零阶优化/强化学习基础

Introduction to Zeroth-Order Optimization and Reinforcement Learning

唐聿劼

Yujie Tang

北京大学工学院工业工程与管理系

Department of Industrial Engineering and Management, Peking University

- 梯度下降与随机梯度下降

  Gradient Descent & Stochastic Gradient Descent

- 零阶优化

  Zeroth-Order Optimization

- 强化学习

  Reinforcement Learning

- 梯度下降与随机梯度下降

  Gradient Descent & Stochastic Gradient Descent

- 零阶优化

  Zeroth-Order Optimization

- 强化学习

  Reinforcement Learning

梯度下降 Gradient Descent

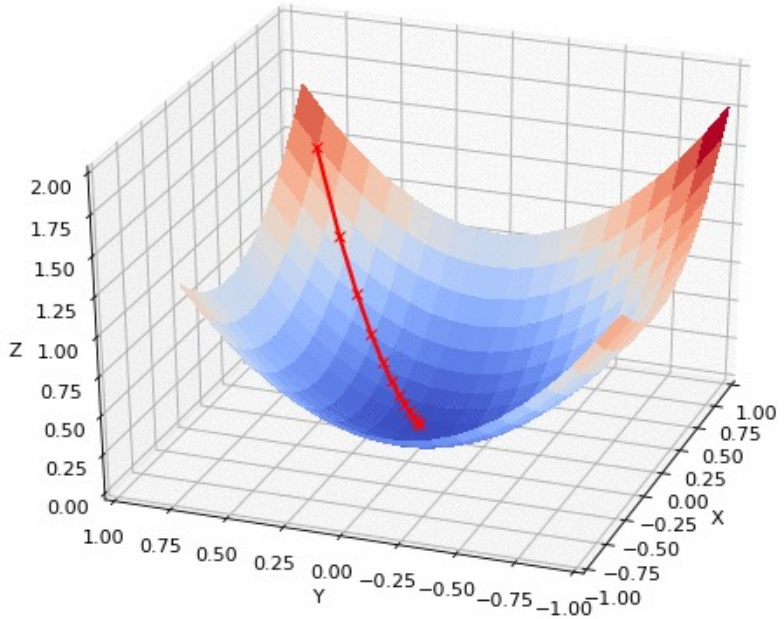
$$\min_{x \in \mathbb{R}^n} \quad f(x)$$

梯度下降：$x_{k+1} = x_k - \eta_k \nabla f(x_k)$

- $-\nabla f(x)$ 是局域上看函数值下降最快的方向
- 当 f 可导时，只要梯度不为零，那么取步长足够小，即有

$$\begin{aligned} f(x_{k+1}) &= f(x_k - \eta_k \nabla f(x_k)) \\ &= f(x_k) - \eta_k \|\nabla f(x_k)\|^2 + o(\eta_k) < f(x_k) \end{aligned}$$

- 单调有界收敛定理 → 只要 f 有下界，则 $f(x_k)$ 收敛



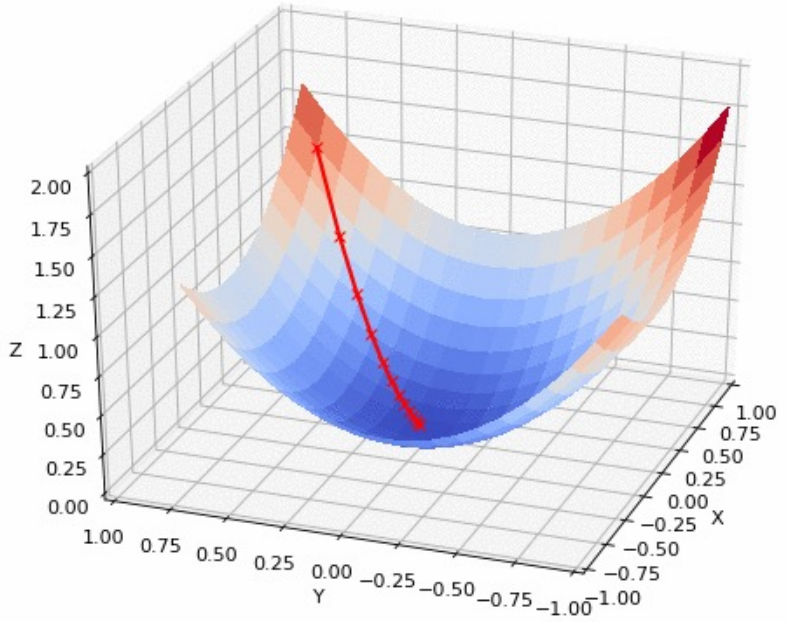
Source: machine-learning.paperspace.com/wiki/gradient-descent

梯度下降 Gradient Descent

$$\min_{x \in \mathbb{R}^n} \quad f(x)$$

梯度下降：$x_{k+1} = x_k - \eta_k \nabla f(x_k)$

- 更强的收敛性保证：
  - x_k 的梯度是否收敛于 0、收敛速度
  - $f(x_k)$ 是否收敛于 f 的最小值（下确界）、收敛速度
  - x_k 是否收敛到最优解（集）、收敛速度
- 更强的收敛性保证需要对 f 施加更强的条件



Source: machine-learning.paperspace.com/wiki/gradient-descent

设 f 在 $\mathbb{R}^n$ 上连续可微.

- 凸函数 Convex function

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle$$

- μ-强凸函数 μ-strongly convex function

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\mu}{2}\|y - x\|^2$$

$$\langle \nabla f(y) - \nabla f(x), y - x \rangle \geq \mu \|x - y\|^2$$

- L-光滑函数 L-smooth function

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|$$

$$\Longrightarrow |f(y) - f(x) - \langle \nabla f(x), y - x \rangle| \leq \frac{L}{2}\|y - x\|^2$$

梯度下降 Gradient Descent

$$\min_{x\in\mathbb{R}^n} \quad f(x) \qquad\qquad x_{k+1} = x_k - \eta_k \nabla f(x_k)$$

定理. 假设 f 是 L-光滑的. 取 $\eta_k = \eta \in (0, 2/L)$，则有

$$\min_{0\le j\le k-1} \|\nabla f(x_j)\|^2 \le \frac{f(x_0) - f^*}{\eta(1-\eta L/2)} \cdot \frac{1}{k} = O\left(\frac{1}{k}\right)$$

若 f 还是凸的且存在最小值点，则

$$f(x_k) - f^* \le \frac{(f(x_0) - f^*)\|x_0 - x^*\|^2}{\|x_0 - x^*\|^2 + \eta k(1-\eta L/2)(f(x_0) - f^*)} = O\left(\frac{1}{k}\right)$$

若 f 还是 μ-强凸的，则

$$\|x_k - x^*\|^2 \le \left(1 - \frac{2\eta\mu L}{\mu + L}\right)^k \|x_0 - x^*\|^2 = O\left(\left(1 - \frac{2\eta\mu L}{\mu + L}\right)^k\right)$$

$$\min_{x \in \mathbb{R}^n} \quad f(x)$$

- What if we can only obtain noisy versions of the gradients?
- Example: Stochastic programming

$$\min_{x \in \mathbb{R}^n} \quad \mathbb{E}_{\xi \sim \rho}[F(x; \xi)]$$

  We can only sample $\xi \sim \rho$ and compute $\nabla_x F(x; \xi)$

- A more specific example: Supervised learning

$$\min_{\theta \in \mathbb{R}^n} \quad \mathbb{E}_{(x,y) \sim \mathcal{P}}[L(\theta; x, y)]$$

  We don't know $\mathcal{P}$ but we can obtain samples $(x, y) \sim \mathcal{P}$

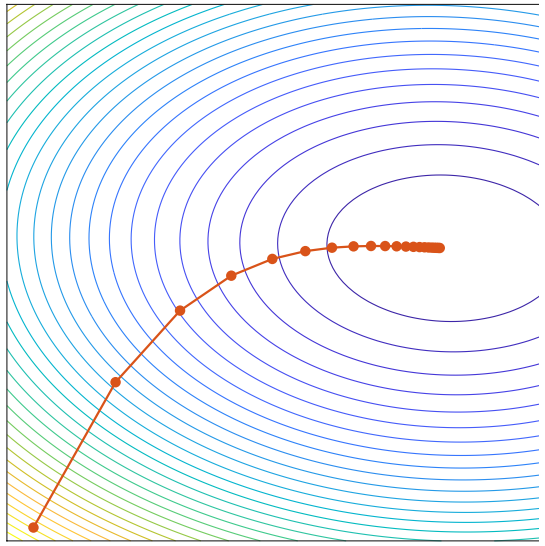
随机梯度下降 Stochastic Gradient Descent

$$\min_{x \in \mathbb{R}^n} \quad f(x)$$

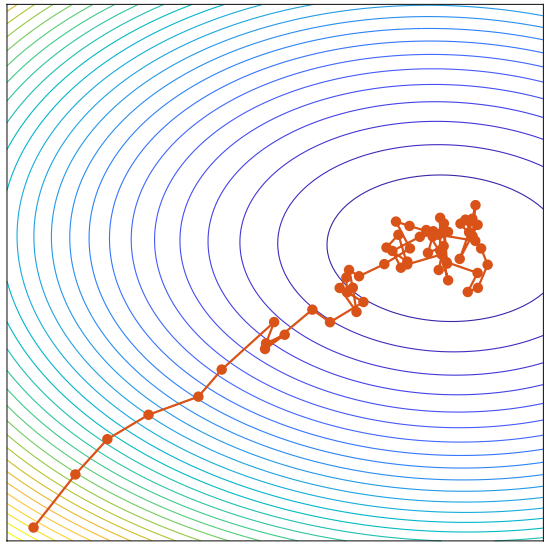
随机梯度下降: $x_{k+1} = x_k - \eta_k \cdot g_k$

- 通常假设 $\mathbb{E}[\, g_k \mid x_{0:k} \,] = \nabla f(x_k)$ (unbiased) $\quad \mathbb{E}\left[\|g_k\|^2 \mid x_{0:k}\right] \leq \sigma^2 + \beta \|\nabla f(x_k)\|^2$ (bounded second moment)

GD



SGD



随机梯度下降 Stochastic Gradient Descent

$$\min_{x\in\mathbb{R}^n} \quad f(x)$$

随机梯度下降: $x_{k+1} = x_k - \eta_k \cdot g_k$

- 通常假设 $\mathbb{E}[\, g_k \mid x_{0:k}\,] = \nabla f(x_k)$ $\mathbb{E}\big[\|g_k\|^2 \mid x_{0:k}\big] \le \sigma^2 + \beta\|\nabla f(x_k)\|^2$
- 收敛性： 设 f 是 L-光滑的. 假设 SGD 共迭代 K 步，取 $\eta_k = \eta = O(1/\sqrt{K})$ ，则

$$\min_{0\le k\le K-1} \mathbb{E}\big[\|\nabla f(x_k)\|^2\big] \le O\left(\frac{1}{\sqrt{K}}\right)$$

若 f 还是凸的，则

$$\mathbb{E}[f(\bar{x}_K) - f^*] \le O\left(\frac{1}{\sqrt{K}}\right)$$

- 梯度下降与随机梯度下降

  Gradient Descent & Stochastic Gradient Descent

- 零阶优化

  Zeroth-Order Optimization

- 强化学习

  Reinforcement Learning

$$\min_{x \in \mathbb{R}^n} \quad f(x)$$

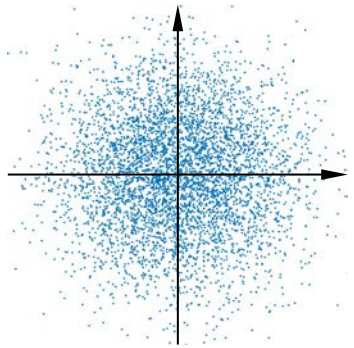
- 如果只能获取 f 在有限个点的取值，无法获取 f 的梯度，该怎么办？
- 无梯度（gradient-free）优化方法
  - 元启发算法（metaheuristics）：模拟退火、遗传算法、……
  - **零阶优化（zeroth-order optimization）**

    基本思想：在 x 附近随机探索 f 的取值，进而构造 f 在 x 处的一个随机梯度

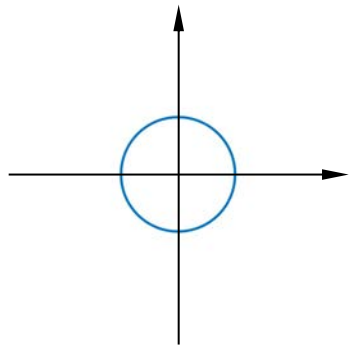
- 令 λ 为 $\mathbb{R}^n$ 上一满足如下性质的概率分布：

  1. 具有旋转与反射不变性
  2. $\mathbb{E}_{z\sim\lambda}\left[\|z\|^2\right]=1$

Examples:



$\mathcal{N}(0,n^{-1}I)$



$\mathrm{Unif}(\mathbb{S}_{n-1})$

$\mathbb{S}_{n-1}=\{x\in\mathbb{R}^n:\|x\|=1\}$

- 我们有 $\mathbb{E}_{z\sim\lambda}\left[zz^\top\right]=\dfrac{1}{n}I$

  从而

$$n\,\mathbb{E}_{z\sim\lambda}[\langle\nabla f(x),z\rangle z]=n\,\mathbb{E}_{z\sim\lambda}\left[zz^\top\nabla f(x)\right]=\nabla f(x)$$

  注意到当 r 足够小时，有

$$\langle\nabla f(x),z\rangle\approx\frac{f(x+rz)-f(x)}{r}\approx\frac{f(x+rz)-f(x-rz)}{2r}$$

  故

$$\nabla f(x)\approx\mathbb{E}_{z\sim\lambda}\left[\frac{n}{r}(f(x+rz)-f(x))z\right]\approx\mathbb{E}_{z\sim\lambda}\left[\frac{n}{2r}(f(x+rz)-f(x-rz))z\right]$$

