Mixture Model POMDPs for Efficient Handling of Uncertainty in Dialogue Management

James Henderson
University of Geneva
Department of Computer Science
James.Henderson@cui.unige.ch

Oliver Lemon
University of Edinburgh
School of Informatics
olemon@inf.ed.ac.uk

Abstract

In spoken dialogue systems, Partially Observable Markov Decision Processes (POMDPs) provide a formal framework for making dialogue management decisions under uncertainty, but efficiency and interpretability considerations mean that most current statistical dialogue managers are only MDPs. These MDP systems encode uncertainty explicitly in a single state representation. We formalise such MDP states in terms of distributions over POMDP states, and propose a new dialogue system architecture (Mixture Model POMDPs) which uses mixtures of these distributions to efficiently represent uncertainty. We also provide initial evaluation results (with real users) for this architecture.

1 Introduction

Partially Observable Markov Decision Processes (POMDPs) provide a formal framework for making decisions under uncertainty. Recent research in spoken dialogue systems has used POMDPs for dialogue management (Williams and Young, 2007; Young et al., 2007). These systems represent the uncertainty about the dialogue history using a probability distribution over dialogue states, known as the POMDP's belief state, and they use approximate POMDP inference procedures to make dialogue management decisions. However, these inference procedures are too computationally intensive for most domains, and the system's behaviour can be difficult to predict. Instead, most current statistical dialogue managers use a single state to represent the dialogue history, thereby making them only Markov Decision Process models (MDPs). These state representations have been fine-tuned over many development cycles so that common types of uncertainty can be encoded in a single state. Examples of such representations include unspecified values, confidence scores, and confirmed/unconfirmed features. We formalise such MDP systems as compact encodings of POMDPs, where each MDP state represents a probability distribution over POMDP states. We call these distributions "MDP belief states".

Given this understanding of MDP dialogue managers, we propose a new POMDP spoken dialogue system architecture which uses mixtures of MDP belief states to encode uncertainty. A Mixture Model POMDP represents its belief state as a probability distribution over a finite set of MDP states. This extends the compact representations of uncertainty in MDP states to include arbitrary disjunction between MDP states. Efficiency is maintained because such arbitrary disjunction is not needed to encode the most common forms of uncertainty, and thus the number of MDP states in the set can be kept small without losing accuracy. On the other hand, allowing multiple MDP states provides the representational mechanism necessary to incorporate multiple speech recognition hypotheses into the belief state representation. In spoken dialogue systems, speech recognition is by far the most important source of uncertainty. By providing a mechanism to incorporate multiple arbitrary speech recognition hypotheses, the proposed architecture leverages the main advantage of POMDP systems while still maintaining the efficiency of MDP-based dialogue managers.

2 Mixture Model POMDPs

A POMDP belief state b_t is a probability distribution $P(s_t|V_{t-1}, u_t)$ over POMDP states s_t given the dialogue history V_{t-1} and the most recent observation (i.e. user utterance) u_t. We formalise the meaning of an MDP state representation r_t as a distribution $b(r_t) = P(s_t|r_t)$ over POMDP states. We represent the belief state b_t as a list of pairs $\langle r_t^i, p_t^i \rangle$ such that $\sum_i p_t^i = 1$. This list is interpreted as a mixture of the $b(r_t^i)$.

$$b_t = \sum_i p_t^i b(r_t^i) \tag{1}$$

State transitions in MDPs are specified with an update function, $r_t = f(r_{t-1}, a_{t-1}, h_t)$, which maps the preceding state r_{t-1}, system action a_{t-1}, and user input h_t to a new state r_t. This function is intended to encode in r_t all the new information provided by a_{t-1} and h_t. The user input h_t is the result of automatic speech recognition (ASR) plus spoken language understanding (SLU) applied to u_t. Because there is no method for handling ambiguity in h_t, h_t is computed from the single best ASR-SLU hypothesis, plus some measure of ASR confidence.

