The Option-Critic Architecture

Pierre-Luc Bacon, Jean Harb, Doina Precup
Reasoning and Learning Lab, School of Computer Science
McGill University
{pbacon, jharb, dprecup}@cs.mcgill.ca

【强化学习算法 20】Option-Critic

张楚珩
清华大学 交叉信息院博士在读

原文传送门：

Bacon, Pierre-Luc, Jean Harb, and Doina Precup. "The Option-Critic Architecture." AAAI. 2017.

特色：

另一大类分层强化学习算法的代表。个人认为这类方法最大的优势是把上层策略和下层策略的控制权移交问题做成一个可以学习的函数（termination function），这样就不再限定上下层策略的temporal resolution是一个固定的c步的缩放关系了。这实际增大了模型容量，为可能学习到的策略提供了更多的可能。（如果不太懂可以看本专栏FuN算法的背景部分）换句话说，可能学到跨度不一样的sub-policy。

过程：

1. option framework

可以简单理解option就和sub-policy差不多的意思。该上层策略做决策的时候，上层策略选择一个option $\omega \in \Omega$，这个option包含两个部分，一个是这个option的行动策略 $\pi_\omega(a|s)$，另一个是这个option的终止函数 $\beta_\omega(s) \to \{0,1\}$。当终止函数返回0的时候，下一步还会由当前的这个option来控制；当终止函数返回1的时候，该option的任务就暂时完成了，控制权就交回给上层策略。把每个option的行动策略和终止函数都用function approximation来参数化表示，即 $\pi_{\omega,\theta}(a|s)$ 和 $\beta_{\omega,\vartheta}(s)$。

有了这些option之后，还需要有用来在这些option之间做选择的上层策略。$\pi_\Omega(\omega|s)$ 表示在状态 s 的时候策略选择option ω 的概率。

在此基础上，我们可以定义各类价值函数。

$$Q_\Omega(s,\omega) = \sum_a \pi_{\omega,\theta}(a \mid s) Q_U(s,\omega,a) \ , \qquad (1)$$

$$Q_U(s,\omega,a) = r(s,a) + \gamma \sum_{s'} \mathrm{P}(s' \mid s,a) U(\omega,s') \ . \quad (2)$$

$$U(\omega, s') = (1 - \beta_{\omega,\vartheta}(s'))Q_\Omega(s', \omega) + \beta_{\omega,\vartheta}(s')V_\Omega(s') \quad (3)$$

分别表示“在某状态下选择某个option之后产生的总收益”、“在某状态、选择某个option时，采取某行动之后产生的总收益”和“在使用某option时到达某状态之后产生的总收益”，这些价值函数都是相对于当前策略而言的，即除了固定规定的变量之外，之后都服从当前策略运行。

2. option内的更新

option主要包含各个option的策略 $\pi_{\omega,\theta}(a|s)$ 和其终止函数 $\beta_{\omega,\vartheta}(s)$。如果我们能推导出最后总discounted return相对于其参数的导数的话，就可以利用类似policy gradient的方法来更新其参数了。文章的主要贡献就是推导出了相对于这两个参数的policy gradient。

Theorem 1 (Intra-Option Policy Gradient Theorem). *Given a set of Markov options with stochastic intra-option policies differentiable in their parameters θ, the gradient of the expected discounted return with respect to θ and initial condition (s_0, ω_0) is:*

$$\sum_{s,\omega} \mu_\Omega(s, \omega \mid s_0, \omega_0) \sum_a \frac{\partial \pi_{\omega,\theta}(a \mid s)}{\partial \theta} Q_U(s, \omega, a) \ ,$$

where $\mu_\Omega(s, \omega \mid s_0, \omega_0)$ is a discounted weighting of state-option pairs along trajectories starting from (s_0, ω_0): $\mu_\Omega(s, \omega \mid s_0, \omega_0) = \sum_{t=0}^{\infty} \gamma^t \mathrm{P}(s_t = s, \omega_t = \omega \mid s_0, \omega_0)$.

Theorem 2 (Termination Gradient Theorem). *Given a set of Markov options with stochastic termination functions differentiable in their parameters ϑ, the gradient of the expected discounted return objective with respect to ϑ and the initial condition (s_1, ω_0) is:*

$$-\sum_{s',\omega} \mu_\Omega\left(s', \omega \mid s_1, \omega_0\right) \frac{\partial \beta_{\omega,\vartheta}(s')}{\partial \vartheta} A_\Omega(s', \omega) \ ,$$

where $\mu_\Omega\left(s', \omega \mid s_1, \omega_0\right)$ is a discounted weighting of state-option pairs from (s_1, ω_0): $\mu_\Omega\left(s, \omega \mid s_1, \omega_0\right) = \sum_{t=0}^{\infty} \gamma^t \mathrm{P}\left(s_{t+1} = s, \omega_t = \omega \mid s_1, \omega_0\right)$.

