Learning Internal States in POMDPs

Andrea Baisero Christopher Amato
baisero.a@husky.neu.edu c.amato@northeastern.edu
CCIS, Northeastern University

Notation: We denote the space of probability (mass or density) distributions over a set $\mathcal{X}$ as $\Delta(\mathcal{X})$.

Partially Observable Markov Decision Processes

A partially observable Markov decision process (POMDP) $\langle \mathcal{S}, \mathcal{A}, \mathcal{O}, T, O, R \rangle$ is composed of:

- State, action and observation spaces $\mathcal{S}, \mathcal{A}$, and $\mathcal{O}$;
- State dynamics $T \colon \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$;
- Observation emissions $O \colon \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \Delta(\mathcal{O})$;
- Reward function $R \colon \mathcal{S} \times \mathcal{A} \to \mathbb{R}$.

We further denote the space of observable histories as $\mathcal{H} \doteq (\mathcal{A} \times \mathcal{O})^*$.

Agent $\doteq$ Internal State Representation + Policy

Partial observability calls for agents capable of summarizing past events into an *internal state* (i-state) representation $\langle \mathcal{N}, n_0, \phi \rangle$ composed of:

- an i-state space $\mathcal{N}$, with initial i-state $n_0 \in \mathcal{N}$;
- i-state dynamics (i-dynamics) $\phi \colon \mathcal{N} \times \mathcal{A} \times \mathcal{O} \to \Delta(\mathcal{N})$ (often deterministic).

A *policy* $\pi \colon \mathcal{N} \to \Delta(\mathcal{A})$ complements the i-state representation and completes the acting agent.

Note

This notion of i-state representation encompasses all acting agents, notably:

- **Belief-MDPs** The i-state space $\mathcal{N} \doteq \Delta(\mathcal{S})$ is the set of belief-states, and the i-dynamics ϕ correspond to the Bayesian belief-update;
- **Memoryless/reactive agents** The i-state space $\mathcal{N} \doteq \mathcal{O} \cup \{n_0\}$ is the observation space extended with a singleton initial i-state n_0, and the i-dynamics $\phi \colon (n, a, o) \mapsto o$ return the observation;
- **Finite state controllers (FSCs)** The i-state space $\mathcal{N}$ is the set of FSC nodes, and the i-dynamics ϕ correspond to the FSC observation-strategy;
- **Recurrent neural networks (RNNs/LSTMs)** The i-state space $\mathcal{N}$ is the set of hidden states afforded by the recurrent network, and the i-dynamics ϕ is the network itself.

Learning with Policy Gradient Methods

- General family of model-free learning methods, e.g. A2C;
- Optimizes the agent performance, a.k.a. the RL objective;
- Applicable to learn i-dynamics in partially observable domains.

Issue with Learning Internal State Representations via Policy Gradient

In practice, learning both i-dynamics and policy by optimizing the RL objective results in tight dependencies:

- Quality of overall policy π depends on quality of overall i-dynamics ϕ;
  i.e. good actions require good context.
- Quality of overall i-dynamics ϕ depends on quality of overall policy π;
  i.e. context is good if good actions can be performed.
- Quality of overall i-dynamics ϕ depends on quality of overall i-dynamics ϕ;
  i.e. informative context is built on top of other informative context.

Issue: The initial i-dynamics provide no context to bootstrap the learning of good policies or better context;
$\Rightarrow$ Convergence to blind local optima.

Solution: Decouple the learning goals, by training i-dynamics based on predictiveness.

Predictive Internal State Models

IDEA

Learn domain structure by training i-dynamics to predict future observations and rewards.

We complement the agent's i-dynamics ϕ with predictive models:

- an observation model (o-model) $m_o \colon \mathcal{N} \times \mathcal{A} \to \Delta(\mathcal{O})$;
- a reward model (r-model) $m_r \colon \mathcal{N} \times \mathcal{A} \to \mathbb{R}$.

