第四讲

蒙特卡洛方法

CMU 10703 Deep Reinforcement Learning & Control

by Professor Ruslan Satakhutdinov

翻译贡献者：

李飞腾，HFUT，Mechatronics (1-9)

李政锴，HIT，CSE (10-16)

王馨，CUHK, rehabilitation robotics (17-21, 39-41)

曹瑾，SJTU，Robotics (22-28)

刘乃龙，SIA, Robotics (29-38)

组长：李宏坤

「机器人学家」授权翻译

目录

蒙特卡洛方法（MC） 2
MC 策略评估 2
First-visit MC 策略评估 3
Every-visit MC 策略评估 3
例子：21 点纸牌游戏 4
● 学习 21 点游戏 状态价值函数 5
蒙特卡洛 backup 图 5
如何方便计算 5
● 递增的均值(Incremental Mean) 5
● 递增的蒙特卡罗更新(Incremental Monte Carlo Updates) 6
动作价值(Q)的蒙特卡洛估计(MC Estimation of Action Values) 6
蒙特卡洛控制(Monte-Carlo Control) 6
贪婪策略(Greedy Policy) 7
蒙特卡洛控制的收敛性 7
● 探索型初始化(Monte Carlo Exploring Starts) 8
举例：21 点游戏 9
同策略 On-policy 蒙特卡罗控制 10
● 同策略(On-policy)蒙特卡罗控制算法实现 11
小结 12
异策略(off-policy)蒙特卡罗控制 13
重要性采样(importance sampling) 13
● 通常的蒙特卡洛采样 13
● 重要性采样 14
● 重要性采样的潜在问题 15
如何使用重要性采样 15
● 重要性采样率 15
● 估计 Value 16
● 普通的重要性采样下方差为无穷大的例子 17
● 单 21 点状态的值的异策略估计例子 18
● 估计 Q 19
异策略 every-visit 蒙特卡洛控制 21

总结 22
拓展：策略形成的所有路径 22
基于模拟的 RL(Simulation-basd RL) 23
基于传统模型的 RL (Conventional Model-based RL) 23

***************李飞腾 P1-9***************

免责说明：本堂课的大部分材料和幻灯片来自于 Rich Sutton 和 Dacid Silver 的增强学习课堂。

蒙特卡洛方法（MC）

- 蒙特卡洛方法是一种学习方法

  经验价值策略

- 蒙塔卡罗方法使用最简单的想法：价值 = 平均返回值(return)
- 有两种方式来应用蒙特卡洛方法

  Model-free:没有必要的模型，但仍然能得到最优性（Optimality）

  Simulated:可以在仿真环境下进行采样估计，而不一定需要在实际环境下。

- MC 方法从完整样本返回值中学习

  只对已定义的周期任务（本课堂）

  所有周期必须可中止（No bootstrapping）

MC 策略评估

- 目标：在策略 π 下从经验周期中学习 Vπ（s）

  S1,A1,R2,…,Sk~π

- 记下整个折扣奖励的返回值

$$G_t = R_{t+1} + \gamma R_{t+2} + \ldots + \gamma^{T-1} R_T$$

- 记下价值函数是期待的返回值

$$v_\pi(s) = \mathbb{E}_\pi [G_t \mid S_t = s]$$

- MC 策略评估使用经验平均返回值而不是预期返回值
- 想法：访问 s 后观察到的平均返回值
- Every-visit MC：对 s 的每次访问得到的返回值的平均值
- First-visit MC： 每个周期内首次访问得到的平均返回值
- 都是逐渐收敛的

First-visit MC 策略评估

- 评估状态 s
- 在一个周期内访问状态 s 的第一个时间步长 t
- 增加计数器：

$$N(s) \leftarrow N(s) + 1$$

- 增加返回值：

$$S(s) \leftarrow S(s) + G_t$$

- 平均返回值：

$$V(s) = S(s)/N(s)$$

- 根据大数定理：

$$V(s) \rightarrow v_\pi(s) \text{ as } N(s) \rightarrow \infty$$

Every-visit MC 策略评估

- 评估状态 s
- 在一个周期内访问状态 s 的每一次时间步长 t
- 增加计数器：

$$N(s) \leftarrow N(s) + 1$$

- 增加返回值：

$$S(s) \leftarrow S(s) + G_t$$

- 平均返回值：

$$V(s) = S(s)/N(s)$$

- 根据大数定理：

$$V(s) \to v_\pi(s) \text{ as } N(s) \to \infty$$

例子：21 点纸牌游戏

- 目标：使你手中牌的点数之和大于庄家的点数，且不能超过 21 点
- 规则：庄家会展示前两张牌中的一张（是 A 到 10），A（Ace）可以当 1，也可以当 11，玩家自己定义。玩家可以选择不断要牌或停牌。停牌后庄家摊牌，开始比牌：自己的牌小于等于 21 就跟庄家比大，大于庄家则赢；大于 21 点就爆了，即输。
- 状态（200）

