Experiential Preference Elicitation for Autonomous HVAC Systems (Supplemental Material)

Andrew Perrault and Craig Boutilier

1 Proofs

Theorem 1. *Any EE instance can be formulated as a POMDP.*

Proof. The fundamental idea of the proof is to embed the set of possible rewards in the POMDP state space. This transformation is related to Poupart et al.'s [3] method of solving Bayesian reinforcement learning problems by formulating them as POMDPs. State transitions act on S, but never change the reward embedded in the state; thus, for any distinct $r, r' \in R$, the corresponding embedded state sets $S^{(r)}$ and $S^{(r')}$ do not communicate. The reward uncertainty distribution may be discrete or continuous (necessitating a POMDP with infinite states in the latter case). The action space A is augmented by the set of queries, so the agent can ask queries or take (original) actions; queries cause no state transition. The observation function for (original) actions reflects full observability of the (original) state space (observations are the states themselves), while for queries, the observation function captures the distribution over responses.

Figures 1 and 2 give a visual example of the transformation. Figure 1 shows the MDP of a two-state EE model where the reward in each state is either 0 or 1. Figure 2 shows the equivalent POMDP representation where there is a single query action available for each state.

We begin by constructing a model without elicitation.

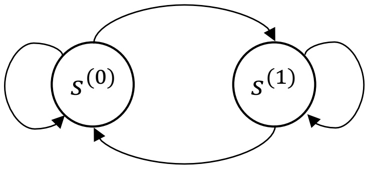

Figure 1: Diagram of a simple two-state MDP with deterministic transitions.

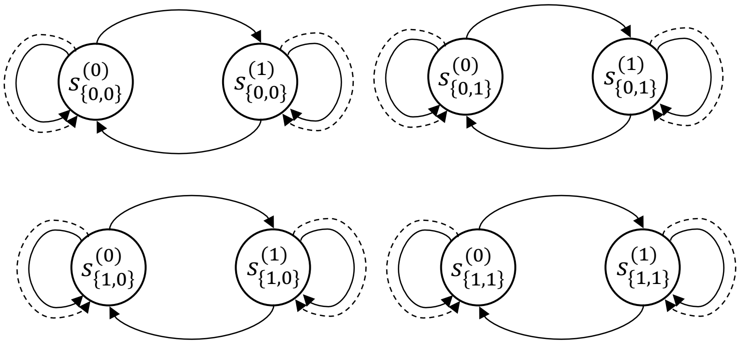

Figure 2: Diagram of the corresponding POMDP, where dashed lines represent query actions. There is one query action available in each state.

Formally, let S be the same as S in the MDP, except each state is duplicated $|R|$ times, with a copy corresponding to each possible reward function. We denote the set of states where r is the true reward function as $S^{(r)}$. Consider a state $s_r \in S^{(r)}$ that corresponds to $s \in S$. $A(s_r)$ is a superset of $A(s)$.

- Transitions: for each action $a \in A(s)$ and each state $s'_r \in S'^{(r)}$ with corresponding original state s', $P_{s_r a}(s'_r) = P_{sa}(s')$. $P'_{s_r a}(s'_r)$ is zero otherwise.

- Observations: each action emits the observation $\omega_{s'}$ with probability 1, which informs the agent of the identity of the state $s' \in S$ corresponding to the state s'_r it has transitioned into.

- Rewards: the reward is $r(s_r, a) = r(s, a)$.

- Initial state distribution: β' is $\beta(s_r) = \Pr(r)\beta(s)$ where $\beta(s)$ is the initial state distribution in the MDP.

The model now matches the EE instance exactly, but there is no way for the agent to gain information about the true reward function. To allow this, we will augment the action set with "querying" actions. These actions do not cause state transitions, but they emit observations that reveal information about the true reward function. The rewards of the actions are negative and given by the query cost function.

Let $q \in Q$ be a query. Create a corresponding action a_q which is available from every state.

