Emotionally Adaptive Intelligent Tutoring Systems using POMDPs

Natasha Jaques

April 24, 2013

Abstract

An emerging field in user-adaptive systems is affect adaptivity: modeling and responding to an estimation of the user's emotional state. This could be particularly useful in the context of Intelligent Tutoring Systems (ITS), where learning gains are sensitive to the user's affect. Previous research used an empirically validated Dynamic Bayesian Network (DBN) to create an ITS affect model, but could not plan affect-sensitive responses. This paper will extend this research by converting the model into a Partially Observable Markov Decision Process (POMDP) representation, in order to compute a plan of interventions for the ITS agent to take given an estimation of the user's mood and goals. Two different methods for solving the POMDP will be compared: Incremental Pruning (an exact method) and SARSOP (an approximate method). Factors affecting the tractability of POMDP representations will also be discussed.

1 Introduction

Developing adaptive computer interfaces that respond to the specific traits of each user is the focus of great interest. Through the use of statistical analysis and probabilistic reasoning, these systems can build a model of the user's goals, preferences, expertise, or cognitive abilities, in order to better serve user needs [16]. The ability to model and adapt to user tastes and preferences has already been extensively explored for commercial purposes [29]. Other applications include helping users perform tasks more efficiently and easily by adapting to their needs and abilities.

An area that has begun to be explored recently involves modeling and adapting to the user's emotional state, or affect [28]. Although affect adaptivity is a difficult problem, it has enormous potential. Not only does emotion have a substantial effect on human behaviour, but during human social interaction a great deal of information is communicated through emotional cues [14]. For a truly advanced artificial intelligence to interact effectively with a human, it should be able to adapt to these cues. Potential uses of an emotionally adaptive interface include helping those with disorders of emotional regulation or expression, such as individuals with Autism Spectrum Disorder (ASD) [25]. Such a system could help these individuals perceive the emotions of others, and also mirror the displayed emotions of an individual with ASD so s/he could learn how to express the emotions s/he wished to convey. Or, affect adaptivity could simply allow a system to react appropriately to, and potentially improve, any user's mood. Emotional sensitivity could be the next step in creating a truly intuitive and interactive computer interface.

Intelligent Tutoring Systems (ITS) provide an excellent application for modeling user affect. An ITS is usually a system with an artificially intelligent pedagogical agent, that adapts learning interactions to individual students' expertise and cognitive abilities [10]. The role of the ITS includes choosing material, planning tests, and giving tailored hints to the tutored students [12]. While research has shown that ITSs have led to increased learning gains, this effect could potentially be magnified by incorporating affect sensitivity [12]. Studies have shown that individuals who are able to maintain a positive affect while studying are able to deal better with confusion, and eventually obtain higher learning gains [8]. Attempts to incorporate affect sensitivity into ITSs have already been met with success; in one experiment, students perceived the emotionally adaptive version of an ITS as more useful and appropriate [26]. Recently, an ITS interface combined data from a camera, eye tracker, and body posture measurement device, in order to detect if the user experienced boredom, frustration or confusion [10]. The system was shown to be effective at promoting learning, especially for low domain knowledge students [10].

Although affect modeling is a promising research direction, especially in regards to ITS, it is an extremely difficult problem involving a great deal of uncertainty [2]. For this reason, current research suffers from several shortcomings. For instance, many affect models rely on data collected from expensive and cumbersome equipment that is only available in a lab. One such system used both eye tracking and electromyography (EMG) sensors placed on the user's forehead in order to determine the user's mood [7]. A superior system would utilize lab equipment and extensive testing to build user models, but only require interaction log data to predict the traits of a new user. A second problem is that because building the affect model can be so time consuming, much research stops short of putting the information obtained about the user's emotional state to good use [8]. Further research into how an agent can utilize the model to plan effective responses and interventions is required. Essentially, the next step is to find a way to translate simple observations about the user's behaviour into a plan of actions for the ITS agent, in the face of a great deal of uncertainty about the user's actual emotional state.

Fortunately, a mechanism for planning in the face of uncertainty based on observations of hidden states exists, and it is known as a Partially Observable Markov Decision Process (POMDP). POMDPs combine the concepts of a Hidden Markov Model (HMM) and a Markov Decision Process (MDP) into a model that is both able to deal with uncertainty about the current state of the world, and make decisions about the best action to take next [6]. Therefore POMDPs appear well-suited to modeling user emotion in an ITS context, and planning the most appropriate tutoring actions. It is the intention of this research to extend an ITS affect model previously developed by Zhou and Conati [28] using a POMDP, and thus incorporate the ability to plan and make decisions.