In POMDPs, belief state transitions are done by changing the distribution over states to take into account the new information from the system action a_{t-1} and an n-best list of ASR-SLU hypotheses h_t^j. This new belief state can be estimated as

$$\begin{aligned}
b_t &= P(s_t|V_{t-1}, u_t) \\
&= \sum_{h_t^j} \sum_{s_{t-1}} \frac{\begin{array}{c} P(s_{t-1}|V_{t-1}) P(h_t^j|V_{t-1}, s_{t-1}) \\ P(u_t|V_{t-1}, s_{t-1}, h_t^j) \\ P(s_t|V_{t-1}, s_{t-1}, h_t^j, u_t) \end{array}}{P(u_t|V_{t-1})} \\
&\approx \sum_{h_t^j} \sum_{s_{t-1}} \frac{\begin{array}{c} P(s_{t-1}|V_{t-2}, u_{t-1}) P(h_t^j|a_{t-1}, s_{t-1}) \\ P(h_t^j|u_t) P(s_t|a_{t-1}, s_{t-1}, h_t^j) \end{array}}{P(h_t^j) Z(V_t)}
\end{aligned}$$

where $Z(V_t)$ is a normalising constant. $P(s_{t-1}|V_{t-2}, u_{t-1})$ is the previous belief state. $P(h_t^j|u_t)$ reflects the confidence of ASR-SLU in hypothesis h_t^j. $P(s_t|a_{t-1}, s_{t-1}, h_t^j)$ is normally 1 for $s_t = s_{t-1}$, but can be used to allow users to change their mind mid-dialogue. $P(h_t^j|a_{t-1}, s_{t-1})$ is a user model. $P(h_t^j)$ is a prior over ASR-SLU outputs.

Putting these two approaches together, we get the following update equation for our mixture of MDP belief states:

$$\begin{aligned}
b_t &= P(s_t|V_{t-1}, u_t) \\
&\approx \sum_{h_t^j} \sum_{r_{t-1}^i} \frac{\begin{array}{c} p_{t-1}^i P(h_t^j|a_{t-1}, r_{t-1}^i) \\ P(h_t^j|u_t) b(f(r_{t-1}^i, a_{t-1}, h_t^j)) \end{array}}{P(h_t^j) Z(V_t)} \\
&= \sum_{i'} p_t^{i'} b(r_t^{i'})
\end{aligned} \tag{2}$$

where, for each i' there is one pair i, j such that

$$\begin{aligned}
r_t^{i'} &= f(r_{t-1}^i, a_{t-1}, h_t^j) \\
p_t^{i'} &= \frac{p_{t-1}^i P(h_t^j|a_{t-1}, r_{t-1}^i) P(h_t^j|u_t)}{P(h_t^j) Z(V_t)}.
\end{aligned} \tag{3}$$

For equation (2) to be true, we require that

$$b(f(r_{t-1}^i, a_{t-1}, h_t^j)) \approx P(s_t|a_{t-1}, r_{t-1}^i, h_t^j) \tag{4}$$

which simply ensures that the meaning assigned to MDP state representations and the MDP state transition function are compatible.

From equation (3), we see that the number of MDP states will grow exponentially with the length of the dialogue, proportionately to the number of ASR-SLU hypotheses. Some of the state-hypothesis pairs r_{t-1}^i, h_t^j may lead to equivalent states $f(r_{t-1}^i, a_{t-1}, h_t^j)$, but in general pruning is necessary. Pruning should be done so as to minimise the change to the belief state distribution, for example by minimising the KL divergence between the pre- and post- pruning belief states. We use two heuristic approximations to this optimisation problem. First, if two states share the same core features (e.g. filled slots, but not the history of user inputs), then the state with the lower probability is pruned, and its probability is added to the other state. Second, a fixed beam of the k most probable states is kept, and the other states are pruned. The probability p_t^i from a pruned state r_t^i is redistributed to unpruned states which are less informative than r_t^i in their core features.[1]

The interface between the ASR-SLU module and the dialogue manager is a set of hypotheses h_t^j paired with their confidence scores $P(h_t^j|u_t)$. These pairs are analogous to the state-probability pairs r_t^i, p_t^i within the dialogue manager, and we can extend our mixture model architecture to cover these pairs as well. Interpreting the set of $h_t^j, P(h_t^j|u_t)$ pairs as a

[1] In the current implementation, these pruned state probabilities are simply added to an uninformative "null" state, but in general we could check for logical subsumption between states.

mixture of distributions over more specific hypotheses becomes important when we consider pruning this set before passing it to the dialogue manager. As with the pruning of states, pruning should not simply remove a hypothesis and renormalise, it should redistribute the probability of a pruned hypothesis to similar hypotheses. This is not always computationally feasible, but all interfaces within the Mixture Model POMDP architecture are sets of hypothesis-probability pairs which can be interpreted as finite mixtures in some underlying hypothesis space.

