Entropy Regularization with Discounted Future State Distribution in Policy Gradient Methods

Riashat Islam
McGill University, Mila
School of Computer Science
riashat.islam@mail.mcgill.ca

Raihan Seraj
McGill Univeristy
raihan.seraj@mail.mcgill.ca

Pierre-Luc Bacon
Stanford Univeristy
plbacon@cs.stanford.edu

Doina Precup
McGill University, Mila
School of Computer Science
dprecup@cs.mcgill.ca

【强化学习 114】State Entropy Regularization

张楚珩
清华大学 交叉信息院博士在读

28 人赞同了该文章

原文传送门

Islam, Riashat, et al. "Entropy Regularization with Discounted Future State Distribution in Policy Gradient Methods." arXiv preprint arXiv:1912.05104 (2019).

特色

强化学习的主要问题是状态空间的探索，策略梯度方法要求在有一个较好的 restart distribution 的基础上才能够表现较好。

这篇文章介绍一种估计 discounted future state distribution 的方法，并且在此基础上做 entropy regularization，从而鼓励状态空间的探索。基于 three time-scale algorithm 证明算法可以收敛到局部最优。在实验上证明该正则项可以帮助更快地覆盖整个状态空间，并且在一些复杂任务上表现更好。

贡献

- 提供一种可操作的估计 discounted state distribution 的方法；
- 在策略梯度更新中加上 state space entropy 的 maximization；
- 证明了收敛到局部最优解；
- 实验上说明加上 state space entropy regularization 的好处；

过程

1、背景

之前的工作有很多基于 maximize entropy 的，但是一般都是 action conditioned on state 的。这里想直接 maximize state space entropy，这个想法是 [Hazan et al 2018] (ICML 2019) 提出的，这是一

篇理论的工作，大致上是把 state space entropy maximization 的目标转化为一个 reward function，然后利用 sota 的方法（SAC）去求解。这篇工作和它稍有区别，这里是想估计一个 discounted future state distribution，然后在学习的基础上加一个探索的 bonus/regularization。

相关的有很多方法也在原本奖励函数的基础上加入 pseudo-reward，以鼓励探索，包括专栏前面讲过的一些方法和 curiosity-driven [Pathak et al 2017], count-based exploration [Bellemare et al 2016] 和 count-based with neural density models [Ostrovski et al 2017]。

2、简介

优化目标

$$\tilde{J}(\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{\infty} \gamma^t r(S_t, A_t) \,\middle|\, S_0\right] + \lambda \mathbb{H}(d_{\pi_\theta}),$$

其中 $d_{\pi_\theta} = d_{\alpha,\gamma,\pi_\theta}$ 代表从初始状态分布 alpha 出发，以 gamma 的 discount rate 衰减，遵循某个被 theta 参数化的策略，得到的 discounted state distribution。

2、估计 state distribution

首先，这里区分一下 stationary state distribution 和 discounted state distribution，具体定义会在后面写出来。这里的目标是想让智能体在状态空间上分布更加均匀，因此不是特别关心状态是先被访问到还是后被访问到；因此，最合适的应该是选择最大化 stationary state distribution 对应的分布。不过有很多情况下，stationary state distribution 并不存在或者并不良好，比如周期性的 MDP 中就不存在，而在 episodic environment 中，如果非要计算 stationary state distribution，那么这个分布会全部集中到终态（吸收态）上。

discounted state distribution

discounted state distribution 的定义如下

$$d_{\alpha,\gamma,\pi_\theta}(s) = (1-\gamma)\alpha^T \sum_{t=0}^{\infty} \gamma^t P(S_t = s), \quad \forall s \in \mathcal{S} \tag{4}$$

上述定义是基于 horizon = infinite 来定义的，近似地用有限的 horizon = T 的样本来估计它，估计方法如下

$$\tilde{p}(s) \overset{(a)}{=} \frac{(1-\gamma)}{T} \sum_{t=0}^{T} \gamma^t \mathbb{1}(S_t = s) \overset{(b)}{=} (1-\gamma) \sum_{t=0}^{T} (\gamma^t P(S_t = s \mid S_0)) \overset{(c)}{\approx} d_{\gamma,\pi_\theta}(s), \tag{5}$$