零阶梯度估计

- 双点梯度估计器 Two-point gradient estimator

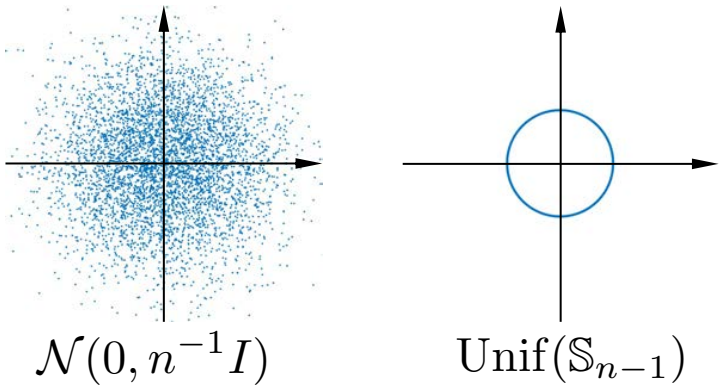
$$\mathsf{G}_f^{(2)}(x; r, z) = \frac{n}{r}(f(x + rz) - f(x))z$$

$$\tilde{\mathsf{G}}_f^{(2)}(x; r, z) = \frac{n}{2r}(f(x + rz) - f(x - rz))z$$

n - 自变量维数 dimension of x

r - 平滑半径 smoothing radius

z - 随机扰动 random perturbation



零阶梯度估计

- 双点梯度估计器 Two-point gradient estimator

$$\mathsf{G}_f^{(2)}(x;r,z)=\frac{n}{r}(f(x+rz)-f(x))z$$

$$\tilde{\mathsf{G}}_f^{(2)}(x;r,z)=\frac{n}{2r}(f(x+rz)-f(x-rz))z$$

- 单点梯度估计器 Single-point gradient estimator

$$\mathsf{G}_f^{(1)}(x;r,z)=\frac{n}{r}f(x+rz)z$$

由 λ 的对称性可知，以上三种梯度估计器的期望相等

零阶优化算法

$$x_{k+1} = x_k - \eta \cdot \mathsf{G}_f^{(2)}(x_k; r, z_k)$$

$$\mathsf{G}_f^{(2)}(x; r, z) = \frac{n}{r}(f(x + rz) - f(x))z$$

$$z_k \sim \mathcal{N}(0, n^{-1}I) \quad \text{or} \quad z_k \sim \mathrm{Unif}(\mathbb{S}_{n-1})$$

- Which distribution should I choose?

$\mathcal{N}(0, n^{-1}I)$
- Independent entries, easy for parallel/distributed computation
- Unbounded exploration, may cause problems if f is defined only on a compact set

$\mathrm{Unif}(\mathbb{S}_{n-1})$
- Bounded exploration, easy to be adapted to bounded feasible region
- Entries are dependent, not that easy to be adapted for distributed computation

零阶梯度估计: Bias

- Zeroth-order gradient estimation is in general biased
- How large is the bias?

引理. 假设 $z \sim \mathcal{N}(0, n^{-1}I)$ 或 $z \sim \mathrm{Unif}(\mathbb{S}_{n-1})$. 则

$$\mathbb{E}\left[\mathsf{G}_f^{(1)}(x; r, z)\right] = \nabla f_r(x)$$ ← 有偏的随机梯度

其中

$$f_r(x) = \mathbb{E}[f(x + ry)]$$ ← 平滑化的 f

$$y \sim \begin{cases} \mathcal{N}(0, n^{-1}I), & \text{if } z \sim \mathcal{N}(0, n^{-1}I) \\ \mathrm{Unif}(\mathbb{B}_n), & \text{if } z \sim \mathrm{Unif}(\mathbb{S}_{n-1}) \end{cases}$$

$$\mathbb{B}_n = \{x \in \mathbb{R}^n : \|x\| \le 1\}$$

$$\begin{aligned}
\|\nabla f_r(x) - \nabla f(x)\| &\le \|\nabla \mathbb{E}[f(x+ry) - f(x)]\| \\
&= \|\mathbb{E}[\nabla_x f(x+ry) - \nabla_x f(x)]\| \\
&\le \mathbb{E}[\|\nabla_x f(x+ry) - \nabla_x f(x)\|] \\
&\le \mathbb{E}[L\|ry\|] \ \le Lr
\end{aligned}$$

交换求期望与求梯度
(控制收敛定理)

f is L-smooth

$$\begin{aligned}
f_r(x) - f(x) &= \mathbb{E}[f(x+ry) - f(x)] \\
&= \mathbb{E}[\langle \nabla f(x), ry\rangle + R(x,y)] \qquad R(x,y) = f(x+ry) - f(x) - \langle \nabla f(x), ry\rangle \\
&= r\langle \nabla f(x), \mathbb{E}[y]\rangle + \mathbb{E}[R(x,y)] \qquad \mathbb{E}[y] = 0 \\
&= \mathbb{E}[R(x,y)]
\end{aligned}$$

$$\begin{aligned}
|f_r(x) - f(x)| &\le \mathbb{E}[|R(x,y)|] \\
&\le \mathbb{E}\left[\frac{L}{2}\|ry\|^2\right] \ \le \frac{L}{2}r^2
\end{aligned}$$

f is L-smooth $\Longrightarrow |R(x,y)| \le \frac{L}{2}\|ry\|^2$

零阶梯度估计: Bias

引理.

假设 $z \sim \mathcal{N}(0, n^{-1}I)$ 或 $z \sim \mathrm{Unif}(\mathbb{S}_{n-1})$.

$$\mathbb{E}\left[\mathsf{G}_f^{(1)}(x; r, z)\right] = \nabla f_r(x)$$

其中

$$f_r(x) = \mathbb{E}[f(x + ry)]$$

$$y \sim \begin{cases} \mathcal{N}(0, n^{-1}I), & \text{if } z \sim \mathcal{N}(0, n^{-1}I) \\ \mathrm{Unif}(\mathbb{B}_n), & \text{if } z \sim \mathrm{Unif}(\mathbb{S}_{n-1}) \end{cases}$$

引理.

设 f 是 L-光滑的. 则

$$\|\nabla f_r(x) - \nabla f(x)\| \le Lr$$

$$|f_r(x) - f(x)| \le \frac{L}{2} r^2$$

← 偏差由 r 控制

若 f 还是凸的, 则

$$f_r(x) - f(x) \ge 0$$

且 f_r 也是凸函数.

Summary: $\mathsf{G}_f^{(1)}(x; r, z)$ 给出了一个有偏的梯度估计器（biased gradient estimator）

偏差可由平滑半径（smoothing radius）r 控制

零阶梯度估计: Variance / 2nd Moment

- 单点梯度估计：$\mathsf{G}_f^{(1)}(x;r,z) = \frac{n}{r} f(x+rz)z$ $\quad \mathbb{E}\left[\left\|\mathsf{G}_f^{(1)}(x;r,z)\right\|^2\right] \sim r^{-2}$ *Large variance*

零阶梯度估计: Variance / 2nd Moment

- 双点梯度估计：$\mathsf{G}_f^{(2)}(x;r,z)=\frac{n}{r}(f(x+rz)-f(x))z \quad \tilde{\mathsf{G}}_f^{(2)}(x;r,z)=\frac{n}{2r}(f(x+rz)-f(x-rz))z$

引理. 设 f 是 L-光滑的. 则

$$\mathbb{E}\left[\left\|\mathsf{G}_f^{(2)}(x;r,z)\right\|^2\right] \leq \begin{cases} 2(n+2)\|\nabla f(x)\|^2+\frac{L^2r^2n^2}{2}\left(\frac{n+6}{n}\right)^2, & z\sim\mathcal{N}(0,n^{-1}I) \\ 2n\|\nabla f(x)\|^2+\frac{L^2r^2n^2}{2}, & z\sim\mathrm{Unif}(\mathbb{S}_{n-1}) \end{cases}$$

Some observations:

1. The bound is monotonically increasing in r and is finite when $r\to 0$

   In practice, r should not be too small due to numerical errors in computing $f(x+rz)-f(x)$

2. As x approaches a stationary point, the bound can be made arbitrarily small.

   Perhaps faster convergence than standard SGD?

Convergence Rate

定理. 设 f 是 L-光滑且凸的，且存在最小值点 x^*. 考虑零阶优化迭代

$$\begin{aligned} x_{k+1} &= x_k - \eta \cdot \mathsf{G}_f^{(2)}(x_k; r_k, z_k) \\ &= x_k - \eta \cdot \frac{n}{r}(f(x_k + r_k z_k) - f(x_k))z_k \qquad z_k \sim \mathrm{Unif}(\mathbb{S}_{n-1}) \end{aligned}$$

令 $\eta = \dfrac{c}{2nL}$，其中 $c \in (0,1)$，并令 r_k 满足 $\sum_{k=0}^{\infty} r_k^2 = R^2 < +\infty$

则

$$\begin{aligned} \mathbb{E}[f(\bar{x}_K) - f(x^*)] &\le \frac{n}{K}\left(\frac{L\|x_0 - x^*\|^2}{c(1-c)} + \frac{R^2 L(c + 4/n)}{8(1-c)}\right) \\ &= O\Big(\frac{n}{K}\Big) \end{aligned}$$

其中 $\bar{x}_K = \dfrac{1}{K}\sum_{k=0}^{K-1} x_k$

Comparison with First-Order Methods

Two-point zeroth-order	First-order GD	First-order SGD
$\mathbb{E}[f(\bar{x}_K) - f(x^*)] \leq O\left(\frac{n}{K}\right)$	$f(x_K) - f(x^*) \leq O\left(\frac{1}{K}\right)$	$\mathbb{E}[f(\bar{x}_K) - f(x^*)] \leq O\left(\frac{1}{\sqrt{K}}\right)$

Extensions

- Noisy function values
- Stochastic problems $\min_x \mathbb{E}_\xi[F(x;\xi)]$
- Escaping saddle points
- Connection with extremum seeking control
- Distributed zeroth-order optimization
- …

Summary

- 双点梯度估计器 Two-point gradient estimator

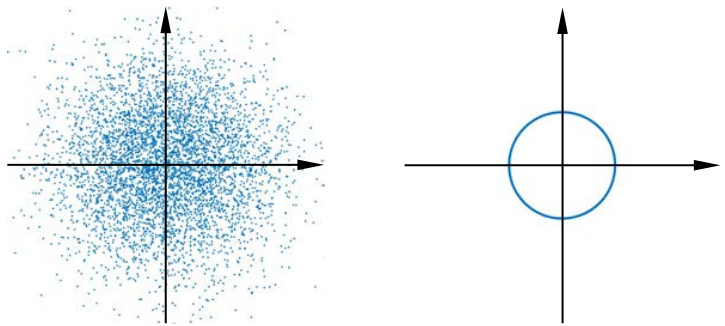
$$\mathsf{G}_f^{(2)}(x;r,z) = \frac{n}{r}(f(x+rz) - f(x))z$$

$$\tilde{\mathsf{G}}_f^{(2)}(x;r,z) = \frac{n}{2r}(f(x+rz) - f(x-rz))z$$

n - 自变量维数 dimension of x

r - 平滑半径 smoothing radius

z - 随机扰动 random perturbation



$\mathcal{N}(0, n^{-1}I)$ $\quad$ $\mathrm{Unif}(\mathbb{S}_{n-1})$

$$x_{k+1} = x_k - \eta \cdot \mathsf{G}_f^{(2)}(x_k; r, z_k)$$

$$z_k \sim \mathcal{N}(0, n^{-1}I) \quad \text{or} \quad z_k \sim \mathrm{Unif}(\mathbb{S}_{n-1})$$

Expectation: $\mathbb{E}\left[\mathsf{G}_f^{(2)}(x;r,z)\right] = \nabla f_r(x)$

2nd moment:

$$\mathbb{E}\left[\left\|\mathsf{G}_f^{(2)}(x;r,z)\right\|^2\right] \le C\left(n\|\nabla f(x)\|^2 + L^2 r^2 n^2\right)$$

Convergence for convex & smooth f:

$$\mathbb{E}[f(\bar{x}_K) - f(x^*)] \le O\left(\frac{n}{K}\right)$$

- 梯度下降与随机梯度下降

  Gradient Descent & Stochastic Gradient Descent

- 零阶优化

  Zeroth-Order Optimization

- 强化学习

  Reinforcement Learning

什么是强化学习?