- Transitions: a_q causes no transitions. $P_{s_r a_q}(s_r) = 1$.

- Rewards: $r(s, a_q) = -C(q, \mathbf{s})$ where $\mathbf{s}$ is equal to the state history. We need to augment the POMDP's state space to track the necessary history for both the query cost function C and the query response function. This augmentation only affects the rewards and observations of querying actions. For example, suppose our query cost and response models depend on the shortest distance between the queried state and any state we have visited in the last 50 states. We would need to augment the state representation with a S^{50} vector representing the past 50 visited states. If the model depends on an infinite state history (as do those in "Query Response and Cost Models" section), the POMDP requires an infinite dimensional state in the worst case.

- Observations: let Ω contain $\bigcup_q N_q$. $O_{s_r a_q} = D_q(r, \mathbf{s})$.

The agent can now perform elicitation. This concludes the construction, except for one issue: discounting. In the POMDP, discounting is applied when any action is taken (including a query), but in the EE model, discounting is only applied when a control action is taken. POMDPs can be extended so that certain actions do not cause discounting (and this does not interfere with many standard results). If we wish to make no such modification, we can augment the state space (storing the number of queries that have been asked so far) to fix the problem.

We show that an optimal policy for the POMDP is optimal for the EE instance. Observe that for a given sequence of actions and queries and a starting state distribution, the expected reward in the EE and POMDP models is identical. The actions have the same transition probabilities except that the POMDP state has the reward information embedded in it, and action trajectories can never cross from a state with a particular reward embedded to a state with a different reward embedded. The queries have the same costs associated with them and do not cause the state to change in either model. Thus, because the set of policies is the same in each model and the expected reward of each policy is the same, the optimal policy in the POMDP is the same as that in the EE.

□

Note that the POMDP belief state for an EE problem will reflect the agent's posterior over R, reflecting information captured about a user's preferences by queries and responses. POMDPs have been used to model elicitation problems in the past [1, 2]. These formulations differ from ours because they require only one state for each potential reward function.

We discuss the three main obstacles in the POMDP reduction and their impacts from a practical and theoretical perspective. The first is that the POMDP may require infinite states if R is continuous. This is not an important issue for two reasons: i) it would occur for any PE scenario where the support of R is infinite, regardless of model; and ii) the state space of POMDPs is often approximated in practice, even when it is finite, because of the high computational complexity of solving POMDPs.

The second, bigger, issue is that the POMDP may require an infinite-dimensional state space to keep track of the state history. EE, as we define it, is not Markov whereas POMDPs must be. Infinite dimensional POMDPs can be a problem because compressing the state dimension while keeping the relevant information is hard. However, it is unlikely in that any practical EE system would require unbounded state history to model responses accurately. Our definition permits unbounded state history, and we make use of it because it is elegant, but it is not necessary from a practical perspective.

The third issue is the different way discounting is handled in the two models. This is quite annoying from a practical perspective because it prevents any EE instance from being solved as a POMDP exactly without requiring an infinite-dimensional state space (unless the discount factor is 1). However, from a theoretical perspective, POMDPs having a fixed discount rate is more for ease of exposition than it is essential. From a practical perspective, we may simply ignore the difference (or tweak the discount rate, taking into account the expected number of queries that will be asked).

Observation 1. *Consider an RL problem $\langle \mathcal{M} \rangle$ that consists of an MDP $\mathcal{M}$ which is known to the agent except for the reward function, and whenever the agent transitions into a state, it receives the reward information for that state. This problem can be reduced to EE and the reduction requires increasing the number of states by a factor of $O(|S| \times |A|)$.*

Proof. Create an EE that retains the model details (states, actions, transitions, rewards, discount). We let reward uncertainty R be an uninformative prior. For each state s, we allow *value queries* for any state-action pair (s, a), which asks a "user" (representing the environment) for the reward for that pair, and the response function represents the RL (stochastic) reward for that pair. Query cost is zero if asking about the action just taken at the previous state, and infinite otherwise. To encode this query cost function, the EE state must include the previous state visited and action taken, which causes a state space blowup of $|S| \times |A|$.