Goal: Train i-dynamics ϕ and predictive models m_o, m_r to match the domain's true predictive distributions,

$$m_o(\phi(n_0, h), a) \overset{!}{=} \Pr(o \mid h, a) \qquad \forall h \in \mathcal{H}, a \in \mathcal{A} \tag{1}$$

$$m_r(\phi(n_0, h), a) \overset{!}{=} \mathbb{E}_{s \sim \Pr(s \mid h)}\left[R(s, a)\right] \qquad \forall h \in \mathcal{H}, a \in \mathcal{A} \tag{2}$$

In the absence of the RHS target distributions, $\Pr(o \mid h, a)$ and $\mathbb{E}_{s \sim \Pr(s \mid h)}\left[R(s, a)\right]$, we propose 2 methods:

- *Experience Replay*: the predictive targets are approximated by sample experiences;
- *Inferential Reference*: the predictive targets are approximated by accumulated statistics.

Method 1: Experience Replay

IDEA

1. Store sample experience into experience replay buffers;
2. Train the predictive models using the replay buffers.

Past experiences are stored into prioritized *experience replay* buffers, and periodically re-sampled for training:

- The i-dynamics ϕ and o-model m_o are trained on a cross-entropy loss;
- The i-dynamics ϕ and r-model m_r are trained on a mean-squared-error loss.

Method 2: Inferential Reference

IDEA

1. Define a statistical model of observations and rewards for each history and action;
2. Update the statistical models using sample experiences;
3. Train the predictive models using the most recent statistical models.

For each history h and action a, we define independent (Bayesian or frequentist) statistical models $\rho_{h,a}$ of observations and rewards; We call the set of all such models, the *inferential reference* model $\rho \doteq \{\rho_{h,a}\}_{h,a}$.

- Experienced history-action-observation-rewards $\langle h, a, o, r \rangle$ are used to update the respective references $\rho_{h,a}$.

Past experiences are summarized by reference models $\rho_{h,a}$, which are periodically re-sampled for training:

- The i-dynamics ϕ and o-model m_o are trained on a loss defined by the observation reference model;
- The i-dynamics ϕ and r-model m_r are trained on a loss defined by the reward reference model.

Note

While the reference model ρ is unable to generalize between histories, the i-dynamics ϕ is still able to do so.

Observation Reference Model and Loss

The Dirichlet-categorical conjugate pair is a natural choice,

$$\omega_{h,a} \sim \text{Dirichlet}(\{\alpha_{h,a,o'}\}_{o'}) \,, \tag{3}$$

$$o_{h,a} \sim \text{Categorical}(\omega_{h,a}) \,, \tag{4}$$

which facilitates Bayesian inference,

$$\omega_{h,a} \mid o \sim \text{Dirichlet}(\{\tilde{\alpha}_{h,a,o'}\}_{o'}) \,, \tag{5}$$

$$\tilde{\alpha}_{h,a,o'} \mid o = \alpha_{h,a,o'} + \mathbb{I}\left[o = o'\right] \,. \tag{6}$$

Models $\langle \phi, m_o \rangle$ are scored via the neg-log-likelihood of the prediction $m_o(\phi(n_0, h), a) \in \Delta(\mathcal{O})$ w.r.t. the reference Dirichilet distribution:

$$\begin{aligned} \mathcal{L}_o(\theta; \rho, \langle h, a \rangle) &= -\log \text{Dirichlet}(x; \{\alpha_{h,a,o'}\}_{o'}) \big|_{x = m_o(\phi(n_0, h), a)} \\ &= \log B(\{\alpha_{h,a,o'}\}_{o'}) + \sum_{o'} (\alpha_{h,a,o'} - 1)\,(-\log x_{o'}) \end{aligned} \tag{7}$$

$$\nabla_\theta \mathcal{L}_o(\theta; \rho, \langle h, a \rangle) = \sum_{o'} (\alpha_{h,a,o'} - 1)\, \nabla_\theta \,(-\log x_{o'}) \tag{8}$$

Note

This loss has the desired effect whereby reoccurring histories influence the learning procedure more heavily as a result of higher accumulated counts $\{\alpha_{h,a,o'}\}_{o'}$.