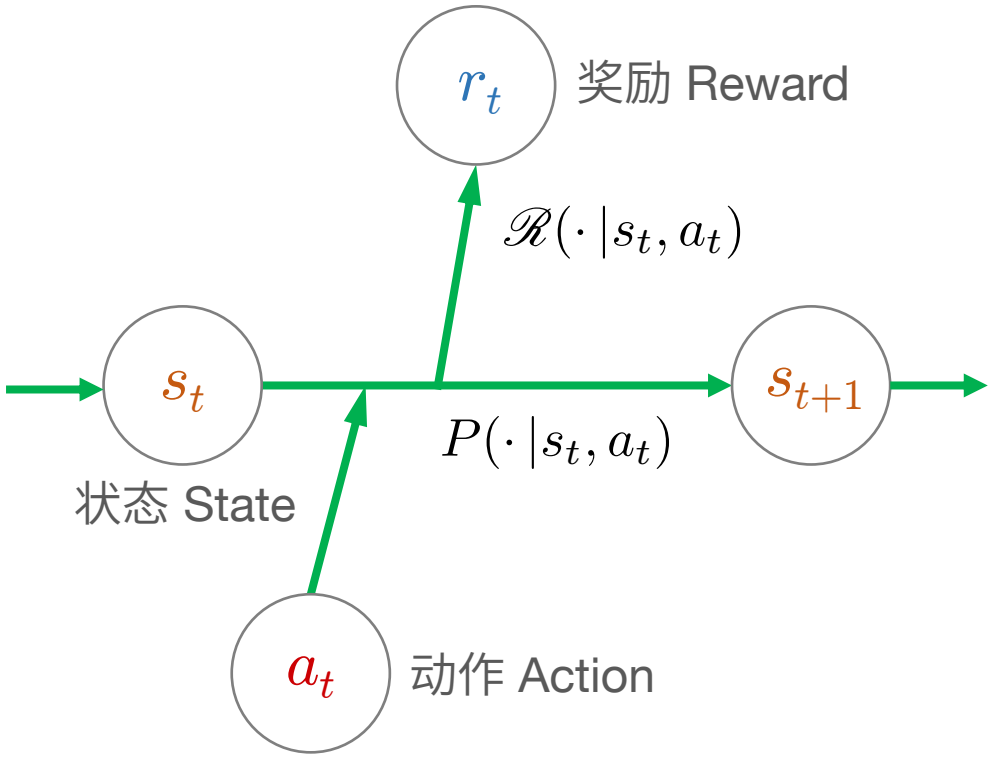
- Data-driven optimization of a Markov Decision Process

马尔可夫决策过程 Markov Decision Process

- 四元组 $(\mathcal{S}, \mathcal{A}, P, \mathscr{R})$
- $\mathcal{S}$ - 状态空间 State space
- $\mathcal{A}$ - 动作空间 Action space
- P - 转移概率 Transition probability

  $P(\cdot|s,a)$ is a probability distribution on $\mathcal{S}$
- $\mathscr{R}$ - 奖励分布 Reward distribution

  $\mathscr{R}(\cdot|s,a)$ is a probability distribution on $\mathbb{R}$



马尔可夫决策过程 Markov Decision Process

- 四元组 $(\mathcal{S}, \mathcal{A}, P, \mathscr{R})$
- $\mathcal{S}$ - 状态空间 State space
- $\mathcal{A}$ - 动作空间 Action space

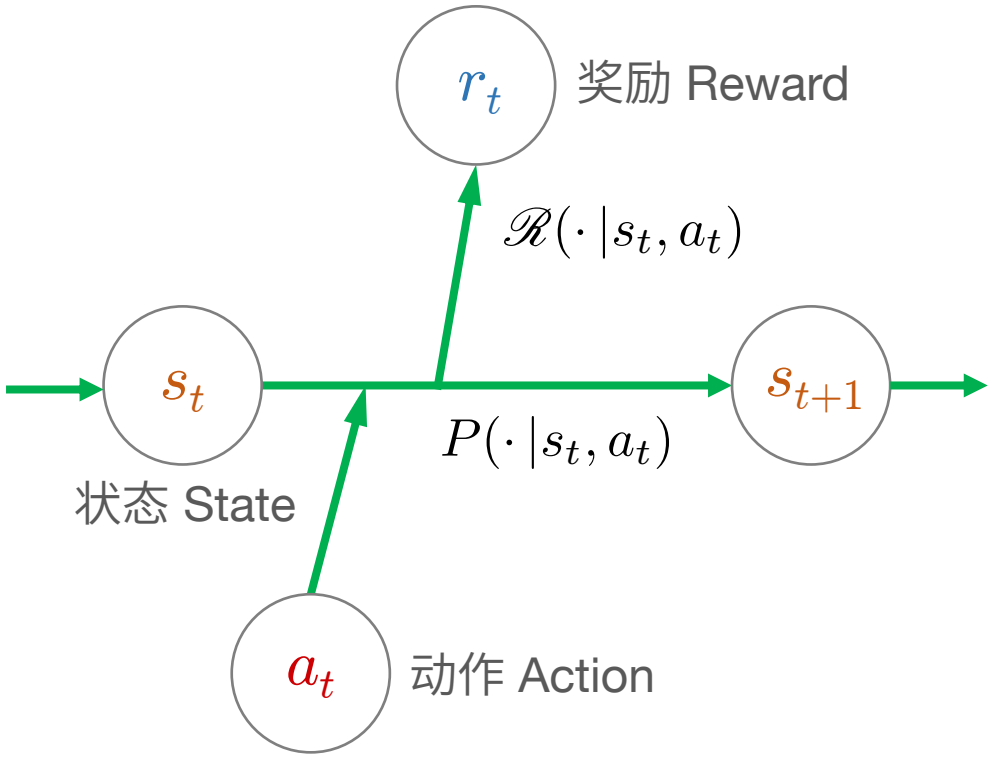
假设为有限集合

- P - 转移概率 Transition probability

  $P(\cdot|s,a)$ is a probability distribution on $\mathcal{S}$

- $\mathscr{R}$ - 奖励分布 Reward distribution

  $\mathscr{R}(\cdot|s,a)$ is a probability distribution on $\mathbb{R}$



马尔可夫决策过程 Markov Decision Process

- 四元组 $(\mathcal{S}, \mathcal{A}, P, \mathscr{R})$
- $\mathcal{S}$ - 状态空间 State space
- $\mathcal{A}$ - 动作空间 Action space

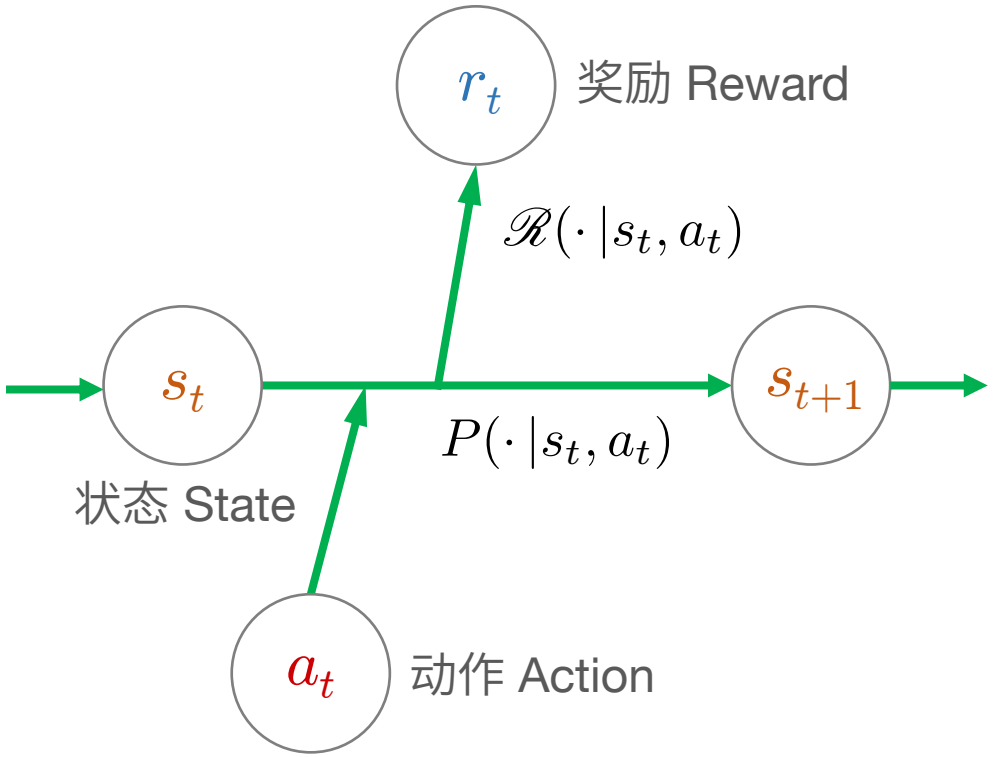
假设为有限集合

- P - 转移概率 Transition probability

$$P(s'|s,a) \geq 0, \quad \sum_{s' \in \mathcal{S}} P(s'|s,a) = 1$$

- $\mathscr{R}$ - 奖励分布 Reward distribution

$\mathscr{R}(\cdot|s,a)$ is a probability distribution on $\mathbb{R}$



马尔可夫决策过程 Markov Decision Process

- 四元组 $(\mathcal{S}, \mathcal{A}, P, r)$
- $\mathcal{S}$ - 状态空间 State space
- $\mathcal{A}$ - 动作空间 Action space

假设为有限集合

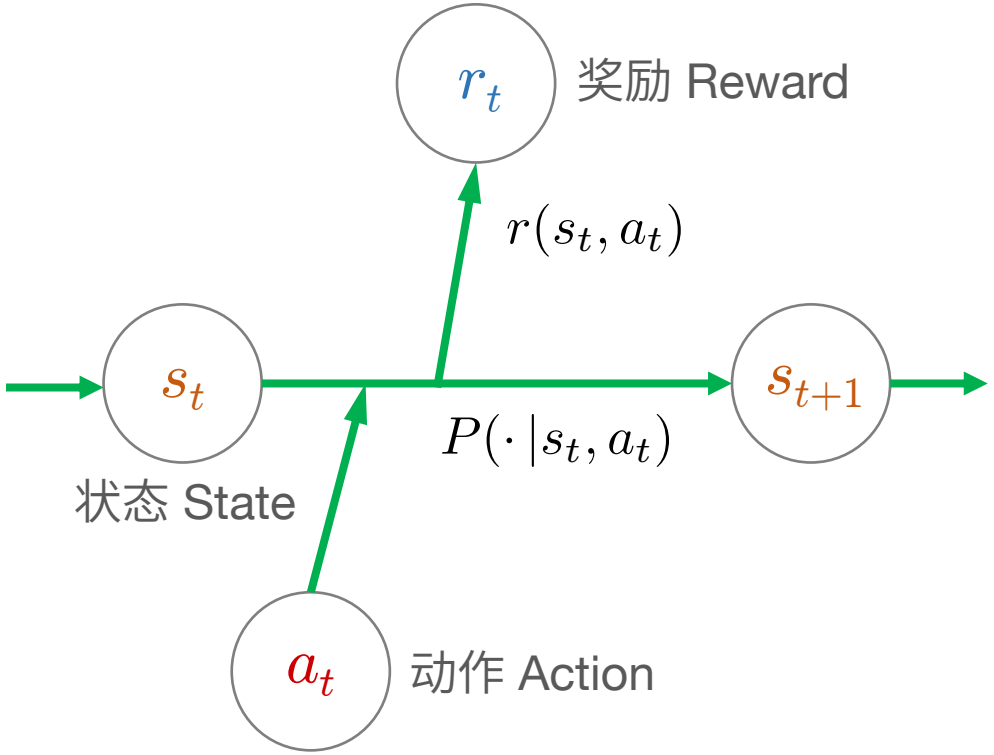
- P - 转移概率 Transition probability

$$P(s'|s,a) \geq 0, \quad \sum_{s' \in \mathcal{S}} P(s'|s,a) = 1$$

- r - 奖励函数 Reward function

$$r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$$

假设 $r(s,a)$ 属于 $[0,1]$



马尔可夫决策过程 Markov Decision Process

- 四元组 $(\mathcal{S}, \mathcal{A}, P, r)$
- 两类任务

1) Episodic tasks

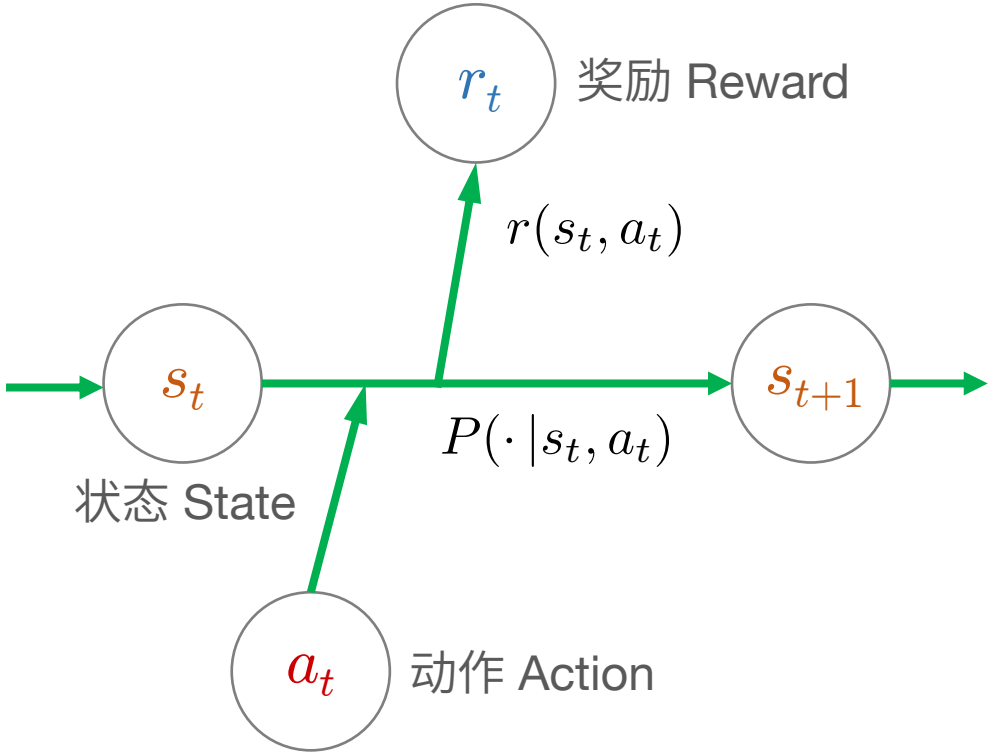
There is a set of *terminal states* $\mathcal{T}$ such that

$$\sum_{s' \in \mathcal{T}} P(s'|s,a) = 1 \quad \text{and} \quad r(s,a) = 0$$

for all $s \in \mathcal{T}$ and $a \in \mathcal{A}$

2) Continuing tasks

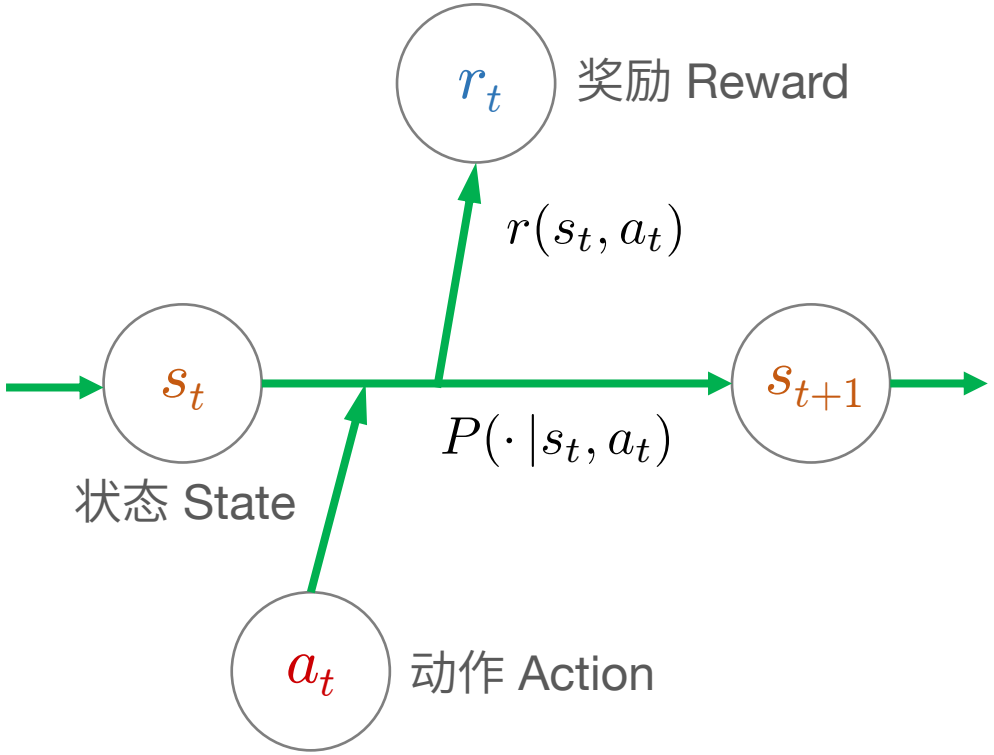
Otherwise



马尔可夫决策过程 Markov Decision Process

- 四元组 $(\mathcal{S}, \mathcal{A}, P, r)$
- 回报（Return）
  - 有限时域 $\sum_{t=0}^{T} r_t$
  - 无限时域
    - 平均回报 $\limsup_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} r_t$
    - 折扣回报 $\sum_{t=0}^{\infty} \gamma^t r_t$

