Learning with Recognizers

Pierre-Luc Bacon and Doina Precup

McGill University

Off-Policy learning and recognizers

This talk is building upon *Off-policy Learning with Recognizers* (Precup et al, 2005), which apparently was conceived around a bottle of rhum in Barbados 2004.

Problem

- Off-policy learning relies on importance sampling weights, which can have high variance
- The recognizer idea is to define a class policies for which the importance sampling corrections have minimum variance.

Recognizers

- A recognizer is a function $c : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^+$
  - Note: the map might not be in $[0, 1]$
  - In this talk, we will however give it a probabilistic interpretation
- A recognizer and a **behavior** policy $b : \mathcal{S} \times \mathcal{A} \to [0, 1]$ induce a **target policy** π as follows:

$$\pi(s, a) = \frac{b(s, a)c(s, a)}{\sum_{a' \in \mathcal{A}} b(s, a')c(s, a')} = \frac{b(s, a)c(s, a)}{\eta(s)}$$

The target policy is not explicitely specified (one of Doina's point on Monday)

- Recognizer functions (c) are **about courses of actions**, but are not policies themselves.
- They let us focus on *things of interest*:
  - They form "tubes"/"highways"/paths of the state-action space (Jan's talk)
- They allows us to learn from the **different ways of behaving** in order to achieve a goal:
  - Eg: grabbing a cup from the left or the right
  - Good for a Horde-like system that is trying to learn the most out of its experience

Options framework

An option is a triple: $\langle \mathcal{I} \subseteq \mathcal{S},\ \pi : \mathcal{S} \times \mathcal{A} \to [0,1],\ \beta : \mathcal{S} \to [0,1]$

- **initiation set** $\mathcal{I}$
- **policy** π (stochastic or deterministic)
- **termination condition** β

I want us to have a more principled approach for **option discovery**

- I think that past work focused too much on **task decomposition**
- We might benefit from **thinking less about subgoals**

Expressing options with recognizers

Initiation

$$\mathcal{I} := \{s \mid \eta(s) > 0\}$$

Policy

$$\pi(s, a) := \frac{b(s, a)c(s, a)}{\eta(s)}$$

Termination

$$\beta(s) := \mathbb{1}_{\eta(s)=0}$$

- Recognizer-induced options de-emphasize termination
  - What matter is the courses of actions
- However, subgoals can still be expressed in this framework
  - They are those states where no action is recognized
  - One could choose to use threholds to express initiation and termination
- We would expect recognizers to be very good in continuous action spaces under continuous dynamics

Recognizers and humans

- Mirror neurons in the premotor area of monkeys:
  - Neurons that activate when observing external actions
  - Involved in *motor understanding*
- Ideomotor principle: a common coding for action and perception
- Affordances: $c(s, a)$ somehow talks about the actions that are "afforded" under the influence of the behavior policy

Learning recognizers

We will parametrize our recognizer and learn with policy gradient methods.

Assumptions

- The behavior policy is known
- Experience is generated from the recognizer-induced policy
  - The stationary distribution is then:

$$d^{\pi}(s) = \sum_{t=0}^{\infty} \gamma^{t} \, \mathbb{P}\{s_t = s \mid s_0, \pi\}$$

- For now, we only consider the single recognizer case

Objective

We want to maximize discounted return while having well-behaved importance sampling corrections:

$$J(\pi) = \mathbb{E}\left\{ \sum_{t=1}^{\infty} \gamma^{t-1} r_t \;\middle|\; s_0, \pi \right\} - \zeta D_{\mathrm{KL}}(\pi || b)$$

ζ is a *knob* for controlling this tradeoff

- A similar D_{KL} term can also found in:
  - Jan's *Relative Entropy Policy Search* (REPS) algorithm,
  - Emanuel Todorov's *linearly solvable MDPs*,
  - recent work by Sergey Levine on *guided policy search*

