

# Maximum a Posteriori Policy Optimisation

Abdolmaleki, Jost Tobias Springenberg, Yuval Tassa, Remi Munos, Heess, Martin Riedmiller

Google Research, London, UK

{abdolmaleki, springenberg, tassa, munos, heess, riedmiller}@google.

## 【强化学习 83】MPO



张楚珩

清华大学 交叉信息院博士在读

18 人赞同了该文章

Maximum a-posterior Policy Optimization 简称 MPO, ICLR 2018。

### 原文传送门

Abdolmaleki, Abbas, et al. "Maximum a posteriori policy optimisation." arXiv preprint arXiv:1806.06920 (2018).

### 特色

之前就见过一族基于 Expectation Maximization (EM) 算法的强化学习算法，这篇文章也是基于这样分析框架得到的算法；同时，也使用了『RL as inference』的想法，每一步优化一条轨迹上能够得到最大收益的概率。基于此得到了两种 MPO 算法，其中一种是和 TRPO、PPO 等类似，另外一种是一种比较新的算法，后面会详细讲到。文章的跑的实验比较丰富，ablation 也比较详细，可惜作者告诉我暂时没有公开提供代码。

### 过程

#### 1. 算法导出

假设事件  $O=1$  表示『选择的行动使得相应的轨迹收益最大』，并且假设  $p(O=1|\tau) \propto \exp(\sum_t r_t/\alpha)$ ，其中  $\alpha$  是一个温度常数。目标是优化策略，使得该事件发生的概率最大，即最大化以下目标

$$\log p_\pi(O=1) = \log \int p_\pi(\tau) p(O=1|\tau) d\tau \geq \int q(\tau) \left[ \log p(O=1|\tau) + \log \frac{p_\pi(\tau)}{q(\tau)} \right] d\tau \quad (1)$$

$$= \mathbb{E}_q \left[ \sum_t r_t / \alpha \right] - \text{KL}(q(\tau) \| p_\pi(\tau)) = \mathcal{J}(q, \pi), \quad (2)$$

使用一个辅助的分布  $q(\tau)$  来替代真实（但是难以计算得到的）分布  $p_\pi(\tau)$ ，并使用 evidence lower bound (ELBO) 可以得到上述下界。于是 RL 问题变为了一个 inference 问题，这样可以使用 EM

算法来尝试解决这个问题。其中 E-step 通过更新  $q$  来优化  $\mathcal{J}$ ，M-step 通过更新参数化的  $\pi$  来优化  $\mathcal{J}$ 。

上述优化目标是基于轨迹来表示的，为了便于优化，我们一般都希望把它写成基于单步 transition 来表示的。令  $q(\tau) = p(a_0) \prod_{t=0}^{\infty} p(a_{t+1}|s_t, a_t) q(a_t|s_t)$ ，上述优化目标可以写为（注意到复合概率分布的 KL 散度可以写成分量的求和）

$$\mathcal{J}(q, \theta) = \mathbb{E}_q \left[ \sum_{t=0}^{\infty} \gamma^t [r_t - \alpha \text{KL}(q(a|s_t) \parallel \pi(a|s_t, \theta))] \right] + \log p(\theta). \quad (3)$$

同时定义一个 regularized Q function

$$Q_{\theta}^q(s, a) = r_0 + \mathbb{E}_{q(\tau), s_0=s, a_0=a} \left[ \sum_{t=1}^{\infty} \gamma^t [r_t - \alpha \text{KL}(q_t \parallel \pi_t)] \right], \quad (4)$$

with  $\text{KL}(q_t \parallel \pi_t) = \text{KL}(q(a|s_t) \parallel \pi(a|s_t, \theta))$ .

和相应的 augmented reward  $\tilde{r}_t = r_t - \alpha \log \frac{q(a_t|s_t)}{\pi(a_t|s_t, \theta)}$ 。

## 2. E-step

E-step 的目标是通过更新  $q$  来优化  $\mathcal{J}$ ，而在目标函数  $\mathcal{J}$  的表达式中，分布  $q$  控制了状态的轨迹，因此要想对它做优化并不容易。因此，没有办法，只有认为状态的轨迹还是原来的轨迹，只是在每个状态上优化选择不同行动的概率（按照  $q$ ），来做一步优化。即优化以下替代的函数

$$\begin{aligned} \max_q \bar{\mathcal{J}}_s(q, \theta_i) &= \max_q T^{\pi, q} Q_{\theta_i}(s, a) \\ &= \max_q \mathbb{E}_{\mu(s)} \left[ \mathbb{E}_{q(\cdot|s)} [Q_{\theta_i}(s, a)] - \alpha \text{KL}(q \parallel \pi_i) \right], \end{aligned} \quad (6)$$

其中，

$$\text{Bellman operator } T^{\pi, q} = \mathbb{E}_{q(a|s)} \left[ r(s, a) - \alpha \text{KL}(q \parallel \pi_i) + \gamma \mathbb{E}_{p(s'|s, a)} [V_{\theta_i}(s')] \right]$$

