Towards Relational POMDPs for Adaptive Dialogue Management

Pierre Lison
Language Technology Lab
German Research Centre for Artificial Intelligence (DFKI GmbH)
Saarbrücken, Germany

Abstract

Open-ended spoken interactions are typically characterised by both structural complexity and high levels of uncertainty, making dialogue management in such settings a particularly challenging problem. Traditional approaches have focused on providing theoretical accounts for either the uncertainty or the complexity of spoken dialogue, but rarely considered the two issues simultaneously. This paper describes ongoing work on a new approach to dialogue management which attempts to fill this gap. We represent the interaction as a Partially Observable Markov Decision Process (POMDP) over a rich state space incorporating both dialogue, user, and environment models. The tractability of the resulting POMDP can be preserved using a mechanism for dynamically constraining the action space based on prior knowledge over locally relevant dialogue structures. These constraints are encoded in a small set of general rules expressed as a Markov Logic network. The first-order expressivity of Markov Logic enables us to leverage the rich relational structure of the problem and efficiently abstract over large regions of the state and action spaces.

1 Introduction

The development of spoken dialogue systems for rich, open-ended interactions raises a number of challenges, one of which is dialogue management. The role of dialogue management is to determine which communicative actions to take (i.e. what to say) given a goal and particular observations about the interaction and the current situation.

Dialogue managers have to face several issues. First, spoken dialogue systems must usually deal with high levels of noise and uncertainty. These uncertainties may arise from speech recognition errors, limited grammar coverage, or from various linguistic and pragmatic ambiguities.

Second, open-ended dialogue is characteristically complex, and exhibits rich relational structures. Natural interactions should be adaptive to a variety of factors dependent on the interaction history, the general context, and the user preferences. As a consequence, the state space necessary to model the dynamics of the environment tends to be large and sparsely populated.

These two problems have typically been addressed separately in the literature. On the one hand, the issue of uncertainty in speech understanding is usually dealt using a range of probabilistic models combined with decision-theoretic planning. Among these, *Partially Observable Markov Decision Process* (POMDP) models have recently emerged as a unifying mathematical framework for dialogue management (Williams and Young, 2007; Lemon and Pietquin, 2007). POMDPs provide an explicit account for a wide range of uncertainties related to partial observability (noisy, incomplete spoken inputs) and stochastic action effects (the world may evolve in unpredictable ways after executing an action).

On the other hand, structural complexity is typically addressed with logic-based approaches. Some investigated topics in this paradigm are pragmatic interpretation (Thomason et al., 2006), dialogue structure (Asher and Lascarides, 2003), or collaborative planning (Kruijff et al., 2008). These approaches are able to model sophisticated dialogue behaviours, but at the expense of robustness and adaptivity. They generally assume complete observability and provide only a very limited account (if any) of uncertainties.

We are currently developing an hybrid approach which *simultaneously* tackles the uncertainty and complexity of dialogue management, based on a

Proceedings of the ACL 2010 Student Research Workshop, pages 7–12,
Uppsala, Sweden, 13 July 2010.

POMDP framework. We present here our ongoing work on this issue. In this paper, we more specifically describe a new mechanism for dynamically constraining the space of possible actions available at a given time. Our aim is to use such mechanism to significantly reduce the search space and therefore make the planning problem globally more tractable. This is performed in two consecutive steps. We first structure the state space using *Markov Logic Networks*, a first-order probabilistic language. Prior pragmatic knowledge about dialogue structure is then exploited to derive the set of dialogue actions which are locally admissible or relevant, and prune all irrelevant ones. The first-order expressivity of Markov Logic Networks allows us to easily specify the constraints via a small set of general rules which abstract over large regions of the state and action spaces.

Our long-term goal is to develop an unified framework for adaptive dialogue management in rich, open-ended interactional settings.

This paper is structured as follows. Section 2 lays down the formal foundations of our work, by describing dialogue management as a POMDP problem. We then describe in Section 3 our approach to POMDP planning with control knowledge using Markov Logic rules. Section 4 discusses some further aspects of our approach and its relation to existing work, followed by the conclusion in Section 5.