Reward term:

$$\mathbb{E}\left\{\sum_{t=1}^{\infty}\gamma^{t-1}r_t \,\middle|\, s_0,\pi\right\} = \sum_s d^{\pi}(s)\sum_a \pi(s,a)r(s,a)$$
$$= \mathbb{E}_{s\sim d^{\pi},a\sim\pi}\left\{r(s,a)\right\}$$

Divergence term:

$$D_{\mathsf{KL}}(\pi||b) = \sum_{(s,a)} d^{\pi}(s)\pi\left(a \mid s\right)\log\frac{\pi\left(s \mid a\right)}{b\left(s \mid a\right)}$$
$$= \sum_{s,a} d^{\pi}(s)\,\frac{b\left(s \mid a\right)c_{\theta}(s,a)}{\eta(s)}\log\frac{c_{\theta}(s,a)}{\eta(s)}$$
$$= \mathbb{E}_{s\sim d^{\pi},a\sim\pi}\left\{\log\frac{c_{\theta}(s,a)}{\eta(s)}\right\}$$

By linearity:

$$J(\pi) = \mathbb{E}_{s \sim d^{\pi}, a \sim \pi} \left\{ r(s, a) - \zeta \log \frac{c_{\theta}(s, a)}{\eta(s)} \right\}$$

Gradient of the action-value function:

$$\nabla_\theta Q^\pi(s,a) = \nabla_\theta \left[r(s,a) - \zeta \log \frac{c_\theta(s,a)}{\eta(s)} + \sum_{s'} \gamma P\left(s' \mid s,a\right) V^\pi(s') \right]$$

Gradient of our objective:

Let $\widetilde{Q}$ be the state-action value function for the modified reward function:

$$
\begin{aligned}
\nabla_\theta J(\pi_\theta) &= \nabla_\theta \widetilde{V}^\pi(s_0) = \nabla_\theta \left[\sum_a \pi(s,a)\widetilde{Q}^\pi(s,a)\right] \\
&= \sum_a \nabla_\theta \pi(s,a)\widetilde{Q}^\pi(s,a) + \pi(s,a)\nabla_\theta \widetilde{Q}^\pi(s,a) \\
&= \sum_a \nabla_\theta \pi(s,a)\widetilde{Q}^\pi(s,a) + \pi(s,a)\left[\sum_{s'} \gamma P\left(s' \mid s,a\right)\nabla_\theta \widetilde{V}^\pi(s')\right] \\
&= \sum_s d^\pi(s)\sum_a \nabla_\theta \pi(s,a)\widetilde{Q}^\pi(s,a)
\end{aligned}
$$

Our last results followed directly from the policy gradient theorem.

$$\nabla J(\pi_\theta) = \mathbb{E}_{s \sim d^\pi, a \sim \pi} \left\{ \nabla \log \pi(s,a) \widetilde{Q}^\pi(s,a) \right\}$$

Demo

- Four state linear chain where the goal state is the leftmost state.
  - 10% chance of staying in the same state
  - Two actions: go left or right
- Behavior policy: biased random walk
- We parametrized the recognizer as a sigmoid function of the form:

$$c(s, a) = \sigma\left(\mathbf{A}\Phi(\mathbf{s}) + \mathbf{b}\right)$$

  where ϕ is a basis function, $\mathbf{A}$ is a map to the action space $\mathbb{R}^{|\mathcal{A}|}$ and $\mathbf{b}$ is a bias term.
- SARSA(λ) was used to learn $\widetilde{Q}(s, a)$

Future

- Problem: the reward function is no longer stationary:
  - Would like theoretical result: under small enough variations, this might be fine
  - Anna H. : could we fix this by a potential-based formulation ?
  - RL is already quite good a tracking, it might just work in practice
- How to extend to multiple options
  - How to obtain *diverse* options
- Application to *learning from demonstration*
- What role can recognizers play in formalizing the idea of **intentions**