Chap 6 SOTA method

2022年2月2日 16:58

-
- 两个 学派

  Two lines of works on policy optimization:

  - ❶ Policy Gradient→Natural Policy Gradient/TRPO→ACKTR→PPO

    ❷ Q-learning→DDPG→TD3→SAC
- 第一派：policy based
  - TRPO
    - 费舍尔信息矩阵
      - Fisher Information Matrix - Agustinus Kristiadi's Blog：主要证明了 费舍尔矩阵 可用于估计 黑塞矩阵。而黑塞矩阵常用于多元泰勒展开函数近似，这是计算TRPO参数更新的基础
      - 费舍尔信息矩阵 定义
        - 首先在极大似然估计原理中，假设有一个概率模型p(x|θ)（θ是模型参数），假设有大量采样xi，则要最大化p(xi|θ)，一般是对这个函数求关于θ的极值，假设函数连续，极值处梯度一定为0
        - 因此定义评估函数s(θ)。θ是向量，s也是向量

          $$s(\theta) = \nabla_\theta \log p(x|\theta) \, ,$$
        - 区分两个概率：客观概率，和我们估计的主观概率模型（p(x|θ)）
        - 本文章中，当使用主观概率采样时，s(θ)的期望是0

          $$\begin{aligned} \mathop{\mathbb{E}}_{p(x|\theta)}[s(\theta)] &= \mathop{\mathbb{E}}_{p(x|\theta)}[\nabla \log p(x|\theta)] \\ &= \int \nabla \log p(x|\theta) \, p(x|\theta) \, \mathrm{d}x \\ &= \int \frac{\nabla p(x|\theta)}{p(x|\theta)} p(x|\theta) \, \mathrm{d}x \\ &= \int \nabla p(x|\theta) \, \mathrm{d}x \\ &= \nabla \int p(x|\theta) \, \mathrm{d}x \\ &= \nabla 1 \\ &= 0 \end{aligned}$$

          而概率统计中学习的 极大似然估计法，是通过真实客观概率采样，当s(θ)的期望是0时，θ就是真实参数的估计
        - 费舍尔信息矩阵 为s(θ) 各分量 的协方差

          $$\mathop{\mathbb{E}}_{p(x|\theta)}\left[(s(\theta) - 0) \, (s(\theta) - 0)^{\mathrm{T}}\right] .$$

$$\mathbf{F} = \mathop{\mathbb{E}}_{p(x|\theta)} \left[\nabla \log p(x|\theta) \, \nabla \log p(x|\theta)^{\mathrm{T}} \right] .$$

- 费舍尔矩阵等于负的黑塞矩阵的期望
  - 黑塞矩阵分解

Proof. The Hessian of the log likelihood is given by the Jacobian of its gradient:

$$\begin{aligned}
\mathrm{H}_{\log p(x|\theta)} &= \mathrm{J}\left(\frac{\nabla p(x|\theta)}{p(x|\theta)}\right) \\
&= \frac{\mathrm{H}_{p(x|\theta)}\, p(x|\theta) - \nabla p(x|\theta)\, \nabla p(x|\theta)^{\mathrm{T}}}{p(x|\theta)\, p(x|\theta)} \\
&= \frac{\mathrm{H}_{p(x|\theta)}\, p(x|\theta)}{p(x|\theta)\, p(x|\theta)} - \frac{\nabla p(x|\theta)\, \nabla p(x|\theta)^{\mathrm{T}}}{p(x|\theta)\, p(x|\theta)} \\
&= \frac{\mathrm{H}_{p(x|\theta)}}{p(x|\theta)} - \left(\frac{\nabla p(x|\theta)}{p(x|\theta)}\right)\left(\frac{\nabla p(x|\theta)}{p(x|\theta)}\right)^{\mathrm{T}},
\end{aligned}$$

  - $$\begin{aligned}
\mathop{\mathbb{E}}_{p(x|\theta)}\left[\mathrm{H}_{\log p(x|\theta)}\right] &= \mathop{\mathbb{E}}_{p(x|\theta)}\left[\frac{\mathrm{H}_{p(x|\theta)}}{p(x|\theta)} - \left(\frac{\nabla p(x|\theta)}{p(x|\theta)}\right)\left(\frac{\nabla p(x|\theta)}{p(x|\theta)}\right)^{\mathrm{T}}\right] \\
&= \mathop{\mathbb{E}}_{p(x|\theta)}\left[\frac{\mathrm{H}_{p(x|\theta)}}{p(x|\theta)}\right] - \mathop{\mathbb{E}}_{p(x|\theta)}\left[\left(\frac{\nabla p(x|\theta)}{p(x|\theta)}\right)\left(\frac{\nabla p(x|\theta)}{p(x|\theta)}\right)^{\mathrm{T}}\right] \\
&= \int \frac{\mathrm{H}_{p(x|\theta)}}{p(x|\theta)} p(x|\theta)\, \mathrm{d}x - \mathop{\mathbb{E}}_{p(x|\theta)}\left[\nabla \log p(x|\theta)\, \nabla \log p(x|\theta)^{\mathrm{T}}\right] \\
&= \mathrm{H}_{\int p(x|\theta)\, \mathrm{d}x} - \mathbf{F} \\
&= \mathrm{H}_1 - \mathbf{F} \\
&= -\mathbf{F}.
\end{aligned}$$