$\gamma \in [0, 1]$ – 折扣因子 discount factor



马尔可夫决策过程 Markov Decision Process

- 四元组 $(\mathcal{S}, \mathcal{A}, P, r)$
- 回报 $\sum_{t=0}^{\infty} \gamma^t r_t$
- 策略 Policy

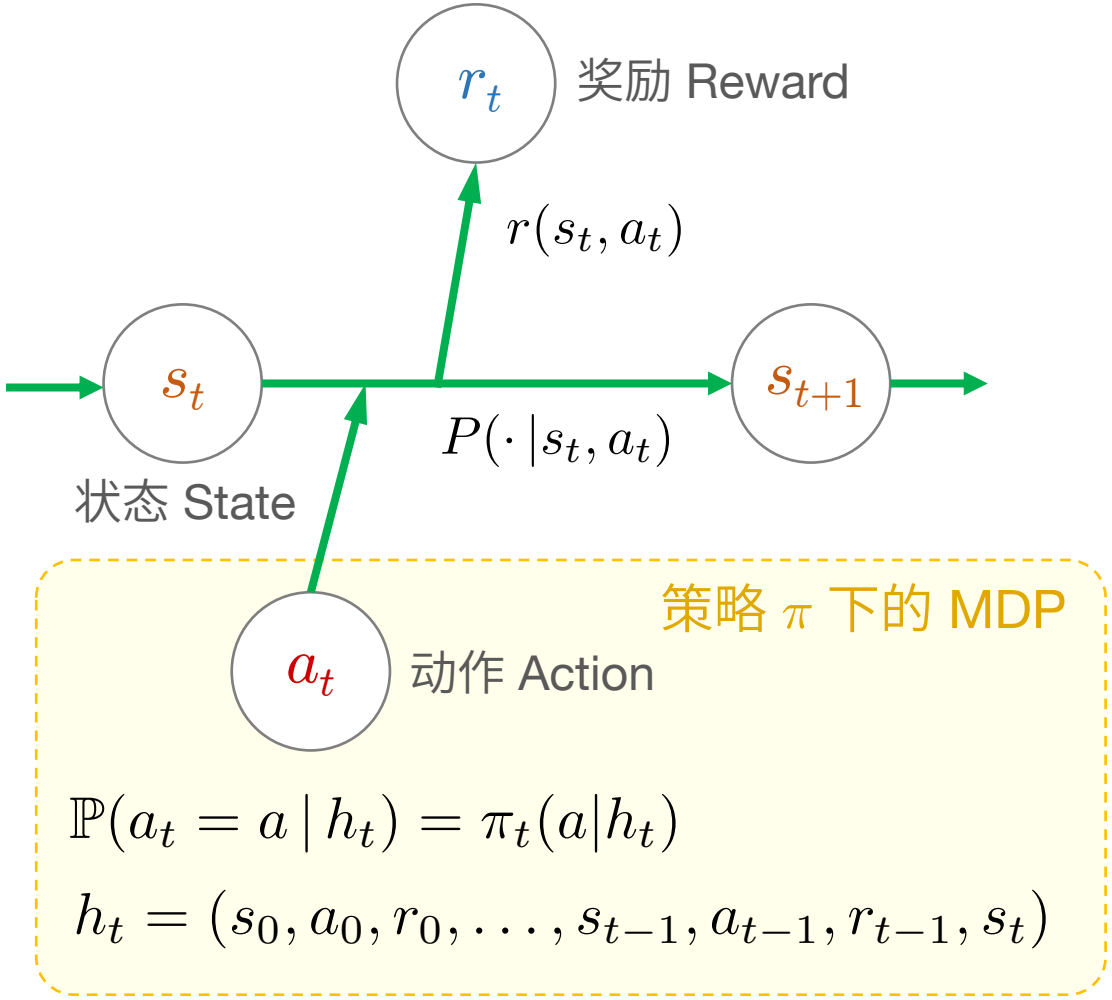
$$\pi = (\pi_0, \pi_1, \pi_2, \ldots) = (\pi_t)_{t \geq 0}$$

$$\pi_t : \mathcal{H}_t \to \Delta(\mathcal{A})$$ 动作空间上的概率分布

$$\mathcal{H}_t = \{(s_0, a_0, r_0, \ldots, s_{t-1}, a_{t-1}, r_{t-1}, s_t)\}$$

所有可能的直到 t 时刻的完整历史

假定状态 S_t 在每个时刻都能被**准确观测**

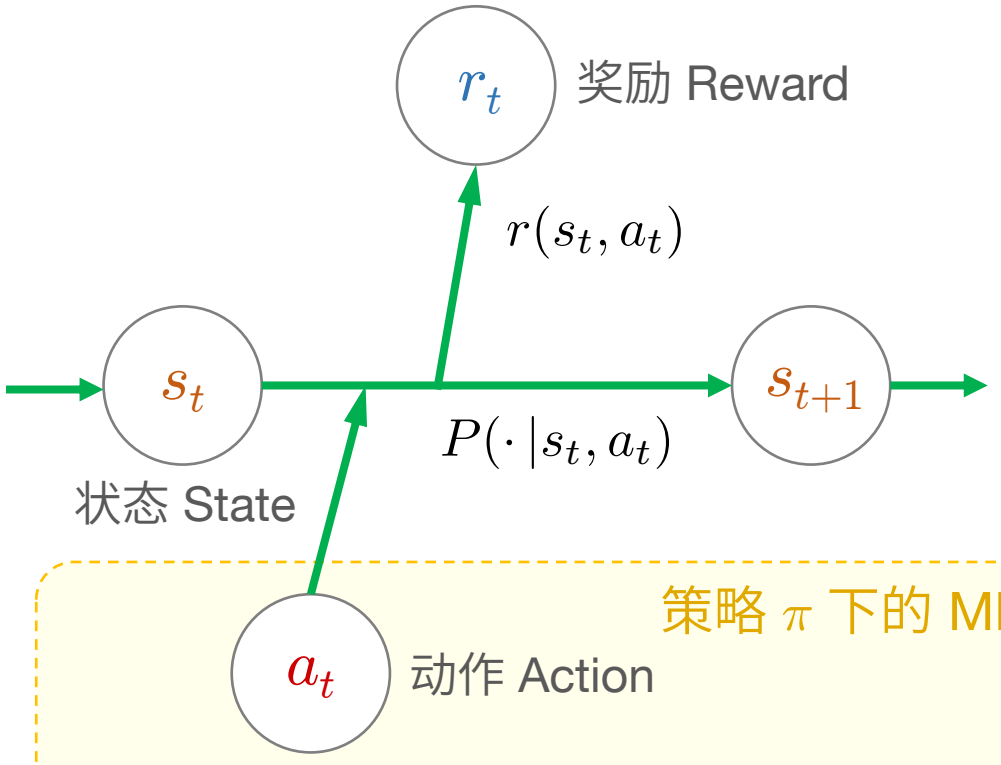


马尔可夫决策过程 Markov Decision Process

- 四元组 $(\mathcal{S}, \mathcal{A}, P, r)$
- 回报 $\sum_{t=0}^{\infty} \gamma^t r_t$
- 策略 Policy
- MDP 的最优控制问题

给定初始状态的概率分布 $s_0 \sim \mu$，
求解使得预期回报达到最大的策略

$$\min_{\pi} \quad \mathbb{E}_{\mu}^{\pi}\left[\sum_{t=0}^{\infty} \gamma^t r_t\right]$$



MDP 最优控制问题的解

定理. 取回报为无限时域折扣回报. 令函数 $V^*: \mathcal{S} \to \mathbb{R}$ 为如下 **Bellman 最优性方程**（Bellman Optimality Equation）的解：

$$V^*(s) = \max_{a \in \mathcal{A}} \left(r(s,a) + \gamma \sum_{s' \in \mathcal{S}} V^*(s')\, P(s'|s,a) \right)$$

则下式给出一个对任意初始状态分布 μ 均为最优的策略：

$$\pi_t(a|s_0, a_0 \ldots, s_t) = \pi^*(a|s_t)$$ 时不变、仅取决于当前状态（无记忆）

$$\pi^*(a|s) = \begin{cases} 1, & a = a^*(s), \\ 0, & \text{otherwise} \end{cases}$$ 确定性策略

$$a^*(s) = \arg\max_{a \in \mathcal{A}} \left(r(s,a) + \gamma \sum_{s' \in \mathcal{S}} V^*(s')\, P(s'|s,a) \right)$$ 贪婪策略

MDP 最优控制问题的解

Implication: 在寻找最优策略时，我们只需考虑**时不变**、**无记忆**的策略.

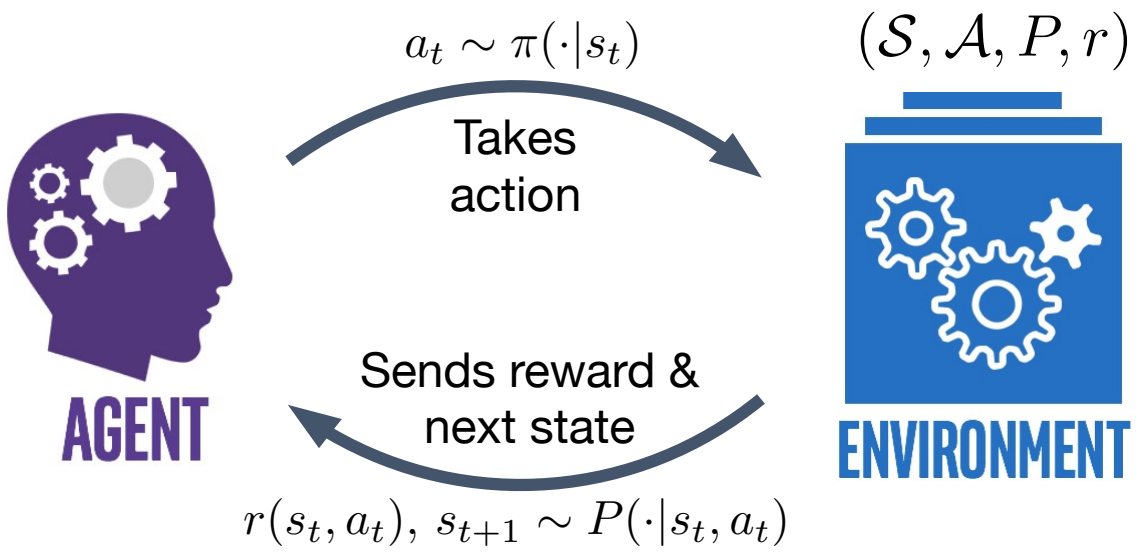
- 我们暂不排除随机性策略以及非贪婪的策略

策略： $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ $\mathbb{P}(a_t = a \,|\, h_t) = \pi(a|s_t)$

Caveat:

1. 对于有限时域问题，其最优策略一般**不再是时不变**的，但在状态可准确观测的情况下依然是**无记忆**的.
2. 若状态无法准确观测，则最优策略一般**不再是无记忆**的. 此时通常需要用部分可观测马尔可夫决策过程（Partially Observable MDP）建模.

Reinforcement Learning Problems



- Some defining features of RL problems:
  - The environment is an MDP
  - The transition probability of the MDP is not available (either because it is unknown or because it is too complicated)
  - We can collect data from the environment through sequential interactions

Further Specifications of the Environment

- The environment has an internal state s, and action input, and a “reset button”
- As long as the reset button is not pushed, whenever an action a is taken, the environment will
  - Sample $s' \sim P(\cdot|s,a)$
  - Output $r(s,a)$ and s'
  - Set its internal state $s \leftarrow s'$
- If the reset button is pushed, the environment will
  - Reset its internal state s to be a new sample from μ
  - Output s

Note: The environment doesn’t need to be a real system. It can be a simulator of a real system.

