Effects of User Modeling on POMDP-based Dialogue Systems

Dongho Kim[1]*, Hyeong Seop Sim*[1]*, Kee-Eung Kim*[1]*, Jin Hyung Kim*[1]
Hyunjeong Kim[2]*, Joo Won Sung*[2]

[1] Department of Computer Science, Korea Advanced Institute of Science and Technology, Korea
[2] HCI Research Department, KT, Korea

Abstract

Partially observable Markov decision processes (POMDPs) have gained significant interest in research on spoken dialogue systems, due to among many benefits its ability to naturally model the dialogue strategy selection problem under the unreliability in automated speech recognition. However, the POMDP approaches are essentially model-based, and as a result, the dialogue strategy computed from POMDP is subject to the correctness of the model. In this paper, we extend some of the previous user models for POMDPs, and evaluate the effects of user models on the dialogue strategy computed from POMDP.

Index Terms: spoken dialogue systems, user modeling, POMDP

1. Introduction

Over the recent years, partially observable Markov decision processes (POMDPs) have been proposed as an attractive framework for modeling spoken dialogue systems [1, 2]. POMDPs are a natural model for sequential decision making problems under partial or uncertain observations, and thus, it is well suited for computing the optimal dialogue strategy under unreliable automatic speech recognition or natural language processing. Although the task of solving POMDPs is known to be intractable, recent advances in approximate algorithms such as point-based value iteration (PBVI) [3], heuristic search value iteration (HSVI) [4], and composite-summary PBVI [5] show great promise for solving real-world scale POMDPs.

However, these POMDP approaches to spoken dialogue systems are essentially model-based: casting the dialogue system as a POMDP requires the model of the user response and the speech recognition error. Hence if the model is poor, the dialogue strategy computed from the corresponding POMDP can be useless. The effect of model quality on the dialogue strategies has been extensively studied in [6], but only in the context of the less expressive Markov decision processes (MDPs).

This paper is about evaluating the effect of user model on POMDP dialogue strategies. Our experiments extend upon the results in [6], comparing the generalization of MDP and POMDP dialogue strategies across different user models. The user models used in our work are the extensions of standard user models to POMDPs, incorporating speech recognition error rates. We build the user models from the DARPA Communicator corpus 2000, and use a symbolic version of HSVI for solving POMDPs [7].

To the best of our knowledge, our work is the first to report the results of POMDP dialogue strategies for a real-world scale corpus such as the DARPA Communicator.

2. MDPs and POMDPs for dialogues

An MDP is defined as $\langle S, A, T, R\rangle$: S is the set of states; A is the set of actions; T is the transition function where $T(s, a, s')$ denotes the probability $P(s'|s, a)$ of changing to state s' from state s by executing action a; R is the reward function where $R(s, a)$ denotes the immediate reward of executing action a in state s. MDPs for human-computer dialogues typically model the dialogue states as S and the system actions as A. The dialogue state keeps track of dialogue progress, incorporating the user response at each step of the iteration. One of the fundamental limitations of MDP modeling for dialogues is that MDP assumes the complete observability: it assumes no error in the automated speech recognition or natural language processing.

POMDPs [8] make the model more expressive by assuming partial or uncertain observations. A POMDP is defined as $\langle S, A, Z, T, O, R, b_0\rangle$: S, A, T, and R are the states, actions, transition function, and reward function as in MDPs. The states are hidden in the sense that the decision making has to depend on observations from the set Z. O is the observation function where $O(s, a, z)$ denotes the probability $P(z|s, a)$ of making observation z when executing action a and arriving in state s. b_0 is the initial belief where $b_0(s)$ is the probability that we start at state s.

A standard approach for casting a spoken dialogue system as a POMDP is to use a factored representation of the state space [2]: state s is factored into three components $\langle s_u, a_u, s_d\rangle$ where s_u is the user goal, a_u is the current user response, and s_d is the dialogue progress. Note that a_u represents the true response, which generates noisy recognition result $z = \tilde{a}_u$. The state space of MDP can also be factored into a number of variables, each representing specific aspects of the dialogue progress. There are a number of algorithms for factored MDPs and POMDPs. In our work, we used SPUDD [9] for factored MDPs and Symbolic HSVI [7] for factored POMDPs. Both algorithms use algebraic decision diagrams for compact representations of intermediate computation results to effectively deal with factored state spaces.

3. User models for MDPs and POMDPs

In this section, we review some of the standard probabilistic approaches for modeling user behavior. These user models are typically used for generating user responses for learning dialogue strategies.

