# Hierarchical iPOMDPs for MTRL

Momchil

December 4, 2017

## 1 Background

Define MDPs, RL framework POMDPs Multi-task RL HRL = sMDPs Option discovery as hidden state inference POMDP formulation for one environment Multi-task POMDP

### 1.1 Reinforcement Learning and Markov Decision Processes

In reinforcement learning (RL), tasks are traditionally represented as Markov Decision Processes (MDPs) [1]. We define an MDP as a tuple $(\mathcal{S}, \mathcal{A}, T, R, I, \mathcal{F})$ where:

- $\mathcal{S}$ is a set of states that the agent can occupy,
- $\mathcal{A}$ is a set of actions that the agent can perform in those states,
- $T(\cdot|s,a)$ is a distribution of next states, such that $T(s'|s,a)$ is the probability that the agent ends up in state $s'$ after executing action $a$ in state $s$,
- $R(s,a)$ is the amount of reward an agent obtains after executing action $a$ in state $s$,
- $I(\cdot)$ is a distribution of initial states in which the agent may start the task,
- $\mathcal{F}$ is a set of terminal states in which the task is completed.

The agent begins the task in state $s_0 \sim I$. At each time step $t$, the agent performs action $a_t$ that takes it from state $s_t$ to state $s_{t+1} \sim T(\cdot|s_t, a_t)$ and receives reward $r_t \sim R(s_t, a_t)$. The agent is said to follow a policy $\pi$ if it chooses its actions according to $a_t \sim \pi(\cdot|s_t)$. For a fixed policy $\pi$, the total expected discounted reward for a given state $s$ is the sum of all future rewards that the agent will receive, on average, if it starts in state $s$ and follows $\pi$:

$$V_\pi(s) = \mathbb{E}_\pi \left[ \sum_{t=0}^{\infty} \gamma^t r_t \middle| s_0 = s \right],$$

where $\gamma$ is the discount rate which determines the present value of future rewards. The goal of the agent is to select an optimal policy $\pi^*$ that maximizes $V_{\pi^*}(s)$. The optimal value function $V_{\pi^*}(s)$ satisfies the Bellman optimality equation [2]:

$$V_{\pi^*}(s) = \max_a \left\{ R(s,a) + \sum_{s'} V_{\pi^*}(s')\ T(s'|s,a) \right\}.$$

which serves as a basis of a number of algorithms for solving optimal control problems. RL algorithms often fall into one of two broad categories: model-free algorithms, in which the agent learns $V_\pi$ directly; and model-based algorithms, in which the agent learns $R$ and $T$ to compute $V_\pi$.

Computational cognitive neuroscience has borrowed this dichotomy as a formal description of the distinction between habitual and goal-directed behaviors [3]. Model-free accounts of habitual behaviors posit that animals learn automatic responses to stimuli if they tend to lead to rewarding outcomes. Model-based accounts of goal-directed behaviors posit that animals learn the statistics of the environment independently of the rewards.

Model-free accounts are less computationally intensive, however they require many training examples and have to be re-trained whenever the reward distribution changes. This is consistent with empirical observations that overtrained animals exhibit fast reaction times and rigid behavioral patterns that tend to persist even reward contingencies change. Conversely, model-based accounts predict that agents will flexibly adapt to changes in reward distributions, at the cost of slower reaction times due to the greater computational complexity.

## 1.2 Hierarchical Reinforcement Learning and semi-Markov Decision Processes

From a computational standpoint,

$$\begin{aligned} \bar{T} &\sim \mathrm{GEM}(...) \\ T(.|s) = T_s &\sim \mathrm{DP}(..., \bar{T}) \\ \phi_s &\sim H \\ s_t &\sim T_{s_{t-1}} \\ o_t &\sim F(\phi_{s_t}) \end{aligned}$$

### 1.3 hierarchical iHMM as HDP

$$\begin{aligned}
\beta &\sim \text{GEM}(...) \\
z_s &\sim \text{Mult}(\beta) \\
\bar{T} &\sim \text{GEM}(...) \\
\bar{T}_c &\sim \text{DP}(..., \bar{T}) \\
T(.|s) = T_s &\sim \text{DP}(..., \bar{T}_{z_s}) \\
\phi_s &\sim H \\
s_t &\sim T_{s_{t-1}} \\
o_t &\sim F(\phi_{s_t})
\end{aligned}$$

$$\begin{aligned}
H_c &\sim ... \\
\phi_s &\sim H_{z_s}
\end{aligned}$$

### 1.4 hierarchical iHMM as HDP, version 2

$$\begin{aligned}
\bar{T} &\sim \text{GEM}(...) \\
T(.|c) = T_c &\sim \text{DP}(..., \bar{T}) \\
\bar{T}_{c,\cdot} &\sim \text{GEM}(...) \\
T(. \in c | s \in c) = T_{c,s} &\sim \text{DP}(..., \bar{T}_{c,\cdot}) \\
T_s(s') = T(s'|s) = &\begin{cases} T_{c,s}(s')\ T_c(c) & \text{if } s, s' \in c \\ \bar{T}_{c',\cdot}(s')\ T_c(c') & \text{if } s \in c, s' \in c' \end{cases} \\
\phi_s &\sim H \\
s_t &\sim T_{s_{t-1}} \\
o_t &\sim F(\phi_{s_t})
\end{aligned}$$

### 1.5 hierarchical iHMM as HDP, version 3

$$\begin{aligned}
\beta &\sim \text{GEM}(...) \\
z_c &\sim \text{Mult}(\beta) \\
\bar{T}_{g,\cdot} &\sim \text{GEM}(...) \\
T_{g,s} &\sim \text{DP}(..., \bar{T}_{g,\cdot}) \\
\bar{T} &\sim \text{GEM}(...) \\
T(.|c) = T_c &\sim \text{DP}(..., \bar{T}) \\
\bar{T}_{c,\cdot} &= \bar{T}_{g=z_c,\cdot} \\
T(. \in c | s \in c) = T_{c,s} &= T_{g=z_c,s} \\
T_s(s') = T(s'|s) = &\begin{cases} T_{c,s}(s')\ T_c(c) & \text{if } s, s' \in c \\ \bar{T}_{c',\cdot}(s')\ T_c(c') & \text{if } s \in c, s' \in c' \end{cases} \\
\phi_s &\sim H \\
s_t &\sim T_{s_{t-1}} \\
o_t &\sim F(\phi_{s_t})
\end{aligned}$$

## Bibliography & References Cited

[1] Richard S. Sutton and Andrew G. Barto. *Introduction to Reinforcement Learning.* MIT Press, Cambridge, MA, USA, 1st edition, 1998.

[2] Richard Bellman. *Dynamic Programming.* Princeton University Press, Princeton, NJ, USA, 1957.

[3] Ray J Dolan and Peter Dayan. Goals and habits in the brain. *Neuron*, 80(2):312–325, 2013.