0.1 Derivation

我们的算法是分别对上下层策略进行更新，接下来我们证明，单独更新上层策略和单独更新下层策略都可以使得整体策略变优。

在接下来的推导过程中，我们用上标或下标 h 和 l 来表示强化学习中的一系列参数，是针对上层策略（high）还是下层策略（low）。

首先，我们证明，在下层策略不变的情况下，利用 TRPO 更新上层策略，可以保证整体策略的单调递增，即 $V_h(s_0^h)$ 单调递增。这一步很简单，既然底层策略不变，那么可以认为底层策略是包含在环境中的，因此 TRPO 去优化上层策略，就相当于 TRPO 直接作用于一个单层的强化学习结构，对应的优化目标必然是越来越好的。注意到对于上层策略来说，它得到的 R 就是上下层策略作为一个整体得到的真实 R。

接下来，我们推导，采用 $\frac{\gamma_h V_h(s_{t+k}^h)-V_h(s_t^h)}{k}$ 作为上层给下层的输入，可以使得整体策略变优。

我们的上下层优化算法采用的均是 TRPO[1]。在 TRPO 当中，我们最终希望去优化的东西是 $V_\pi(s_0)$。TRPO 算法等价于在基于策略梯度定理给出的策略梯度优化中，增加了对步长的限制，从而确保策略更新的单调性。在策略梯度方法的论文[2] 中，Sutton 指出，对一个策略优化的目标函数 J 既可以采用 $V_\pi(s_0)$ 的定义，也可以采用 R 的平均值的定义，它们可以分别推导得出策略梯度定理，而这个定理中的策略梯度，与 TRPO 试图优化的替代函数的梯度是相等的。因此，TRPO 也可以理解成是在最大化 $\mathbb{E}_{s,a\sim\pi}[R(s,a)]$。

如果把 R 的均值作为优化目标，我们有

$$\nabla J(\theta) = \nabla\mathbb{E}_{s\sim\pi}[\sum_a \pi(s,a)R(s,a)] = \nabla\mathbb{E}_{s,a\sim\pi}[R(s,a)]. \tag{1}$$

我们定义了

$$R_l(s_{t+i}^l, a_{t+i}^l)|_{i=0,1,\ldots,k-1} = \frac{V_h(s_{t+k}^h) - V_h(s_t^h)}{k}. \tag{2}$$

前面我们提到，TRPO 完成的任务是最大化 $\mathbb{E}_{s,a\sim\pi}[R(s,a)]$，因此我们在利用 TRPO 针对底层策略进行优化的时候，相当于最大化 $\mathbb{E}_{s,a\sim\pi_l,\pi_h}[R_l(s_l,a_l)]$。注意这里，我们把 π_h 看做是一个固定的概率分布函数，而不对它进行优化。π_h 可以看做是环境的一部分，而它输出的隐式编码（latent code）则是我们观察（observation）的一部分。

根据 (2) 中的结果，我们两边取期望，可以得到

$$\mathbb{E}_{s^l,a^l\sim\pi_l,\pi_h}[R_l(s^l,a^l)] = \frac{1}{k}\mathbb{E}_{s^h,a^h\sim\pi_l,\pi_h}[\gamma_h V_h(s^h_{t+k}) - V_h(s^h_t)]. \tag{3}$$

回忆在强化学习中，我们关于优势函数 $A(s,a)$ 的定义，为

$$A(S_t,A_t) = Q(S_t,A_t) - V(S_t) = R(S_t,A_t) + \gamma V(S_{t+1}) - V(s). \tag{4}$$

由于在稀疏强化学习问题中有稀疏奖励条件:

$$R(S_t,A_t) = 0, \forall t \neq t_{end}. \tag{5}$$

因此，(4) 变为

$$A(S_t,A_t) = \gamma V(S_{t+1}) - V(s). \tag{6}$$

由 (3) 和 (6) 可知，TRPO 优化下层策略的结果，等效于优化了上层的优势函数，也就是

$$\mathbb{E}_{s^l,a^l\sim\pi_l,\pi_h}[R_l(s^l,a^l)] = \frac{1}{k}\mathbb{E}_{s^h,a^h\sim\pi_l,\pi_h}[A^h(s^h_t,a^h_t)]. \tag{7}$$

接下来我们证明，通过优化这个期望值，就近似等效于优化了整体策略的表现。这里不能直接把 TRPO 的（13）搬过来用在上层 policy 上！TRPO 是建立在环境动态不变的基础上的，但是我们的环境变了。必须 follow TRPO 论文的推导，看一下我们能不能推出一样的结果。下面做的就是这个事情。

注意到，对于上层策略来说，底层策略相当于环境动力学（dynamics）。当底层策略发生了变化时，可以看做是环境动力学发生了变化。因此，即使上层策略不变，S^h_t 的分布也可能发生改变。我们用 $\mathcal{E}(\pi_l)$ 表示环境，并且环境与 π_l 有关。定义 π_h 带来的折扣奖励期望

$$\begin{gathered}\eta(\pi_h) = \mathbb{E}_{s^h_0,a^h_0,\ldots}\left[\sum_{t=0,k,2k,\ldots}\gamma_h^{t/k} r_h(s^h_t,a^h_t)\right], \text{where} \\ s^h_0\sim\rho^h_0(s^h_0), a^h_t\sim\pi_h(s^h_t), s_{t+1}\sim P(s^h_{t+1}|s^h_t,a^h_t,\mathcal{E}(\pi_l))\end{gathered} \tag{8}$$