- 自然梯度下降 Natural Gradient Descent
  - [Natural Gradient Descent - Agustinus Kristiadi's Blog]
  - Parameter Space versus Distribution Space
    - 一般梯度下降通过 欧氏范数 控制梯度下降步长。

$$\frac{-\nabla_\theta \mathcal{L}(\theta)}{\|\nabla_\theta \mathcal{L}(\theta)\|} = \lim_{\epsilon \to 0} \frac{1}{\epsilon} \mathop{\arg\min}_{d \text{ s.t. } \|d\| \leq \epsilon} \mathcal{L}(\theta + d) .$$

    - 自然梯度下降，根据更新前后概率模型分布的差别控制步长，具体是使用 KL散度
- Importance Sampling
  - we can sample data from another distribution q and use the probability ratio between p and q to re-calibrate the result

$$\mathbb{E}_{x\sim p}[f(x)] = \int p(x)f(x)dx = \int q(x)\frac{p(x)}{q(x)}f(x)dx = \mathbb{E}_{x\sim q}[\frac{p(x)}{q(x)}f(x)]$$

  - 用在策略函数：

$$J(\theta) = \mathbb{E}_{a\sim\pi_\theta}[r(s,a)] = \mathbb{E}_{a\sim\hat{\pi}}[\frac{\pi_\theta(s,a)}{\hat{\pi}(s,a)}r(s,a)]$$

- **Trust Region Policy Optimization（TRPO）**
  - 结合了 IS 和 Natural Gradient
  - $$J_{\theta_{old}}(\theta) = \mathbb{E}_t\left[\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}R_t\right]$$
    subject to $KL(\pi_{\theta_{old}}(.|s_t)||\pi_\theta(.|s_t)) \leq \delta$
  - 估计，使用多元函数的泰勒展开，并化简，
    $$J_{\theta_t}(\theta) \approx g^T(\theta - \theta_t)$$
    $$KL(\theta_t||\theta) \approx \frac{1}{2}(\theta - \theta_t)^T H(\theta - \theta_t)$$
    where $g = \nabla_\theta J_{\theta_t}(\theta)$ and $H = \nabla_\theta^2 KL(\theta_t||\theta)$ and θ_t is the old policy parameter
  - 根据估计，写成
    $$\theta_{t+1} = \arg\max_\theta g^T(\theta - \theta_t) \text{ s.t. } \frac{1}{2}(\theta - \theta_t)^T H(\theta - \theta_t) \leq \delta$$
  - 解出参数更新的解析解
    $$\theta_{t+1} = \theta_t + \sqrt{\frac{2\delta}{g^T H^{-1} g}} H^{-1} g$$
    ？怎么算出来的
  - 自然梯度

    Algorithm 1 Natural Policy Gradient

    Input: initial policy parameters θ_0
    for $k = 0, 1, 2, \ldots$ **do**
    Collect set of trajectories $\mathcal{D}_k$ on policy $\pi_k = \pi(\theta_k)$
    Estimate advantages $\hat{A}_t^{\pi_k}$ using any advantage estimation algorithm
    Form sample estimates for
    - policy gradient $\hat{g}_k$ (using advantage estimates)
    - and KL-divergence Hessian / Fisher Information Matrix $\hat{H}_k$

    Compute Natural Policy Gradient update:
    $$\theta_{k+1} = \theta_k + \sqrt{\frac{2\delta}{\hat{g}_k^T \hat{H}_k^{-1} \hat{g}_k}} \hat{H}_k^{-1} \hat{g}_k$$
    end for

    [1] Sham Kakade. “A Natural Policy Gradient.” NIPS 2001
  - 计算H的逆，使用Conjugate Gradient迭代

1. FIM and its inverse are very expensive to compute
2. TRPO estimates the term $x = H^{-1}g$ by solving the following linear equation $Hx = g$
3. Consider the optimization for a quadratic equation

$$\text{Solving } Ax = b \text{ is equivalent to}$$
$$x = \arg\max_x f(x) = \frac{1}{2}x^T Ax - b^T x$$
$$\text{since } f'(x) = Ax - b = 0$$

4. Thus we can optimize the quadratic equation as

$$\min_x \frac{1}{2}x^T Hx - g^T x$$

5. Use conjugate gradient method to solve it. It is very similar to the gradient ascent but can be done in fewer iterations

- TRPO总算法

1. Resulting algorithm is a refined version of natural policy gradient

Algorithm 3 Trust Region Policy Optimization

Input: initial policy parameters θ_0
for $k = 0, 1, 2, \ldots$ **do**
  Collect set of trajectories $\mathcal{D}_k$ on policy $\pi_k = \pi(\theta_k)$
  Estimate advantages $\hat{A}_t^{\pi_k}$ using any advantage estimation algorithm
  Form sample estimates for

- policy gradient $\hat{g}_k$ (using advantage estimates)
- and KL-divergence Hessian-vector product function $f(v) = \hat{H}_k v$

  Use CG with n_{cg} iterations to obtain $x_k \approx \hat{H}_k^{-1}\hat{g}_k$
  Estimate proposed step $\Delta_k \approx \sqrt{\frac{2\delta}{x_k^T \hat{H}_k x_k}} x_k$
  Perform backtracking line search with exponential decay to obtain final update

$$\theta_{k+1} = \theta_k + \alpha^j \Delta_k$$

end for