Let’s move on to some fundamental terminologies in RL

- Value function
- Q function and advantage function
- Occupancy measure

值函数 Value Function

- 如何衡量一个策略的好坏?
- 给定策略 π，定义其值函数 $V^\pi : \mathcal{S} \to \mathbb{R}$ 为

$$V^\pi(s) := \mathbb{E}^\pi \left[\sum_{t=0}^{\infty} \gamma^t \, r_t \,\middle|\, s_0 = s \right]$$

- 对于任意初始状态分布 μ，有

$$\mathbb{E}_\mu^\pi \left[\sum_{t=0}^{\infty} \gamma^t \, r_t \right] = \sum_{s \in \mathcal{S}} \mathbb{P}_\mu(s_0 = s) \cdot \mathbb{E}^\pi \left[\sum_{t=0}^{\infty} \gamma^t \, r_t \,\middle|\, s_0 = s \right]$$

$$= \sum_{s \in \mathcal{S}} \mu(s) V^\pi(s) \quad =: \mu V^\pi$$

值函数 Value Function

$$V^{\pi}(s) \coloneqq \mathbb{E}^{\pi}\left[\sum_{t=0}^{\infty}\gamma^{t}\, r_t \,\middle|\, s_0 = s\right] = \mathbb{E}^{\pi}\left[r_0 + \gamma\sum_{t=1}^{\infty}\gamma^{t-1}\, r_t \,\middle|\, s_0 = s\right]$$

$$\gamma\,\mathbb{E}^{\pi}\left[\sum_{t=1}^{\infty}\gamma^{t-1} r_t \,\middle|\, s_0 = s\right] = \gamma\,\mathbb{E}^{\pi}\left[\mathbb{E}^{\pi}\left[\sum_{t=1}^{\infty}\gamma^{t-1} r_t \,\middle|\, s_1\right] \,\middle|\, s_0 = s\right]$$

$$= \gamma\,\mathbb{E}^{\pi}[V^{\pi}(s_1) \,|\, s_0 = s]$$

Bellman 方程（Bellman Equation）

$$V^{\pi}(s) = \mathbb{E}^{\pi}[r_0 + \gamma V^{\pi}(s_1) \,|\, s_0 = s]$$

值函数 Value Function

$$V^{\pi}(s) \coloneqq \mathbb{E}^{\pi}\left[\sum_{t=0}^{\infty}\gamma^{t}\, r_t \,\middle|\, s_0 = s\right] = \mathbb{E}^{\pi}\left[r_0 + \gamma\sum_{t=1}^{\infty}\gamma^{t-1}\, r_t \,\middle|\, s_0 = s\right]$$

$$\gamma\,\mathbb{E}^{\pi}\left[\sum_{t=1}^{\infty}\gamma^{t-1} r_t \,\middle|\, s_0 = s\right] = \gamma\,\mathbb{E}^{\pi}\left[\mathbb{E}^{\pi}\left[\sum_{t=1}^{\infty}\gamma^{t-1} r_t \,\middle|\, s_1\right] \,\middle|\, s_0 = s\right]$$

$$= \gamma\,\mathbb{E}^{\pi}[V^{\pi}(s_1) \,|\, s_0 = s]$$

Bellman 方程（Bellman Equation）

$$V^{\pi}(s) = \mathbb{E}^{\pi}[r_t + \gamma V^{\pi}(s_{t+1}) \,|\, s_t = s]$$

$$= \sum_{a}\left(r(s,a) + \gamma\sum_{s'} V^{\pi}(s')\, P(s'|s,a)\right)\pi(a|s)$$

Q 函数与优势函数

- Q 函数 $\quad Q^\pi(s,a) \coloneqq r(s,a) + \gamma \sum_{s' \in \mathcal{S}} V^\pi(s')\, P(s'|s,a)$

$$= \mathbb{E}^\pi\left[r_t + \gamma V^\pi(s_{t+1}) \,\middle|\, s_t = s, a_t = a\right]$$

$$= \mathbb{E}^\pi\left[\sum_{t'=t}^{\infty} \gamma^{t'-t}\, r_{t'} \,\middle|\, s_t = s, a_t = a\right]$$

- Q 函数的 Bellman 方程

$$Q^\pi(s,a) = r(s,a) + \gamma \sum_{s',a'} Q^\pi(s',a')\, \pi(a'|s')\, P(s'|s,a)$$

$$= \mathbb{E}^\pi\left[r_t + \gamma\, Q^\pi(s_{t+1}, a_{t+1}) \,\middle|\, s_t = s, a_t = a\right]$$

- 优势函数 Advantage function $\quad A^\pi(s,a) \coloneqq Q^\pi(s,a) - V^\pi(s)$

Occupancy Measure

$$d^{\pi}_{\mu}(s,a) := \sum_{t=0}^{\infty} \gamma^t \, \mathbb{P}^{\pi}_{\mu}(s_t = s, a_t = a)$$

$$\tilde{d}^{\pi}_{\mu}(s) := \sum_{a \in \mathcal{A}} d^{\pi}_{\mu}(s,a) = \sum_{t=0}^{\infty} \gamma^t \, \mathbb{P}^{\pi}_{\mu}(s_t = s)$$

$$\sum_{s,a} d^{\pi}_{\mu}(s,a) = \frac{1}{1-\gamma}$$

Sometimes also called *(discounted) visitation frequencies*

Occupancy Measure

$$d_\mu^\pi(s,a) \coloneqq \sum_{t=0}^{\infty} \gamma^t \, \mathbb{P}_\mu^\pi(s_t = s, a_t = a)$$

$$\tilde{d}_\mu^\pi(s) \coloneqq \sum_{a \in \mathcal{A}} d_\mu^\pi(s,a) = \sum_{t=0}^{\infty} \gamma^t \, \mathbb{P}_\mu^\pi(s_t = s)$$

引理. 对任意 $q : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$，有

$$\mathbb{E}_\mu^\pi \left[\sum_{t=0}^{\infty} \gamma^t \, q(s_t, a_t) \right] = \sum_{s,a} q(s,a) \, d_\mu^\pi(s,a)$$

- How to obtain a sample of the quantity $\sum_{s,a} q(s,a) \, d_\mu^\pi(s,a)$?

  令 T 充分大，重置环境并在策略 π 下运行 T 步，获取 $s_0, a_0, \ldots, s_T, a_T$，再构造

$$\sum_{t=0}^{T} \gamma^t \, q(s_t, a_t)$$

Occupancy Measure

$$d_\mu^\pi(s,a) := \sum_{t=0}^{\infty} \gamma^t \, \mathbb{P}_\mu^\pi(s_t = s, a_t = a)$$

$$\tilde{d}_\mu^\pi(s) := \sum_{a \in \mathcal{A}} d_\mu^\pi(s,a) = \sum_{t=0}^{\infty} \gamma^t \, \mathbb{P}_\mu^\pi(s_t = s)$$

引理. $d_\mu^\pi(s,a) = \pi(a|s)\, \tilde{d}_\mu^\pi(s)$

策略 π 几乎可由 occupancy measure 唯一确定

引理. 若 $\sum_{s,a} |\pi_1(a|s) - \pi_2(a|s)|\, \tilde{d}_\mu^{\pi_2}(s) \le \delta$，则 $\left\| \tilde{d}_\mu^{\pi_1} - \tilde{d}_\mu^{\pi_2} \right\|_{\ell_1} \le \frac{\delta\gamma}{1-\gamma}$

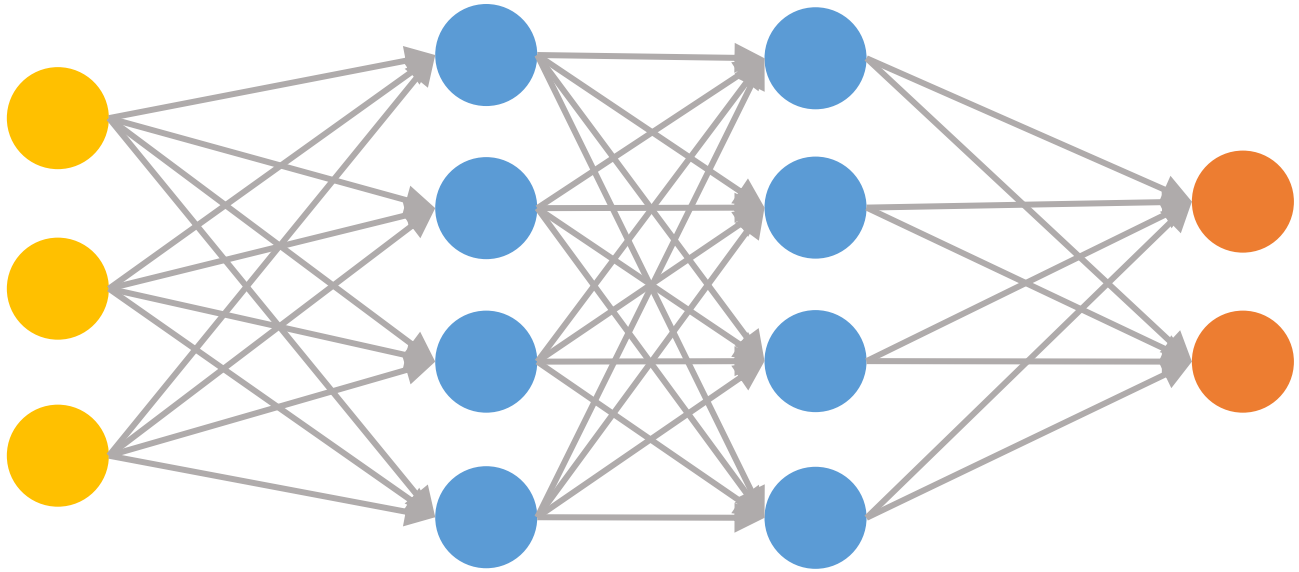
若两个策略很接近，那么相应的 occupancy measure of the state 同样也很接近

[arXiv:1705.10528]

Policy Evaluation

- How to estimate the value function of a policy using only samples?
- A fundamental building block for most RL algorithms
- We use a parameterized family of functions $\{v_w : \mathcal{S} \to \mathbb{R} \mid w \in \mathcal{W}\}$, and search for the w that can best approximate the true value function.
  - Q: Why parameterized functions?
  - A: To handle situations with enormous $|\mathcal{S}|$
  - Typical examples:
    - $\mathcal{W} = \mathbb{R}^{\mathcal{S}}$, $v_w(s) = w(s)$ (tabular RL)
    - $\mathcal{W} = \mathbb{R}^d$, $v_w(s) = w^\top \phi(s)$ (linear parameterization, $\phi : \mathcal{S} \to \mathbb{R}^d$ is a feature mapping)
    - $\mathcal{W} = \mathbb{R}^d$, $v_w(s)$ is a neural network parameterized by w

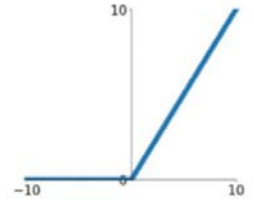
A Slight Digression: Neural Networks



σ : Activation function

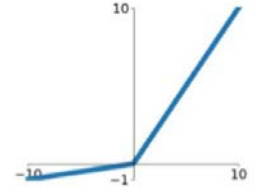
ReLU

$\max(0, x)$



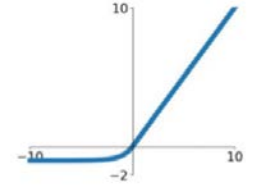
Leaky ReLU

$\max(0.1x, x)$



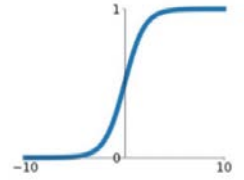
ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



Sigmoid

$\sigma(x) = \frac{1}{1+e^{-x}}$



Input Layer

Hidden Layers

$$h^{(1)} = \sigma(W^{(1)}x + b^{(1)})$$
$$h^{(k)} = \sigma(W^{(k)}h^{(k-1)} + b^{(k)})$$

Output Layer

$$y = W^{(L)}h^{(L)} + b^{(L)}$$

The parameters of a neural net: $w = (W^{(1)}, \ldots, W^{(L)}, b^{(1)}, \ldots, b^{(L)})$

The gradient with respect to w: Computed by *back propagation*

A Slight Digression: Neural Networks

Universal Approximation Theorems

[arXiv:2006.08859] Suppose $f : \mathbb{R}^{d_x} \to \mathbb{R}^{d_y}$ satisfies

$$\int_{\mathbb{R}^{d_x}} |f_i(x)|^p \, dx < +\infty$$

for some $p \in [1, \infty)$. Let $w \geq \max\{d_x + 1, d_y\}$ be arbitrary. Then for any $\epsilon > 0$, there exists a neural network $f_{\text{NN}} : \mathbb{R}^{d_x} \to \mathbb{R}^{d_y}$ with ReLU as its activation function, satisfying

1) Every hidden layer of f_{NN} has at least w neurons

2) The neural network f_{NN} approximates f in the L^p-norm:

$$\|f(x) - f_{\text{NN}}(x)\|_p = \left(\int_{\mathbb{R}^{d_x}} \sum_{i=1}^{p} |f_i(x) - f_{\text{NN},i}(x)|^p \right)^{1/p} dx \leq \epsilon$$

Policy Evaluation

- How to estimate the value function of a policy using only samples?
- A fundamental building block for most RL algorithms
- We use a parameterized family of functions $\{v_w : \mathcal{S} \to \mathbb{R} \mid w \in \mathcal{W}\}$, and search for the w that can best approximate the true value function.
  - Typical examples:
    - $\mathcal{W} = \mathbb{R}^{\mathcal{S}}$, $v_w(s) = w(s)$ (tabular RL)
    - $\mathcal{W} = \mathbb{R}^d$, $v_w(s) = w^\top \phi(s)$ (linear parameterization, $\phi : \mathcal{S} \to \mathbb{R}^d$ is a feature mapping)
    - $\mathcal{W} = \mathbb{R}^d$, $v_w(s)$ is a neural network parameterized by w
  - What does “best approximation” mean? – Minimize the mean square error

$$\min_{w \in \mathcal{W}} \quad \frac{1}{2} \mathbb{E}_{s \sim \tilde{d}_\mu^\pi} \left[(v_w(s) - V^\pi(s))^2 \right]$$

Policy Evaluation

$$\min_{w \in \mathcal{W}} \quad \frac{1}{2}\,\mathbb{E}_{s\sim\tilde{d}_\mu^\pi}\left[\left(v_w(s) - V^\pi(s)\right)^2\right] = \frac{1}{2}\,\mathbb{E}_\mu^\pi\left[\sum_{t=0}^{\infty}\gamma^t\left(v_w(s_t) - V^\pi(s_t)\right)^2\right]$$

- Apply stochastic gradient descent

$$w \leftarrow w + \alpha \sum_{t\geq 0}\gamma^t\left(V^\pi(s_t) - v_w(s_t)\right)\nabla_w v_w(s_t) \qquad \tau = (s_t, a_t, r_t)_{t\geq 0} \sim \mathbb{P}_\mu^\pi$$

A randomly sampled trajectory under π

- We don't know the value function → Replace it by an estimate (also called the *target*)

$$w \leftarrow w + \alpha \sum_{t\geq 0}\gamma^t\left(\hat{v}(\tau;t) - v_w(s_t)\right)\nabla_w v_w(s_t) \qquad \tau = (s_t, a_t, r_t)_{t\geq 0} \sim \mathbb{P}_\mu^\pi$$

  - Intuitively, $\hat{v}(\tau;t)$ should provide a "better" estimate than $v_w(s_t)$

$$\sup_s \left|\mathbb{E}^\pi[\hat{v}(\tau;t)|s_t = s] - V^\pi(s)\right| < \sup_s \left|v_w(s) - V^\pi(s)\right|$$

$\hat{v}(\tau;t)$ should have a smaller "bias" than $v_w(s)$

Policy Evaluation

$$w \leftarrow w + \alpha \sum_{t \geq 0} \gamma^t \big(\hat{v}(\tau; t) - v_w(s_t)\big) \nabla_w v_w(s_t)$$

- How to choose the target?