2 Background and Related Work

2.1 Partially Observable Markov Decision Processes

POMDPs have been used to model a variety of real world problems. The ability to handle uncertainty in movement and sensing makes POMDPs popular in robotics and navigation [18], but they have also been used for dialogue control, pursuit evasion, sensor placement, games, trust/reputation reasoning, planetary rover control, map learning [20], and aiding people with disabilities such as dementia [15].

A POMDP is a stochastic state process, occurring over multiple time steps, that can be described as a tuple $< S, A, Z, T, O, R >$, where: S is a finite set of states; A is the set of actions; Z is the set of observations; $T(s' \mid s, a)$ is the transition function, and gives the probability of transitioning from state *s* to state *s'* after taking action *a*; $O(z \mid a, s')$ is the observation function, and gives the probability of seeing observation *z* after taking action *a* and ending up in state *s'*; and $R(s, a)$ is the reward function, which gives the expected reward for taking action *a* in state *s* [17]. A *belief state b* is a probability distribution over all states, where $b(s)$ gives the probability of being in state *s*. The initial belief state b_0 gives the probability of starting in each state. A *discount factor* γ can be used to calculate the value of rewards in the future. A reward *t* time steps in the future is worth $\gamma^t r$ [20].

In an MDP, the solution consists of a policy π that simply dictates which action to take in every state [17]. The value of a policy $V_\pi(s)$ can be determined based solely on the expected reward. However, in the case of a POMDP, a policy π is a mapping from a belief state to an action. Because the dimensionality of a belief state grows with the number of states, explicit representation of a policy for every possible belief state becomes intractable for all but the smallest problems [1]. Finding the optimal policy involves finding the maximum set of *alpha vectors*, where an alpha vector is a linear function that gives the expected reward of taking a given action over all belief states. Together, the alpha vectors form a piecewise linear and convex value function, which maps belief states into expected total discounted reward [5] (see Figure 1).

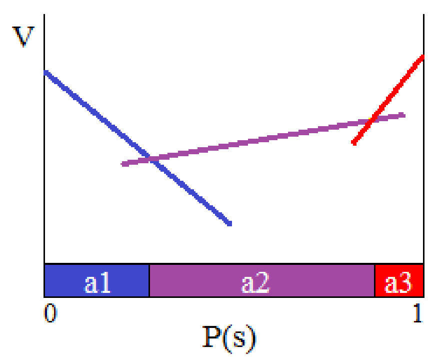

Figure 1: A value function with three alpha vectors, showing how the value of each action changes with the belief in the probability of being in state *s*

As the POMDP steps through time and *t* increases, the complexity of the value function and the number of alpha vectors needed to represent it can grow doubly exponentially in *t*. When the optimal policy is not *finitely transient* [20], an infinite number of linear functions may be needed to describe the optimal value function, so the number of alpha vectors can actually grow without bound. Therefore, solving POMDPs may actually be impossible in the worse case. In fact, a theoretical result has shown that a completely general exact POMDP solution algorithm could be used to solve the halting problem [20]. Essentially, the complexity of solving a POMDP *exactly* is "forbidding", and makes all but the smallest example problems intractable [19]. Ironically, although POMDPs are able to represent real world problems and their inherent uncertainty much more accurately than MDPs, they are so much more complex that only simple, toy examples that do not reflect the real world are able to be solved exactly [3]! Fortunately, many approximate algorithms exist that can efficiently compute an estimate of the optimal policy. The intent of this paper is to compare the practicality and accuracy of both an exact and approximate solution to a POMDP used in an affect-sensitive ITS context.

There are a number of methods for solving a POMDP exactly, including the value iteration, policy iteration, witness, linear support, and one pass algorithms [5]. Many of these utilize dynamic programming to gradually improve the estimation of the value function each iteration. Perhaps the most efficient method currently available is Incremental Pruning (IP), developed by Zhang in 1996 [27]. IP builds on the idea that many alpha vectors in the value function become completely dominated by other vectors, and are thus irrelevant to the optimal solution. Pruning these vectors leads to a more compact representation of the value function at time t, making it more tractable to compute the value function at time $t+1$. Although IP is not the only method that prunes these extraneous alpha vectors, it goes one step farther by breaking down the equation for the value function into three parts, and optimizing each part using findings from linear algebra and proofs developed by Zhang [27]. Empirically, over 95% of the total execution time of exact POMDP algorithms is spent solving linear programs, so these optimizations are extremely beneficial [5]. This paper will apply Incremental Pruning to the affect sensitive ITS model and demonstrate that despite its high efficiency, only the simplest formulations of the problem are able to be solved exactly.