2 Background

2.1 Partially Observable Markov Decision Processes (POMDPs)

POMDPs are a mathematical model for sequential decision-making in partially observable environments. It provides a powerful framework for control problems which combine partial observability, uncertain action effects, incomplete knowledge of the environment dynamics and multiple, potentially conflicting objectives.

Via reinforcement learning, it is possible to automatically *learn* near-optimal action policies given a POMDP model combined with real or simulated user data (Schatzmann et al., 2007).

2.1.1 Formal definition

A POMDP is a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{Z}, T, \Omega, R \rangle$, where:

- $\mathcal{S}$ is the **state space**, which is the model of the world from the agent's viewpoint. It is defined as a set of mutually exclusive states.

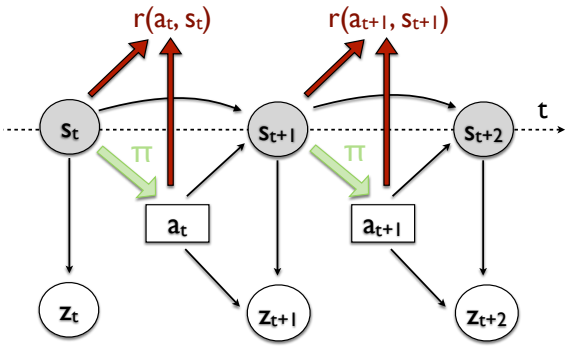

Figure 1: *Bayesian decision network* corresponding to the POMDP model. Hidden variables are greyed. Actions are represented as rectangles to stress that they are system actions rather than observed variables. Arcs into circular nodes express influence, whereas arcs into squared nodes are informational. For readability, only one state is shown at each time step, but it should be noted that the policy π is function of the full belief state rather than a single (unobservable) state.

- $\mathcal{A}$ is the **action space**: the set of possible actions at the disposal of the agent.
- $\mathcal{Z}$ is the **observation space**: the set of observations which can be captured by the agent. They correspond to features of the environment which can be directly perceived by the agent's sensors.
- T is the **transition function**, defined as $T : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$, where $T(s, a, s') = P(s'|s, a)$ is the probability of reaching state s' from state s if action a is performed.
- Ω is the **observation function**, defined as $\Omega : \mathcal{Z} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$, with $\Omega(z, a, s') = P(z|a, s')$, i.e. the probability of observing z after performing a and being now in state s'.
- R is the **reward function**, defined as $R : \mathcal{S} \times \mathcal{A} \rightarrow \Re$, $R(s, a)$ encodes the utility for the agent to perform the action a while in state s. It is therefore a model for the goals or preferences of the agent.

A graphical illustration of a POMDP model as a Bayesian decision network is provided in Fig. 1.

In addition, a POMDP can include additional parameters such as the horizon of the agent (num-

ber of look-ahead steps), and the discount factor (weighting scheme for non-immediate rewards).

2.1.2 Beliefs and belief update

A key idea of POMDP is the assumption that the state of the world is not directly accessible, and can only be inferred via observation. Such uncertainty is expressed in the **belief state** b, which is a probability distribution over possible states, that is: $b : \mathcal{S} \rightarrow [0, 1]$. The belief state for a state space of cardinality n is therefore represented in a real-valued simplex of dimension $(n-1)$.

This belief state is dynamically updated before executing each action. The belief state update operates as follows. At a given time step t, the agent is in some unobserved state $s_t = s \in \mathcal{S}$. The probability of being in state s at time t is written as $b_t(s)$. Based on the current belief state b_t, the agent selects an action a_t, receives a reward $R(s, a_t)$ and transitions to a new (unobserved) state $s_{t+1} = s'$, where s_{t+1} depends only on s_t and a_t. The agent then receives a new observation o_{t+1} which is dependent on s_{t+1} and a_t.