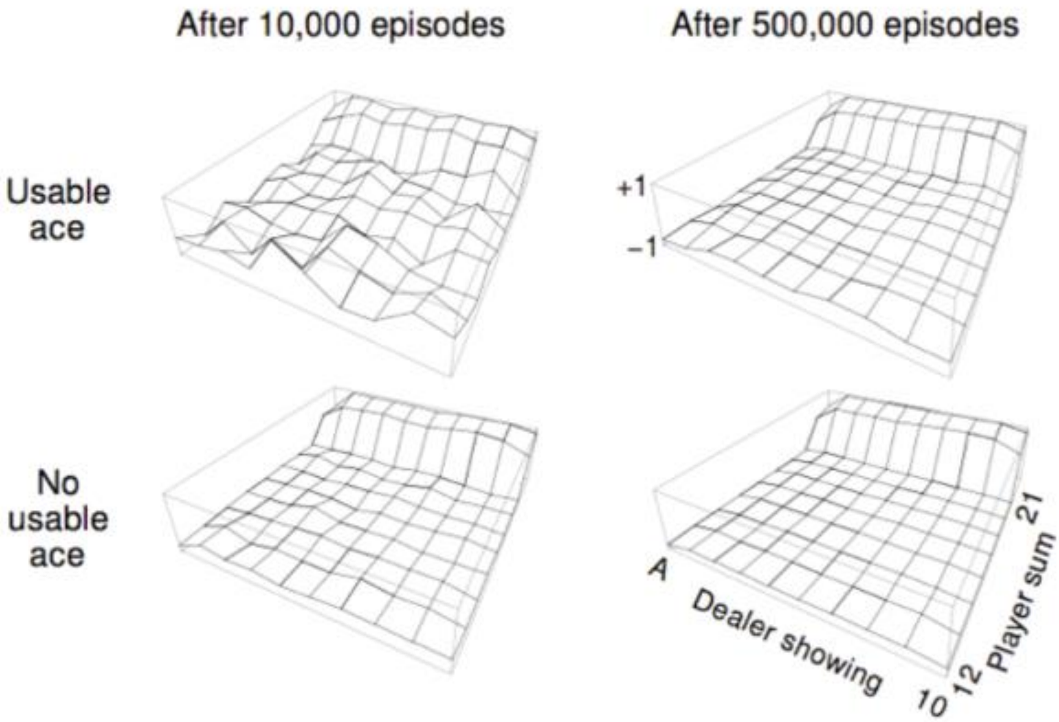
  现在总和（12-21）

  庄家的牌（ace-10）

  我有可用的 ace 吗？

- 奖励：赢者+1，平得 0，输者-1
- 动作：stick(停止接受牌),hit（继续接受牌）
- 策略：如果我的牌点数之和是 20 或 21，stick；否则 hit.
- 没有折扣（discounting）:$\gamma = 1$

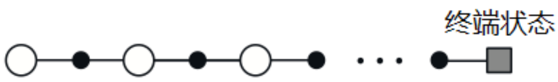
- 学习 21 点游戏 状态价值函数



/********** 李政锴(10-16) **********/

蒙特卡洛 backup 图

- ➢ 包括了一次试验中，从当前状态之后的所有状态。
- ➢ 在每个状态下，仅仅考虑一种选择（不同于 DP（动态规划））。
  - ○ 故将出现进行探索 explore(try different strategy)还是开发利用 exploit(use the best known strategy)的困境
- ➢ 不用后续状态的价值进行自举采样 bootstrap（不同于 DP）。
- ➢ 用平均返回值估计价值。
- ➢ 估计一个状态用时不取决于状态总数。



如何方便计算

- 递增的均值(**Incremental Mean)**

- ➢ 要解决的问题：在新数据一个一个到来的情况下，如何方便地计算均值；

➢ 序列 x1,x2...的均值 u1,u2...可以递增地被算得：

$$\begin{aligned}\mu_k &= \frac{1}{k}\sum_{j=1}^{k} x_j \\ &= \frac{1}{k}\left(x_k + \sum_{j=1}^{k-1} x_j\right) \\ &= \frac{1}{k}\left(x_k + (k-1)\mu_{k-1}\right) \\ &= \mu_{k-1} + \frac{1}{k}\left(x_k - \mu_{k-1}\right)\end{aligned}$$

- 递增的蒙特卡罗更新(**Incremental Monte Carlo Updates)**

➢ 在周期$S_1, A_1, R_2, ..., S_T$ 之后，递增地更新 V(s)。

➢ 对于每个伴随着返回值 Gt 的状态 St。

$$\begin{aligned}&N(S_t) \leftarrow N(S_t) + 1 \\ &V(S_t) \leftarrow V(S_t) + \frac{1}{N(S_t)}\left(G_t - V(S_t)\right)\end{aligned}$$

➢ 在非静态问题中，跟踪一个滑动平均值 a running mean 将很有用，即忘掉旧的周期。

$$V(S_t) \leftarrow V(S_t) + \alpha\left(G_t - V(S_t)\right)$$

动作价值(Q)的蒙特卡洛估计(MC Estimation of Action Values)

➢ 在模型不可获取的情况下，使用蒙特卡洛是最有效的。

 ○ 我们想要习得 q*(s,a)

➢ qπ(s,a) - 从状态 s 和动作 a 下采用策略 π 起始，获得的平均返回值。

$$\begin{aligned}q_\pi(s,a) &= \mathbb{E}_\pi[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s, A_t = a] \\ &= \sum_{s',r} p(s',r|s,a)\Big[r + \gamma v_\pi(s')\Big].\end{aligned}$$

➢ 如果每个状态-动作对被访问，将渐进收敛。

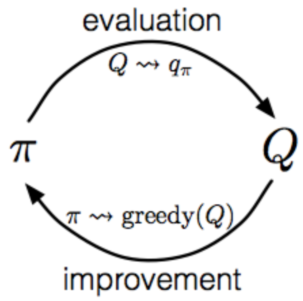
➢ 探索开始(Exploring starts)：每个状态-动作对都有一定几率作为起始对。

蒙特卡洛控制(**Monte-Carlo Control)**

➢ 前面都是评估(evaluation)，即给定一个策略(policy)，估计在该策略下的价值(value)。

- 完整的策略迭代包括评估和控制(control)二者，你需要一边估计当前策略的价值，一边设法根据当前的估计来改进策略。
- 对于蒙特卡洛方法来说，这个过程称为蒙特卡洛控制。

$$\pi_0 \xrightarrow{\text{E}} q_{\pi_0} \xrightarrow{\text{I}} \pi_1 \xrightarrow{\text{E}} q_{\pi_1} \xrightarrow{\text{I}} \pi_2 \xrightarrow{\text{E}} \cdots \xrightarrow{\text{I}} \pi_* \xrightarrow{\text{E}} q_*$$