One of the earliest user models is the Bigram model [10], which is a simple stochastic model for predicting the user response to the given system action. The Bigram model is specified as the probability $p(a_u|a)$ for every possible pair of system action a and user response a_u. The Bigram model has the advantage of being purely probabilistic and domain-independent,

although it often fails to accurately model the slot-filling dialogues.

The Levin model [11] is a modification to the pure Bigram model, which reduces the number of model parameters by limiting to admittable user responses. For instance, the ATIS corpus has three types of system actions: *greeting*, *constraining question* and *relaxation prompt*. The constraining questions are the set of actions each requesting a value for a particular slot from the user and the relaxation prompts are the actions requesting the user to relax a particular constraint that was specified earlier. The user response for greeting is parameterized by $P(n)$, $n = 0, 1, 2, \ldots$, the probability of providing n slots in the same response, and $P(k)$, the probability distribution on each slot k. The user response of the constraining questions is similarly parameterized by $P(n|k)$ and $P(k'|k)$, the probability of the user specifying a value for slot k' when asked for the value of slot k, and n is the number of additional unsolicited slots in the same response. The user is only allowed to either accept or reject the proposed relaxation of slot k, hence the user response is parameterized by $P(\text{yes}|k) = 1 - P(\text{no}|k)$.

The Bigram and the Levin model both suffer from a lack of goal consistency in user behavior. To overcome this problem, the Pietquin model [12] extends the Levin model by conditioning the probabilities in the Levin model on the user goal s_u, *i.e.*, $P(a_u|s_u, a)$.

All of the above user models should be used for generating the true user response. Hence, if we model the dialogue problem as an MDP, these models are used in the transition probabilities. Note that the reinforcement learning approach in [6] is equivalent to solving MDPs, where the next state is sampled using the user model rather than calculating the exact transition probability. In the context of POMDPs, we additionally need the observation probabilities representing the unreliability of the automated speech recognition results. These probabilities can be easily obtained from an annotated corpus. Our detailed methodologies for obtaining the models are described in section 4.3

4. Experimental setup

4.1. Dataset

The DARPA Communicator corpus used in our experiments contains 648 real human-computer dialogues recorded using different dialogue managers. We selected 100 dialogues among all dialogues for manually tagging the true user response with semantic information such as the type of the user response and its corresponding slot. We also constrained ourselves to the task of completing the first leg flight reservations. As a result, the dialogue manager has to fill out four slots: *orig_city*, *dest_city*, *depart_date* and *depart_time*, which means the starting location, the destination city, the departure date, and the departure time, respectively. The slot values are ignored, hence the dialogue happens at the intention level.

4.2. State space, action set and reward function

The factored state space $\langle s_u, a_u, s_d \rangle$ of the dialogue management POMDP is specified as follows: The dialogue process s_d represents that a particular slot is *unknown*, *known*, or *confirmed*, resulting in a total number of $3^4 = 81$ possible combinations. Similarly, the current user response a_u is determined by the user response for each slot: $a_u = \{a_{u,\text{orig\_city}}, a_{u,\text{dest\_city}}, a_{u,\text{depart\_date}}, a_{u,\text{depart\_time}}\}$. We have eight types of user responses for each slot: *provide_info*, *reprovide_info*, *correct_info*, *reject_info*, *yes_answer*, *no_answer*, *question*, and *null* (no mention of the slot in the response).

The action set is determined by the combination of the system actions for each slot (*null*, *request_info*, *implicit_confirm*, and *explicit_confirm*), resulting in a total of $4^4 - 1$ system actions. We added a *hangup* action to the action set for finishing the dialogue. Ideally, we would like to use all of 256 system actions, but we used 30 system actions which are appeared in the corpus due to the large memory requirement of the POMDP algorithm. We treated *request_info* actions for all slots as *greeting* system action.

The reward function is selected so that it penalizes long dialogues with -1 for every interaction and awards $+25$ for successful slot-filling or confirmation. The positive rewards are given only when the system executes *hangup* action, in order not to provide clues on how to complete the task. We also penalize some inappropriate system actions to block subsets of the action set for certain states: -10 for *request_info* on a known or confirmed slot and -10 for requesting confirmation for an unknown or confirmed slot. The reward function also assigned $+100$ for taking the *greeting* action in the first turn and -100 for *greeting* when not in the first turn. This represents that the *greeting* action may only be taken in the first turn. The discount factor of $\gamma = 0.95$ was used for all experiments.

