实验三 - 马尔科夫决策

李一鸣

1160300625

2018 年 10 月 26 日

马尔科夫决策过程

马尔科夫随机过程（MDP, Markov Decision Process）是一种离散时间随机控制过程。

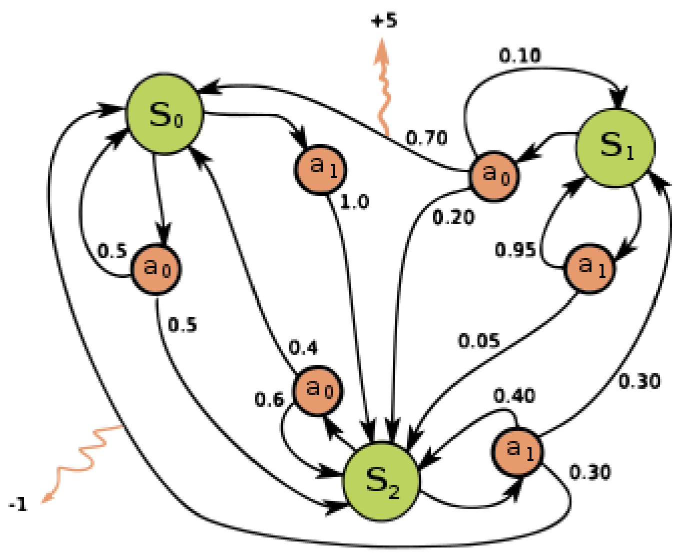
在每一步中，随机过程处于状态 s，决策者此时采取一个可行的决策 a。在下一步随机过程将转移到新的状态 s'，并给决策者相应的回报 $R_a(s,s')$。

随机过程进入新状态 s' 的概率取决于所采取的决策。具体来说，其概率大小有转移矩阵中的 $P_a(s,s')$ 定义。因此，下一个状态 s' 取决于当前状态 s 和决策者采取的决策 a。在已知 s 和 a 的情况之下，状态 s' 与所有之前的状态都是（条件）独立的。也就是说 MDP 具有马尔科夫性质，即随机过程后续的状态的条件概率分布只与当前状态有关，与之前的状态无关。

定义

马尔科夫过程是一个 5 元组 $(S, A, P_a, R_a; \gamma)$，其中：

- S 是有穷状态空间集合
- A 是有穷决策集合，A_s 表示状态 s 下可以采取的决策集合
- $P_a(s, s') = Pr(s_{t+1} = s' | s_t = s, a_t = a)$ 表示在 t 时刻处于状态 s 并采取决策 a 将在 $t+1$ 进入状态 s' 的概率，此概率与时间 t 无关
- $R_a(s, s')$ 是从状态 s 经过决策 a 到达状态 s' 时的期望瞬时收益
- $\gamma \in [0, 1)$ 是决策因子，表示将来状态与当前状态的重要度差别



三个状态、两个决策的 MDP

问题

假定决策的方案：一个函数 π，其中 $\pi(s)$ 表示在状态 s 下采取的决策。一旦有了决策方案 π，再联合在给定方案下到达下一状态的概率 $Pr(s_{t+1} = s' | s_t = s, a_t = a)$，我们可以得到到达下一状态的概率 $Pr(s_{t+1} = s' | s_t = s)$，这就是一个马尔科夫转移矩阵。

MDP 的核心问题在于找到最优的决策方案。我们的目标是选择最优的 π，使得期望收益最大化，通常我们会将将来的收益打一个折扣，因为无穷远的时间收益即使很大也没有意义：

$$W = \sum_{t=0}^{\infty} \gamma^t R_{a_t}(s_t, s_t + 1) \quad \text{这里 } a_t = \pi(s_t)\text{，也就是我们采取的决策方案} \tag{1}$$

其中 γ 用来表示折扣因子并且满足 $\gamma \in [0, 1)$。例如，当折扣率为 r 时，即后一时刻 $1 + r$ 块钱的收益只相当于当前时刻 1 块钱的收益，那么就有 $\gamma = \frac{1}{1+r}$。