- 蒙特卡洛策略迭代步：使用蒙特卡洛方法进行策略评估，随后进行策略改进。
- 策略改进步：针对价值（或者是动作价值）的贪婪函数。

贪婪策略(Greedy Policy)

- 对于任何动作价值函数 q，相应的贪婪策略如下：
  - 对于每个状态 s，通过最大化动作价值，来确定如何选择动作

$$\pi(s) \doteq \arg\max_a q(s,a).$$

- 使用针对$q_{\pi k}$的贪婪策略构建π_{k+1}，即可完成策略改进。
- 贪婪策略是一种最简单的方式。

蒙特卡洛控制的收敛性

- 贪婪策略满足策略改进的条件

$$\begin{aligned} q_{\pi_k}(s, \pi_{k+1}(s)) &= q_{\pi_k}(s, \arg\max_a q_{\pi_k}(s,a)) \\ &= \max_a q_{\pi_k}(s,a) \\ &\geq q_{\pi_k}(s, \pi_k(s)) \\ &\geq v_{\pi_k}(s). \end{aligned}$$

- 因此必有≥ πk
- 这个证明的一个假设是每个状态都被经历过。
- 保证上述假设的方式是探索开始(Exploring start)且蒙特卡洛策略评估迭代无穷个周期。

/********** 王馨 (17-21) **********/

- ## 探索型初始化(Monte Carlo Exploring Starts)

对蒙特卡罗策略评估(policy evaluation)来说, 在基于一个试验接试验(episode-by-episode) 的评估(evaluation)和改进(improvement)间交替是很自然的。在每一个试验(episode)之后，观测到的回报(observed returns)被用于策略评估。然后该策略就能在这个试验中所有被访问过的状态(states)下进行改进(Sutton 2017)。一个完整的简单的算法过程如下（后附图片版）：

初始化所有的 $s \in \mathcal{S}, a \in \mathcal{A}(s)$：

$Q(s,a) \leftarrow arbitrary$任意取值

$\pi(s) \leftarrow arbitrary$任意取值

$Returns(s,a) \leftarrow empty\ list$赋以空列表

循环：

随机选择 $S_0 \in \mathcal{S}, A_0 \in \mathcal{A}(S_0)$, 所有状态-动作对(state-action pairs)的概率都大于 0

从S_0, A_0开始生成一个试验(episode)，以策略 π进行

对每一对在这个试验中出现的状态和动作，s, a：

$G \leftarrow s, a$第一次出现后的回报(return)

将G附加于回报$Returns(s,a)$上

$Q(s,a) \leftarrow average(Returns(s,a))$对回报取均值

对该试验中的每一个s:

$\pi(s) \leftarrow arg\ max_a Q(s,a)$ 取最大值

初始化所有的 $s \in \mathcal{S}, a \in \mathcal{A}(s)$：

　　$Q(s,a) \leftarrow arbitrary$ 任意取值

　　$\pi(s) \leftarrow arbitrary$ 任意取值

　　$Returns(s,a) \leftarrow empty\ list$ 赋以空列表

循环：

　　随机选择 $S_0 \in \mathcal{S}, A_0 \in \mathcal{A}(S_0)$，所有状态-动作对(state-action pairs)的概率都大于0

　　从 S_0, A_0 开始生成一个试验(episode)，以策略 π 进行

　　对每一对在这个试验中出现的状态和动作，s, a：

　　　　$G \leftarrow s, a$ 第一次出现后的回报(return)

　　　　将 G 附加于回报 $Returns(s,a)$ 上

　　　　$Q(s,a) \leftarrow average(Returns(s,a))$ 对回报取均值

　　对该试验中的每一个 s:

　　　　$\pi(s) \leftarrow arg\ max_a Q(s,a)$ 取最大值

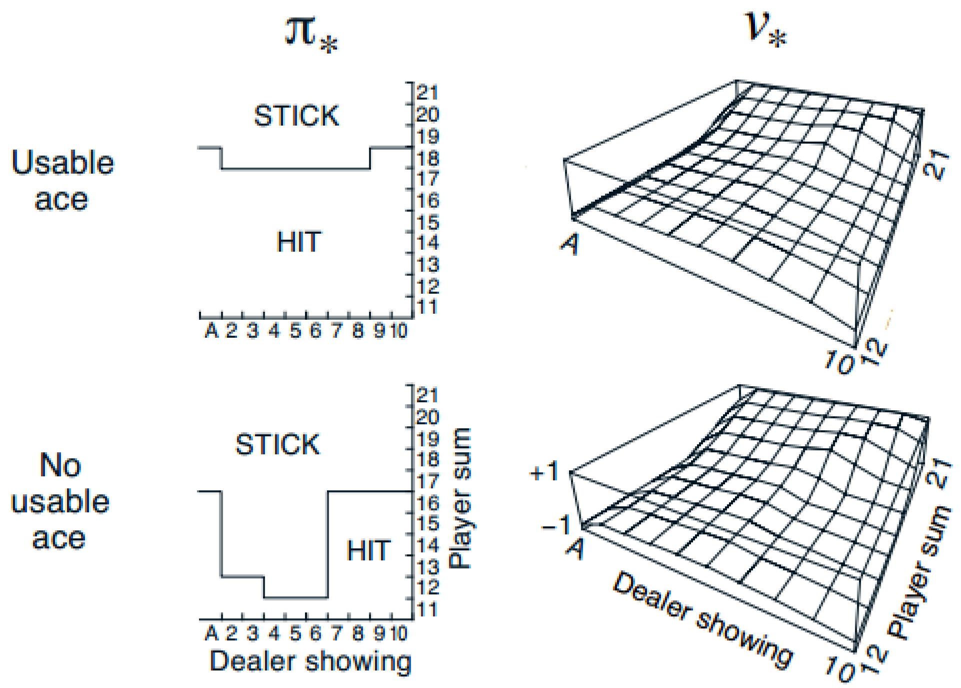
在该算法中，每个状态-动作对(state–action pair)的所有回报(returns)都会被累积(accumulated)和平均(averaged)，不管当前他们观测到的是哪种在执行的策略。