4.3. User model implementation

There are 8 possible user responses for each slot, and hence the number of the combined user response is $8^4 = 4096$. To deal with data sparsity problem when building the Bigram model, we made the naive Bayes assumption, *i.e.*, the user response for each slot is conditionally independent of others given the system action: $P(a_u|a) = \prod_k P(a_k|a)$

We made the same conditional independence assumption for the Levin model. Furthermore, we assumed that the user response for a slot depends only on the system action associated with the slot. The admissible user responses for each system action for the slot was: (1) *null* and *provide_info* for *null* and *request_info*; (2) *null*, *reprovide_info*, *correct_info* and *reject_info* for *implicit_confirm*; (3) *yes_answer* and *no_answer* for *explicit_confirm*.

The original Pietquin model conditions the model parameters on the user goal. The user goal is represented as a table of slot-value pairs, but our dialogues are at the intention level ignoring the actual slot values. As a result, we worked around the problem by having boolean values representing whether the information regarding the slot has been provided or not, instead of the full slot-value table. Note that this workaround was also used in [13].

While obtaining the observation probabilities which represent the inaccurate results from the automated speech recognizer, we also had to deal with the data sparsity problem. The observation z is only dependent on the true user response, hence $O(s, a, z) = P(z|a_u)$. However the numbers of possible observations and user responses are 4096 each, and the table representation of the observation function would require 4096×4096 parameters. Hence, we made the assumption that the observation for a particular slot is only dependent on the user response in the set of related (frequently confused) slots: the observations for *orig_city* and *dest_city* are only dependent on the user responses in *orig_city* and *dest_city*, and those for *depart_date* and *depart_time* are only dependent on the user responses in *depart_date* and *depart_time*. This is a reasonable assumption because, for example, the observation values that *orig_city* and

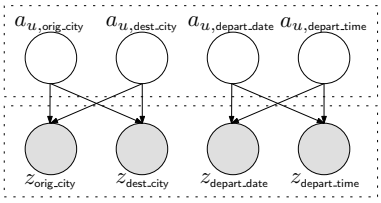

Figure 1: *Graphical model of the observation probability.*

dest_city are the same, and these two slots are often confused with each other. Fig. 1 shows the graphical model representation of the observation probability. This model is used in all of the three user models.

When constructing user models for MDPs, we used the output from the automated speech recognizer (available in the corpus), rather than the transcription (manually tagged true user response). Hence, we assumed that the output from the automated speech recognizer was true user response.

5. Experimental results

5.1. Dialogue strategy evaluation metric

The performance measure used in [6] rewards the filling and confirmation of slots while penalizing long dialogues. We extended this metric to be able to penalize misunderstandings of the system due to the speech recognition error, *i.e.*, wrong filling or confirmation. For example, the dialogue system can conclude that the slot is confirmed even though the user has not provided information or confirmed it when the dialogue is over. In this situation, the system could issue a wrong ticket.

Hence, for each of the four slots, the metric rewards $+25$ for the properly *known* slot and another $+25$ for the properly *confirmed* slot. The metric assigned -25 for the *known* slot without information provided, -25 for the *confirmed* slot without confirmation, and another -25 for the *confirmed* slot without information provided. Every interaction is penalized with -1. We calculate the reward using the dialogue state with the highest probability when the system takes *hangup* action.

5.2. Cross-model evaluation

To investigate the effect of the user model on the learned strategy, we tested the learned strategy across different user models. Fig. 2 shows the cross-evaluation result averaged over 1000 runs (reset after *hangup*, terminate after 70 steps). The results of MDP strategies are similar to the results reported previously in [6]: a strategy learned with a poor user model may appear to perform well when evaluated on the same user model used during learning but shows poor performance when tested on a different user model. Performance of the Bigram strategy degraded drastically when tested on the other user models. However, POMDP strategies significantly outperform MDP strategies when tested on the same user model, with a lower number of steps (Fig. 3). POMDP strategies also show better generalization performance than MDP strategies when evaluated on user models that were not used for computing the strategies.