Finally, the belief distribution b_t is updated, based on o_{t+1} and a_t as follows[1].

$$b_{t+1}(s') = P(s'|o_{t+1}, a_t, b_t) \quad (1)$$

$$= \frac{P(o_{t+1}|s', a_t, b_t)P(s'|a_t, b_t)}{P(o_{t+1}|a_t, b_t)} \quad (2)$$

$$= \frac{P(o_{t+1}|s', a_t) \sum_{s \in \mathcal{S}} P(s'|a_t, s)P(s|a_t, b_t)}{P(o_{t+1}|a_t, b_t)} \quad (3)$$

$$= \alpha\, \Omega(o_{t+1}, s', a_t) \sum_{s \in \mathcal{S}} T(s, a_t, s')b_t(s) \quad (4)$$

where α is a normalisation constant. An initial belief state b_0 must be specified at runtime as a POMDP parameter when initialising the system.

2.1.3 POMDP policies

Given a POMDP model $\langle \mathcal{S}, \mathcal{A}, \mathcal{Z}, T, Z, R \rangle$, the agent should execute at each time-step the action which maximises its expected cumulative reward over the horizon. The function $\pi : \mathcal{B} \rightarrow \mathcal{A}$ defines a *policy*, which determines the action to perform for each point of the belief space.

The expected reward for policy π starting from belief b is defined as:

$$J^{\pi}(b) = E\Big[\sum_{t=0}^{h} \gamma^t R(s_t, a_t) \mid b, \pi\Big] \quad (5)$$

[1] As a notational shorthand, we write $P(s_t{=}s)$ as $P(s)$ and $P(s_{t+1}{=}s')$ as $P(s')$.

The optimal policy π^* is then obtained by optimizing the long-term reward, starting from b_0:

$$\pi^* = \underset{\pi}{\text{argmax}}\ J^{\pi}(b_0) \quad (6)$$

The optimal policy π^* yields the highest expected reward value for each possible belief state. This value is compactly represented by the optimal value function, noted V^*, which is a solution to the Bellman optimality equation (Bellman, 1957).

Numerous algorithms for (offline) policy optimisation and (online) planning are available. For large spaces, exact optimisation is impossible and approximate methods must be used, see for instance grid-based (Thomson and Young, 2009) or point-based (Pineau et al., 2006) techniques.

2.2 POMDP-based dialogue management

Dialogue management can be easily cast as a POMDP problem, with the *state space* being a compact representation of the interaction, the *action space* being a set of dialogue moves, the *observation space* representing speech recognition hypotheses, the *transition function* defining the dynamics of the interaction (which user reaction is to be expected after a particular dialogue move), and the *observation function* describing a "sensor model" between observed speech recognition hypotheses and actual utterances. Finally, the *reward function* encodes the utility of dialogue policies – it typically assigns a big positive reward if a long-term goal has been reached (e.g. the retrieval of some important information), and small negative rewards for minor "inconveniences" (e.g. prompting the user to repeat or asking for confirmations).

Our long-term aim is to apply such POMDP framework to a rich dialogue domain for human-robot interaction (Kruijff et al., 2010). These interactions are typically open-ended, relatively long, include high levels of noise, and require complex state and action spaces. Furthemore, the dialogue system also needs to be *adaptive* to its user (attributed beliefs and intentions, attitude, attentional state) and to the current situation (currently perceived entities and events).

As a consequence, the state space must be expanded to include these knowledge sources. Belief monitoring is then used to continuously update the belief state based on perceptual inputs (see also (Bohus and Horvitz, 2009) for an overview of techniques to extract such information). These requirements can only be fullfilled if we address the

"curse of dimensionality" characteristic of traditional POMDP models. The next section provides a tentative answer.

3 Approach

3.1 Control knowledge

Classical approaches to POMDP planning operate directly on the full action space and select the next action to perform based on the maximisation of the expected cumulative reward over the specified horizon. Such approaches can be used in small-scale domains with a limited action space, but quickly become intractable for larger ones, as the planning time increases exponentially with the size of the action space. Significant planning time is therefore spend on actions which should be directly discarded as irrelevant[2]. Dismissing these actions *before* planning could therefore provide important computational gains.