很明显蒙特卡罗探索型初始化算法不可能收敛到一个次优的策略，假如它能够收敛到一个次优的策略的话，那么价值函数(value function)将会最终收敛到该次优策略下的价值函数，这样反过来又会改变这个次优策略。仅当策略和价值函数同时达到最优时，结果达到稳定。

随着动作-价值函数(action-value function)的改变越来越少，收敛到这个最优的固定点(fixed point，即最优策略π*)似乎是必然的，但这个问题还没有被正式证明过(Sutton 2017)。

举例：21 点游戏

由于试验都是模拟的游戏，很容易让探索型初始化(exploring starts)包含所有的可能性。在这种情况下，我们能够假设出庄家(dealer)展示出的牌、玩家的总和，以及玩家是否有可用的 A 牌的所有情况，这些值都是任意且等概率的。我们用之前 21 点例子中的策略作为初始策略，即仅当在 20 或 21 时 stick。所有状态-动作对(state–action pair)的初始动作-价值函数(action-value function)都可设为 0，下图展示了通过探索型初始化算法找到的 21 点游戏的最优策略。



同策略 On-policy 蒙特卡罗控制

- 同策略(On-policy)：从当前执行的策略中学习 （与 off-policy 相对，之后会讲）
- 我们如何摆脱探索型初始化算法？（译者注：探索型初始化蒙特卡罗控制算法是在一定假设(assumption)情况下得到的。这种假设就是所有动作(actions)都被无限频繁选中。此处即指摆脱这种假设）

  答：策略必须是永久温和(eternally soft)的，即：

  对所有的s和a都满足：$\pi(a|s) > 0$

- 例如，对ε- soft策略，某一动作的概率，$\pi(a|s)$等于

$$\frac{\epsilon}{|\mathcal{A}(s)|} \quad 或 \quad 1-\epsilon+\frac{\epsilon}{|\mathcal{A}(s)|}$$

(非最大)　（最大(贪婪策略)）

这里贪婪是指，大多数时候策略都会选择有最大估计动作值(maximal estimated action value)的动作(action)，但是仍然有ϵ这么大的概率去任意选择一个动作。其中所有的非贪婪动作都予以$\frac{\epsilon}{|\mathcal{A}(s)|}$这么小的概率，而贪婪动作则予以$1-\epsilon+\frac{\epsilon}{|\mathcal{A}(s)|}$这样大的概率。

- 与一般策略迭代(generalized policy iteration (GPI))相似，即：

  将策略逐渐变化移至贪婪策略，而不是整个过程一直采用贪婪策略。在 on-policy 方式中就是将一策略逐渐移至ϵ-贪婪 策略。

- 能收敛到最佳ε- soft策略

注：去除前面的假设的唯一方法就是，让智能体连续去选择这些动作(actions)。这种方法有两种形式去实现，一种是 on-policy 的，另一种是 off-policy 的。

On-policy 方式希望评估(evaluate)或者改进(improve)当前被用来做决定的策略，而 off-policy 方式则是评估或者改进一个不同于当前做决定的策略的策略(Sutton 2017)。

上面提出的探索型初始化蒙特卡罗方法就是一种 on-policy 的方式。接下来的算法是在没有探索型初始化这个假设下的 on-policy 蒙特卡罗控制方法。

- ## 同策略(On-policy)蒙特卡罗控制算法实现

初始化所有的 $s \in \mathcal{S}, a \in \mathcal{A}(s)$：

$Q(s,a) \leftarrow \text{arbitrary}$任意取值

$Returns(s,a) \leftarrow \text{empty list}$ 空列表

$\pi(a|s) \leftarrow$任意的一个ε- soft策略

循环：

(a) 用策略π生成一次试验

(b) 对每一对在试验中出现的s, a：

$G \leftarrow s, a$第一次出现后的回报(return)

将G附加于回报$Returns(s,a)$上

$Q(s,a) \leftarrow average(Returns(s,a))$对回报取均值

(c) 对试验中的每一个状态s:

$A^* \leftarrow arg\ max_a Q(s,a)$ 表最大的均值回报

对所有的$a \in \mathcal{A}(s)$:

$$\pi(a|s) \leftarrow \begin{cases} 1-\epsilon+\frac{\epsilon}{|\mathcal{A}(s)|} & \text{if } a = A^* \\ \frac{\epsilon}{|\mathcal{A}(s)|} & \text{if } a \neq A^* \end{cases}$$

初始化所有的 $s \in \mathcal{S}, a \in \mathcal{A}(s)$：

$Q(s,a) \leftarrow \text{arbitrary}$ 任意取值

$Returns(s,a) \leftarrow \text{empty list}$ 空列表

$\pi(a|s) \leftarrow$ 任意的一个 ε- soft 策略

循环：

(a) 用策略 π 生成一次试验

(b) 对每一对在试验中出现的 s, a：

$G \leftarrow s, a$ 第一次出现后的回报(return)

将 G 附加于回报 $Returns(s,a)$ 上

$Q(s,a) \leftarrow average(Returns(s,a))$ 对回报取均值

(c) 对试验中的每一个状态 s:

$A^* \leftarrow arg\ max_a Q(s,a)$ 表最大的均值回报

对所有的 $a \in \mathcal{A}(s)$:

$$\pi(a|s) \leftarrow \begin{cases} 1-\epsilon+\frac{\epsilon}{\mathcal{A}(s)} & \text{if } a = A^* \\ \frac{\epsilon}{\mathcal{A}(s)} & \text{if } a \neq A^* \end{cases}$$