5.3. Evaluation on real dialogue data

Using a user model for evaluating dialogue strategies inevitably introduces a bias. In [6], a technique was proposed for evaluating the learned dialogue strategy directly on the corpus: we first compute the similarity score $Sim(\pi_d, \pi^*)$ for each dialogue d

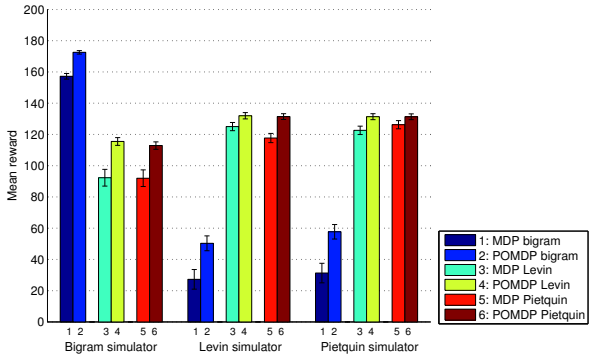

Figure 2: *Average rewards obtained by MDP and POMDP strategies across different user models. The error bars indicate 95% condfidence intervals.*

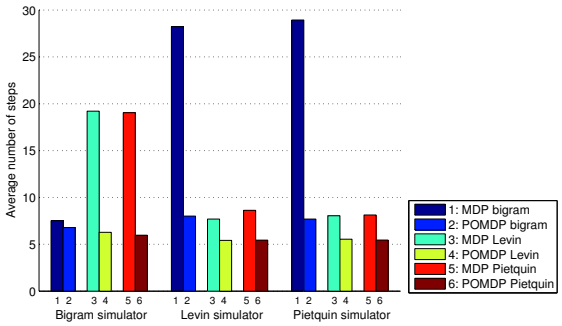

Figure 3: *Average number of steps.*

in real data based on how similar the strategy π_d followed in this dialogue d is to the learned strategy π^*. We then measure the correlation between $Sim(\pi_d, \pi^*)$ and the reward $Rew(d)$ of the dialogue d. This correlation is expected to reflect the quality of the learned strategy because a high correlation means that dialogues with a high similarity to the learned strategy tend to achieve higher rewards.

The similarity score between the MDP/POMDP strategy π^* and the strategy from the data π_d is the average similarities between two system actions,

$$Sim(\pi_d, \pi^*) = \frac{1}{n} \sum_{i=1}^{n} \theta_\pi(a_i) \tag{1}$$

The similarity between two system actions could be defined in various ways. Three methods were proposed in [6]: the reciprocal rank of the system action according to the ordering of Q-values, the ratio of Q-value to the sum of all Q-values, and the ratio of $\langle$system speech act, slot$\rangle$ pairs present in both a_i and a_{π^*} to pairs present in a_i or a_{π^*} defined as

$$\theta_\pi(a_i) = \frac{|\{a \in a_i\} \cap \{a \in a_{\pi^*}\}|}{|\{a \in a_i\} \cup \{a \in a_{\pi^*}\}|} \tag{2}$$

where a_{π^*} is the best action according to π^*, and a_i is the system action in the data. In this paper, we only used the third measure of similarity because computing Q-values for novel belief point was computationally prohibitive.

Table 1 shows the evaluation result on 100 dialogues in the Communicator corpus. Unfortunately, the average reward of the POMDP strategy is slightly lower than that of the MDP

Table 1: *Evaluation result on real dialogue data.*

	sim. score	avg. reward	corr.	area under line
MDP Bigram	0.342	95.99	0.528	105.07
MDP Levin	0.329	90.49	0.484	96.44
MDP Pietquin	0.328	90.49	0.484	96.58
POMDP Bigram	0.257	95.24	0.273	87.37
POMDP Levin	0.276	85.49	0.513	101.22
POMDP Pietquin	0.229	85.24	0.405	99.32

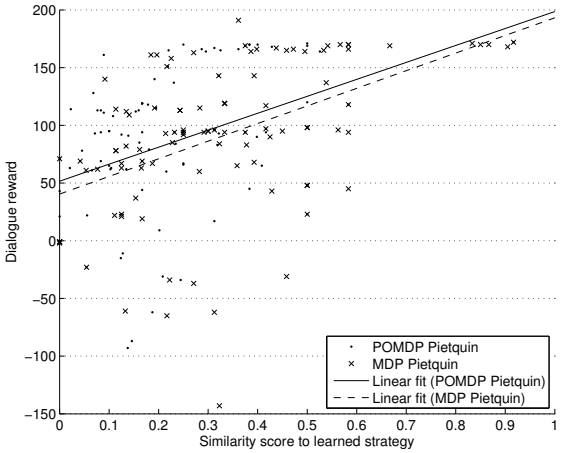

Figure 4: *The scatter plot with linear fits show that POMDP Pietquin strategy dominates MDP Pietquin strategy in spite of lower correlation coefficient.*

strategy trained on the same user model. We obtain the correlation coefficients for POMDP Bigram and POMDP Pietquin strategies which are lower than those for MDP strategies. Interestingly, the average similarity scores of POMDP strategies are also lower than MDP strategies and this indicates clearly that the handcrafted strategies used for collecting the corpus are more similar to the learned MDP strategies.

