Chapter 5

Monte Carlo Methods

5.1 Blackjack

5.1.1 问题描述

Blackjack 是简化版的 21 点问题，规则如下：

- 牌面大小范围是 $[1, 10]$，花牌 (J,Q,K,Joker) 被认为是十点。
- Ace 是一张特殊的牌，玩家 (player) 可以决定 Ace 代表一点还是十点。
- 游戏开始时玩家 (player) 和庄家 (dealer) 各拿两张牌，庄家展示手中的一张牌。
- 玩家 (player) 可以考虑再拿一张牌 (hit)，或者保留当前手牌状态 (strike) 并结束自己的回合；若当前手牌之和小于 12，则无条件进行一次 hit。
- 庄家 (dealer) 可以考虑再拿一张牌 (hit)，或者保留当前手牌状态 (strike) 并结束自己的回合；若当前手牌之和小于 12，则无条件进行一次 hit。
- 无论是庄家还是玩家，如果手牌点数之和大于 21，那么称为手牌爆炸 (bust), 直接输掉这局游戏。

- 若玩家和庄家都选择 strike，并且没有 bust，那么根据两者手牌点数之和比较大小，决定胜负平。(点数之和大者获胜)
- 若玩家 (player) 胜利，则 reward=+1；若平局，则 reward=0；若失败，则 reward=-1。

5.1.2 问题分析

本题使用蒙特卡洛抽样，根据大数定律，抽样的期望可以近似看作全部样本的期望。首先我们需要模拟一局游戏，获得该游戏的轨迹。由于本题逻辑复杂，涉及很多 3D 图。所以参考github上的代码，写成了如下代码。

定义：

- state=[玩家是否使用 Ace 当作 11，游戏结束时玩家的手牌总和，庄家亮出的第一张牌的大小]；
- reward=[-1,0,1], 分别代表负/平/胜；
- 蒙特卡洛游戏轨迹 (episode)=[(hit/strike),state]

这样，我们就获得了蒙特卡洛所有需要的参数。根据不同的策略选择不同的参数更新策略。

5.1.3 实验结果

on-policy

on-policy 是十分简单的蒙特卡洛学习法，它的思想就是大量进行蒙特卡洛采样，根据伯努利大数定律:

$$\lim_{n\to\infty} P(\left|\frac{\mu_n}{n} - p\right| < \epsilon) = 1 \tag{5.1}$$