Instead of a direct policy optimisation over the full action space, our approach formalises action selection as a *two-step* process. As a first step, a set of *relevant dialogue moves* is constructed from the full action space. The POMDP planner then computes the optimal (highest-reward) action on this reduced action space in a second step.

Such an approach is able to significantly reduce the dimensionality of the dialogue management problem by taking advantage of prior knowledge about the expected relational structure of spoken dialogue. This prior knowledge is to be encoded in a set of general rules describing the admissible dialogue moves in a particular situation.

How can we express such rules? POMDPs are usually modeled with Bayesian networks which are inherently propositional. Encoding such rules in a propositional framework requires a distinct rule for every possible state and action instance. This is not a feasible approach. We therefore need a first order (probabilistic) language able to express generalities over large regions of the state action spaces. Markov Logic is such a language.

3.2 Markov Logic Networks (MLNs)

Markov Logic combines first-order logic and probabilistic graphical models in a unified representation (Richardson and Domingos, 2006). A Markov Logic Network L is a set of pairs (F_i, w_i), where F_i is a formula in first-order logic and w_i is a real number representing the formula weight.

A Markov Logic Network L can be seen as a *template* for constructing markov networks[3]. To construct a markov network from L, one has to provide an additional set of constants $C = \{c_1, c_2, ..., c_{|C|}\}$. The resulting markov network is called a *ground markov network* and is written $M_{L,C}$. The ground markov network contains one feature for each possible grounding of a first-order formula in L, with the corresponding weight. The technical details of the construction of $M_{L,C}$ from the two sets L and C is explained in several papers, see e.g. (Richardson and Domingos, 2006).

Once the markov network $M_{L,C}$ is constructed, it can be exploited to perform *inference* over arbitrary queries. Efficient probabilistic inference algorithms such as Markov Chain Monte Carlo (MCMC) or other sampling techniques can then be used to this end (Poon and Domingos, 2006).

3.3 States and actions as relational structures

The specification of Markov Logic rules applying over complete regions of the state and action spaces (instead of over single instances) requires an explicit relational structure over these spaces.

This is realised by factoring the state and action spaces into a set of distinct, conditionally independent features. A state s can be expanded into a tuple $\langle f_1, f_2, ...f_n \rangle$, where each sub-state f_i is assigned a value from a set $\{v_1, v_2, ...v_m\}$. Such structure can be expressed in first-order logic with a binary predicate $f_i(s, v_j)$ for each sub-state f_i, where v_j is the value of the sub-state f_i in s. The same type of structure can be defined over actions. This factoring leads to a relational structure of arbitrary complexity, compactly represented by a set of unary and binary predicates.

For instance, (Young et al., 2010) factors each dialogue state into three independent parts $s = \langle s_u, a_u, s_d \rangle$, where s_u is the user goal, a_u the last user move, and s_d the dialogue history. These can be expressed in Markov Logic with predicates such as $\texttt{UserGoal}(\texttt{s}, \texttt{s}_\texttt{u})$, $\texttt{LastUserMove}(\texttt{s}, \texttt{a}_\texttt{u})$, or $\texttt{History}(\texttt{s}, \texttt{s}_\texttt{d})$.

[2]For instance, an agent hearing a user command such as "Please take the mug on your left" might spent a lot of planning time calculating the expected future reward of dialogue moves such as "Is the box green?" or "Your name is John", which are irrelevant to the situation.

[3]Markov networks are undirected graphical models.

3.4 Relevant action space

For a given state s, the relevant action space $RelMoves(\mathcal{A}, s)$ is defined as:

$$\{a_m : a_m \in \mathcal{A} \wedge \texttt{RelevantMove}(\texttt{a}_\texttt{m}, \texttt{s})\} \quad (7)$$

The truth-value of the predicate $\texttt{RelevantMove}(\texttt{a}_\texttt{m}, \texttt{s})$ is determined using a set of Markov Logic rules dependent on both the state s and the action a_m. For a given state s, the relevant action space is constructed via probabilistic inference, by estimating the probability $P(\texttt{RelevantMove}(\texttt{a}_\texttt{m}, \texttt{s}))$ for each action a_m, and selecting the subset of actions for which the probability is above a given threshold.