Given an MDP state representation, this formalisation allows us to convert it into a Mixture Model POMDP. The only additional components of the model are the user model $P(h_t^j|a_{t-1}, r_{t-1}^i)$, the ASR-SLU prior $P(h_t^j)$, and the ASR-SLU confidence score $P(h_t^j|u_t)$. Note that there is no need to actually define $b(r_t^i)$, provided equation (4) holds.

3 Decision Making with MM POMDPs

Given this representation of the uncertainty in the current dialogue state, the spoken dialogue system needs to decide what system action to perform. There are several approaches to POMDP decision making which could be adapted to this representation, but to date we have only considered a method which allows us to directly derive a POMDP policy from the policy of the original MDP.

Here again we exploit the fact that the most frequent forms of uncertainty are already effectively handled in the MDP system (e.g. by filled vs. confirmed slot values). We propose that an effective dialogue management policy can be created by simply computing a mixture of the MDP policy applied to the MDP states in the belief state list. More precisely, we assume that the original MDP system specifies a Q function $Q_{\text{MDP}}(a_t, r_t)$ which estimates the expected future reward of performing action a_t in state r_t. We then estimate the expected future reward of performing action a_t in belief state b_t as the mixture of these MDP estimates.

$$Q(a_t, b_t) \approx \sum_i p_t^i Q_{\text{MDP}}(a_t, r_t^i) \qquad (5)$$

The dialogue management policy is to choose the action a_t with the largest value for $Q(a_t, b_t)$. This is known as a Q-MDP model (Littman et al., 1995), so we call this proposal a Mixture Model Q-MDP.

4 Related Work

Our representation of POMDP belief states using a set of distributions over POMDP states is similar to the approach in (Young et al., 2007), where POMDP belief states are represented using a set of partitions of POMDP states. For any set of partitions, the mixture model approach could express the same model by defining one MDP state per partition and giving it a uniform distribution inside its partition and zero probability outside. However, the mixture model approach is more flexible, because the distributions in the mixture do not have to be uniform within their non-zero region, and these regions do not have to be disjoint. A list of states was also used in (Higashinaka et al., 2003) to represent uncertainty, but no formal semantics was provided for this list, and therefore only heuristic uses were suggested for it.

5 Initial Experiments

We have implemented a Mixture Model POMDP architecture as a multi-state version of the DIPPER "Information State Update" dialogue manager (Bos et al., 2003). It uses equation (3) to compute belief state updates, given separate models for MDP state updates (for $f(r_{t-1}^i, a_{t-1}, h_t^j)$), statistical ASR-SLU (for $P(h_t^j|u_t)/P(h_t^j)$), and a statistical user model (for $P(h_t^j|a_{t-1}, r_{t-1}^i)$). The state list is pruned as described in section 2, where the "core features" are the filled information slot values and whether they have been confirmed. For example, the system will merge two states which agree that the user only wants a cheap hotel, even if they disagree on the sequence of dialogue acts which lead to this information. It also never prunes the "null" state, so that there is always some probability that the system knows nothing.

The system used in the experiments described below uses the MDP state representation and update function from (Lemon and Liu, 2007), which is designed for standard slot-filling dialogues. For the ASR model, it uses the HTK speech recogniser (Young et al., 2002) and an n-best list of three ASR hypotheses on each user turn. The prior over user inputs is assumed to be uniform. The ASR hypotheses are passed to the SLU model from (Meza-Ruiz et al., 2008), which produces a single user input for each ASR hypothesis. This SLU model was trained on

	TC %	Av. length (std. deviation)
Handcoded	56.0	7.2 (4.6)
MDP	66.6	7.2 (4.0)
MM Q-MDP	73.3	7.3 (3.7)