In this EE instance, the only "available" query at a state is "free," so it is optimal to always ask it, giving an EE agent the same information as an RL agent. □

Observation 2. *Given a risk-neutral agent and an MDP with uncertain reward R, its optimal policy is that of an MDP where each state-action reward is its expected reward under R. (This holds even if rewards are correlated under R).*

Proof. This follows from a simple argument using linearity of expectation. Let $R_{s,a}$ be a random variable representing the (possibly correlated) state-action reward according to the agent's current beliefs. Given a risk-neutral agent, the optimal policy π^* of an MDP with uncertain reward R satisfies

$$\pi^*(s) = \operatorname*{argmax}_{a \in A(s)} \mathbb{E}\left[R_{s,a} + \gamma \sum_{s' \in S} P_{sa}(s')V^*(s')\right] \tag{1}$$

where

$$V^*(s) = \mathbb{E}\left[R_{s,\pi^*(s)} + \gamma \sum_{s' \in S} P_{s\pi^*(s)}(s')V^*(s')\right] \tag{2}$$

We can rewrite $\pi^*(s)$ using linearity of expectation and using the fact that $V^*(s)$ is already an expectation w.r.t. to R:

$$\pi^*(s) = \operatorname*{argmax}_{a \in A(s)}(\mathbb{E}[R_{s,a}] + \gamma \sum_{s' \in S} P_{sa}(s')V^*(s')) \tag{3}$$

We can rewrite $V^*(s)$ likewise:

$$V^*(s) = \mathbb{E}[R_{s,\pi^*(s)}] + \gamma \sum_{s' \in S} P_{s\pi^*(s)}(s')V^*(s') \tag{4}$$

Combining the two equations, we get the standard Bellman equation for the value function, but with $R_{s,a}$ replaced with $\mathbb{E}[R_{s,a}]$:

$$V^*(s) = \max_{a \in A(s)}\left(\mathbb{E}[R_{s,a}] + \gamma \sum_{s' \in S} P_{sa}(s')V^*(s')\right) \tag{5}$$

Since the optimal policy is the only policy that satisfies the Bellman equation, π^* is the optimal policy of the MDP with reward replaced by its expected value. □

Observation 3. *An MDP can be solved optimally given only information about the differences in rewards between a collection of state-action pairs.*

Proof. Knowing the difference in reward between all state-action pairs is equivalent to knowing the reward function up to an additive factor. Thus, it suffices to show that adding c to the reward of each state-action pair does not change the optimal policy.

The optimal policy π^* of an MDP is a policy that satisfies

$$\pi^*(s) = \underset{a \in A(s)}{\operatorname{argmax}} \left(r(s,a) + \gamma \sum_{s' \in S} P_{sa}(s') V^*(s') \right) \tag{6}$$

where $V^*(s) = r(s, \pi^*(s)) + \gamma \sum_{s' \in S} P_{s\pi^*(s)}(s') V^*(s')$.

Consider the effect on Equation 6 of replacing $r(s,a)$ with $r'(s,a) = r(s,a) + c$. Each $V^*(s)$ will increase by $\sum_t c\gamma^t$, where t is the number of steps remaining in the MDP. Because each $V^*(s)$ increases by the same amount, $\pi^*(s)$ remains the same. □

References

[1] Boutilier, C. 2002. A POMDP formulation of preference elicitation problems. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI-02)*, 239–246.

[2] Holloway, H. A., and White, III, C. C. 2003. Question selection for multiattribute decision-aiding. *European Journal of Operational Research* 148:525–543.

[3] Poupart, P.; Vlassis, N.; Hoey, J.; and Regan, K. 2006. An analytic solution to discrete Bayesian reinforcement learning. In *Proceedings of the Twenty-third International Conference on Machine Learning (ICML-06)*, 697–704.