上式表明若 n 足够大的时候，事件 A 发生的频率将接近它的概率。换句话说，大量采样的期望逼近样本的期望。

当前的 value 的更新可以采用增量法：

$$V_{t+1}(s) = \frac{n \times V_t(s)}{n+1} \tag{5.2}$$

基于这个公式，我们可以得到书上的图 5.1：

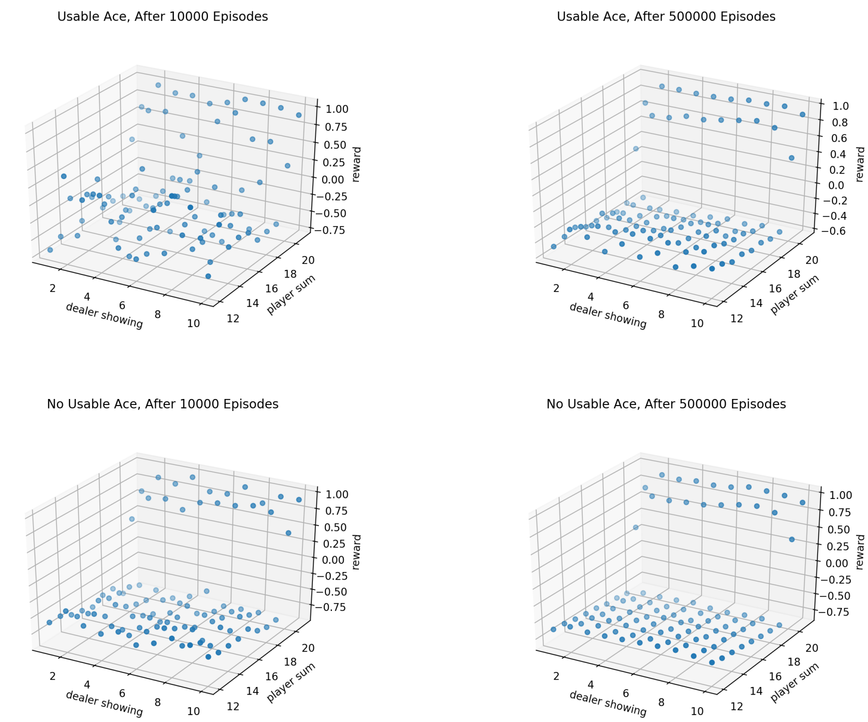


图 5.1: 蒙特卡洛同策略算法

上图我们可以得到 Blackjack 游戏的四个规律：

1. 随着轨迹 (episodes) 的增加，reward 将趋近于稳定，符合大数定律。
2. 在采样数相同的情况下，有 Ace 的情况波动较大，说明不确定性更加高。换句话说，策略选择越多，则需要更多的采样才能使得 reward 趋于稳定。
3. 在 Blackjack 游戏中，庄家亮的牌对于最后的 reward 没有影响，只是一个游戏中的干扰项。
4. 玩家可以在手中点数较少时选择 hit 来增加自己 reward 的期望，但是随着玩家手牌的增多，hit 导致 bust 的概率会增大，从而导致 reward

期望降低，但是若玩家手牌之和超过 20 点，那么胜利的概率会非常大。

off-policy

蒙特卡洛异策略更新是由 Monte Carlo ES (Exploring Starts) 算法实现，这个算法也十分简单：

Algorithm 1 ES 算法

1. 初始化 state 全部采用随机数；
2. **repeat**
3. 获得本局的 reward：r;
4. 增量更新 $Q(s_{t+1}, a_{t+1}) = \frac{Q(s_t, a_t) \times count + r}{count + 1}$;
5. $count = count + 1$;
6. 对于每个状态 s，更新当前的策略:
$$\pi(s) = \arg\max_a Q(s, a)$$
7. **until** 随机玩 BlackJack500,000 局

通过上面的算法，我们可以得到书上的图 5.3:

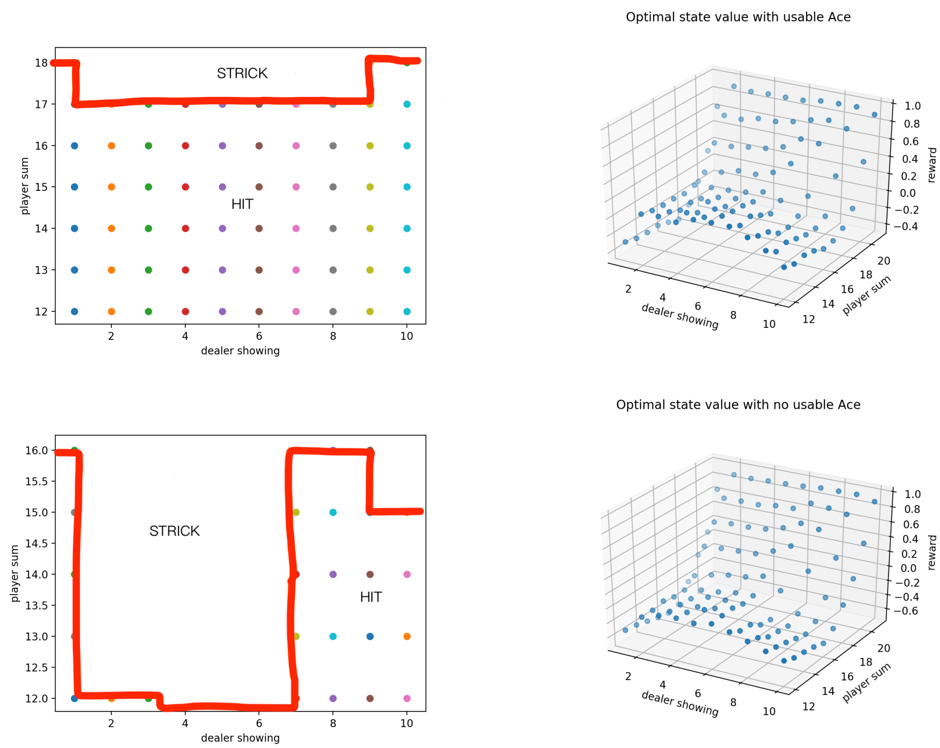


图 5.2: Monte Carlo ES (Exploring Starts)