Because exact solution methods can fail with less than a dozen states, they can be "woefully inadequate" for handling real world problems [18]. Therefore this paper will also present the results of using an approximate solver known as SARSOP, or Successive Approximations of the Reachable Space under Optimal Policies [18]. As mentioned before, one of the biggest roadblocks in computing POMDPs involves representing the value function at each possible point in the exponentially sized belief space. This insight has led to the development of point-based methods, which represent the value function at only a few sampled points in the belief space - preferably those points which are reachable from the initial belief state under an arbitrary sequence of actions. SARSOP goes one step farther by attempting to represent only the subset of points in the belief space that are reachable under the *optimal* sequence of actions [18]. Since the optimal sequence of actions cannot be known ahead of time (after all, that is what the algorithm is trying to compute), SARSOP uses a predictive heuristic to initially guide exploration of the belief space. Then, pruning is used to avoid sampling the belief space in regions that are unlikely to be optimal, and instead focus on those that appear to be most relevant to the final solution. This type of belief space pruning also enables efficient alpha vector pruning [18].

2.2 Affect Modeling and Intelligent Tutoring Systems

This research intends to expand upon that of [28], in which Zhou and Conati built a probabilistic model of the affective states of students using an ITS called Prime Climb. The model is based on findings from psychology, including the OCC theory of emotions [24]. This theory states that emotional reactions are contingent on an individual's goals, and outcomes in the world that affect these goals. Therefore Zhou and Conati [28] used empirical survey results to develop a set of goals that students using Prime Climb tend to have, and model how events in the software affect those goals. OCC theory dictates that when a goal becomes satisfied, the student will be in a state of positive affect. Conversely, if an event opposes the student's goal, it will lead to a negative affective state. [24]. In order to create a model that can map new users to goals, the Five Factor Personality model was leveraged [9]. In the end, a probabilistic model was developed in which dependencies among goals, personality, and interface actions were established through correlation analysis between the goal questionnaire results, personality test results, and interaction logs [28]. The model is also dynamically updated; as more evidence becomes available, the posterior probabilities are updated in real time, during the game [8]. Although the model achieved an accuracy of 68.72% in identifying user emotions, it was not able to use this estimate to plan which action to take [8]. The next step in this work is to use the modeled relationships between the students' behaviour, goals, and emotions to allow the agent to plan how to intervene and instruct more effectively. A POMDP may be an excellent tool for this task, as POMDPs have already been used in work on goal recognition [11]

In fact, POMDPs have also been applied to modeling affect, although not in the context of an ITS. Bui et al. [3] used a POMDP to model the stress level of rescue workers issuing voice commands and queries to an automated system. Like [28], this system also modeled both goals and emotions, but not the relationship between them. Only the effect of emotions on actions was represented [3]. The authours assert that a POMDP is well suited to the task of responding appropriately to uncertain, ambiguous, and constantly changing emotion. This claim was supported by the fact that their POMDP strategy outperformed three handcrafted strategies when the user committed error induced by stress [3]. The authours hope that a POMDP-based strategy will improve the weaknesses of previous affect modeling techniques, which have poor confidence estimates, cannot cope with dynamic, changing user goals, and are labour intensive to create [3].

Work on using POMDPs in an ITS context has also begun, but these systems do not incorporate affect sensitivity. Folsom-Kovarik et al. [12] created a system which used a POMDP to model students' domain knowledge, so that an ITS could plan which topic to teach next. Because of the ability of a POMDP to plan and handle uncertainty, the authours felt it could be beneficial in an ITS situation in which material and hint selection requires planning and timing, and interventions can be more or less helpful depending on the cognitive states and traits of the user [12]. However, when attempting to create an ITS POMDP, they found that the largest possible model they could create had only eight states; any more and the SARSOP implementation they were using exhausted all 8G of available memory [12]. These authours have demonstrated that although POMDPs may be theoretically appropriate for an ITS, in practice modeling them presents many challenges - a theme which will also be reflected in the results of this paper.