我们用 $\tilde{\eta}(\pi_h)$ 来表示 π_l 发生变化（变为 $\tilde{\pi}_l$）而 π_h 没有发生变化以后的折扣奖励值期望。将 TRPO 根据[3] 得到的引理 1（**Lemma 1**）稍加变形，我们可以得

到类似的结果:（具体推导参见附录中证明 1）

$$\tilde{\eta}(\pi_h) = \eta(\pi_h) + \mathbb{E}_{s_0^h, a_0^h, \ldots \sim \pi_h, \mathcal{E}(\tilde{\pi}_l)} \left[\sum_{t=0,k,2k,\ldots} \gamma_h^{t/k} A_{\pi_h}(s_t^h, a_t^h) \right]. \tag{9}$$

等式 (9) 右边第二项的含义就是，新的整体策略相对于旧的整体策略的优势。我们想要最大化的就是这个优势。

接下来，我们定义折扣访问频率 ρ 为

$$\rho_{\pi_h}(s^h) = \sum_{t=0,k,2k,\ldots} \gamma_h^{t/k} P(s_t^h = s|\mathcal{E}(\pi_l)) \tag{10}$$

当底层策略发生改变而上层策略不变的时候，ρ_{π_h} 相应地变为 $\tilde{\rho}_{\pi_h}$。

表达式 (9) 中，等式右侧第二项可以变形为（参见 TRPO 论文中的等式（2）推导，写在附录中证明 2）

$$\sum_{s^h} \tilde{\rho}_{\pi_h}(s^h) \sum_{a^h} \pi_h(a^h|s^h) A_{\pi_h}(s^h, a^h). \tag{11}$$

进一步，由于 $s^h \sim \tilde{\rho}_{\pi_h}(s^h)$ 难以采样，我们采用 $s^h \sim \rho_{\pi_h}(s^h)$ 近似，从而定义出一个需要最大化的函数，为

$$\sum_{s^h} \rho_{\pi_h}(s^h) \sum_{a^h} \pi_h(a^h|s^h) A_{\pi_h}(s^h, a^h). \tag{12}$$

注意到在这里，我们采用 $s^h \sim \rho_{\pi_h}(s^h)$ 来近似 $s^h \sim \tilde{\rho}_{\pi_h}(s^h)$，与 TRPO 采用 $s^h \sim \rho_{\pi_h}(s^h)$ 近似 $s^h \sim \rho_{\tilde{\pi}_h}(s^h)$ 是不同的。是否可以证明 (12) 与表达式 (11) 在 π_h 处的一阶泰勒展开相等? 这里不会证!! 如果这样，只要我们优化 (12) 的步长足够小，就能够确保 (11) 也变大了。（此处还有一个问题：无法证明我们的 step size 可以保证单调性。因为底层的 TRPO 算法做的事情是让 (14) 单调变大，但是并没有限制它能变大多少。也就是说，并没有满足 KL constraint。也就是说，底层的 TRPO 最大化的 objective，不等效于上层的 TRPO 应该最大化的 objective（少了一个 constraint）。这就导致我们无法证明底层的 TRPO 能够让上层的结果也单调递增）。

这个替代函数的形式与 TRPO 中提出的替代函数形式类似，只不过在这里我们是利用旧环境的采样来替代新环境的采样，而 TRPO 是利用旧策略的采样来替代新策略的采样。我们可以用与 TRPO 相同的方法把式 (12) 变为期望的形式，注意到与 $\mathcal{E}(\pi_{\theta^l_{old}})$ 相关其实就是与 $\pi_{\theta^l_{old}}$ 相关，因此我们简化表达式，略去

$\mathcal{E}$，同时我们引入 $\pi(a|s)$ 的参数 θ，得到

$$\mathbb{E}_{s^h\sim\rho_{\pi_h},a^h\sim\pi_h}[\frac{\pi_\theta^h(a^h|s^h)}{\pi_{\theta_{old}}^h(a^h|s^h)}A_{\theta_{old}}^h(s^h,a^h)]. \tag{13}$$

在我们实际的算法中，状态 s 可以直接来自于采样。这是因为，当我们的训练样本足够大，并且选择随机起始时，可以认为 $P(s_t=s)$ 对于不同的 t 是相同的值，因此，训练样本中的 s 也服从 $\rho_{\pi_h}(s^h)$ 的分布。

$$\mathbb{E}_{s^h,a^h\sim\pi_{\theta_{old}^h},\pi_{\theta_{old}^l}}[\frac{\pi_\theta^h(a^h|s^h)}{\pi_{\theta_{old}}^h(a^h|s^h)}A_{\theta_{old}}^h(s^h,a^h)]. \tag{14}$$