Reward Reference Model and Loss

A simpler frequentist approach is used, whereby only the empirical cumulative average of rewards is maintained. The model parameters are initialized to contain no prior knowledge,

$$\mu_{h,a} = 0.0 \,, \tag{9}$$

$$\nu_{h,a} = 0 \,, \tag{10}$$

and updated, upon observing a new reward, via the cumulative average equation,

$$\tilde{\mu}_{h,a} \mid r = \frac{\mu_{h,a}\nu_{h,a} + r}{\nu_{h,a} + 1} \,, \tag{11}$$

$$\tilde{\nu}_{h,a} \mid r = \nu_{h,a} + 1 \,. \tag{12}$$

Models $\langle \phi, m_r \rangle$ are scored via the squared difference between prediction $m_r(\phi(n_0, h), a)$ and reference:

$$\mathcal{L}_r(\theta; \rho, \langle h, a \rangle) = (x - \mu_{h,a})^2 \Big|_{x = m_r(\phi(n_0, h), a)} \,. \tag{13}$$

Evaluation

We compare the performance of 4 methods:

- **True-Belief** (Soft Upper Bound)
  - Uses the true (unavailable) belief-state;
  - Trains policy π with A2C.
- **Recurrent A2C**
  - Trains both i-dynamics ϕ and policy π with A2C.
- **Experience Replay**
  - Trains i-dynamics ϕ with the experience replay method;
  - Trains policy π with A2C.
- **Inferential Reference**
  - Trains i-dynamics ϕ with the inferential reference method;
  - Trains policy π with A2C.

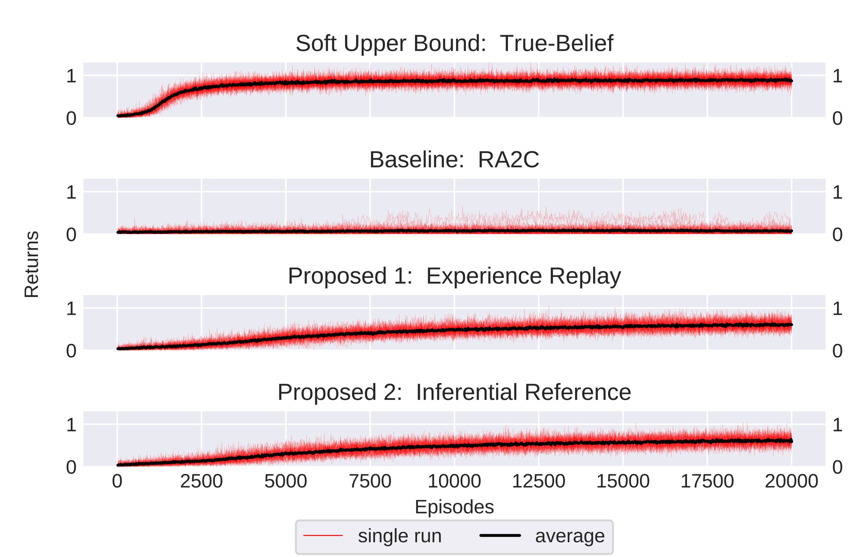
Architectures

Recurrent models:

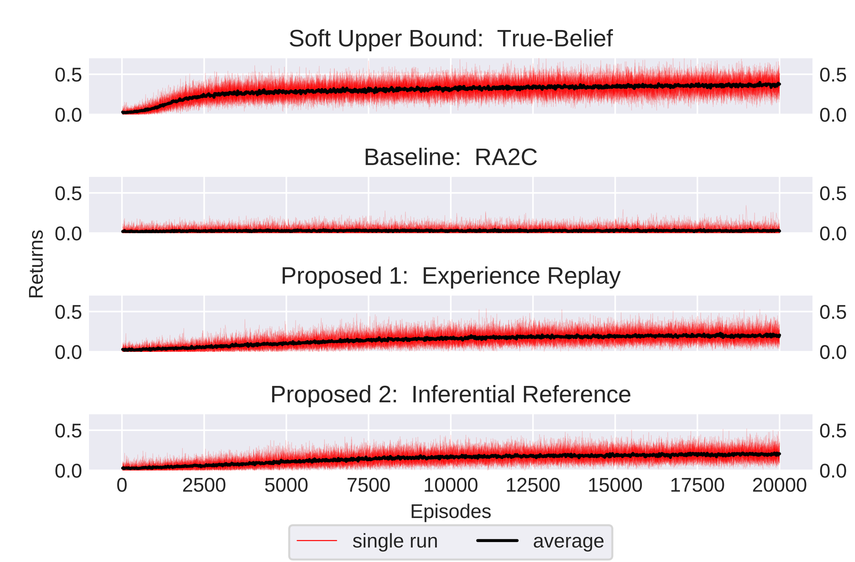
- I-dynamics;
- 2-layer LSTM, *tanh*;
- N° hidden state units = N° environment states.