  Opt. 1: Use the return starting from s_t $\qquad \hat{v}(\tau; t) = \sum_{t' \geq t} \gamma^{t'-t} r_{t'}$

  - Q: Why is it a reasonable option?
  - A: We have $\mathbb{E}_\mu^\pi \left[\sum_{t'=t}^{\infty} \gamma^{t'-t} r_{t'} \,\middle|\, s_t \right] = V^\pi(s_t)$ *unbiased estimation*

Monte Carlo Method

$$w \leftarrow w + \alpha \sum_{t \geq 0} \gamma^t \left(\sum_{t' \geq t} \gamma^{t'-t} r_{t'} - v_w(s_t) \right) \nabla_w v_w(s_t) \qquad \tau = (s_t, a_t, r_t)_{t \geq 0} \sim \mathbb{P}_\mu^\pi$$

Monte Carlo Method in Practice

From OpenAI Spinning Up (simplified ver.):[1]

Loop

Reset the environment, observe s_0

For $t = 0$ to $T - 1$:

Take action $a_t \sim \pi(\cdot|s_t)$

Record the reward r_t and the next state s_{t+1}

End for

For $t = T - 1$ downto 0:

$G_t \leftarrow R_t + \gamma G_{t+1}$

$$G_T = \begin{cases} v_w(S_T), & \text{continuing} \\ 0, & \text{episodic} \end{cases}$$

End for

End loop

Loop for sufficiently many iterations:

$$w \leftarrow w + \alpha \sum_i \sum_t \left(G_t^{(i)} - v_w\left(s_t^{(i)}\right)\right) \nabla_w v_w\left(s_t^{(i)}\right)$$

End loop

Bootstrapping: Update estimates on the basis of other estimates

$$\min_w \ \mathbb{E}_{s\sim\nu}\left[\left(v_w(s) - V^\pi(s)\right)^2\right]$$

$$\nu(s) \propto \sum_{t=0}^{T-1} \mathbb{P}_\mu^\pi(s_t = s)$$

[1]https://github.com/openai/spinningup

Policy Evaluation

$$w \leftarrow w + \alpha \sum_{t \geq 0} \gamma^t \big(\hat{v}(\tau; t) - v_w(s_t)\big) \nabla_w v_w(s_t)$$

- How to choose the target?

Opt. 2: Use the temporal difference (TD) target $\hat{v}(\tau; t) = r_t + \gamma \cdot v_{w_{\text{old}}}(s_{t+1})$

bootstrapping

- Q: Why is it a reasonable option?
- A: Suppose $v_{w_{\text{old}}}(s)$ is sufficiently close to $V^\pi(s)$. Then by the Bellman equation,

$$\mathbb{E}^\pi_\mu[r_t + \gamma \cdot v_{w_{\text{old}}}(s_{t+1}) \,|\, s_t] \approx \mathbb{E}^\pi_\mu[r_t + \gamma \cdot V^\pi(s_{t+1}) \,|\, s_t] = V^\pi(s_t)$$ *biased estimation*

More detailed calculation:

$$\begin{aligned} |\mathbb{E}^\pi[\hat{v}(\tau; t) | s_t = s] - V^\pi(s)| &= \gamma \mathbb{E}^\pi[v_w(s_{t+1}) - V^\pi(s_{t+1}) \,|\, s_t = s] \\ &\leq \gamma \sup_{s'} |v_w(s') - V^\pi(s')| \end{aligned}$$

smaller "bias" than $v_w(s)$

Policy Evaluation

$$w \leftarrow w + \alpha \sum_{t \geq 0} \gamma^t \big(\hat{v}(\tau; t) - v_w(s_t)\big) \nabla_w v_w(s_t)$$

- How to choose the target?

Opt. 2: Use the temporal difference (TD) target $\hat{v}(\tau; t) = r_t + \gamma \cdot v_{w_{\text{old}}}(s_{t+1})$

bootstrapping

- Q: Why is it a reasonable option?
- A: Suppose $v_{w_{\text{old}}}(s)$ is sufficiently close to $V^\pi(s)$. Then by the Bellman equation,

$$\mathbb{E}^\pi_\mu[r_t + \gamma \cdot v_{w_{\text{old}}}(s_{t+1}) \,|\, s_t] \approx \mathbb{E}^\pi_\mu[r_t + \gamma \cdot V^\pi(s_{t+1}) \,|\, s_t] = V^\pi(s_t)$$ *biased estimation*

TD(0) Learning

$$w \leftarrow w + \alpha \sum_t \gamma^t \underbrace{\big(r_t + \gamma \cdot v_w(s_{t+1}) - v_w(s_t)\big)}_{\text{temporal difference}} \nabla_w v_w(s_t) \qquad \tau = (s_t, a_t, r_t)_{t \geq 0} \sim \mathbb{P}^\pi_\mu$$

TD(0) in Practice

From Stable Baselines 3 (simplified ver.):[1]

Loop for sufficiently many iterations:

Reset the environment, observe s_0

For $t = 0$ to $T - 1$:

Take action $a_t \sim \pi(\cdot|s_t)$

Record the reward r_t and the next state s_{t+1}

$\delta_t \leftarrow r_t + \gamma\, v_w(s_{t+1}) - v_w(s_t)$

End for

$w \leftarrow w + \alpha \sum_t \delta_t \cdot \nabla_w v_w(s_t)$

End loop

[1]https://github.com/DLR-RM/stable-baselines3

TD(0) in Practice

From Sutton's book:

Reset the environment, observe s_0

For $t = 0$ to $T - 1$:

Take action $a_t \sim \pi(\cdot|s_t)$

Observe the reward r_t and the next state s_{t+1}

$\delta_t \leftarrow r_t + \gamma\, v_w(s_{t+1}) - v_w(s_t)$

$w \leftarrow w + \alpha \cdot \delta_t \nabla_w v_w(s_t)$

End for

- The classical version of TD(0)
- Updates concurrently with the MDP
- Under certain conditions, with properly chosen diminishing step sizes, TD(0) converges to a neighborhood of the optimal solution to

$$\min_w \ \mathbb{E}_{s\sim\nu}\left[(v_w(s) - V^\pi(s))^2\right]$$

for *linear parameterization*, where ν is the *stationary distribution* of the MDP's state under policy π.

Summary of Policy Evaluation

- 用参数化的一族函数近似实际的值函数——避免状态数过大造成的维数灾难
- 如何近似——最小化均方误差

$$\min_{w\in\mathcal{W}} \quad \frac{1}{2}\mathbb{E}_{s\sim\tilde{d}_\mu^\pi}\left[\left(v_w(s)-V^\pi(s)\right)^2\right]=\frac{1}{2}\mathbb{E}_\mu^\pi\left[\sum_{t=0}^{\infty}\gamma^t\left(v_w(s_t)-V^\pi(s_t)\right)^2\right]$$

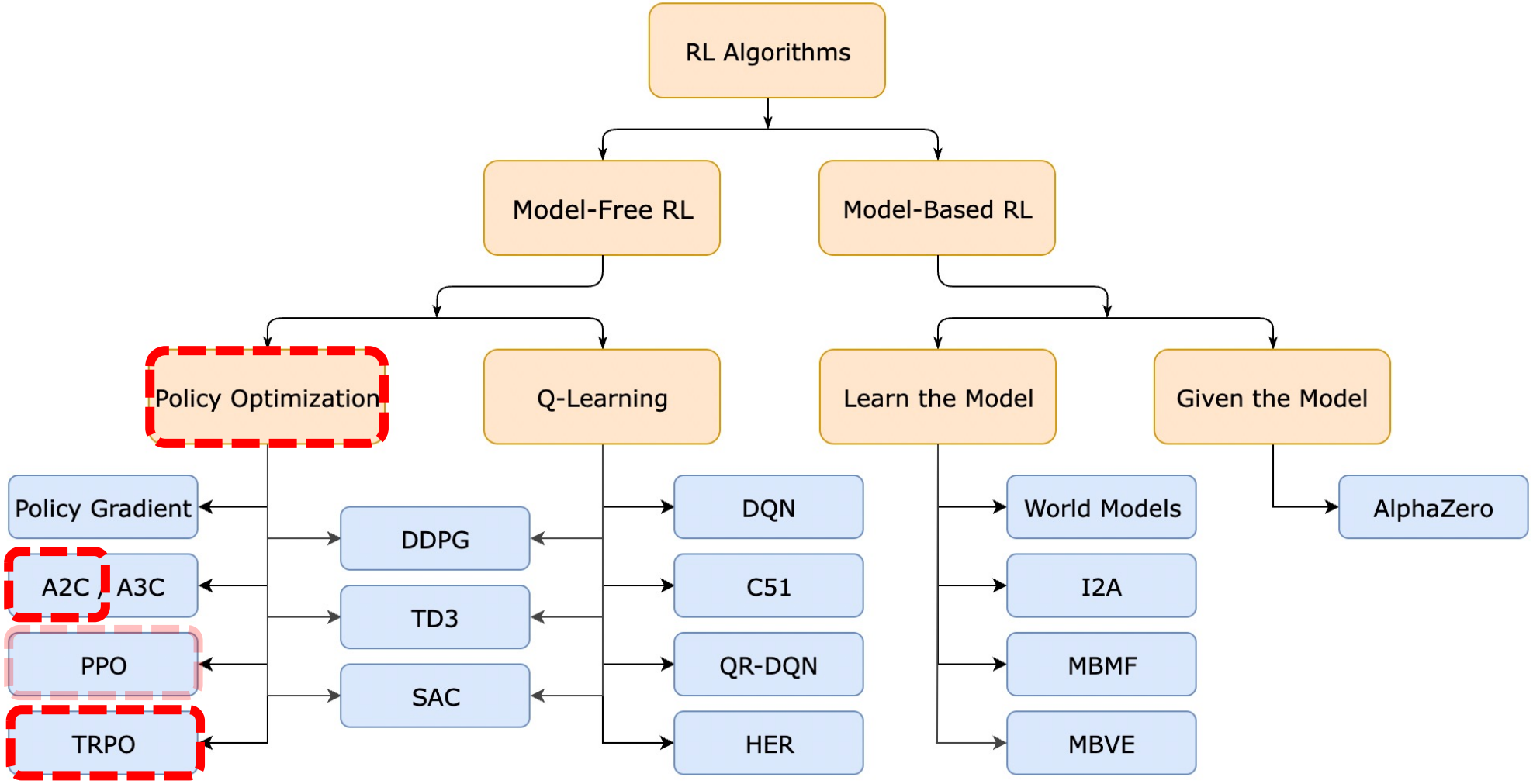
- 如何求解——随机梯度下降 $w \leftarrow w+\alpha\sum_{t\geq 0}\gamma^t\big(\hat{v}(\tau;t)-v_w(s_t)\big)\nabla_w v_w(s_t)$
- 如何选择 target?

Opt. 1: 蒙特卡洛方法
- 无偏，方差较大
- Convergence guarantees by the theory of SGD

Opt. 2: 时序差分方法
- 有偏，方差较小
- Convergence established for Sutton's version with linear parameterization

其它方法: TD(λ), LSTD, LSPE

强化学习算法



A non-exhaustive, but useful taxonomy of algorithms in modern RL. (from OpenAI Spinning Up)

Parameterized Policies

- 引入参数 $\theta \in \Theta$ ，每一个参数 θ 对应一个策略 π_θ
- $\{\pi_\theta : \theta \in \Theta\}$ 不一定包含所有策略，但最好能任意逼近最优策略.
- 对于离散的动作空间，常用如下参数化方法：

$$\pi_\theta(a|s) = \frac{\exp(f_\theta(s,a))}{\sum_{a' \in \mathcal{A}} \exp(f_\theta(s,a'))}$$

其中 $f_\theta(s,a)$ 可以是

- $\theta_{s,a}$ ，即取 $\Theta = \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$ ，而 $f_\theta(s,a)$ 为矩阵 $\theta \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$ 的 (s,a) 号元素 (tabular RL, $|\mathcal{S}||\mathcal{A}|$should not be very large)
- $\theta^\top \phi(s,a)$ ，其中 $\theta \in \mathbb{R}^d$ 而 $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$ 为 feature mapping
- 参数为 θ 的神经网络，输入为 $s \in \mathcal{S}$，共 $|\mathcal{A}|$ 个输出（Actor network）

Two Useful Theoretical Results

Performance Difference Lemma.

对任意两个策略 π, π'，有

$$\mu V^{\pi'} - \mu V^{\pi} = \sum_{s,a} A^{\pi}(s,a)\,\pi'(a|s)\,\tilde{d}_{\mu}^{\pi'}(s)$$

Policy Gradient Theorem.

令 $\{\pi_\theta : \theta \in \Theta\}$ 为一族参数化的策略. 则在一定条件下，有

$$\nabla_\theta(\mu V^{\pi_\theta}) = \sum_{s,a}(Q^{\pi_\theta}(s,a) - b(s))\nabla_\theta \pi_\theta(a|s)\,\tilde{d}_{\mu}^{\pi_\theta}(s)$$

其中 $b(s)$ 可以是任意（只依赖于 s）的函数.