上图的左边指的是返回当前对应的状态值最大时使用的策略 π:

- 当我们手中有 Ace 时，我们不用太在乎庄家亮的牌面；我们手中的牌面超过 17 的时候可以 strike。
- 当我们手中的没有 Ace 时，我们做出的策略有点匪夷所思，在庄家的牌面比较小 ($[2, 6]$) 并且玩家手中的牌面之和也较小 ($[11, 21]$) 就采取 Stick，这显然不是一个很好的策略。

上图的右边两个图则是最优的状态值的分布，从这两幅图我们可以看到:

- 两幅图的走势几乎相同，说明 Ace 对获胜的分布没有太大的影响。

- 有 Ace 在手上时的最优状态值略高，说明有 Ace 时我们获胜的期望略高。

- 对于庄家亮的名牌，呈现两头低中间高的趋势，说明庄家亮的牌越接近 6 我们获胜的期望略高。

- 对于两张图，我们手牌之和在 $[11, 17]$ 之间是，reward 变化的趋势不明显，还略有降低，估计是 bust 造成的。但是在区间 $[18, 21]$，reward 迅速上升。说明无论我们手中是否有 Ace，只要我们手牌之和超过 17 点，获胜的概率就会比较大。

off-policy Estimation of a state value with importance sampling

在蒙特卡洛采样中，我们根据相关轨迹在 target policy 和 behavior policy 中出现的概率来进行加权的重要性采样 (importance-sampling), 我们可以定义出这个权值：

$$\rho_{t:T-1} = \frac{\prod_{k=t}^{T-1} \pi(A_k|S_k)p(S_{k+1}|S_k, A_k)}{\prod_{k=t}^{T-1} b(A_k|S_K)p(S_{k+1}|S_k, A_k)} = \prod_{k=t}^{T-1} \frac{\pi(A_k|S_k)}{b(A_k|S_k)} \tag{5.3}$$

其中 $\pi(s, a)$ 表示 target policy，$b(s, a)$ 表示 behavior policy.

我们将初始状态设置为：

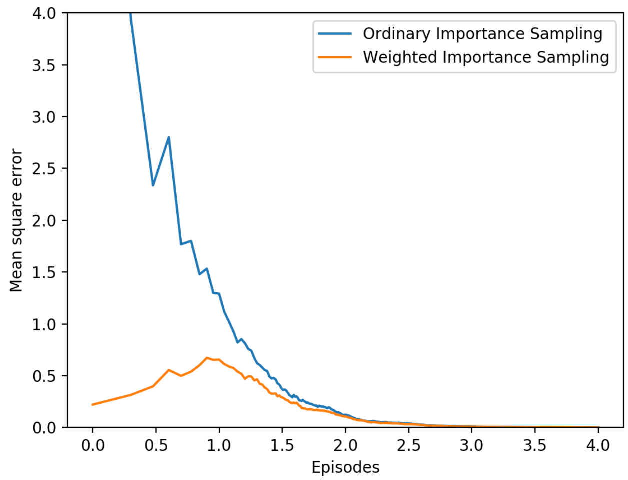
- state=[玩家使用 Ace 当作 11，玩家手中牌面之和为 13，庄家亮的牌面为 2]；

- 玩家随机选择动作 hit 和 action 进行游戏，获得相应的蒙特卡洛轨迹。

- 设置加权重要性采样拟合策略条件：玩家仅在手中牌面之和为 20 或者 21 时采用 strike，否则均采用 hit。

- 若蒙特卡洛轨迹和预设的玩法一致，那么根据 (5.3) 更新权值；否则该次轨迹的权值设为 0；

最后的权值函数与真实值-0.27726 求均方误差即可得到图 (5.4):



从上图我们可以看出：

- 当模拟估计次数少于 10^2 时，是否采用加权重要性采样对均方误差的影响很大，但是加权重要性采样 (橙线) 能够通过加权的方式较快地逼近需要的参考的策略真实值 (-0.27726)。
- 不采用权值的重要性采样只能够在不断迭代中根据 (5.3) 计算重要值，通过类似同策略的增量式来逼近 reward，所以收敛速度较慢。
- 但是我们可以看出是否采用权值仅仅能够加快 reward 的收敛速度，但是当采样次数足够多时，对收敛的结果没有太大影响。