The research presented in this paper attempts to build on and combine the work of [28], [3] and [12] in order to build an affect-sensitive Intelligent Tutoring System using a POMDP. The next two sections will discuss how the problem was modeled and the results that were obtained.

3 Approach

A diagram of a portion of the affect model developed in [28] is presented in Figure 2. The top row of personality traits represent those components of the Five Factor model that were found to be most relevant to the goals of students using Prime Climb. The next row represents the goals themselves, which include *Have Fun, Avoid Falling, Beat Partner, Learn Math,* and *Succeed By Myself.* Some implementations of this model also included a goal called *Want Help*. These goals tend to manifest themselves as interac-

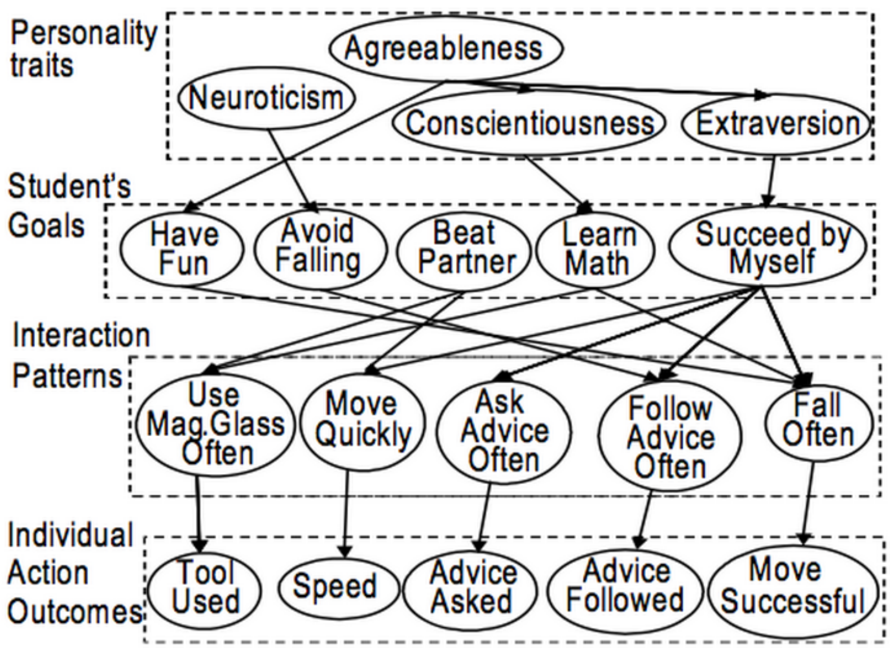

Figure 2: Affect Model as described in [28]

tion patterns within the Prime Climb game. The object of the Prime Climb game is to climb to the top of a mountain of prime numbers, by moving only to numbers that are coprime with the partner's number. If a student moves to a number that shares a common factor with his or her partner's number, s/he will fall. There is also a magnifying glass tool that lets the user click on a number to see its factors, and a help dialog box that lets the user ask for advice.

This portion of the model is responsible solely for predicting the user's goals; the rest of the affect model is not pictured. However, it can be described simply as follows: the user's goal, the user's action outcome (including climb or fall), and the agent's action predict whether the user's goal is satisfied. If it is satisfied, the user will be happy; if it is opposed, the user will be unhappy. The actions the agent can take include providing a hint, providing help when the user is stuck, teaching a concept, encouraging the user, or doing nothing.

To model this network as a POMDP, I treated the goals as possible states. Observations are the interface behaviours exhibited by the student: *use tool, move quickly, use help, fall, climb*. The observation probabilities were based on the conditional dependencies in the original model [28]. The actions the agent can take are: *hint, help, teach, encourage, none*. Probabilities from the original model that relate to the student's goal being satisfied were used to create reward values. For example, if the student is in state *Learn Math*, and the agent chooses action *teach*, the student's goal will be satisfied with probability 0.6. Therefore, a reward for choosing *teach* given *Learn Math* could be modeled as 0.6 (or 6 if integer rewards are required). All models used a discount factor of 0.95, which was based on an example presented in [20].