由于 MDP 具有马尔科夫性质，W 的值与时间是无关的，因此其只是 s 的函数。

算法

已知：转移函数 P，收益函数 R

求解：最优决策方案以最大化期望收益

动态规划算法：

定义两个数组 V 和 π，其中 V 作为暂存数组，π 包含采取的决策。在算法结束时，π 将包含最优决策方案，$V(s)$ 将包含折后收益和。递归计算如下：

$$\pi(s) := \arg\max_a \left\{ \sum_{s'} P_a(s,s') \left(R_a(s,s') + \gamma V(s') \right) \right\} \tag{2}$$

这一步主要是寻求一个决策 a 使得收益最大化。

$$V(s) := \sum_{s'} P_{\pi(s)}(s,s') \left(R_{\pi(s)}(s,s') + \gamma V(s') \right) \tag{3}$$

这一步根据决策累积计算收益。

随机需求的单商品存贮决策

问题描述

每个月，仓库经理都会清点某种商品的当前库存量，从而决定是否要从供应商那里进货，进货的话要进多少。在此过程中，他需要权衡该商品库存带来的成本，和不能满足消费者对该商品的需求所带来的损失。他的目标就是最大化各月所得收益和期望值。我们设商品的需求量是一个已知概率分布的随机变量，且积压订单是不允许的，故库存量不会为负数。

- s_t 是第 t 个月的初库存量，它是状态变量
- a_t 是第 t 个月的订货量，它是决策变量
- D_t 是第 t 个月的随机需求量，假定该需求满足一个时间齐次的分布 $p_j = p(D_t = j), j = 0, 1, 2...$，也就是说需求量的分布与时间 t 无关

由于库存量非负，得到状态转移方程：

$$s_{t+1} = max\{s_t + a_t - D_t, 0\} \equiv [s_t + a_t - D_t]^+ \tag{4}$$

假设：

- 每个月月初做出是否订货和订货数量的决策，并假定订货可以及时送到
- 对商品的需求贯穿整个月，但是在该月的最后一天所有订单必须得到满足
- 如果顾客对某商品的需求超过该商品的库存量，即顾客的需求得不到满足，顾客可以到别处去购买他所需的商品。因此不会有因供货不足而造成订单积压的问题
- 收益、成本和需求分布不会按月改变
- 产品售出量都是整数
- 仓库容量为 M 个单位

建立模型

决策阶段：

$$t = 1, 2, ..., T \tag{5}$$

状态空间：

$$S = \{0, 1, 2, ..., s\} \tag{6}$$

决策集合：

$$A(i) = \{0, 1, 2, ..., s-i\}, i \in S \tag{7}$$

表示在状态 i 下可供选择的有限个决策的集合，由于最多只能存有 $s = M$ 件商品，在状态 i 时最多购进 $s-i$ 件商品。$A = \cup_{i \in S} A(i)$ 表示决策集合。

转移概率：

$$P_a(i,j) = \begin{cases} 0, & j \in (i+a, M], i+a < M \\ p_{i+a-j}, & j \in (0, i+a], i+a \le M \\ q_{i+a}, & j = 0, i+a \le M \end{cases} \tag{8}$$

解释如下：

1. 因为购进了 a 件商品，那么在下一个状态最多只能有 $i+a$ 件商品，这就是需求量 $d=0$ 的情况，发生的概率记为 p_{i+a}。
2. 如果在下一个状态只剩下了 $j \in (0, i+a]$ 件商品，说明卖出了 $d = i+a-j$ 件商品，此事件概率记为 p_{i+a-j}。
3. 如果下一个状态只剩下 0 件商品，这可能是刚好需求量是 $d = i+a$ 全部卖出了，也可能是需求量 $d > i+a$ 但是由于供不应求，顾客转到其他商家去购买了，此事件概率记为 $q_{i+a} = p_{i+a} + p_{i+a+1} + ... + p_\infty = \sum_{d=i+a}^{\infty} p_d$。

期望报酬：

$$\sum_j R_a(i,j) = \begin{cases} F(i+a) - O(a) - h(i+a), & t \in [1, T-1] \\ g(i), t = T \end{cases} \tag{9}$$