Hence, we claim that the similarity-reward correlation measure cannot capture the performance of the learned strategy. For example, the strategy which outperforms other strategies in the criterion of the average dialogue rewards can have a lower similarity-reward correlation (Fig. 4). We propose the linear least square fit to the similarity-reward data of the learned strategy and the area under the linear fit to evaluate the strategy on the real data. The linear fits of all learned strategies and the area under the linear fit of each strategy show that POMDP Levin and Pietquin strategy dominate MDP strategy except for MDP Bigram (Fig. 5, Table 1). All POMDP strategy shows higher dialogue rewards when the similarity score is less than 0.3502.

6. Conclusion

In this paper, we presented experiments that investigate the effect of the user model on POMDP-based dialogue systems and showed that POMDP strategies significantly outperform MDP strategies. POMDP strategies show better generalization performance than MDP strategies. For these experiments, we applied a POMDP to real-world scale dialogue management problems

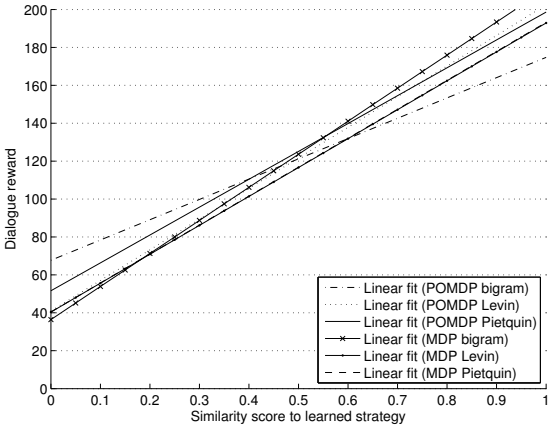

Figure 5: *Linear fits of similarity-reward.*

by using Symbolic HSVI for factored POMDPs. We proposed an appropriate dialogue strategy evaluation metric and a technique for evaluating the strategy directly on the corpus.

7. References

[1] J. D. Williams, P. Poupart, and S. Young, "Factored partially observable Markov decision processes for dialogue management," in *Proceedings of IJCAI-2005 Workshop on Knowledge and Reasoning in Practical Dialogue Systems*, 2005.

[2] J. D. Williams and S. Young, "Partially observable Markov decision processes for spoken dialog systems," *Computer Speech and Language*, vol. 27, 2007.

[3] J. Pineau, G. Gordon, and S. Thrun, "Anytime point-based approximation for large POMDPs," *Journal of Artificial Intelligence Research*, vol. 27, 2006.

[4] T. Smith and R. Simmons, "Point-based POMDP algorithms: Improved analysis and implementation," in *Proceedings of UAI-2005*, 2005.

[5] J. D. Williams and S. Young, "Scaling POMDPs for dialog management with composite summary point-based value iteration (CSPBVI)," *Proceedings of AAAI-2006 Workshop on Statistical and Empirical Approaches for Spoken Dialogue Systems*, 2006.

[6] J. Schatzmann, M. N. Stuttle, K. Weilhammer, and S. Young, "Effects of the user model on simulation-based learning of dialogue strategies," in *Proceedings of IEEE ASRU-2005*, 2005.

[7] H. S. Sim, K.-E. Kim, J. H. Kim, D.-S. Chang, M.-W. Koo, "Symbolic heuristic value iteration for factored POMDPs," in *Proceedings of AAAI-2008*, 2008.

[8] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial Intelligence*, vol. 101, 1998.

[9] J. Hoey, R. St-Aubin, A. Hu, and C. Boutilier, "SPUDD: Stochastic planning using decision diagrams," in *Proceedings of UAI-1999*, 1999.

[10] W. Eckert, E. Levin, and R. Pieraccini, "User modeling for spoken dialogue system evaluation," in *Proceedings of IEEE ASRU-1997*, 1997.

[11] E. Levin, R. Pieraccini, and W. Eckert, "A stochastic model of human-machine interaction for learning dialog strategies," in *IEEE Trans. on Speech and Audio Processing*, vol. 8, no. 1, pp. 11-23, 2000.

[12] O. Pietquin, *A Framework for Unsupervised Learning of Dialogue Strategies*, Ph.D. thesis, Faculte Polytechnique de Mons, 2004.

[13] J. Schatzmann, K. Georgila, and S. Young, "Quantitative evaluation of user simulation techniques for spoken dialogue systems," in *Proceedings of SIGDial-2005*, 2005.