固定 $\theta_0 \in \Theta$，定义

$$L(\theta;\theta_0) := \sum_{s,a} A^{\pi_{\theta_0}}(s,a)\,\pi_\theta(a|s)\,\tilde{d}_{\mu}^{\pi_{\theta_0}}(s)$$

则

1. $\nabla_\theta L(\theta;\theta_0)|_{\theta=\theta_0} = \nabla_\theta(\mu V^{\pi_\theta})|_{\theta=\theta_0}$
2. 当 π_θ 与 π_{θ_0} 足够接近时，有

$$L(\theta;\theta_0) \approx \mu V^{\pi_\theta} - \mu V^{\pi_{\theta_0}}$$

$\Longrightarrow$ $L(\theta;\theta_0)$ 可作为 μV^{π_θ} 的局域近似

A Tentative Framework

for $k = 0, 1, 2, \ldots$ **do**

$$\theta_{k+1} \approx \arg\max_{\theta \in \Theta} \; L(\theta; \theta_k)$$
$$\text{s.t.} \;\; \pi_\theta \approx \pi_{\theta_k}$$

end for

$$L(\theta; \theta_k) = \sum_{s,a} A^{\pi_{\theta_k}}(s,a)\, \pi_\theta(a|s)\, \tilde{d}_\mu^{\pi_{\theta_k}}(s)$$

仍待解决的问题：

1. 怎样定量刻画 $\pi_\theta \approx \pi_{\theta_k}$？
2. 如何（近似）求解当中的优化问题?
3. 如何从观测数据估计 $A^{\pi_{\theta_k}}(s,a)$ 等求解优化问题所需的量?

怎样刻画 $\pi_\theta \approx \pi_{\theta_k}$？

for $k = 0, 1, 2, \ldots$ **do**

$$\theta_{k+1} \approx \arg\max_{\theta \in \Theta} \; L(\theta; \theta_k)$$
$$\text{s.t.} \;\; \pi_\theta \approx \pi_{\theta_k}$$

end for

$$L(\theta; \theta_k) = \sum_{s,a} A^{\pi_{\theta_k}}(s,a)\, \pi_\theta(a|s)\, \tilde{d}_\mu^{\pi_{\theta_k}}(s)$$

- 方案 1：$\frac{1}{2}\|\theta - \theta_k\|^2 \le \delta$
- 方案 2：利用 KL-divergence

$$\sum_s D_{\mathrm{KL}}(\pi_{\theta_k}(\cdot|s) \,\|\, \pi_\theta(\cdot|s))\, \tilde{d}_\mu^{\pi_{\theta_k}}(s) \le \frac{\delta}{1-\gamma}$$

$$D_{\mathrm{KL}}(p \,\|\, q) = \sum_a p(a) \ln \frac{p(a)}{q(a)}$$

- 方案 3：Clip the ratio

$$1 - \epsilon \le \frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)} \le 1 + \epsilon, \quad \forall s \in \mathcal{S}, a \in \mathcal{A}$$

方案 1

$$\max_{\theta\in\Theta} \quad L(\theta;\theta_k) = \sum_{s,a} A^{\pi_{\theta_k}}(s,a)\,\pi_\theta(a|s)\,\tilde{d}_\mu^{\pi_{\theta_k}}(s)$$

$$\text{s.t.} \quad \frac{1}{2}\|\theta-\theta_k\|^2 \le \delta$$

约束替换为惩罚函数

目标函数线性化

$$\max_{\theta\in\Theta} \quad \left\langle \nabla_\theta L(\theta;\theta_k)|_{\theta=\theta_k}, \theta-\theta_k \right\rangle - \frac{1}{2\eta}\|\theta-\theta_k\|^2$$

闭式解

$$\begin{aligned}\theta_{k+1} &= \theta_k + \eta\,\nabla_\theta L(\theta;\theta_k)|_{\theta=\theta_k} \\ &= \theta_k + \eta\,\nabla_\theta(\mu V^{\pi_\theta})|_{\theta=\theta_k}\end{aligned}$$

Policy Gradient Theorem.

$$\nabla_\theta(\mu V^{\pi_\theta}) = \sum_{s,a} A^{\pi_\theta}(s,a)\,\nabla_\theta\pi_\theta(a|s)\,\tilde{d}_\mu^{\pi_\theta}(s)$$

This is just gradient ascent!

方案 1

$$\sum_{s,a} A^{\pi_\theta}(s,a) \cdot \nabla_\theta \pi_\theta(a|s) \cdot \tilde{d}_\mu^{\pi_\theta}(s)$$

$$= \sum_{s,a} A^{\pi_\theta}(s,a) \cdot \frac{\nabla_\theta \pi_\theta(a|s)}{\pi_\theta(a|s)} \cdot \pi_\theta(a|s) \tilde{d}_\mu^{\pi_\theta}(s)$$

$$= \sum_{s,a} A^{\pi_\theta}(s,a) \cdot \nabla_\theta \ln \pi_\theta(a|s) \cdot d_\mu^{\pi_\theta}(s,a)$$

$$= \mathbb{E}_\mu^\pi \left[\sum\nolimits_{t \geq 0} \gamma^t A^{\pi_\theta}(s_t, a_t) \cdot \nabla_\theta \ln \pi_\theta(a_t|s_t) \right]$$

引理. $d_\mu^\pi(s,a) = \pi(a|s)\, \tilde{d}_\mu^\pi(s)$

引理. $\mathbb{E}_\mu^\pi \left[\sum\nolimits_{t \geq 0} \gamma^t\, q(s_t, a_t) \right] = \sum_{s,a} q(s,a) d_\mu^\pi(s,a)$

$$\theta \leftarrow \theta + \eta \sum_{t \geq 0} \gamma^t\, \hat{A}(\tau; t) \cdot \nabla_\theta \ln \pi_\theta(a_t|s_t)$$

$\tau = (s_t, a_t, r_t)_{t \geq 0} \sim \mathbb{P}_\mu^{\pi_\theta}$

$\hat{A}(\tau; t)$ is an estimate of $A^{\pi_\theta}(s_t, a_t)$

How to form an estimate of $A^{\pi_\theta}(s_t, a_t)$?

- **Opt. 1**: Monte Carlo method $\hat{A}(\tau; t) = \sum_{t' \geq t} \gamma^{t'-t} r_{t'} - v_{w_{\text{old}}}(s_t)$
  - Why is it a valid option?
  - Is it an unbiased estimate of $A^{\pi_\theta}(s_t, a_t)$?

$$\mathbb{E}_{\mu}^{\pi_\theta}\left[\sum_{t' \geq t} \gamma^{t'-t} r_{t'} - v_{w_{\text{old}}}(s_t) \,\middle|\, s_t, a_t\right] = Q^{\pi_\theta}(s_t, a_t) - v_{w_{\text{old}}}(s_t)$$ *biased*

$$Q^{\pi}(s, a) = \mathbb{E}^{\pi}\left[\sum_{t'=t}^{\infty} \gamma^{t'-t} r_{t'} \,\middle|\, s_t = s, a_t = a\right]$$

How to form an estimate of $A^{\pi_\theta}(s_t, a_t)$?

- **Opt. 1**: Monte Carlo method $\hat{A}(\tau; t) = \sum_{t' \geq t} \gamma^{t'-t} r_{t'} - v_{w_{\text{old}}}(s_t)$
  - Why is it a valid option?
  - Is it an unbiased estimate of $A^{\pi_\theta}(s_t, a_t)$?

$$\mathbb{E}_{\mu}^{\pi_\theta}\left[\sum_{t' \geq t} \gamma^{t'-t} r_{t'} - v_{w_{\text{old}}}(s_t) \,\middle|\, s_t, a_t\right] = Q^{\pi_\theta}(s_t, a_t) - v_{w_{\text{old}}}(s_t) \quad \textit{biased}$$

Policy Gradient Theorem.

$$\nabla_\theta(\mu V^{\pi_\theta}) = \sum_{s,a}(Q^{\pi_\theta}(s,a) - b(s))\nabla_\theta \pi_\theta(a|s)\, \tilde{d}_{\mu}^{\pi_\theta}(s)$$

其中 $b(s)$ 可以是任意（只依赖于 s）的函数.

How to form an estimate of $A^{\pi_\theta}(s_t, a_t)$?

- **Opt. 1**: Monte Carlo method $\hat{A}(\tau; t) = \sum_{t' \geq t} \gamma^{t'-t} r_{t'} - v_{w_{\text{old}}}(s_t)$
  - Why is it a valid option?
  - We have

    $$\mathbb{E}_{\mu}^{\pi_\theta}\left[\sum_{t\geq 0} \gamma^t \left(\sum_{t'\geq t} \gamma^{t'-t} r_{t'} - v_{w_{\text{old}}}(s)\right) \nabla_\theta \ln \pi_\theta(a_t|s_t)\right] = \nabla_\theta(\mu V^{\pi_\theta})$$

    i.e., the policy gradient estimation is unbiased

Note: The trajectory τ in $\hat{A}(\tau; t)$ should be independent from the trajectories used in constructing $b(s)$.

Policy Gradient Theorem.

$$\nabla_\theta(\mu V^{\pi_\theta}) = \sum_{s,a}(Q^{\pi_\theta}(s,a) - b(s))\nabla_\theta \pi_\theta(a|s)\, \tilde{d}_{\mu}^{\pi_\theta}(s)$$

其中 $b(s)$ **可以是任意（只依赖于 s）的函数.**

How to form an estimate of $A^{\pi_\theta}(s_t, a_t)$?

- **Opt. 2**: Use the temporal difference $\hat{A}(\tau; t) = r_t + \gamma \cdot v_{w_{\text{old}}}(s_{t+1}) - v_{w_{\text{old}}}(s_t)$
  - This time the policy gradient estimation is biased

$$\nabla_\theta(\mu V^{\pi_\theta}) = \mathbb{E}_\mu^{\pi_\theta}\left[\sum\nolimits_{t\geq 0} \gamma^t \left(r_t + \gamma \cdot V^{\pi_\theta}(s_{t+1}) - v_{w_{\text{old}}}(s)\right) \nabla_\theta \ln \pi_\theta(a_t|s_t)\right]$$

$$Q^\pi(s,a) = \mathbb{E}^\pi[r_t + \gamma V^\pi(s_{t+1}) \mid s_t = s, a_t = a]$$

How to form an estimate of $A^{\pi_\theta}(s_t, a_t)$?

- **Opt. 2**: Use the temporal difference $\hat{A}(\tau; t) = r_t + \gamma \cdot v_{w_{\text{old}}}(s_{t+1}) - v_{w_{\text{old}}}(s_t)$
  - This time the policy gradient estimation is biased

$$\nabla_\theta(\mu V^{\pi_\theta}) = \mathbb{E}_\mu^{\pi_\theta}\left[\sum\nolimits_{t\geq 0} \gamma^t \left(r_t + \gamma \cdot V^{\pi_\theta}(s_{t+1}) - v_{w_{\text{old}}}(s)\right) \nabla_\theta \ln \pi_\theta(a_t|s_t)\right]$$

$$\approx \mathbb{E}_\mu^{\pi_\theta}\left[\sum\nolimits_{t\geq 0} \gamma^t \left(r_t + \gamma \cdot v_{w_{\text{old}}}(s_{t+1}) - v_{w_{\text{old}}}(s)\right) \nabla_\theta \ln \pi_\theta(a_t|s_t)\right]$$

  - But the variance is in general lower than Monte Carlo method

Policy Gradient Estimation

$$\widehat{\nabla_\theta}(\mu V^{\pi_\theta}) = \sum_{t \geq 0} \gamma^t \hat{A}(\tau; t)\, \nabla_\theta \ln \pi_\theta(a_t | s_t)$$

- **Opt. 1**: Monte Carlo method $\hat{A}(\tau; t) = \sum_{t' \geq t} \gamma^{t'-t} r_{t'} - v_{w_{\text{old}}}(s_t)$
- **Opt. 2**: Use the temporal difference $\hat{A}(\tau; t) = r_t + \gamma \cdot v_{w_{\text{old}}}(s_{t+1}) - v_{w_{\text{old}}}(s_t)$
- **A general method**: Generalized Advantage Estimation (GAE) [arXiv:1506.02438]

GAE is able to tune the bias-variance tradeoff

The Advantage Actor-Critic (A2C) Method

1. Under policy π_θ, sample a batch of trajectories $\mathcal{D} = \left\{ \tau^{(i)} = \left(s_t^{(i)}, a_t^{(i)}, r_t^{(i)} \right)_{t \geq 0} \,\middle|\, 1 \leq i \leq |\mathcal{D}| \right\}$

2. Update the policy (**actor**) by

$$\theta \leftarrow \theta + \eta \cdot \frac{1}{|\mathcal{D}|} \sum_{i=1}^{|\mathcal{D}|} \sum_{t \geq 0} \gamma^t \, \hat{A}(\tau^{(i)}; t) \, \nabla_\theta \ln \pi_\theta \left(a_t^{(i)} | s_t^{(i)} \right)$$

where $\hat{A}(\tau; t) = \begin{cases} \sum_{t' \geq t} \gamma^{t'-t} \, r_{t'} - v_w(s_t), & \text{Monte Carlo advantage estimation} \\ r_t + \gamma \, v_w(s_{t+1}) - v_w(s_t), & \text{TD advantage estimation} \end{cases}$

3. Update the value estimation (**critic**) by repeating

$$w \leftarrow \begin{cases} w + \alpha \cdot \frac{1}{|\mathcal{D}|} \sum_{i=1}^{|\mathcal{D}|} \sum_{t \geq 0} \gamma^t \left(\sum_{t' \geq t} \gamma^{t'-t} \, r_{t'} - v_w(s_t) \right) \nabla_w v_w(s_t), & \text{Monte Carlo} \\ w + \alpha \cdot \frac{1}{|\mathcal{D}|} \sum_{i=1}^{|\mathcal{D}|} \sum_{t \geq 0} \gamma^t \left(r_t + \gamma \, v_w(s_{t+1}) - v_w(s_t) \right) \nabla_w v_w(s_t), & \text{TD(0)} \end{cases}$$