小结

- MC 相对于 DP(Dynamic Programming, 动态规划)具有很多优点：
  - 可以直接从环境交互中学习(interaction with environment)
  - 不需要完整的模型
  - 不需要学习所有的状态（即不需要引导(bootstrapping)）
  - 能较少地受到违背了马尔科夫特性(Markov property，之后会讲)带来的影响
- MC 方法提供了一种交替策略评估过程(alternate policy evaluation process)
- 需要注意的一个问题： 需维持足够的探索(maintaining sufficient exploration):
  探索型初始化(exploring starts)，软策略(soft policies)

为了让策略评估能效力于动作值(action value)，我们必须确保连续的探索，以上两者都是以这个为前提的。

译者注：在 RL 中，bootstrapping 指某一估计值的更新是基于其他的估计值。详见 https://omarsbrain.wordpress.com/2010/01/22/bootstrapping-and-artificial-intelligence/

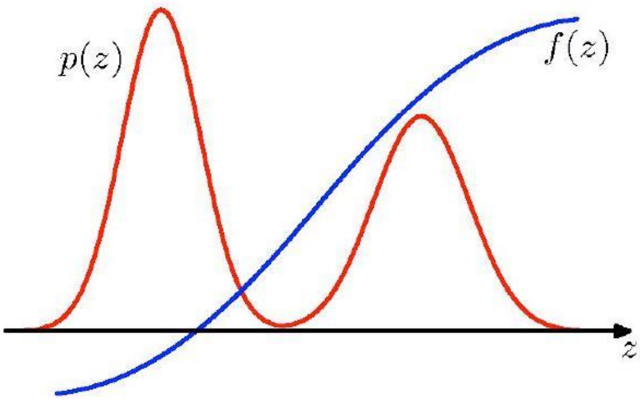
/********** 曹瑾 (22-28) **********/

异策略(off-policy)蒙特卡罗控制

- 为了保证探索性，学习或评估的目标策略 π 与使用的行为策略 μ 不同，根据行为策略 μ 中获得的经验更新目标策略 π 的值
- 举例：π 是贪婪策略（最终的最优策略），μ 是探索策略（如 ε-soft 策略）
- 总的来说，只需要满足覆盖性(coverage)，即：μ 产生的行为覆盖或包含 π 可能产生的行为（满足 π(a|s)>0 的任何 s,a 均满足 μ(a|s)>0）
- 思想：重要性采样(importance sampling)
  - 给每个回报(return)赋以一定的权重，该权重为两个策略下轨迹可能性的比值(ratio of probability)

重要性采样(importance sampling)

- 通常的蒙特卡洛采样



$$\mathbb{E}[f] = \int f(z)p(z)dz \approx \frac{1}{N}\sum_{n=1}^{N} f(z^n) = \hat{f}.$$

- 总体思想：从分布 p(z)中得到独立采样$\{z^1,\ldots,z^n\}$从而近似期望

注意：

$$\mathbb{E}[f] = \mathbb{E}[\hat{f}].$$

所以估计量具有正确的均值（无偏）

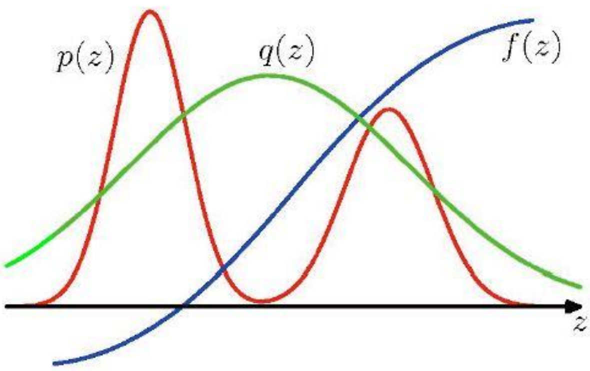
- 方差：

$$\mathrm{var}[\hat{f}] = \frac{1}{N}\mathbb{E}[(f - \mathbb{E}[f])^2].$$

- 方差与 1/N 成正比，随样本量增加方差减小
- 估计量的精度不依赖于 z 的维度
- 较少的独立采样次数 N 也可能从分布 p(z)中得到较高的估计精度

- 简单的采样估计方法有两个问题：
  - **问题 1**：我们可能得不到独立的样本
  - **问题 2**：如果在 p(z)较小的区域 f(z)较大（反之亦然），那么期望可能会被概率小的区域主导，因而需要更大的样本

- ## 重要性采样
- 基于一个分布的采样来估计另一个分布下的期望，从而解决直接采样的困难
- 假设我们提出一个容易采样的分布 q(z)，



$$\mathbb{E}[f] = \int f(z)p(z)dz$$
$$= \int f(z)\frac{p(z)}{q(z)}q(z)dz$$
$$\approx \frac{1}{N}\sum_n \frac{p(z^n)}{q(z^n)}f(z^n), \quad z^n \sim q(z).$$

满足当 p(z)>0 时 q(z)>0

其中 $p(z^n)$指的是 z 按照 p 分布时，出现 $z^1,z^2,\ldots,z^n$ 的概率

- 那么定义 $\omega^n=p(z^n)/q(z^n)$为重要性权重
- 令我们提出的分布 q(z)具有形式$q(z) = \tilde{q}(z)/\mathcal{Z}_q.$（译者注：其中 Z_q 为 q~关于 z 的积分值，

$$\mathbb{E}[f] = \int f(z)p(z)dz \quad = \int f(z)\frac{p(z)}{q(z)}q(z)dz = \frac{\mathcal{Z}_q}{\mathcal{Z}_p}\int f(z)\frac{\tilde{p}(z)}{\tilde{q}(z)}q(z)dz$$
$$\approx \frac{\mathcal{Z}_q}{\mathcal{Z}_p}\frac{1}{N}\sum_n \frac{\tilde{p}(z^n)}{\tilde{q}(z^n)}f(z^n) = \frac{\mathcal{Z}_q}{\mathcal{Z}_p}\frac{1}{N}\sum_n w^n f(z^n),$$