Unfortunately, in the original research no data about how the user's goals change over time was collected [8]. To deal with this problem, I created several models with different assumptions. In models with *identity* transition probabilities (see Figure 3), the assumption is that the user's goals are static, and do not change over time. The opposite assumption, that the user's goals are constantly changing and each goal has equal probability of transitioning to every other goal, is described by *uniform* transition probabilities. To create versions with goal transitions based on data presented in [28], I used the relationship between the agent's actions and the student affect outcomes to design transition probabilities (shown in Figure 3 as a *small number* of transitions). I accomplished this by adding a *Goal Satisfied* state, and having transitions to that state based on the previous state and the agent's actions (the probabilities formerly used for rewards). I then simply added reward values for being in the *Goal Satisfied* state.

Figure 3 presents the various model formulations and their relevant characteristics. *Affect1* is the original model, in *Affect2* the goal *Want Help* was removed, *Affect3* incorporated personality as states, *Affect4, Affect6, Test2* and *Test3* add a small number of transition probabilities by having a *Goal Satisfied* state, *Affect5* makes the assumption of uniform transition probabilities and incorporates integer rewards, and *Affect6* is identical to *Affect5* except without uniform transitions, designed to test the assumption that transitions strongly affect tractability. In all but one version of the model (*Affect3*), I discarded the information about personality traits. Although models with this simplification do not truly encompass all information gathered in [28], I wished to create a model that was not dependent on unrealistic measures such as a personality test, which the user might find intrusive or time consuming and be unwilling to provide. I felt that a model that decided the appropriate response based solely on readily available data about the user's interface actions would be more useful.

Figure 3 also includes test models, which are greatly simplified models that involve a limited number of states, actions, and observations, designed to test the effect of the magnitude of these variables against the effect of the transition probabilities. The number of states, actions, and rewards, and whether reward values are represented as rational or whole numbers can all affect tractability [21]. Another factor is the degree of "observability", where a completely unobservable model has only one observation that is equally likely in every state, and a fully observable model has one observation for each state [21]. Therefore the ratio of observations to states in also an important factor.

The exact solution algorithm used for this research was Incremental Pruning, as implemented by Anthony Cassandra in his publicly available pomdp-solve-5.3 software [4]. Tests using this algorithm were performed on an iMac with a 2.4GHz Intel Core 2 Duo processor and 2GB 667 MHz DDR2 SDRAM. The approximate solution algorithm used was SARSOP, and the implementation used was APPL [23]. Experiments using this algorithm were performed on a PC with a 2GHz AMD Dual-Core processor with 4GB RAM. The approximate solver was set with a timeout of five hours, after which the best solution obtained was recorded.

4 Results

						Exact Method - Incremental Pruning			Approximate Method - SARSOP		
Model	**# States**	**# Actions**	**# Obs**	**Rewards**	**Transitions**	**Solution Time (s)**	**# Alpha Vectors**	**Optimal Expected Reward**	**Solution Time (s)**	**# Alpha Vectors**	**Expected Reward**
Affect1	6	5	5	Real	Identity	Did not finish	496	?	18000*	1,228	3.317
Affect2	5	5	5	Real	Identity	Did not finish	973	?	18000*	1,497	4.395
Affect3	10	5	5	Real	Small number	Did not finish	3269	?	18000*	1,272	3.432
Affect4	7	5	5	Real	Small number	Did not finish	521	?	18000*	743	8.761
Affect5	**6**	**5**	**5**	**Integer**	**Uniform**	**0.062**	**2**	**32.757**	**0.639**	**2**	**32.477**
Affect6	6	5	5	Integer	Small number	Did not finish	6453	?	18000*	6,941	10.206
Test1	2	3	2	Real	Identity	Did not finish	278	?	0.745	774	10.670
Test2	3	3	2	Real	Small number	Did not finish	596	?	18000*	14,613	1.845
Test3	3	3	2	Integer	Small number	Did not finish	614	?	18000*	9,234	11.386
Test4	**2**	**3**	**2**	**Integer**	**Uniform**	**0.060**	**1**	**99.990**	**0.003**	**1**	**99.365**

Figure 3: The various POMDP models, their characteristics, and the results of solving them with both an exact and approximate method. An exact solution was found for rows in bold. Solution times followed by a * indicate that the algorithm timed out before the minimum error threshold was reached.

Figure 3 presents the different models and the results of solving them. It is clear from the table that the biggest factor in the tractability of the models relates to the transition probabilities. Those models with *identity* transition probabilities are actually impossible to solve exactly. In order for a POMDP solution to converge to a unique stationary probability distribution, the state transition diagram must be a singly connected component, or *irreducible* [22]. With no transition probabilities, not only will the state transition diagram not have this property, but it will in fact consist of a number of completely disconnected components, one for each state.