Table 1: Initial test results for human-machine dialogues, showing task completion and average length.

the TownInfo corpus of dialogues, which was collected using the TownInfo human-machine dialogue systems of (Lemon et al., 2006), transcribed, and hand annotated. ASR hypotheses which result in the same user input are merged (summing their probabilities), and the resulting list of at most three ASR-SLU hypotheses are passed to the dialogue manager. Thus the number of MDP states in the dialogue manager grows by up to three times at each step, before pruning. For the user model, the system uses an n-gram user model, as described in (Georgila et al., 2005), trained on the annotated TownInfo corpus.[2]

The system's dialogue management policy is a Mixture Model Q-MDP (MM Q-MDP) policy. As with the MDP states, the MDP Q function is from (Lemon and Liu, 2007). It was trained in an MDP system using reinforcement learning with simulated users (Lemon and Liu, 2007), and was not modified for use in our MM Q-MDP policy.

We tested this system with 10 different users, each attempting 9 tasks in the TownInfo domain (searching for hotels and restaurants in a fictitious town), resulting in 90 test dialogues. The users each attempted 3 tasks with the MDP system of (Lemon and Liu, 2007), 3 tasks with a state-of-the-art hand-coded system (see (Lemon et al., 2006)), and 3 tasks with the MM Q-MDP system. Ordering of systems and tasks was controlled, and 3 of the users were not native speakers of English. We collected the Task Completion (TC), and dialogue length for each system, as reported in table 1. Task Completion is counted from the system logs when the user replies that they are happy with their chosen option. Such a small sample size means that these results are not statistically significant, but there is a clear trend showing the superiority of the the MM Q-MDP system, both in terms of more tasks being completed and less variability in overall dialogue length.

[2] Thanks to K. Georgilla for training this model.

6 Conclusions

Mixture Model POMDPs combine the efficiency of MDP spoken dialogue systems with the ability of POMDP models to make use of multiple ASR hypotheses. They can also be constructed from MDP models without additional training, using the Q-MDP approximation for the dialogue management policy. Initial results suggest that, despite its simplicity, this approach does lead to better spoken dialogue systems than MDP and hand-coded models.

Acknowledgments

This research received funding from UK EPSRC grant EP/E019501/1 and the European Community's FP7 under grant n^{o} 216594 (CLASSIC project: www.classic-project.org).

References

J Bos, E Klein, O Lemon, and T Oka. 2003. DIPPER: Description and Formalisation of an Information-State Update Dialogue System Architecture. In *Proc. SIGdial Workshop on Discourse and Dialogue*, Sapporo.

K Georgila, J Henderson, and O Lemon. 2005. Learning User Simulations for Information State Update Dialogue Systems. In *Proc. Eurospeech*.

H Higashinaka, M Nakano, and K Aikawa. 2003. Corpus-based discourse understanding in spoken dialogue systems. In *Proc. ACL*, Sapporo.

O Lemon and X Liu. 2007. Dialogue policy learning for combinations of noise and user simulation: transfer results. In *Proc. SIGdial*.

O Lemon, K Georgila, and J Henderson. 2006. Evaluating Effectiveness and Portability of Reinforcement Learned Dialogue Strategies with real users: the TALK TownInfo Evaluation. In *Proc. ACL/IEEE SLT*.

ML Littman, AR Cassandra, and LP Kaelbling. 1995. Learning policies for partially observable environments: Scaling up. In *Proc. ICML*, pages 362–370.

I Meza-Ruiz, S Riedel, and O Lemon. 2008. Accurate statistical spoken language understanding from limited development resources. In *Proc. ICASSP*. (to appear).

JD Williams and SJ Young. 2007. Partially Observable Markov Decision Processes for Spoken Dialog Systems. *Computer Speech and Language*, 21(2):231–422.

S Young, G Evermann, D Kershaw, G Moore, J Odell, D Ollason, D Povey, V Valtchev, and P Woodland. 2002. *The HTK Book*. Cambridge Univ. Eng. Dept.

SJ Young, J Schatzmann, K Weilhammer, and H Ye. 2007. The Hidden Information State Approach to Dialog Management. In *Proc. ICASSP*, Honolulu.