保证 q 关于 z 的积分为 1，对 p 同理，推导中会把 Z_p,Z_q 消掉，故选取 q~时可以只考虑分布的形状）

- 我们可以用同样的权重（即重要性权重）去近似 Z_q/Z_p
- 因此

$$\mathbb{E}[f] \approx \sum_{n=1}^{N} \frac{w^n}{\sum_{m=1}^{N} w^m} f(z^n), \qquad z^n \sim q(z).$$

$$\frac{\mathcal{Z}_p}{\mathcal{Z}_q} = \frac{1}{\mathcal{Z}_q}\int \tilde{p}(z)dz = \int \frac{\tilde{p}(z)}{\tilde{q}(z)}q(z)dz \approx \frac{1}{N}\sum_n \frac{\tilde{p}(z^n)}{\tilde{q}^n(z)} = \frac{1}{N}\sum_n w^n.$$

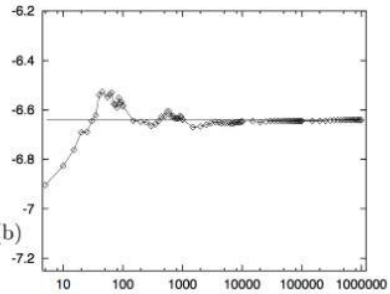
- 重要性采样的潜在问题

例子：

$$\hat{f} = \sum_{n=1}^{N} \frac{w^n}{\sum_{m=1}^{N} w^m} f(z^n), \qquad \mathbb{E}[f] = \int f(z)\frac{p(z)}{q(z)}q(z)dz$$

- 用重要性采样的方法，很难评价估计量的可靠性
- 如果在|f(z)p(z)|较大的区域中提出的分布 q(z)较小将会产生巨大的方差

- 采用高斯分布作为 q(z)（一维的情况）
    - 即使在一百万次采样之后，估计量也没有收敛到真值上



(b)

- 采用柯西分布作为 q(z)（一维的情况）
- 500 次采样过后，估计量出现收敛
- 提出的分布 q(z)应为重尾(heavy tails)分布

/*********** 曹瑾 (22-28)DRAFT END ***********/

/*********************************** START P29-38 **/

如何使用重要性采样

- 重要性采样率
- 在策略 π 下，t 时刻的状态 S 后，剩余轨迹的概率为

$$\begin{aligned}
&\Pr\{A_t, S_{t+1}, A_{t+1}, \ldots, S_T \mid S_t, A_{t:T-1} \sim \pi\} \\
&\quad = \pi(A_t \mid S_t) p(S_{t+1} \mid S_t, A_t) \pi(A_{t+1} \mid S_{t+1}) \cdots p(S_T \mid S_{T-1}, A_{T-1}) \\
&\quad = \prod_{k=t}^{T-1} \pi(A_k \mid S_k) p(S_{k+1} \mid S_k, A_k)
\end{aligned}$$

- **重要性采样**：在重要性采样中，在目标和行为策略的条件下，每个回报都使用轨迹的相对概率进行加权。

$$\rho_t^T = \frac{\prod_{k=t}^{T-1} \pi(A_k \mid S_k) p(S_{k+1} \mid S_k, A_k)}{\prod_{k=t}^{T-1} \mu(A_k \mid S_k) p(S_{k+1} \mid S_k, A_k)} = \prod_{k=t}^{T-1} \frac{\pi(A_k \mid S_k)}{\mu(A_k \mid S_k)}$$

这个比率就是**重要性采样率**。

- 所有的重要性采样率都有期望值 1。

$$\mathbb{E}_{A_k \sim \mu}\left[\frac{\pi(A_k \mid S_k)}{\mu(A_k \mid S_k)}\right] = \sum_a \mu(a \mid S_k) \frac{\pi(a \mid S_k)}{\mu(a \mid S_k)} = \sum_a \pi(a \mid S_k) = 1.$$

需要注意的是，重要性采样会有较高（甚至是无穷大）的方差。

● 估计 Value

普通的重要性采样估计

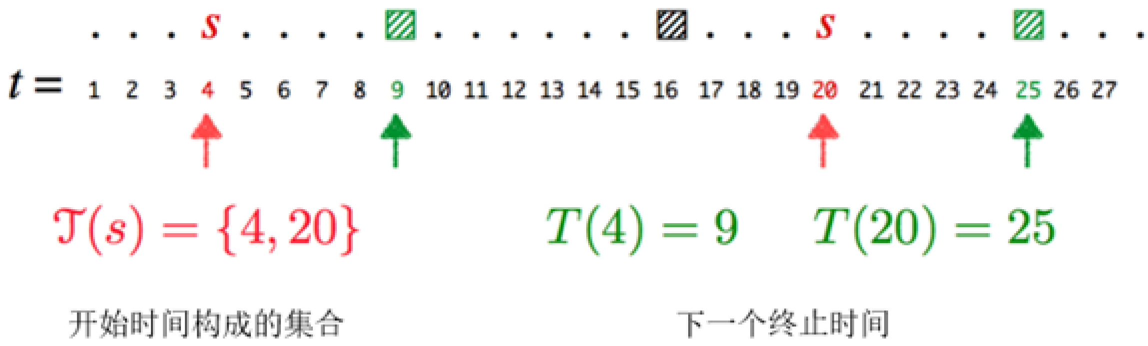
时间 t 后的第一次终止时刻

从 t 到 T(t) 的返回

$$V(s) \doteq \frac{\sum_{t \in \mathcal{T}(s)} \rho_t^{T(t)} G_t}{|\mathcal{T}(s)|}.$$