On the other hand, making the uniform transition probabilities assumption essentially means that each state is equally likely at every time step. This means that the belief about the current state is not dependent on the entire history of actions and observations, but only on the most recent observation. Thus, a model with uniform transition probabilities will not have an exponentially expanding number of alpha vectors, one of the biggest drawbacks to the POMDP model. Similarly, actions that are purely information seeking will not be useful in this type of POMDP model. Yet it is still distinct from a fully observable MDP, because observations must be used to make a probabilistic estimate of the unknown, hidden state, before a decision can be made. It should be noted that uniform transition probabilities have been used in classic POMDP example problems [20].

If information about how goals evolved over time was known, the assumption that they change constantly might not be realistic. However, in the absence of any such information it appears more likely that goals will change rather than remain static. Indeed, Zhou and Conati point out that because of the dynamic nature of their model, new evidence is used to update the conditional probabilities, which will most likely result in a new and different estimate of the user's goal during the course of the interaction [28]. Previous work on goal recognition also makes the assumption that goals change with respect to time [13]. Therefore, in the face of no evidence about how goals change over time, the assumption that they change constantly appears more realistic than the assumption goals are static, and produces simple and interpretable results.

An example of a simple solution obtained for the *Affect5* model is shown in Figure 4. This solution consists of a policy graph, which dictates the best actions to take in each belief state, and is one of the outputs produced by the SARSOP algorithm. This policy involves only two belief states, *B1* corresponds to a state in which the goals *Learn Math*, and *Want Help* are the most likely and *Succeed By Myself* is the least likely, and in state *B2*, the goal *Have Fun* is the most likely and everything else is equally less likely. As shown in Figure 4, the prescribed action for *B1* is to teach the student, which makes sense given that the rewards for teaching when the user's goal is *Learn Math* or *Want Help* are 6 and 8, respectively, and the reward for teaching when

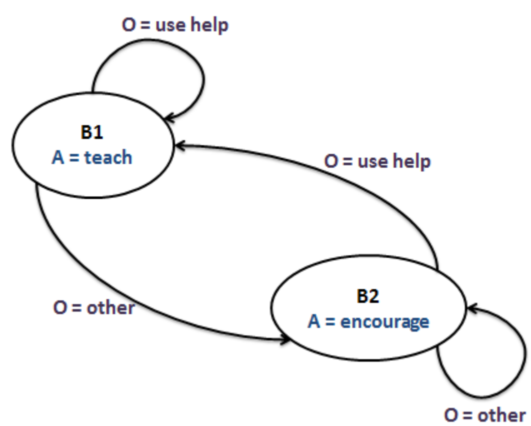

Figure 4: The policy graph computed for the Affect5 model, containing only two belief states. A indicates a recommended action, and O an observation.

the user's goal is *Succeed By Myself* is -9. The prescribed action for belief state *B2* is to encourage, which reflects the fact that there is a reward of 9 for encouraging a student whose goal is *Have Fun*, and there are no negative rewards for encouraging the student in any state. Recall from section 3 that these reward values are based on empirically validated probabilities of the student's goal becoming satisfied given the agent's action. While this policy is extremely simple, I would argue it is a completely reasonable and acceptable approach. A human tutor is most likely to teach her student when she perceives that the student wants help or wants to learn, and in the absence of these observations she may simply encourage the student.

Nearly all of the models (except those with uniform transitions) were not able to be solved exactly. Although an approximate solution was obtained for each model, for many of them it could not converge to a stationary probability distribution within the five hour time limit. The difficulty of solving POMDPs is a common problem, also experienced by [3], [19], [12], [15], [5] and [18]. It seems that an approximate solution may be the only realistic and feasible option for any sufficiently complex problem. Fortunately, most approximate solutions are highly accurate [18]. This is also reflected in Figure 3; for the two models in which the exact solver finished, and the optimum expected reward is known, the approximate solution's expected reward is only off by a small margin. It is clear, then, that an approximate method like SARSOP is a practical and useful way to approach solving POMDPs. This is because SARSOP is an anytime algorithm, so even for those models on which it timed it, it was still able to output and save the best solution discovered within the time limit [18]. Any of these solutions can therefore be incorporated into the ITS model of [28] in order to help the agent in Prime Climb decide on the best intervention at each time step.

