Option-critic Architecture

论文链接：www.aaai.org/ocs/index.php/AAAI/AAAI17/paper/download/14858/14328

主要内容

- 分层强化学习自动学习时间抽象是很重要的
- 设计了基于策略梯度理论的option-critic方法来学习内部策略和option的终点约束。We derive policy gradient theorems for options and propose a new option-critic architecture capable of learning both the internal policies and the termination conditions of options
- 在离散和连续的任务中都取得了好的表现。 discrete and continuous environments
- 时间抽象允许表示关于发生在不同时间尺度上的行动过程的知识。Temporal abstraction allows representing knowledge about courses of action that take place at different time scales.
- Discovering temporal abstractions autonomously 成为了一个持续了15年的研究主题
- 大部分的工作致力于发现子目标，并在后续的策略学习中实现它们。这个其实就是分层结构，并且和肌肉骨骼的运动控制很像。The majority of the existing work has focused on finding subgoals (useful states that an agent should reach) and subsequently learning policies to achieve them
- 问题在于求解子目标时的探索时间和计算开销非常大。Additionally, learning policies associated with subgoals can be expensive in terms of data and computation time; in the worst case, it can be as expensive as solving the entire task.
- 本文的方法模糊了发现option和学习option的界限，
- In contrast, we show that our approach is capable of successfully learning options within a single task without incurring any slowdown and while still providing benefits for **transfer learning**.（看着有点像我的Speed-accuracy Trade-off那篇文章，但它的是直接说在抢一学习上有好处的）

- core ideas：the **intra-option policy** and **termination gradient theorems**
- 方法需要的特殊条件少。As opposed to other methods, we only need to specify **the number of desired options**; it is **not** necessary to have **subgoals**, **extra rewards**, **demonstrations**, **multiple problems** or any other **special accommodations** (however, the approach can take advantage of pseudo-reward functions if desired)

Option Framework

- option框架假设对于option $\omega \in \Omega$ 由三元组 $(I_\omega, \pi_\omega, \beta_\omega)$ 组成。
    - $I_\omega \subseteq S$ 是初始的状态集合
    - π_ω 是intra-option策略
    - $\beta_\omega : S \to [0,1]$ 是终止函数
- 当使用option框架时， MDP就变成了Semi-MDP，它有相应的最优值函数$V_\Omega(s)$ 和option-value函数 $Q_\Omega(s,\omega)$ 。所谓的Semi-MDP就是指从状态s到下一个状态s‘要经过 τ 步的MDP，大致就是状态之间存在时间上的不连续。
- 通过一些针对MDP的算法实现多个option并行地学习，这就是 **intra-option learning** 的主要思路
- 接下来就是要实现两个关键任务：learning **option policies** and **termination functions**
- 算法流程：一个外部的策略 π_Ω 选择option来执行控制， intra-option policy π_ω 开始执行，直到终点（用终点函数来判断停止执行）
- 定义option-value function can be written as:

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

其中 $Q_U(s,\omega,a)$ 是在state-option对下执行行为的评估值。

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

注意 (s,ω) 组导致了一个扩张的状态空间 an augmented state space, cf. (Levy and Shimkin 2011)

$U : \Omega \times S \to \mathbb{R}$ 是 the option-value function upon arrival，The value of executing ω upon entering a state s' is given by:

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

如果option ω_t 在时刻 t 被初始化并被执行，此时状态为 s_t ，然后在1步之后，状态转移到 (s_{t+1}, ω_{t+1}) 的概率是：

$$P(s_{t+1}, \omega_{t+1} \mid s_t, \omega_t) = \sum_a \pi_{\omega_t,\theta}(a|s_t)P(s_{t+1} \mid s_t, a)$$
$$((1 - \beta_{\omega_t,\vartheta}(s_{t+1}))1_{\omega_t=\omega_{t+1}} + \beta_{\omega_t,\vartheta}(s_{t+1})\pi_\Omega(\omega_{t+1} \mid s_{t+1}))$$

显然，给出计算过程是均匀的。在温和的条件下，由于选项到处可用，它实际上是遍历的，并且在 state-option 对上存在唯一的平稳分布。

接下来计算期望累积回报关于intra-option策略的参数 θ 的梯度，假设它是随机的可微分的，我们得到

$$\frac{\partial Q_\Omega(s,\omega)}{\partial\theta} = \left(\sum_a \frac{\partial \pi_{\omega,\theta}(a \mid s)}{\partial\theta} Q_U(s,\omega,a)\right)$$
$$+ \sum_a \pi_{\omega,\theta}(a \mid s) \sum_{s'} \gamma P(s'|s,a) \frac{\partial U(\omega, s')}{\partial\theta}$$

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:

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

其中 μ_Ω 是状态-选项对的折扣权重

$$\mu_\Omega(s,\omega \mid s_0,\omega_0) = \sum_{t=0}^{\infty} \gamma^t P(s_t = s, \omega_t = \omega \mid s_0,\omega_0)$$

下面计算终点函数的梯度，假设时间是随机化的并且可微的关于参数 ϑ

$$\frac{\partial Q_\Omega(s,\omega)}{\partial\vartheta} = \sum_a \pi_{\omega,\theta}(a \mid s) \sum_{s'} \gamma P(s' \mid s,a) \frac{\partial U(\omega, s')}{\partial\vartheta}$$

关于 U 的梯度，可以结合又是函数 A_Ω 来计算：

$$\begin{aligned}\frac{\partial U(\omega, s')}{\partial \vartheta} = & -\frac{\partial \beta_{\omega,\vartheta}(s')}{\partial \vartheta} A_\Omega(s', \omega) \\ & + \gamma \sum_{\omega'} \sum_{s''} P(s'', \omega' \mid s', \omega) \frac{\partial U(\omega', s'')}{\partial \vartheta}\end{aligned}$$

其中，优势函数 $A_\Omega(s', \omega) = Q_\Omega(s', \omega) - V_\Omega(s')$.

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:

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

其中 $\mu_\Omega(s', \omega \mid s_1, \omega_0)$ 是状态-选项对 (s_1, ω_0) 的折扣权重

$$\mu_\Omega(s, \omega \mid s_1, \omega_0) = \sum_{t=0}^{\infty} \gamma^t P(s_{t+1} = s, \omega_t = \omega | s_1, \omega_0)$$

- 算法结构：

- 算法伪代码：

- 算法的问题：假设了所有的options适用于所有的地方。Perhaps the biggest remaining limitation of our work is the assumption that all options apply everywhere.

总结

option-critic算法框架是典型的HRL实现方式之一，上下两层都使用策略梯度优化的证明过程看着还是很不错的。具体公式和推导可以参考原文以及附件。相关代码可以在paper with code找出来看看，估计会比FuN的结构的HRL更容易复现吧。