每个时刻：所有状态 s 被访问过的时间步的集合

注：跨过边界到下一个试验时，时间步将继续增加

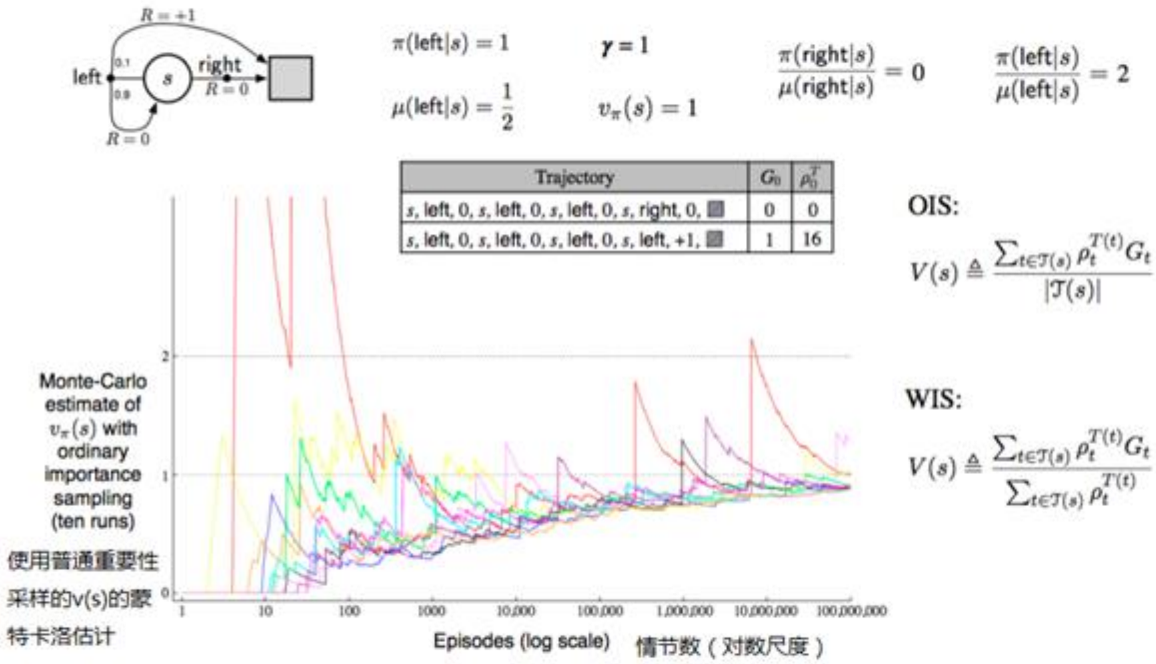


译者注：为了方便重要性采样，需要将各次试验的时间连起来。

加权的重要性采样估计

$$V(s) \doteq \frac{\sum_{t \in \mathcal{T}(s)} \rho_t^{T(t)} G_t}{\sum_{t \in \mathcal{T}(s)} \rho_t^{T(t)}}$$

- 普通的重要性采样下方差为无穷大的例子



其中，状态转移图为图中左上角部分。分布和折扣系数如图中上方所示。

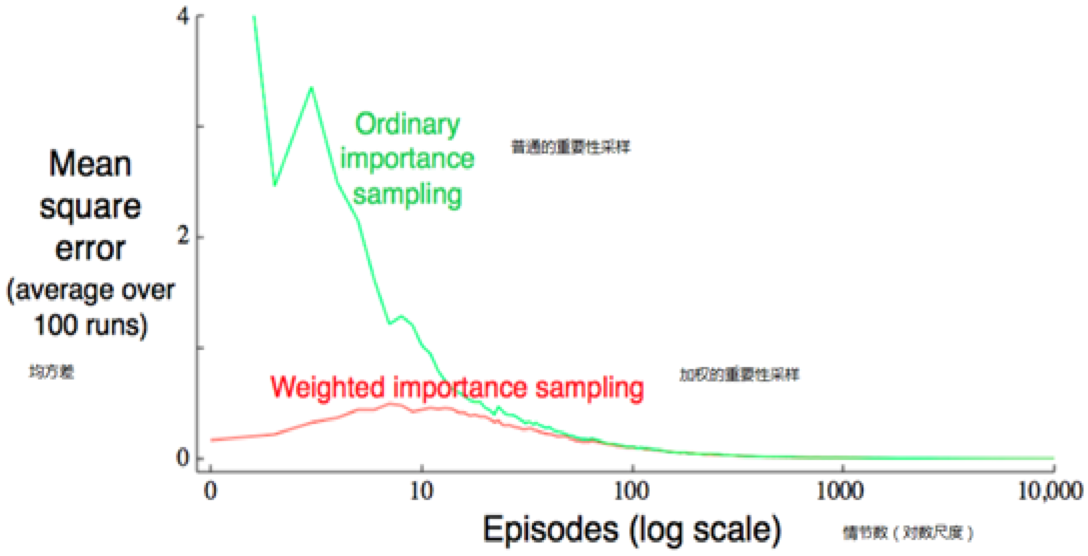
• 单 21 点状态的值的异策略估计例子

这里的状态为：玩家数 13，庄家展示 2，可用的王牌

目标策略是保持在 20 或者 21 上。

行为策略是等概率的。

真实值约为-0.27726。



Mean square error (average over 100 runs)
均方差
Ordinary importance sampling
普通的重要性采样
Weighted importance sampling
加权的重要性采样
Episodes (log scale)
情节数（对数尺度）
4
2
0
0
10
100
1000
10,000

- 估计 Q

输入：任意目标策略π

初始化，对所有的$s \in S$, $a \in A(s)$:

