An I-POMDP Based Multi-Agent Architecture for Dialogue Tutoring

Fangju Wang

School of Computer Science, University of Guelph, Guelph, Canada N1G 2W1

Abstract

Dialogue systems have been widely considered as useful tools for education. The challenging tasks in developing a dialogue tutoring system include correctly interpreting student input and choosing appropriate responses. In this paper, we present a two-agent architecture for addressing the challenges. The two agents are learner agents in a reinforcement learning algorithm, which is based on the interactive partially observable Markov decision process (I-POMDP). One agent learns user behavior for disambiguating student input, and the other learns the optimal teaching strategies.

Keywords: Computer based education, dialogue system, interactive partially observable Markov decision process.

1. Introduction

A *dialogue system* interacts with its user in a text or spoken natural language. It interprets user input and responds to the user based on its interpretation. Dialogue systems have been widely considered as useful tools for education, like tutoring, for the advantages of higher flexibility, lower costs, and easier update of knowledge bases [1]. In the following, we refer to a dialogue system for tutoring as *dialogue tutoring system* (*DTS*).

The challenging tasks in developing a dialogue system include correctly interpreting user input and choosing appropriate responses. A natural language is ambiguous. Ambiguity may cause great difficulties in interpreting user input. In a tutoring system, system responses are controlled by a *teaching strategy*. It is impractical to set up a fixed, universal strategy. An effective strategy is one that may be adaptive to the student, and control the teaching based on the student's study states.

In our research, we develop a two-agent architecture to address the challenges. The two agents are learner agents in a *reinforcement learning* (*RL*) algorithm. One agent learns the student behavior, and uses the knowledge about student behavior to predict student actions (e.g. questions) in different states. When ambiguity occurs in a student input, information about the predicted actions can be used for disambiguation. The other agent learns the teaching strategy that may best satisfy the student. We develop the RL algorithm on *the interactive partially observable Markov decision process* (*I-POMDP*).

This work is a continuation of our research to develop RL algorithms for building DTSs. In our previous work, the RL algorithms were based on Markov decision process (MDP) and then based on partially observable Markov decision process (POMDP).

In this paper, we review the related work, and introduce the technical background. Then we present our RL algorithm on I-POMDP, and explain how the two agents model student behavior and system teaching strategy.

2. Related Work

RL has been an effective tool for developing dialogue systems. In [2], the recent research advances in applying RL in dialogue systems are reviewed. RL has been applied in tutoring systems. In the work reported in [6], Litman and Silliman developed an RL algorithm in building a DTS called ITSPOKE. In [1], Forbes-Riley and colleagues applied RL to learn from two corpuses. In the existing work, RL algorithms are mainly used to learn optimal dialogue strategies.

In the RL algorithms for building dialogue systems, some are based on the POMDP, rather than on the conventional MDP. The algorithm developed by Williams and Young was based on POMDP [9]. In the algorithm, a dialogue system was modeled as an SDS-POMDP, in which states are not observable. Other work to apply POMDP in dialogue systems can be found in, e.g. [4] and [7].

In [10], the authors argue for formulating the problem of an agent learning interactively from a human teacher as an I-POMDP, where the agent programming to be learned is captured by random variables in the agent's state space. In the algorithm, the human teacher is modeled as a distinct agent.

3. Technical Background

We briefly introduce the technical background of RL, POMDP, and I-POMDP.

RL is a machine learning technique [8]. In an RL algorithm, one or more agents learn knowledge through interactions with the *environment*. An RL algorithm can be represented by tuple (S, A, T, ρ), where S is a set of *states*, A is a set of *actions*, T is a *transition probability* function, and ρ is a *reward function*. A learner agent has a *policy*, denoted by π. The policy is used to guide the agent to select action $a \ni A$ in state $s \ni S$ to take, to maximize the long term *return* R_t, which is defined as $R_t = \sum_{k=0}^{n} \gamma^k r_{t+k+1}$, where γ is a discount factor, and r is a *reward* returned by ρ. Policy π can be defined as

$$\pi(s) = \text{argmax}_a \, Q(s, a), \tag{1}$$

Or

$$\pi(s, a_k) = \frac{Q(s,a_k)}{\sum_{i=1}^{n} Q(s,a_i)} \tag{2}$$

where

$$Q(s, a) = \sum_{s'} T(s, a, s') V(s') \tag{3}$$

where $T(s, a, s')$ is the probability of transiting from s to s' after a is taken, and $V(s)$ is

$$V(s) = \sum_a \pi(s, a) \sum_{s'} T(s, a, s')[R_{ss'}^a + \gamma V(s')], \tag{4}$$

where $R_{ss'}^a$ is the expected reward when transiting from s to s' after a is taken. Q and V are called *value functions*.

From time to time, the agent evaluates the performance of π while it uses π to select optimal actions. If the agent does not satisfy the performance, it improves π by updating the value functions. The agent learns the optimal policy by repeated evaluation and improvement.

RL is based on MDP. In MDP, state s represents the information that is needed to make a decision in s. This requires that all the states are fully observable, which means that we know exactly which state the system is in at any point of time.

In applications where states are not fully observable, an RL algorithm based on MDP has difficulties to make decisions, as can be seen in Eqn (1). For such applications, we need POMDP, the partially observable MDP [5].