Feedforward models:

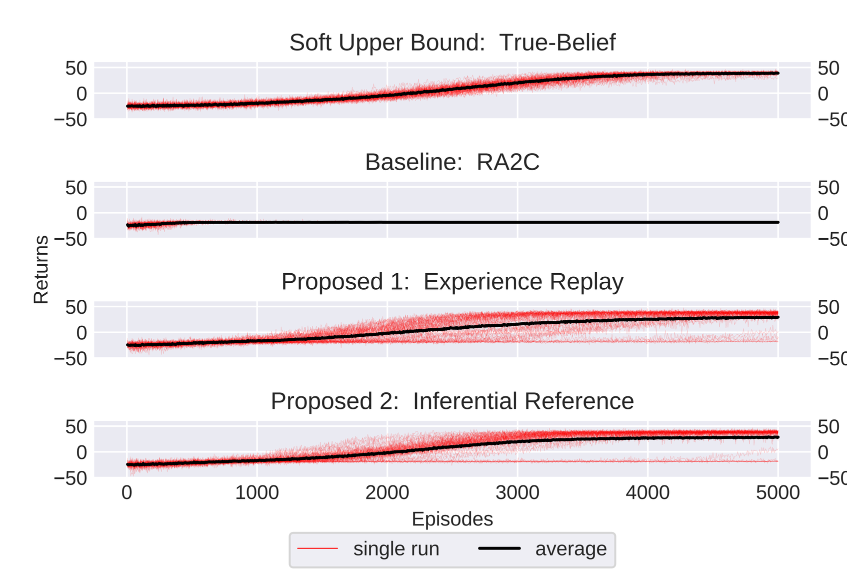
- Policy, critic, o-model, r-model;
- Single-hidden-layer MLP, *leaky-ReLU*;
- N° hidden units = N° input units.



(a) Hallway, $|\mathcal{S}| = 60$, $|\mathcal{A}| = 5$, $|\mathcal{O}| = 21$, (smoothened)



(b) Hallway2, $|\mathcal{S}| = 92$, $|\mathcal{A}| = 5$, $|\mathcal{O}| = 17$, (smoothened)



(c) Shopping, $|\mathcal{S}| = 16$, $|\mathcal{A}| = 6$, $|\mathcal{O}| = 4$, (smoothened)

Figure: Results for Hallway, Hallway2, and Shopping domains.

In the hallway domains (figs. 1a and 1b):

- RA2C is largely unable to improve upon the initial policy;
- Experience Replay and Inferential Reference objectively outperform RA2C.

In the shopping domain (fig. 1c):

- RA2C quickly converges to a blind local optimum;
- Experience Replay and Inferential Reference are able (about 88% of the time) to avoid the local optimum;
- Most runs either fully succeed or fail to learn useful i-dynamics:
  - Likely, policy converges faster than i-dynamics;
  - $\Rightarrow$ more exploration required.

Conclusions

Learning useful state representations is a fundamental necessity for agents operating under partial observability. We can summarize our contributions as follows:

- Learning i-dynamics via the RL objective suffers from convergence to blind local optima;
- Learning i-dynamics via the predictive objective helps learn domain structure and avoid blind local optima;
- The proposed methods are able to learn i-states as useful as the true belief-state;

Future Work

- Enforce exploration by hindering policy convergence to be slower than i-dynamics;
- More sophisticated (Bayesian) reward reference model and loss;
- Scale proposed methods to larger domains;
- The learned i-dynamics and predictive models form an "i-state"-MDP $\Rightarrow$ solve it via planning.