解释如下：

1. 从状态 i 选择策略 a 进入下一个状态的总收益等于总营业额 $F(i+a)$ 减去订购 a 件商品的总成本 $O(a)$，再减去 $h(i+a)$，对应于 $i+a$ 件商品的每个月的库存费用。

   其中：

   $$F(u) = \sum_{j=0}^{u-1} p_j f(j) + q_u f(u) \tag{10}$$

   其中又有 $f(u)$ 表示卖出 u 件商品时的收入。

   $O(a)$ 表示当前订购 a 件商品的成本。

   $h(i+a)$ 表示库存量为 $i+a$ 的库存费用。
2. 当处于最后一个时刻时，我们即使采取任何策略也得不到收益了，因此收益就等于 $g(i)$ 表示库存量为 i 时的剩余库存价值。

策略：

选取每个阶段决策的规则为一个策略。一个有限阶段的马尔科夫策略可以写成：

$$\pi = (d_1(i), d_2(i), ..., d_T(i)) \tag{11}$$

其中 $d_t(i)$ 是阶段 t 下状态为 i 时采用的决策。

动态规划递归方程：

- $u_t^*(i)$ 表示第 t 阶段状态是 i 时，采取最优策略，从第 t 阶段到第 T 阶段的最大总期望收益。

$$u_t^*(i) = \begin{cases} max_{a\in A_s}\{\sum_j R_a(i,j) + \sum_{j=0}^{s} P_a(i,j)u_{t+1}^*(j)\}, & t = T-1, T-2, ..., 1 \\ g(i), & t = T \end{cases} \tag{12}$$

  可以看出我们想要计算 $u_1^*(i)$，必须先计算 $u_2^*(i)$，如此递推到需要最先计算 $u_T^*(i) = g(i)$。

- $a_t^*(t)$ 表示使式 (12) 最大化的决策。
- $v^*(i)$ 表示当第 1 阶段状态为 i 时，采用最优策略获得的第 T 阶段最大总期望收益。

实例计算

对参数赋值，令

$$\begin{gathered} o(u) = 2u, \quad g(u) = 0, \quad h(u) = u, \quad s = 3, \quad T = 3, \quad f(u) = 8u \\ p_j = \begin{cases} \frac{1}{4}, & d = 0 \\ \frac{1}{2}, & d = 1 \\ \frac{1}{4}, & d = 2 \end{cases} \end{gathered} \tag{13}$$

用自然语言解释为：

库存量不能多于 3 件，所有成本和收益都是线性的，这意味着每订购一件商品花费为 2，每件商品每月的库存费用为 1，每单位商品售出的收益为 8。根据 (10) 式，可向顾客供应的商品数量为 u 时的期望收益 $F(u)$ 如下所示：

$$F(u) = \begin{cases} 0, & u = 0 \\ \frac{1}{4}\times 0 + (\frac{1}{2}+\frac{1}{4})\times 8 = 6, & u = 1 \\ \frac{1}{4}\times 0 + \frac{1}{2}\times 8 + \frac{1}{4}\times 16 = 8, & u = 2 \\ \frac{1}{4}\times 0 + \frac{1}{2}\times 8 + \frac{1}{4}\times 16 = 8, & u = 3 \end{cases} \tag{14}$$

如果在第 t 月初库存量为 s_t，购进 a 件新商品，结合订购商品的花费以及库存持有成本，我们可以得到期望收益。

先计算转移概率表：

$$P_a(s_t, j)$$

s_t+a\j	0	1	2	3
0	1	0	0	0
1	$\frac{3}{4}$	$\frac{1}{4}$	0	0
2	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{4}$	0

s +a\j	0	1	2	3
3	0	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{4}$

表 1

$\sum_j R_a(i,j)$

s_t\a	0	1	2	3
0	0-0-0=0	6-2-1=3	8-4-2=2	8-6-3=-1
1	6-0-1=5	8-2-2=4	8-4-3=1	8-6-4=-2 ✕
2	8-0-2=6	8-2-3=3	8-4-4=0 ✕	8-6-5=-3 ✕
3	8-0-3=5	8-2-4=2 ✕	8-4-5=-1 ✕	8-6-6=-4 ✕

