1711.03817 - Learning with Options that Terminate Off-Policy

- **Yunqiu Xu**

- NIPS 2017 HRL Workshop

---

1. Introduction

- A revision of option framework $\omega \in \Omega$:
  - Initiation set $I \subseteq S$
  - Intra-option policy $\pi_\omega : S \rightarrow A$, this is **sub-policy**
  - Termination condition $\beta_\omega : S \rightarrow [0, 1]$
  - Given a state, master policy π select an option (suitable initiation set), then its intra-option policy will be executed to reach terminate state of this subtask $\rightarrow$ a new state for next iteration until final end
- **Learning with longer options is more efficient, why?**
  - Termination condition β is similar to learning rate (λ) in TD-learning
  - Thus can make it faster to converge
- Challenges:
  - β will not only influent learning rate but also affect the solution
  - So if the option set is not ideal, the performance will be affected
  - **In this condition, shorter options can be better (more flowxible)**
- Our work:
  - Try to terminate options "off-policy"
  - **Decouple the behavior termination condition from target termination condition**
    - Behavior TC: options execute with this TC, which influence the **convergence speed**
    - Target TC: factored into the solution

  - Q(β):
    - learn to evaluate a task w.r.t. options terminating off-policy
    - learn an optimal solution from suboptimal options quicker than the alternatives

2. Framework and Notation

2.1 Multi-step off-policy TD learning

- Multi-step TD learning:

$$T_\lambda^\pi q = (1-\lambda)\sum_{t=0}^{\infty}\lambda^n (T^\pi)^n q = q + (I - \lambda\gamma P^\pi)^{-1}(T^\pi q - q)$$

- What is off-policy learning:
  - Behavior and target policies are decoupled
  - $\pi^b \neq \pi$
  - **此处存疑, off-policy是不是类似我之前学DQN时的target net和eval net, target net用于选择动作但每隔一定步数才会更新参数**
- Multi-step off-policy TD learning:
  - Munos et al. 2016. Safe and efficient off-policy reinforcement learning
  - Asis et al. 2017. Multi-step reinforcement learning: A unifying algorithm.

$$\begin{aligned}
\delta_t^{\sigma,\pi} &= R_{t+1} + \gamma(\sigma q(S_{t+1}, A_{t+1}) \\
&\qquad + (1-\sigma)\mathbb{E}_\pi q(S_{t+1}, \cdot)) - q(S_t, A_t), \\
c_i &= \lambda((1-\sigma)\pi_b(A_i|S_i) + \sigma).
\end{aligned}$$

In particular, $\sigma = 1$ corresponds to the on-policy SARSA(λ) algorithm, while $\sigma = 0$ to Tree-Backup(λ).

2.2 Options

- Similar to introduction (initiation set + option policy + terminition set), but some symbles may be different

- **这里暂略**

3. Call-and-return operator

4. Off-policy option termination

Algorithm 1 Q(β) algorithm

Given: Option set $\mathcal{O}$, target termination function β, initial Q-function q_0, step-sizes $(\alpha_k)_{k\in\mathbb{N}}$, start state s_0

$S_0 \leftarrow s_0$
for $k = 0, 1, \ldots$ **do**
  Sample an option o from $\mu_k(\cdot|S_0)$
  Sample the return $R_1, S_1, R_2 \ldots, S_{D_k}$ from π^o. D_k is determined by sampling $1 - \zeta^o(S_i)$.
  for $t = 0, 1, \ldots D_k - 1$ **do**
    $\delta_t^{\beta,\mu_k} = R_{t+1} + \gamma \tilde{q}_{\mu_k}(S_{t+1}, o) - q(S_t, o)$
    $\tilde{q}_{\mu_k}(s, o) \stackrel{\text{def}}{=} (1 - \beta^o(s)) q(s, o) + \beta^o(s) \mathbb{E}_{\mu_k} q(s, \cdot)$
    $c_j^o = 1 - \beta^o(S_j) + \beta^o(S_j) \mu(o|S_j)$
    $\Delta_t = \sum_{i=t}^{D_k - 1} \gamma^{i-t} \left(\prod_{j=t+1}^{i} c_j^o \right) \delta_t^{\beta,\mu_k}$
    $q_{k+1}(S_t, o) \leftarrow q_k(S_t, o) + \alpha_k \Delta_t$
  end for
  $S_0 \leftarrow S_{D_k}$
end for

5. Experiment and Analysis

6. Summary

- 本工作致力于改进option framework
  - Longer option is faster to converge but will affect performance when option set is not ideal
  - We decouple the behavior and target terminations (similar to off-policy learning)
  - Learn the solution with respect to any termination condition, regardless of how the options terminate
- 看得比较粗略, **to be continued**