$Q(s,a) \leftarrow$ 任意

$C(s,a) \leftarrow 0$

重复如下步骤：

$\mu \leftarrow$ 任意覆盖π的策略

使用μ产生一个情节：

$S_0, A_0, R_1, ..., S_{T-1}, A_{T-1}, R_T, S_T$

$G \leftarrow 0$

$W \leftarrow 1$

对$t = T-1, T-2, \ldots, 0$:

$G \leftarrow \gamma G + R_{t+1}$

$C(S_t, A_t) \leftarrow C(S_t, A_t) + W$

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \frac{W}{C(S_t, A_t)}[G - Q(S_t, A_t)]$$

$$W \leftarrow W \frac{\pi(A_t \mid S_t)}{\mu(A_t \mid S_t)}$$

如果$W = 0$,则退出循环。

异策略 every-visit 蒙特卡洛控制

初始化，对所有的$s \in S$，$a \in A(s)$：

$Q(s,a) \leftarrow$ 任意

$C(s,a) \leftarrow 0$

$\pi(s) \leftarrow \arg\max_a Q(s,a)$

重复如下步骤：

$\mu \leftarrow$ 任意软策略

使用μ产生一个情节：

$S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T, S_T$

$G \leftarrow 0$

$W \leftarrow 1$

对$t = T-1, T-2, \ldots, 0$：

$G \leftarrow \gamma G + R_{t+1}$

$C(S_t, A_t) \leftarrow C(S_t, A_t) + W$

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \frac{W}{C(S_t, A_t)}[G - Q(S_t, A_t)]$$

$\pi(S_t) \leftarrow \arg\max_a Q(S_t, a)$ （一致地）

如果 $A_t \neq \pi(S_t)$,则退出循环。

$$W \leftarrow W \frac{1}{\mu(A_t \mid S_t)}$$

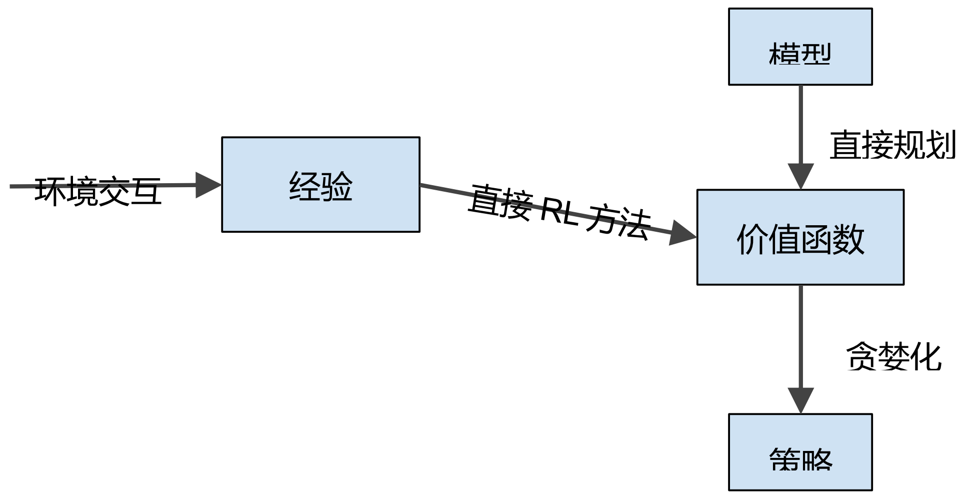
注：目标策略是贪心的确定的。行为策略是软的，典型的是 ε-greedy。

总结

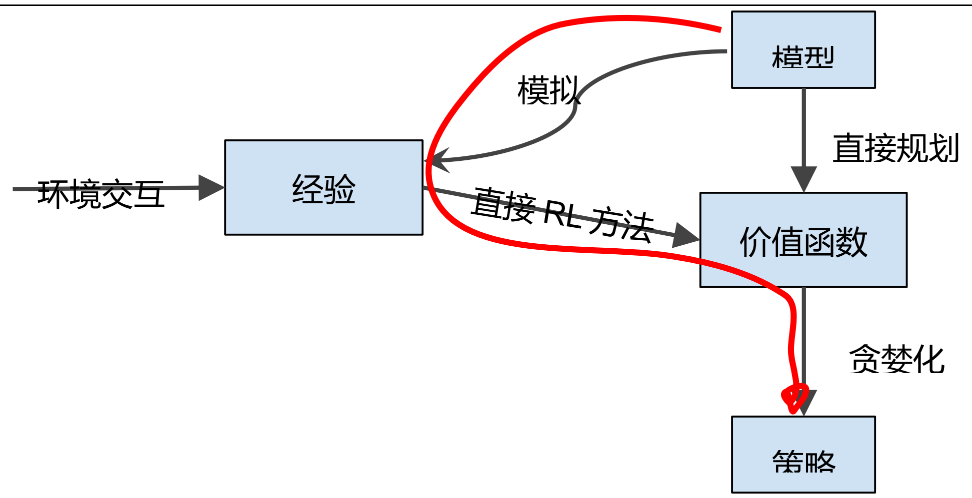
- MC 相对于 DP 有很多优势：
  - 可以直接从环境的交互中学习
  - 不需要完整的模型
  - 通过违反 Markov 性获得较少的损失（后续课程介绍）
- MC 方法提供了另一种策略评估流程
- 一个值得注意的问题是：保持充分的探索
  - 可以直接从与环境的交互中进行学习
- 介绍了区别同策略和异策略方法的不同
- 介绍了异策略学习中的重要性采样
- 介绍了一般和加权重要性采样的不同

/********** 王馨 (39-41) **********/

拓展：策略形成的所有路径



基于模拟的 RL(Simulation-basd RL)



基于传统模型的 RL (Conventional Model-based RL)