因此可以每一个 E-step 中，都选择  $q_i = \arg \max_q \bar{\mathcal{J}}_s(q, \theta_i)$ 。

（注：这里其实有很多可以继续做的事情，比如能不能不只看一步呢？把推理或者 graph 的东西放进来，能够更有效率地做每步的策略更新）

由于惩罚系数  $\alpha$  不好确定，相比之下，KL 散度更 invariant（在不同的情形下）一些，因此写成下面这样的 hard constraint。（类似地，TRPO 也有这么做）

$$\begin{aligned} \max_q \mathbb{E}_{\mu(s)} \left[ \mathbb{E}_{q(a|s)} [Q_{\theta_i}(s, a)] \right] \\ \text{s.t. } \mathbb{E}_{\mu(s)} \left[ \text{KL}(q(a|s), \pi(a|s, \theta_i)) \right] < \epsilon. \end{aligned} \quad (7)$$

如何具体地去解这个优化问题，有以下两种途径：

- Use parametric variational distribution: 认为  $q$  是参数化的模型，这样就直接优化上面的目标，得到新的  $q$ ；观察到  $q$  其实就可以作为新的策略，这样就不需要再做 M-step 了，因为 M-step 就是最小化分布  $q$  和策略  $\pi$  的 KL 散度，我们直接把策略  $\pi$  设置为分布  $q$  即可。TRPO、PPO 其实就是基于上式来做优化的。
- Use non-parametric representation: 用 non-parametric 的方式来表示  $q(a|s)$ ，即在采集的样本上把它们的数值表示出来；这样，还需要一个另外的 M-step 来得到一个关于策略的可泛化的表示。这种方法相比于 parametric 的表示方法来说，是比较创新的。下面主要就讲这种方法如何实现。

优化问题 (7) 有闭式解：

$$q_i(a|s) \propto \pi(a|s, \theta_i) \exp\left(\frac{Q_{\theta_i}(s, a)}{\eta^*}\right), \quad (8)$$

where we can obtain  $\eta^*$  by minimising the following convex dual function,

$$g(\eta) = \eta \epsilon + \eta \int \mu(s) \log \int \pi(a|s, \theta_i) \exp\left(\frac{Q_{\theta_i}(s, a)}{\eta}\right) da ds \quad \text{知乎 @张楚琦}$$

实际的算法中可以对于每一个 state sample 都采集若干个 action，然后先计算 (9) 的极值点，然后再利用 (8) 计算每一个 state-action sample 上的  $q$ 。接下来，根据这些 state-action sample 上的  $q$  通过有监督学习得到一个可泛化的策略网络，即 M-step。

### 3. M-step

回顾 E-step 的目标是  $q_i = \arg \max_q \mathcal{J}(q, \theta_i)$ ，相应地 M-step 的目标是  $\theta_{i+1} = \arg \max_{\theta} \mathcal{J}(q_i, \theta)$ ，即：

$$\max_{\theta} \mathcal{J}(q_i, \theta) = \max_{\theta} \mathbb{E}_{\mu_{q_i}(s)} \left[ \mathbb{E}_{q_i(a|s)} \left[ \log \pi(a|s, \theta) \right] \right] + \log p(\theta), \quad (10)$$

选择一个高斯先验

$p(\theta) \approx \mathcal{N}\left(\mu = \theta_i, \Sigma = \frac{F_{\theta_i}}{\lambda}\right)$ , where  $\theta_i$  are the parameters of the current policy distribution,  $F_{\theta_i}$  is the empirical Fisher information matrix and  $\lambda$  is a positive scalar.

上述优化问题可以变为：

$$\max_{\pi} \mathbb{E}_{\mu_{q_i}(s)} \left[ \mathbb{E}_{q_i(a|s)} \left[ \log \pi(a|s, \theta) \right] - \lambda \text{KL}\left(\pi(a|s, \theta_i), \pi(a|s, \theta)\right) \right] \quad (11)$$

同样地，还是使用 hard constraint 然后迭代地优化拉格朗日乘子和参数。

### 4. Policy evaluation

可以看到，本文中的 policy improvement step 完全依赖于 Q 函数给出的信息，因此 Q 函数估计的是否准确直接影响到算法是否成功。由于本文要做 off-policy 的算法提高 sample efficiency，因此对于 Q 函数的估计使用 Retrace 来估计（本专栏之前有讲过）：

$$\min_{\phi} L(\phi) = \min_{\phi} \mathbb{E}_{\mu_b(s), b(a|s)} \left[ (Q_{\theta_i}(s_t, a_t, \phi) - Q_t^{\text{ret}})^2 \right], \text{ with}$$

$$Q_t^{\text{ret}} = Q_{\phi'}(s_t, a_t) + \sum_{j=t}^{\infty} \gamma^{j-t} \left( \prod_{k=t+1}^j c_k \right) \left[ r(s_j, a_j) + \mathbb{E}_{\pi(a|s_{j+1})} [Q_{\phi'}(s_{j+1}, a)] - Q_{\phi'}(s_j, a_j) \right],$$

$$c_k = \min \left( 1, \frac{\pi(a_k|s_k)}{b(a_k|s_k)} \right),$$

知乎 @张楚珩

发布于 2019-08-03

强化学习 (Reinforcement Learning)

赞同 18



添加评论

分享

喜欢

收藏



文章被以下专栏收录



强化学习前沿  
读呀读paper

进入专栏