注意到，由式 (7)，在 π_h 不变的条件下，我们针对底层策略 π^l 的更新，增大了 $\mathbb{E}_{s^h,a^h\sim\pi_l,\pi_h}[A^h(s_t^h,a_t^h)]$，却没有影响 $\frac{\pi_\theta^h(a^h|s^h)}{\pi_{\theta_{old}}^h(a^h|s^h)}$ 这一项（由于 $\theta^h=\theta_{old}^h$，因此该项等于 1）。这样，针对底层的更新确实使得表达式 (14) 变大了。但是这个表达式并不等于加了 KL-constraint 的替代函数，因此它的变大，并不能保证单调性。但在实际中，我们可以忽略掉这个限制。因此认为，底层策略的优化，就优化了整体策略度量 $\eta(\pi_h)$ 的替代函数，从而优化了整体策略。

参考文献

[1] Schulman J, Levine S, Moritz P, et al. Trust region policy optimization[C]//ICML. [S.l.: s.n.], 2015.

[2] Sutton R S, McAllester D A, Singh S P, et al. Policy gradient methods for reinforcement learning with function approximation[M]//Solla S A, Leen T K, Müller K. Advances in Neural Information Processing Systems 12. [S.l.]: MIT Press, 2000: 1057-1063

[3] Kakade S, Langford J. Approximately optimal approximate reinforcement learning[C]//ICML '02: Proceedings of the Nineteenth International Conference on Machine Learning. [S.l.: s.n.], 2002: 267-274.

证明 1 源自 TRPO 论文的引理 1，在表达式上有区别，但是思路相同。

证明 1:

$$\tilde{\eta}(\pi_h) = \eta(\pi_h) + \mathbb{E}_{s_0^h, a_0^h, \ldots \sim \pi_h, \mathcal{E}(\tilde{\pi}_l)}\left[\sum_{t=0,k,2k,\ldots} \gamma_h^{t/k} A_{\pi_h}(s_t^h, a_t^h)\right]. \tag{15}$$

证:

$$\mathbb{E}_{s_0^h, a_0^h, \ldots \sim \pi_h, \mathcal{E}(\tilde{\pi}_l)}\left[\sum_{t=0,k,2k,\ldots} \gamma_h^{t/k} A_{\pi_h}(s_t^h, a_t^h)\right] \tag{16}$$

$$= \mathbb{E}_{s_0^h, a_0^h, \ldots \sim \pi_h, \mathcal{E}(\tilde{\pi}_l)}\left[\sum_{t=0,k,2k,\ldots} \gamma_h^{t/k} (r_h(s_t^h) + \gamma_h V_{\pi_h}(s_{t+k}^h) - V_{\pi_h}(s_t^h))\right] \tag{17}$$

$$= \mathbb{E}_{s_0^h, a_0^h, \ldots \sim \pi_h, \mathcal{E}(\tilde{\pi}_l)}\left[- V_{\pi_h}(s_0^h) + \sum_{t=0,k,2k,\ldots} \gamma_h^{t/k} r_h(s_t^h)\right] \tag{18}$$

$$= -\eta(\pi_h) + \tilde{\eta}(\pi_h) \tag{19}$$

存疑！这里的推导，前提条件是 t 一直走到无穷。对于我们这种 episode 会结束，并且 sparse reward 的情况，能否这么写??

证明 2:

$$\mathbb{E}_{s_0^h, a_0^h, \ldots \sim \pi_h, \mathcal{E}(\tilde{\pi}_l)}\left[\sum_{t=0,k,2k,\ldots} \gamma_h^{t/k} A_{\pi_h}(s_t^h, a_t^h)\right] = \sum_{s^h} \tilde{\rho}_{\pi_h}(s^h) \sum_{a^h} \pi_h(a^h|s^h) A_{\pi_h}(s^h, a^h). \tag{20}$$

证：

$$\mathbb{E}_{s_0^h, a_0^h, \ldots \sim \pi_h, \mathcal{E}(\tilde{\pi}_l)}\left[\sum_{t=0,k,2k,\ldots} \gamma_h^{t/k} A_{\pi_h}(s_t^h, a_t^h)\right] \tag{21}$$

$$= \sum_{t=0,k,2k,\ldots} \sum_{s^h} P(s_t^h = s^h | \pi_h, \mathcal{E}(\pi_l)) \sum_{a^h} \pi_h(a^h|s^h) \gamma_h^{t/k} A_{\pi_h}(s^h, a^h) \tag{22}$$

$$= \sum_{s^h} \sum_{t=0,k,2k,\ldots} \gamma_h^{t/k} P(s_t^h = s^h | \pi_h, \mathcal{E}(\pi_l)) \sum_{a^h} \pi_h(a^h|s^h) A_{\pi_h}(s^h, a^h) \tag{23}$$

$$= \sum_{s^h} \tilde{\rho}_{\pi_h}(s^h) \sum_{a^h} \pi_h(a^h|s^h) A_{\pi_h}(s^h, a^h). \tag{24}$$