5 Future Work

Obviously the major difficulty in this research was guessing at how to model transition probabilities in the absence of evidence about how goals change over time. Therefore, the next step is to conduct a study about how users' goals relate to each other and evolve over time while playing Prime Climb. This information can be used to create a more realistic POMDP model.

Another interesting direction would be to extend the network described in [28] into a true Dynamic Decision Network, in which the utility of various actions is modeled explicitly, and the net could be used to calculate decisions. In fact, there is a great deal of similarity between DDNs and POMDPs; a DDN can be used as input to an exact pomdp solver [2]. In [2], DDNs were actually explicitly combined with POMDPs to form a probabilistic estimate of emotion and decide how to act.

To make the POMDP models more efficient, it is also possible to decompose the problem into modules which are handled separately from the computation of the POMDP solution. [15] uses a module called a *belief monitor* to process camera images and estimate which step of a handwashing task an elderly individual is currently performing. Similarly, [3] uses a module called a *state estimator* to compute the internal belief about the user's stress level based on their vocal patterns, such as pitch, volume, etc. Essentially, supplementing a POMDP with external modules which implement techniques such as particle filters or Bayesian networks to interpret observations or decide actions can improve efficiency [12].

Finally, another promising direction that could be particularly beneficial to this research is Limited Memory Influence Diagrams, or LIMIDs [19]. These are essentially decision networks which relax the "no forgetting" assumption: that the current state is dependent on the entire history of actions and observations. Relaxing this assumption leads to increased computational efficiency [19]. Coincidentally, making all transition probabilities uniform in this research was a crude form of relaxing the no forgetting assumption. This is because if it is equally likely that the previous state was any of the other states, then the previous state provides no information about the likelihood of the current state. This could be seen as an extreme form of "limited memory" - complete amnesia. Another innovation proposed in [19] relates to pruning *non-requisite* nodes; these are nodes that do not affect the outcome or potential utility of a decision variable. This technique could also be usefully applied to the affect model presented here. For example, there is no action that the agent can take to help a student accomplish the goal *Beat Partner*, so this information appears to be largely irrelevant to the model. Calculating and pruning irrelevant nodes could greatly decrease the complexity of the model.

6 Conclusion

Modeling user affect could potentially benefit a number of applications, particularly Intelligent Tutoring Systems in which learning gains are sensitive to the emotional state of the user. Since affect modeling is a difficult problem involving a great deal of uncertainty, and ITSs require that the pedagogical agent choose the correct action at the correct time, this problem seems like the ideal application for a POMDP. However, many real world problems are almost impossible to represent as a POMDP that can be solved exactly. This paper presented the results of modeling an existing probabilistic network as a POMDP, and the difficulties therein. It demonstrated that by making a simplifying assumption that the user's goals are constantly changing and do not depend on their previous goals, the problem becomes quite tractable. However, even without this assumption, multiple approximate solutions exist that will allow a pedagogical agent to select the most emotionally appropriate action while tutoring a student. This work could be extended by researching how goals relate to each other over time, as well as incorporating new innovations related to decision-theoretic planning and inference in stochastic domains.

References

[1] X. Boyen and D. Koller. Tractable inference for complex stochastic processes. In *14th conference on Uncertainty in Artificial Intelligence*, pages 33–42, 1998.

[2] Trung H Bui, Mannes Poel, Anton Nijholt, and Job Zwiers. A tractable ddn-pomdp approach to affective dialogue modeling for general probabilistic frame-based dialogue systems. 2006.

[3] Trung H Bui, Job Zwiers, Mannes Poel, and Anton Nijholt. Toward affective dialogue modeling using partially observable markov decision processes. 2006.

[4] Anthony Cassandra. Pomdp solver software, 2009. `http://www.cassandra.org/pomdp/code/`.

[5] Anthony Cassandra, Michael L Littman, and Nevin L Zhang. Incremental pruning: A simple, fast, exact method for partially observable markov decision processes. In *Proceedings of the Thirteenth conference on Uncertainty in artificial intelligence*, pages 54–61. Morgan Kaufmann Publishers Inc., 1997.

[6] Anthony R Cassandra, Leslie Pack Kaelbling, and Michael L Littman. Acting optimally in partially observable stochastic domains. In *Proceedings of the National Conference on Artificial Intelligence*, pages 1023–1023. JOHN WILEY & SONS LTD, 1995.

[7] Cristina Conati. Combining cognitive appraisal and sensors for affect detection in a framework for modeling user affect. In *New Perspectives on Affect and Learning Technologies*, pages 71–84. Springer, 2011.