4. Go back to Step 1 unless stopping criterion is satisfied

Policy Gradient Estimation in Practice

- What the theory tells us: $\widehat{\nabla_\theta}(\mu V^{\pi_\theta}) = \sum_{t \geq 0} \gamma^t \hat{A}(\tau; t)\, \nabla_\theta \ln \pi_\theta(a_t|s_t)$
- What most RL libraries use: $\widehat{\nabla_\theta}(\mu V^{\pi_\theta}) = \sum_{t \geq 0} \cancel{\gamma^t}\, \hat{A}(\tau; t)\, \nabla_\theta \ln \pi_\theta(a_t|s_t)$ *theoretically incorrect*
- See [arXiv:1906.07073] [arXiv:2010.01069] for relevant discussions

怎样刻画 $\pi_\theta \approx \pi_{\theta_k}$？

for $k = 0, 1, 2, \ldots$ **do**

$$\theta_{k+1} \approx \arg\max_{\theta \in \Theta} \; L(\theta; \theta_k)$$

$$\text{s.t.} \quad \pi_\theta \approx \pi_{\theta_k}$$

end for

$$L(\theta; \theta_k) = \sum_{s,a} A^{\pi_{\theta_k}}(s,a)\, \pi_\theta(a|s)\, \tilde{d}_\mu^{\pi_{\theta_k}}(s)$$

- 方案 1：$\frac{1}{2}\|\theta - \theta_k\|^2 \le \delta$ **Advantage Actor Critic**
- 方案 2：利用 KL-divergence

$$\sum_s D_{\mathrm{KL}}(\pi_{\theta_k}(\cdot|s) \,\|\, \pi_\theta(\cdot|s))\, \tilde{d}_\mu^{\pi_{\theta_k}}(s) \le \frac{\delta}{1-\gamma}$$

$$D_{\mathrm{KL}}(p \,\|\, q) = \sum_a p(a) \ln \frac{p(a)}{q(a)}$$

- 方案 3：Clip the ratio

$$1 - \epsilon \le \frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)} \le 1 + \epsilon, \quad \forall s \in \mathcal{S}, a \in \mathcal{A}$$

方案 2

$$\theta_{k+1} \approx \arg\max_{\theta\in\Theta} \sum_{s,a} A^{\pi_{\theta_k}}(s,a)\,\pi_\theta(a|s)\,\tilde{d}_\mu^{\pi_{\theta_k}}(s)$$

$$\text{s.t.} \sum_s D_{\mathrm{KL}}(\pi_{\theta_k}(\cdot|s) \,\|\, \pi_\theta(\cdot|s))\,\tilde{d}_\mu^{\pi_{\theta_k}}(s) \le \frac{\delta}{1-\gamma}$$

$$D_{\mathrm{KL}}(p \,\|\, q) = \sum_a p(a) \ln \frac{p(a)}{q(a)}$$

- Why is KL-divergence a reasonable choice?
- Pinsker's inequality $\sum_a |p(a) - q(a)| \le \sqrt{2\, D_{\mathrm{KL}}(p \,\|\, q)}$

  Lemma. If $\sum_{s,a} |\pi(a|s) - \pi'(a|s)|\,\tilde{d}_\mu^{\pi}(s) \le \varepsilon$, then $\left\|\tilde{d}_\mu^{\pi'} - \tilde{d}_\mu^{\pi}\right\|_{\ell_1} \le \frac{\varepsilon\gamma}{1-\gamma}$

  Performance difference lemma: $\mu V^{\pi'} - \mu V^{\pi} = \sum_{s,a} A^{\pi}(s,a)\,\pi'(a|s)\,\tilde{d}_\mu^{\pi'}(s)$

$$\mu V^{\pi_\theta} - \mu V^{\pi_{\theta_k}} \ge L(\theta;\theta_k) - C_{\pi_{\theta_k}} \sqrt{\sum_s D_{\mathrm{KL}}(\pi_{\theta_k}(\cdot|s) \,\|\, \pi_\theta(\cdot|s))\,\tilde{d}_\mu^{\pi_{\theta_k}}(s)}$$

penalty on the KL divergence

[arXiv:1705.10528]

方案 2

$$\theta_{k+1} \approx \arg\max_{\theta \in \Theta} \sum_{s,a} A^{\pi_{\theta_k}}(s,a)\, \pi_\theta(a|s)\, \tilde{d}_\mu^{\pi_{\theta_k}}(s)$$

$$\text{s.t.} \sum_{s} D_{\mathrm{KL}}(\pi_{\theta_k}(\cdot|s) \,\|\, \pi_\theta(\cdot|s))\, \tilde{d}_\mu^{\pi_{\theta_k}}(s) \le \frac{\delta}{1-\gamma}$$

$$D_{\mathrm{KL}}(p \,\|\, q) = \sum_a p(a) \ln \frac{p(a)}{q(a)}$$

- Solving the optimization problem exactly is prohibitive
- We do some approximations:

Objective function → Approximate to first order (linearize)

Constraint → Approximate to second order

Q: Why approximate constraint to second order?

A: Its first order approximation is zero:

$$\nabla_\theta \left[\sum_s D_{\mathrm{KL}}(\pi_{\theta_k}(\cdot|s) \,\|\, \pi_\theta(\cdot|s))\, \tilde{d}_\mu^{\pi_{\theta_k}}(s) \right]_{\theta=\theta_k} = 0$$

方案 2

$$\theta_{k+1} \approx \arg\max_{\theta\in\Theta} \sum_{s,a} A^{\pi_{\theta_k}}(s,a)\,\pi_\theta(a|s)\,\tilde{d}_\mu^{\pi_{\theta_k}}(s)$$

$$\text{s.t.} \sum_{s} D_{\mathrm{KL}}(\pi_{\theta_k}(\cdot|s) \,\|\, \pi_\theta(\cdot|s))\,\tilde{d}_\mu^{\pi_{\theta_k}}(s) \le \frac{\delta}{1-\gamma}$$

$$D_{\mathrm{KL}}(p \,\|\, q) = \sum_a p(a) \ln \frac{p(a)}{q(a)}$$

approximations

$$\theta_{k+1} \approx \arg\max_{\theta\in\Theta} \left\langle \nabla_\theta L(\theta;\theta_k)|_{\theta=\theta_k}, \theta-\theta_k \right\rangle$$

$$\text{s.t.} \quad \frac{1}{2}(\theta-\theta_k)^\top H_k (\theta-\theta_k) \le \delta$$

where

$$(H_k)_{ij} = \sum_{s,a} \frac{\partial \ln \pi_\theta(a|s)}{\partial\theta_i} \frac{\partial \ln \pi_\theta(a|s)}{\partial\theta_j}\bigg|_{\theta=\theta_k} (1-\gamma) d_\mu^{\pi_{\theta_k}}(s,a)$$

Fisher information matrix

Closed-form solution:

$$\theta_{k+1} = \theta_k + \sqrt{\frac{2\delta}{g_k^\top H_k^{-1} g_k}} H_k^{-1} g_k$$

$$g_k = \nabla_\theta L(\theta;\theta_k)|_{\theta=\theta_k}$$

A version of
natural policy gradient

Trust Region Policy Optimization

1. Under policy π_θ, sample a batch of trajectories $\mathcal{D} = \left\{ \tau^{(i)} = \left(s_t^{(i)}, a_t^{(i)}, r_t^{(i)} \right)_{t \geq 0} \,\middle|\, 1 \leq i \leq |\mathcal{D}| \right\}$
2. Construct the gradient estimate

$$\hat{g} \leftarrow \frac{1}{|\mathcal{D}|} \sum_{i=1}^{|\mathcal{D}|} \sum_{t \geq 0} \gamma^t \hat{A}(\tau^{(i)}; t) \nabla_\theta \ln \pi_\theta \left(a_t^{(i)} | s_t^{(i)} \right)$$

In practice, we don't store the matrix but construct the mapping $x \mapsto \hat{H} x$ from data

3. Construct the Hessian estimate

$$\hat{H}_{jk} \leftarrow \frac{\partial^2}{\partial \theta'_j \, \partial \theta'_k} \left[\frac{1}{|\mathcal{D}|} \sum_{i=1}^{|\mathcal{D}|} \sum_{t \geq 0} (1-\gamma) \gamma^t \sum_a \pi_\theta \left(a | s_t^{(i)} \right) \ln \frac{\pi_\theta \left(a | s_t^{(i)} \right)}{\pi_{\theta'} \left(a | s_t^{(i)} \right)} \right] \Bigg|_{\theta' = \theta}$$

4. Solve $\Delta \leftarrow \hat{H}^{-1} \hat{g}$ Calculated by conjugate gradient descent
5. Update the policy (**actor**) by

$$\theta \leftarrow \theta + \sqrt{\frac{2\delta}{\Delta^\top \hat{H} \Delta}} \cdot \Delta$$

We can use backtracking to ensure the new θ improves the (sampled) objective value and is in the (sampled) trust region

6. Update the value estimation (**critic**) $v_w(s)$ using TD(0)/Monte Carlo/etc.
7. Go back to Step 1 unless stopping criterion is satisfied

怎样刻画 $\pi_\theta \approx \pi_{\theta_k}$？

for $k = 0, 1, 2, \ldots$ **do**

$$\theta_{k+1} \approx \arg\max_{\theta \in \Theta} \; L(\theta; \theta_k)$$

$$\text{s.t.} \quad \pi_\theta \approx \pi_{\theta_k}$$

end for

$$L(\theta; \theta_k) = \sum_{s,a} A^{\pi_{\theta_k}}(s,a)\, \pi_\theta(a|s)\, \tilde{d}_\mu^{\pi_{\theta_k}}(s)$$

- 方案 1：$\frac{1}{2}\|\theta - \theta_k\|^2 \le \delta$ **Advantage Actor Critic**

Trust Region Policy Optimization

- 方案 2：利用 KL-divergence

$$\sum_s D_{\mathrm{KL}}(\pi_{\theta_k}(\cdot|s) \,\|\, \pi_\theta(\cdot|s))\, \tilde{d}_\mu^{\pi_{\theta_k}}(s) \le \frac{\delta}{1-\gamma}$$

$$D_{\mathrm{KL}}(p \,\|\, q) = \sum_a p(a) \ln \frac{p(a)}{q(a)}$$

- 方案 3：Clip the ratio **Proximal Policy Optimization**

$$1 - \epsilon \le \frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)} \le 1 + \epsilon, \quad \forall s \in \mathcal{S}, a \in \mathcal{A}$$

On Policy & Off Policy Methods

- A2C, TRPO, PPO are all **on-policy** methods.
- **On-policy** methods: Evaluate or improve the policy that is used to make decisions and generate the data
- **Off-policy** methods: Evaluate or improve a policy that is different from the one used to generate the data
- Typical off-policy methods: Q-leaning and its variants (DDQN, DDPG, TD3, etc.)
- Q-leaning: Learn the optimal Q-function that satisfies the Bellman optimality equation

$$Q^*(s,a) = \mathbb{E}_{s' \sim P(\cdot|s,a)} \left[r(s,a) + \gamma \max_{a'} Q^*(s',a') \right]$$

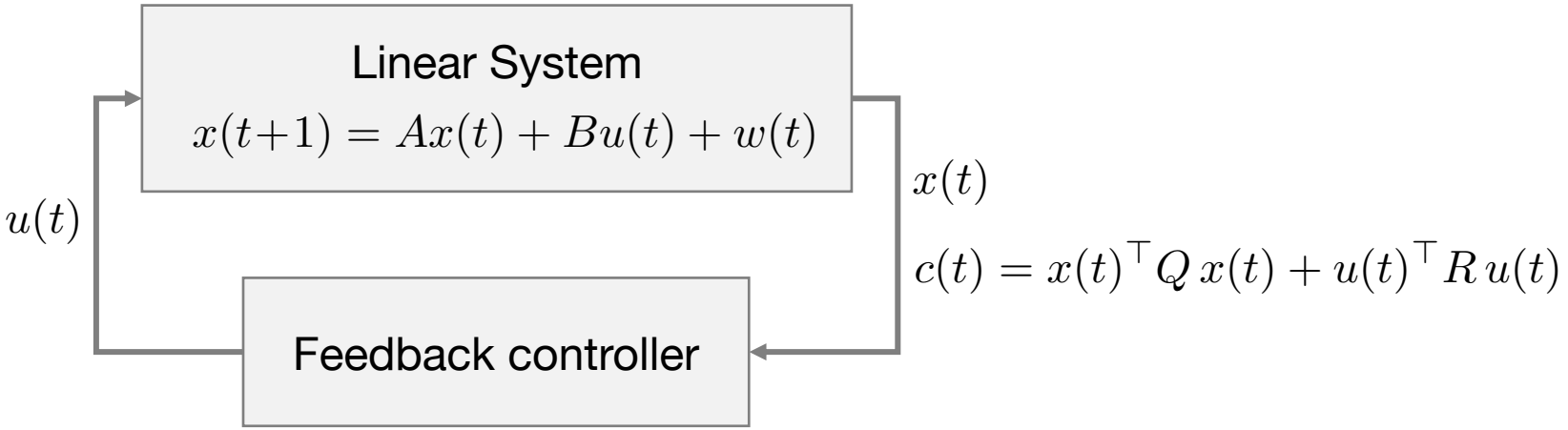
RL Theory

- Sample complexity analysis
- Still many open problems
- Many works focus on the finite-horizon setting
- Algorithms with desirable sample complexities are very different from algorithms used in practice

- `https://rltheorybook.github.io/rltheorybook_AJKS.pdf`

RL for Control Systems

- Recent advances in RL for linear quadratic control



- Convergence rate & sample complexity have been analyzed for algorithms based on zeroth-order optimization

Some RL Resources

- Richard Sutton's classical book

- OpenAI Spinning Up (https://spinningup.openai.com)
- Stable Baselines 3 (https://stable-baselines3.readthedocs.io)

- Slides and lecture notes of RL courses
  - It's interesting to see how different instructors emphasize different aspects of RL