再使用TD的方式学习到critic Q_U 和 A_Ω 的话，就能更新各个option内的参数了，从而进行option的学习了。

3. 上层策略

上层策略主要使用 $\pi_\Omega(\omega|s)$ 来描述，可以认为它是一个greedy的策略，即选择在所有option中价值函数最大的option，即

$$\max_{\omega} \sum_{a} \pi_{\omega,\theta}\left(a \mid s_{t+1}\right) Q_U(s_{t+1}, \omega, a)$$

算法：

Algorithm 1: Option-critic with tabular intra-option Q-learning

$s \leftarrow s_0$
Choose ω according to an ϵ-soft policy over options $\pi_\Omega(s)$
repeat
  Choose a according to $\pi_{\omega,\theta}(a \mid s)$
  Take action a in s, observe s', r

  1. Options evaluation:
  $\delta \leftarrow r - Q_U(s,\omega,a)$
  if s' *is non-terminal* **then**
    $\delta \leftarrow \delta + \gamma(1-\beta_{\omega,\vartheta}(s'))Q_\Omega(s',\omega) + \gamma\beta_{\omega,\vartheta}(s')\max_{\bar{\omega}} Q_\Omega(s',\bar{\omega})$
  end
  $Q_U(s,\omega,a) \leftarrow Q_U(s,\omega,a) + \alpha\delta$

  2. Options improvement:
  $\theta \leftarrow \theta + \alpha_\theta \frac{\partial \log \pi_{\omega,\theta}(a \mid s)}{\partial\theta} Q_U(s,\omega,a)$
  $\vartheta \leftarrow \vartheta - \alpha_\vartheta \frac{\partial \beta_{\omega,\vartheta}(s')}{\partial\vartheta}(Q_\Omega(s',\omega) - V_\Omega(s'))$

  if $\beta_{\omega,\vartheta}$ *terminates in* s' **then**
  choose new ω according to ϵ-soft$(\pi_\Omega(s'))$
  $s \leftarrow s'$
until s' *is terminal*

注意到算法里面只学习了一个critic Q_U，另一个critic可以根据(1)式算出来。

结构如下

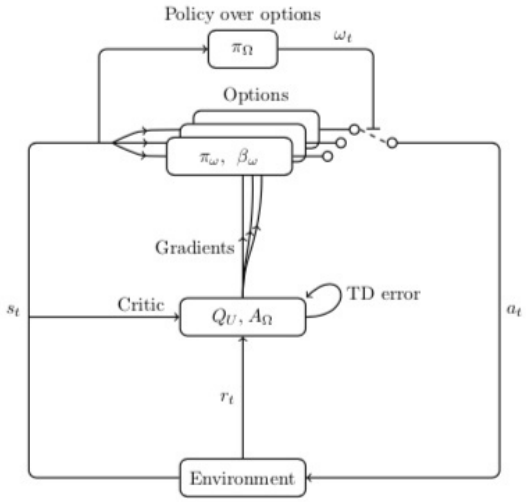

Figure 1: Diagram of the option-critic architecture. The option execution model is depicted by a *switch* $\perp$ over the *contacts* $-\circ$. A new option is selected according to π_Ω only when the current option terminates.

实验结果：

实验主要研究了以下几个问题：

1. 由于使用了option，在外界环境发生变化的时候，使用到的基本的option应该还能使用，只是组合他们的方式（在上层策略）需要进行调整，因此它能比普通方法更快地转移适应新的环境。
2. option的termination function发生的位置是option的交接点，如果学习的比较好，应该发生在子问题的交界处。把termination function热力图打印出来，看看是否满足此假设，如果满足，说明确实学习到了有语义的option。
3. HRL能解决复杂RL问题，一个好的HRL算法应该能更快地解决复杂RL问题。
4. 学习到的option应该能完成一些相对独立的子任务，具有一些较为明确的含义。在Seaquest游戏中，两个option的学习可以观察到一个option完成大致向下的动作、另一个option完成上浮换气的动作。

option常常会收敛到“每个option仅代表一个action”或者“一个option干了所有的事情”的情况，如何避免？

可以通过regularizer来调节。文中遇到了第一个问题，然后他们使用了这样的regularizer $A_\Omega(s,\omega) \to A_\Omega(s,\omega)+\xi = Q_\Omega(s,\omega) - V_\Omega(s) + \xi$ ，当 $\xi > 0$ 的时候实际上奖励了延续当前option的选择，从而避免第一类问题。如果出现第二类问题，应该可以通过 $\xi < 0$ 来调节吧（个人猜想）。

Semi- Markov Decision Process（SMDP）？

分层强化学习里面，这里的上层策略其实是定义在一个SMDP上的，引用一下俞扬老师文章里面的一句话来解释一下。

> 马尔可夫决策过程中，选择一个动作后，agent 会立刻根据状态转移方程 P 跳转到下一个状态，而在半马尔可夫决策过程(SMDP)中，当前状态到下一个状态的步数是个随机变量 τ，即在某个状态 s 下选择一个动作 a 后，经过 τ 步才会以一个概率转移到下一个状态 s'。此时的状态转移概率是 s 和 τ 的联合概率 $P(s',\tau|s,a)$ 。

文献：周文吉, and 俞扬. "分层强化学习综述." *智能系统学报*12.5 (2017): 590-594.

发布于 2018-10-19

机器学习　算法　强化学习 (Reinforcement Learning)

文章被以下专栏收录

强化学习前沿
读呀读paper

进入专栏