[8] Cristina Conati and Heather Maclaren. Empirically building and evaluating a probabilistic model of user affect. *User Modeling and User-Adapted Interaction*, 19(3):267–303, 2009.

[9] Paul T Costa and Robert R McCrae. Four ways five factors are basic. *Personality and individual differences*, 13(6):653–665, 1992.

[10] Sidney D'Mello, Blair Lehman, Jeremiah Sullins, Rosaire Daigle, Rebekah Combs, Kimberly Vogt, Lydia Perkins, and Art Graesser. A time for emoting: When affect-sensitivity is and isn't effective at promoting deep learning. In *Intelligent tutoring systems*, pages 245–254. Springer, 2010.

[11] Alan Fern, Sriraam Natarajan, Kshitij Judah, and Prasad Tadepalli. A decision-theoretic model of assistance. In *Proc. of AAAI*, volume 6, page 96, 2007.

[12] J Folsom-Kovarik, Gita Sukthankar, Sae Schatz, and Denise Nicholson. Scalable pomdps for diagnosis and planning in intelligent tutoring systems. In *AAAI Fall Symposium on Proactive Assistant Agents*, 2010.

[13] Eun Y Ha, Jonathan P Rowe, Bradford W Mott, and James C Lester. Goal recognition with markov logic networks for player-adaptive games. In *Seventh Artificial Intelligence and Interactive Digital Entertainment Conference*, 2011.

[14] James V Haxby, Elizabeth A Hoffman, M Ida Gobbini, et al. Human neural systems for face recognition and social communication. *Biological psychiatry*, 51(1):59–67, 2002.

[15] Jesse Hoey, Axel Von Bertoldi, Pascal Poupart, and Alex Mihailidis. Assisting persons with dementia during handwashing using a partially observable markov decision process. In *Proc. Int. Conf. on Vision Systems*, volume 65, page 66, 2007.

[16] Eric Horvitz, Jack Breese, David Heckerman, David Hovel, and Koos Rommelse. The lumiere project: Bayesian user modeling for inferring the goals and needs of software users. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pages 256–265. Morgan Kaufmann Publishers Inc., 1998.

[17] Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1):99–134, 1998.

[18] Hanna Kurniawati, David Hsu, and Wee Sun Lee. Sarsop: Efficient point-based pomdp planning by approximating optimally reachable belief spaces. In *Proc. Robotics: Science and Systems*, volume 62, 2008.

[19] Steffen L Lauritzen and Dennis Nilsson. Representing and solving decision problems with limited information. *Management Science*, 47(9):1235–1251, 2001.

[20] Michael L Littman. A tutorial on partially observable markov decision processes. *Journal of Mathematical Psychology*, 53(3):119–125, 2009.

[21] Martin Mundhenk, Judy Goldsmith, and Eric Allender. The complexity of policy evaluation for finite-horizon partially-observable markov decision processes. In *Mathematical Foundations of Computer Science 1997*, pages 129–138. Springer, 1997.

[22] Kevin P Murphy. *Machine learning: a probabilistic perspective*. The MIT Press, 2012.

[23] University of Singapore. Approximate pomdp planning toolkit, 2012. `http://bigbird.comp.nus.edu.sg/pmwiki/farm/appl/index.php?n=Main.HomePage`.

[24] Andrew Ortony, Gerald L Clore, and Allan Collins. *The cognitive structure of emotions*. Cambridge university press, 1990.

[25] Stela H Seo, James E Young, and Andrea Bunt. Exploring user attitudes toward affect recognition in web-capable applications. 2012.

[26] Konstantin Zakharov, Antonija Mitrovic, and Lucy Johnston. Towards emotionally-intelligent pedagogical agents. In *Intelligent Tutoring Systems*, pages 19–28. Springer, 2008.

[27] Nevin L Zhang and Wenju Liu. Planning in stochastic domains: Problem characteristics and approximation. Technical report, Citeseer, 1996.

[28] Xiaoming Zhou and Cristina Conati. Inferring user goals from personality and behavior in a causal model of user affect. In *Proceedings of the 8th international conference on Intelligent user interfaces*, pages 211–218. ACM, 2003.

[29] Yunhong Zhou, Dennis Wilkinson, Robert Schreiber, and Rong Pan. Large-scale parallel collaborative filtering for the netflix prize. In *Algorithmic Aspects in Information and Management*, pages 337–348. Springer, 2008.