中间有一个因为有限位置截断之后产生的近似误差。接下来可以写出 entropy，注意到前面那一坨相当于 $\mathbb{E}_{S_t \sim p(\cdot)}$ 。

$$\mathbb{H}(d_{\alpha,\gamma,\pi_\theta}) \approx -\frac{1}{T} \sum_{t=0}^{T} \log \tilde{p}(S_t).$$

知乎 @张楚珩

stationary state distribution

stationary state distribution 是根据在相应策略下的 state transition matrix P_{π_θ} 来定义的

$$d_{1,\pi_\theta} = P_{\pi_\theta}^{\intercal} d_{1,\pi_\theta}, \tag{7}$$

其对应的状态空间上的熵可以写作

$$\mathbb{H}(d_{1,\pi_\theta}) \overset{(a)}{=} -\sum_{s\in\mathcal{S}} d_{1,\pi_\theta}(s)\log(d_{1,\pi_\theta}(s)) \overset{(b)}{\approx} -\frac{1}{T}\sum_{t=0}^{T}\log d_{1,\pi_\theta}(S_t) \overset{(c)}{=} -\frac{1}{T}\sum_{t=0}^{T}\log p(S_t), \quad (8)$$

估计 entropy regularization

加上 entropy regularization 之后的策略梯度算法可以写成如下形式（考虑 discounted case）

$$\tilde{J}(\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{\infty}\gamma^t r(S_t, A_t) - \lambda\log d_{\alpha,\gamma,\pi_\theta}(s_t)\ \middle|\ S_0\right] \quad (9)$$

下面就要估计对于任意策略 theta 对应的上述 log 项。

因为 state distribution 是依赖策略的，因此需要训练一个接受策略神经网络参数 theta 作为输入的 state density estimator 。这里使用 VAE 来建模，其中 encoder q 输入为策略网络的参数 theta，输出为一个隐变量 z 的分布；decoder p 的输入为一个隐变量 z，输出为状态 s 的分布；p(z) 为固定的 decoder 的先验分布，一般为一个标准正态分布。VAE 的网络参数为 phi。目标函数为

$$\mathcal{L}_\gamma(\phi,\theta) = (1-\gamma)\gamma^k\mathbb{E}_{q_\phi(Z|\theta)}\left[\log p_\phi(S|\theta)\right] - KL(q_\phi(Z|\theta)||p(\theta)) \quad (10)$$

文章似乎打印错了

3、策略梯度上升

策略网络的更新公式如下

$$\nabla_\theta\tilde{J}(\theta) = \mathbb{E}_{\pi_\theta}\left[\nabla_\theta\log\pi(A_t \mid S_t)Q^\pi(S_t,A_t) - \lambda\nabla_\theta\mathcal{L}_\gamma(\phi,\theta).\right], \text{where } \mathcal{L}_\gamma(\phi,\theta) = (1-\gamma)\gamma^t\mathcal{L}(\phi,\theta) \quad (11)$$

最后的算法如下所示

Algorithm 1: Entropy regularization with $\mathbb{H}(\hat{d_\pi})$

Require: A policy π_θ and critic $Q_\psi(s,a)$
Require: A density estimator $p_\phi(s)$ and regularization weight λ
for episodes = 1 to E **do**
  Take action a_t,get reward r_t and observe next state s_{t+1}
  Store tuple $(s_t, a_t, r(s_{t+1}), s_{t+1},)$ in $\mathcal{D}$
  if *mod(t,N)* **then**
   Update critic parameters ψ as policy evaluation
   Update density estimator ϕ to estimate $\log d_\pi(s)$ or $\log d_{\gamma,\pi}(s)$ by maximizing variational lower bound $\mathcal{L}(\phi,\theta)$
   Update policy parameters θ folowing *any* policy gradient method according to
   $\nabla_\theta\tilde{J}(\theta) = \left[\nabla_\theta\log\pi_\theta(A_t|S_t)\,Q^\psi(A_t,S_t) - \lambda\nabla_\theta\mathcal{L}_\gamma(\phi,\theta)\right]$
end for

知乎 @张楚珩

4、实验

本文做了格子世界和 Mujoco 上的实验，说明在已有的策略梯度算法上加上这一项 regularization 的提升，特别是对于 sparse reward 任务上的提升。

发布于 2020-03-28

强化学习 (Reinforcement Learning) 机器学习 算法

▲ 赞同 28 ▼ 1 条评论 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