UM using machine learning

The user preference varies in detailed levels of actions and a stochastic approach is taken to learn user models. However, those actions share features: same abstraction, shared subplans, or same goals. We use templates as abstractions of actions, to facilitate the explanation of a user's plan with procedural structures[1]. While each action can be explained as an abstract class, some actions are optional to complete a plan or shared by multiple plans which are hidden from the observation. We use hidden Markov models (HMMs) to represent the ambiguity of actions towards plans within underlying structure. Due to the simplicity and efficiency of its parameter estimation algorithm, the HMM has emerged for modeling discrete time series of stochastic processes and sequences and speech recognition[2]. In the double embedded structure of this model, the outcomes of user actions are observable random variables, called the *observation sequence*, that each depend only on an underlying state variable (*output probability*). The hidden state variables which represent plans in turn depend only on their immediate predecessors (*Markov chains*).

Suppose the recognizing agent has two resources to manage: printer and router resources having three abstract plans, namely, "Printing" plan, "SendingOff" plan, and "Others" plan which pertains to actions other than actions under the "Printing" and "SendingOff" plans. The plan states are interconnected in such a way that any state can be reached from any other state, and the underlying stochastic process of the states has another set of stochastic processes that produce the sequence of observations. The outcome sequence depends very much on the parameter sets which characterize the HMM: the individual bias(preferences) of abstract plans, the transition probabilities between various states(plans), as well as initial states chosen as the starting points of observations. In Fig. 1, the likelihood of a partial sequence: {`ls, vi, latex`}, based on their correlation, to be in a "Printing" plan given a model can be computed with the sum of the joint probabilities, that is, the product of individual state transitions and output probabilities of observable actions.

[1] Abstract classes of actions are defined depending on the effects of actions on files. For example, *find* class of actions has no effects on files but gathers information such as *'ls'* or *'more'* and both *'latex'* and *'compress'* actions are a *format* class for actions of changing source files to destination files and/or creating auxiliary files, etc.

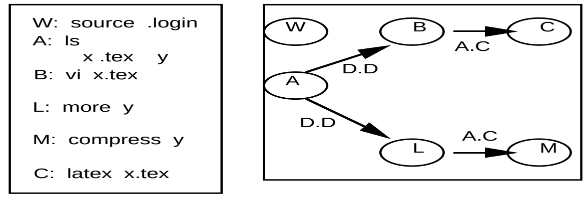

Figure 1. Action sequence and correlations

4 Acquisition of user knowledge

In command-driven systems like UNIX, observation of action sequence only is not enough to acquire the knowledge of user behavior[4]. To typify a user's behavior and intentions, *relational* information such as system's responses and arguments to the commands are considered besides *sequential* information from the command sequence. D.D. and A.C. in Fig. 1 represent *data dependency* and *argument coherence* rules respectively. In respect to learning user behavior, issues of *ambiguity, distraction* and *interleaved execution* are examined from the automatic analysis of recognizing shared actions, excluding extraneous actions related to resouce use and finding correlations and disclosing hidden states and taken into account to improve the accuracy of model parameters in HMMs. A graph-based analysis algorithm is used to disclose hidden states and shared actions.

5 Current status

The objective of this work is to develope a reactive agent in a real-world application, solving plan recognition problems using user models. The work is still in a training phase to produce reliable and stable models for predictions. A *Perl* script is used for automatic on-line data collection, capturing both sequential and relational information. These data are used for off-line analysis and evaluation of the predictions.

References

[1] J.J. Lee. Case-based plan recognition in computing domains, (Doctoral Consortium), In *Proceedings of the Fifth International Conference on User Modeling* pp. 234–236. Kailua-Kona, Hawaii 1996.

[2] L. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, vol. 77, No. 2, pp. 257–286, 1989.

[3] W. Van de Velde et al. Competition for Attention. in *Preproceedings of the 4th Int., Workshop on Agent Theories, Architectures, and Languages*, pp 282–296, Providence, RI. 1997.

[4] K. Yoshida and H. Motoda. Automated User Modeling for Intelligent Interface. in *Int. J. of Human Computer Interaction*, 8(3):237–258, 1996.