Eq. 8 provides a simple example of such Markov Logic rule:

$$\texttt{LastUserMove}(\texttt{s}, \texttt{a}_\texttt{u}) \wedge \texttt{PolarQuestion}(\texttt{a}_\texttt{u}) \wedge \texttt{YesNoAnswer}(\texttt{a}_\texttt{m}) \rightarrow \texttt{RelevantMove}(\texttt{a}_\texttt{m}, \texttt{s}) \quad (8)$$

It defines an admissible dialogue move for a situation where the user asks a polar question to the agent (e.g. "do you see my hand?"). The rule specifies that, if a state s contains a_u as last user move, and if a_u is a polar question, then an answer a_m of type yes-no is a relevant dialogue move for the agent. This rule is (implicitly) universally quantified over s, a_u and a_m.

Each of these Markov Logic rules has a weight attached to it, expressing the strength of the implication. A rule with infinite weight and satisfied premises will lead to a relevant move with probability 1. Softer weights can be used to describe moves which are less relevant but still possible in a particular context. These weights can either be encoded by hand or learned from data (how to perform this efficiently remains an open question).

3.5 Rules application on POMDP belief state

The previous section assumed that the state s is known. But the real state of a POMDP is never directly accessible. The rules we just described must therefore be applied on the belief state. Ultimately, we want to define a function $Rel : \Re^n \rightarrow \mathcal{P}(\mathcal{A})$, which takes as input a point in the belief space and outputs a set of relevant moves. For efficiency reasons, this function can be precomputed offline, by segmenting the state space into distinct regions and assigning a set of relevant moves to each region. The function can then be directly called at runtime by the planning algorithm.

Due to the high dimensionality of the belief space, the above function must be approximated to remain tractable. One way to perform this approximation is to extract, for belief state b, a set S_m of m most likely states, and compute the set of relevant moves for each of them. We then define the global probability estimate of a being a relevant move given b as such:

$$P(\texttt{RelevantMove}(\texttt{a}) \mid b, a) \approx \sum_{s \in S_m} P(\texttt{RelevantMove}(\texttt{a}, \texttt{s}) \mid s, a) \times b(s) \quad (9)$$

In the limit where $m \rightarrow |S|$, the error margin on the approximation tends to zero.

4 Discussion

4.1 General comments

It is worth noting that the mechanism we just outlined does not intend to *replace* the existing POMDP planning and optimisation algorithms, but rather *complements* them. Each step serves a different purpose: the action space reduction provides an answer to the question "Is this action relevant?", while the policy optimisation seeks to answer "Is this action useful?". We believe that such distinction between relevance and usefulness is important and will prove to be beneficial in terms of tractability.

It is also useful to notice that the Markov Logic rules we described provides a "positive" definition of the action space. The rules were applied to produce an exhaustive list of all admissible actions given a state, all actions outside this list being *de facto* labelled as non-admissible. But the rules can also provide a "negative" definition of the action space. That is, instead of generating an exhaustive list of possible actions, the dialogue system can initially consider all actions as admissible, and the rules can then be used to prune this action space by removing irrelevant moves.

The choice of action filter depends mainly on the size of the dialogue domain and the availability of prior domain knowledge. A "positive" filter is a necessity for large dialogue domains, as the action space is likely to grow exponentially with the domain size and become untractable. But the positive definition of the action space is also significantly more expensive for the dialogue developer. There is therefore a trade-off between the costs of tractability issues, and the costs of dialogue domain modelling.

4.2 Related Work

There is a substantial body of existing work in the POMDP literature about the exploitation of the problem structure to tackle the curse of dimensionality (Poupart, 2005; Young et al., 2010), but the vast majority of these approaches retain a propositional structure. A few more theoretical papers also describe first-order MDPs (Wang et al., 2007), and recent work on Markov Logic has extended the MLN formalism to include some decision-theoretic concepts (Nath and Domingos, 2009). To the author's knowledge, none of these ideas have been applied to dialogue management.