表二

因为仓库容量为 3 所以打 ✕ 的决策是不可能出现的。

表中每一项都写成了营业额-订购成本-库存成本的形式。营业额通过查询式 (13) 得来，订购成本通过查询 $o(u)$ 得来，库存成本通过查询 $h(u)$ 得来。

Step 1

令 $t=4$

$$u_4^*(i)=g(i)=0, i\in[0,3] \tag{15}$$

Step 2

令 $t=3$

$$\begin{aligned} u_3^*(i) &= \max_{a\in A_s}\{\sum_j R_a(i,j)+\sum_{j=0}^{s}P_a(i,j)u_4^*(j)\} \\ u_3^*(i) &\xlongequal{u_4^*=0} \max_{a\in A_s}\{\sum_j R_a(i,j)\} \\ &i\in[0,3] \end{aligned} \tag{16}$$

根据期望收益表 $\sum_j R_a(i,j)$ 得到如下的决策表：

i	$a_3^*(t)$	$u_3^*(t)$
0	1	3
1	0	5
2	0	6
3	0	5

表三

Step 3

令 $t=2$

$$u_2^*(i)=\max_{a\in A_s}\{\sum_j R_a(i,j)+\sum_{j=0}^{s}P_a(i,j)u_3^*(j)\} \tag{17}$$

例如，对 $u_2(0)$ 在 $a=0$ 时的计算过程如下：

$$u_2(0)=0+1\times u_3^*(0)=3 \tag{18}$$

对 $u_2(1)$ 在 $a=1$ 时的计算过程如下：

$$\begin{aligned}u_2(1)&=3+\frac{1}{4}\times u_3^*(0)+\frac{1}{2}\times u_3^*(1)+\frac{1}{4}\times u_3^*(2)\\&=3+\frac{1}{4}\times 3+\frac{1}{2}\times 5+\frac{1}{4}\times 6\\&=7.75\end{aligned} \tag{19}$$

对每个 i 取不同的 a 计算得到最大的 $u_2^*(i)$，将此时的 a 记为 $a_2^*(i)$。

Step 4

令 $t=1$，重复上述过程，得到 $u_1^*(i)$ 和 $a_1^*(i)$。当初始状态为 i 时，最优策略 $\pi(i)=(a_1^*(i),a_2^*(i),a_3^*(i))$，从初始时刻到结束时刻的总报酬 $v^*(i)$。

实验结果

执行 `make` 得到如下的计算结果：

```
Transition probablity P[before_sold][after_sold] =
[[1.   0.   0.   0.  ]
 [0.75 0.25 0.   0.  ]
 [0.25 0.5  0.25 0.  ]
 [0.   0.25 0.5  0.25]]

Reward matrix R[current_stock][action] =
[[ 0.  3.  2. -1.]
 [ 5.  4.  1.  0.]
 [ 6.  3.  0.  0.]
 [ 5.  0.  0.  0.]]

Cumulative maximum reward u[time][current_stock] =
[[ 9.      11.625  14.4375 15.125 ]
 [ 6.       8.5    10.75   10.5   ]
 [ 3.       5.      6.      5.    ]
 [ 0.       0.      0.      0.    ]]

Best action a[time][current_stock] =
[[1. 0. 0. 0.]
 [1. 0. 0. 0.]
 [1. 0. 0. 0.]
 [0. 0. 0. 0.]]
```

其中 $u[0][0]$ 表示在第 1 天库存量为 0 时最大的收益，即前文所说的 $u_1^*(0)$，这时需要采取的决策是 $a[0][0]$，也就是购进 1 件商品。

从整体上看 $u[v][m]$ 表示第 $v+1$ 天库存量为 m 时采取最佳策略 $a[v][m]$ 得到的最大累计收益。

可以看到计算结果中 P、R 与手动计算的表一、二完全相符，同时 u 和 a 中第 3 行与表三完全相符。不难看出计算结果完全正确。

源代码

upupming/Lab3-markov-decision-process

参考文献

1. Markov decision process - Wikipedia