POMDP has an important component called *belief state*. POMDP can be represented as tuple $(S, A, T, \rho, O, Z, b_0)$, where

S, A, T, and ρ are the same as described above, O is a set of observations, Z is an observation probability function, and b_0 is the initial belief state. A brief state, denoted by b, is a set of the probabilities over all the (physical) states:

$$b = [b(s_1), b(s_2), \dots, b(s_N)] \quad (5)$$

where $s_i \in S$ is the ith physical state, N is the number of states, and $b(s_i)$ is the probability of being in s_i. $b(s_i)$ is a function of the previous b, transition probability T, and observation probability Z. In POMDP, π is a function of b, since s is not fully observable. π can be defined as

$$\pi(b) = \text{argmax}_{p \in P} V_p(b) \quad (6)$$

where $V_p(b)$ is the value function of b:

$$V_p(b) = \sum_{s \in S} b(s) V_p(s) \,. \quad (7)$$

4. I-POMDP for a Tutoring System

4.1. I-POMDP

I-POMDP was recently developed for applying POMDP in multi-agent settings [3]. The I-POMDP of agent i is represented by tuple $(IS_i, A, T_i, O_i, Z_i, \rho_i)$. I-POMDP is characterized by IS_i, which is a set of *interactive states*. In the case of two agents, i and j, $IS_i = S \times M_j$, where S is the set of physical states, and M_j is the set of possible models of agent j. $m_j \in M_j$ consists of a sub-intentional model, and an intentional model of j. It is a model of the belief, preference and rationality in action selection. Note the discussion regarding i is also applicable to j.

In I-POMDP, a belief state of agent i is a set of probabilities over the interactive states. Thus, it includes information about agent j's belief, which includes j's belief about i's belief, and so on. The policy of agent i is a function of i's belief state. This implies that when i chooses an action, it takes into consideration not only the state it is in, but also j's belief, preference, rationality, and so on. I-POMDP provides a better framework for RL in some multi-agent applications. It enables an agent to make a decision based on knowledge about other agents' intention.

4.2. Modeling student and teacher

In our research, we develop an I-POMDP architecture for applying RL to a dialogue tutoring system. The RL algorithm has two agents: the teacher agent i and student agent j. The policy of the teacher agent π_i is the teaching (dialogue) strategy, and the policy of the student agent π_j models the behavior of the student. The two agents share the physical state space S, and have individual interactive state spaces IS_i and IS_j. As described before, IS_i contains information about the student agent's belief, preference, rationality, and so on, and vice versa.

We define the physical states in term of the related concepts in the subject that the system teaches. For example, "limit", "quotient difference" are such concepts in calculus. A physical state represents a "study state" of the student: What concepts the student understands and what the student does not. When the system teaches a student, it does not know exactly the student's study states, especially at the beginning. A suitable way to represent its belief about the study states is thus a set of probabilities over the study states. This is the belief state in POMDP. The dialogues are taken as observations.

As mentioned, the policy of the teacher agent $\pi_i(b)$ is actually the teaching strategy. It is used to select the most appropriate response to each student input. Belief state b contains the probabilistic information about the student's current study state and preference, as well as rationality. π_i is created and repeatedly improved to

select the responses that satisfy the student best. The reward function ρ_i is defined to measure the satisfaction, in terms of the number of times the student explicitly rejects the responses, or asks about some "prerequisite" concepts.

The policy of the student agent $\pi_j(b)$ models the behavior of the student. It can be used to predict the most probable action, e.g. question, by the student in a given b. The prediction is based on the student's current study state, as well as information about the teacher agent, like observation history, preference, and rationality. When interpreting an ambiguous student input, the agent may use the prediction to help resolve the ambiguity. π_j is created and repeatedly improved for best match between predicted and actual student actions. Reward function ρ_j is defined in terms of the similarity between predicted and actual actions.

5. Summary

We propose an I-POMDP architecture for building a dialogue tutoring system, in which two agents model the teacher (system) and the human student, addressing the two challenging tasks of disambiguating student input and selecting appropriate responses. The major advantage of the architecture is that, in a state space where the states are not fully observable, the two agents can make decisions based on their beliefs about their states and each other's preferences and rationality. This fits well the nature of tutoring between a teacher and a student.

6. References

[1] K. Forbes-Riley, D. Litman, A. Huettner, and A. Ward, "Dialogue-learning correlations in spoken dialogue tutoring". In *Proc. of the 2005 conf. on AI in Education*, 2005.

[2] M. Frampton and O. Lemon, "Recent research advances in reinforcement learning in spoken dialogue systems", *The Knowledge Engineering Review*, Vol. 24(4), pp. 375-408, 2009.

[3] P. Gmytrasiewicz and P. Doshi, "A framework for sequential planning in multi-agent settings", *Journal of Artificial Intelligence Research*, Vol. 24, pp. 49-79, 2005.

[4] F. Jurcicek, B. Thomson, S. Keizer, M. Gasic, F. Mairesse, K. Yu, and S. Young, "Natural Belief-Critic: a reinforcement algorithm for parameter estimation in statistical spoken dialogue systems" *Proc. s of Interspeech10*, pp 90-93, 2010.

[5] L. Kaelbling, M. Littman, and A. Cassandra, "Planning and acting in partially observable stochastic domains", *Artificial Intelligence*, Vol. 101, pp. 84-98, 1998.

[6] D. Litman, and S. Silliman, "Itspoke: an intelligent tutoring spoken dialogue system". In *Proc. of Human Language Technology Conf. 2004*.

[7] S. Png and J. Pineau, "Bayesian Reinforcement Learning for POMDP-based dialogue systems", *Proc. Of Conf. on Acoustics, Speech and Signal Processing*, pp. 2156-2159, 2011.

[8] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, Massachusetts: The MIT Press, 2005.

[9] J. Williams and S. Young, "Partially observable Markov decision processes for spoken dialog systems", *Elsevier Computer Speech and Language*, Vol 21, pp. 393-422, 2007.

[10] M. Woodword and R. Wood, "Learning from Humans as an I-POMDP", http://www.eecs.harvard.edu/~woodword/papers/woodword_2010_ipomdp_position_paper.pdf, 2010.