5 Conclusions

This paper described a new approach to exploit relational models of dialogue structure for controlling the action space in POMDPs. This approach is part of an ongoing work to develop a unified framework for adaptive dialogue management in rich, open-ended interactional settings. The dialogue manager is being implemented as part of a larger cognitive architecture for talking robots.

Besides the implementation, future work will focus on refining the theoretical foundations of relational POMDPs for dialogue (including how to specify the transition, observation and reward functions in such a relational framework), as well as investigating the use of reinforcement learning for policy optimisation based on simulated data.

References

N. Asher and A. Lascarides. 2003. *Logics of Conversation*. Cambridge University Press.

R. Bellman. 1957. *Dynamic Programming*. Princeton University Press.

Dan Bohus and Eric Horvitz. 2009. Dialog in the open world: platform and applications. In *ICMI-MLMI '09: Proceedings of the 2009 international conference on Multimodal interfaces*, pages 31–38, New York, NY, USA. ACM.

G.J.M. Kruijff, M. Brenner, and N.A. Hawes. 2008. Continual planning for cross-modal situated clarification in human-robot interaction. In *Proceedings of the 17th International Symposium on Robot and Human Interactive Communication (RO-MAN 2008)*, Munich, Germany.

G.-J. M. Kruijff, P. Lison, T. Benjamin, H. Jacobsson, H. Zender, and I. Kruijff-Korbayova. 2010. Situated dialogue processing for human-robot interaction. In H. I. Christensen, A. Sloman, G.-J. M. Kruijff, and J. Wyatt, editors, *Cognitive Systems*. Springer Verlag. (in press).

O. Lemon and O. Pietquin. 2007. Machine learning for spoken dialogue systems. In *Proceedings of the European Conference on Speech Communication and Technologies (Interspeech'07)*, pages 2685–2688, Anvers (Belgium), August.

A. Nath and P. Domingos. 2009. A language for relational decision theory. In *Proceedings of the International Workshop on Statistical Relational Learning*.

J. Pineau, G. Gordon, and S. Thrun. 2006. Anytime point-based approximations for large pomdps. *Artificial Intelligence Research*, 27(1):335–380.

H. Poon and P. Domingos. 2006. Sound and efficient inference with probabilistic and deterministic dependencies. In *AAAI'06: Proceedings of the 21st national conference on Artificial intelligence*, pages 458–463. AAAI Press.

P. Poupart. 2005. *Exploiting structure to efficiently solve large scale partially observable markov decision processes*. Ph.D. thesis, University of Toronto, Toronto, Canada.

M. Richardson and P. Domingos. 2006. Markov logic networks. *Machine Learning*, 62(1-2):107–136.

Jost Schatzmann, Blaise Thomson, Karl Weilhammer, Hui Ye, and Steve Young. 2007. Agenda-based user simulation for bootstrapping a POMDP dialogue system. In *HLT '07: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies*, pages 149–152, Rochester, New York, April. Association for Computational Linguistics.

R. Thomason, M. Stone, and D. DeVault. 2006. Enlightened update: A computational architecture for presupposition and other pragmatic phenomena. In Donna Byron, Craige Roberts, and Scott Schwenter, editors, *Presupposition Accommodation*. Ohio State Pragmatics Initiative.

B. Thomson and S. Young. 2009. Bayesian update of dialogue state: A pomdp framework for spoken dialogue systems. *Computer Speech & Language*, August.

Ch. Wang, S. Joshi, and R. Khardon. 2007. First order decision diagrams for relational mdps. In *IJCAI'07: Proceedings of the 20th international joint conference on Artifical intelligence*, pages 1095–1100, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

J. Williams and S. Young. 2007. Partially observable markov decision processes for spoken dialog systems. *Computer Speech and Language*, 21(2):231–422.

S. Young, M. Gašić, S. Keizer, F. Mairesse, J. Schatzmann, B. Thomson, and K. Yu. 2010. The hidden information state model: A practical framework for pomdp-based spoken dialogue management. *Computer Speech & Language*, 24(2):150–174.