一个问题复习各类收敛，随机过程

Yutong Wang
London School of Economics

Dec 2024

Contents

1 原问题

原问题如下：初始资产10000，现在有一个游戏，扔公平的硬币，正面则资产增值80%，变为18000，反面，贬值50%，变为5000，问是否会一直玩下去。

题意理解：有两个晦涩的点，一是'一直玩下去'是指我们可以设置一个停止条件，玩到该条件达成为止，还是指没有任何停止条件地玩下去；二则是，每次我们是可以选择下注金额，还是必须使用全部资金来玩这个游戏。

虽然在现实中，对于两个问题而言我们都有选择（可以选择什么时候停止，也可以选择每次下注多少）。但为了探究清楚问题本质，我们从严到宽来看这个问题。我们先来看两者都没有选择权时的情况。

2 每次下注全部，无停止条件

2.1 期望收益

假设玩第i次后能将资产变为之前的X_i倍，那么X_i作为一个随机变量，它的分布为：$\mathrm{P}(X_i = 1.8) = 1/2$, $\mathrm{P}(X_i = 0.5) = 1/2$. 简单计算可知$\mathbb{E}X_i = \frac{1.8+0.5}{2} = 1.15$.

设玩n次后的总资产为V_n, 那么初始资产$V_0 = 10000$. (后面会看到，在这个情况下，初始资产无关紧要).

根据定义，我们有递推：$V_k = X_k \cdot V_{k-1}$.

那么，

$$\begin{aligned}\mathbb{E}[V_n] &= \mathbb{E}[X_n X_{n-1} \cdot \cdots \cdot X_1 V_0] \\ &= \mathbb{E}[X_n]\mathbb{E}[X_{n-1}] \cdot \cdots \cdot \mathbb{E}[X_1] V_0 \\ &= 1.15^n V_0.\end{aligned}$$

As a result, $\mathbb{E}[V_n] \to \infty$ as $n \to \infty$.

2.1.1 关于期望的一个误区

有人认为：玩$2n$次以后，平均来说，n次会赢，n次会输，那么那时候的资产$V_n = 1.8^n \cdot 0.5^n \cdot V_0 = 0.9^n \cdot V_0$，从而得到期望趋于0的结论。

这样算出来的确实也是一个有意义的量，但它不是期望，这是沿着概率最大的path得到的一个realisation。但它也不能叫大概率结果，因为这个情况发生的概率本身也是趋于0的：$\frac{\binom{2n}{n}}{2^n} \to 0$

根据最大概率path的输赢来做决策显然是有问题的，我们很容易构造出一个最大概率path输一点点，但其他情况都赢非常多，且最大概率path的概率收敛到0的游戏。例如，我们先构造一个序列Y_n, $\mathrm{P}(Y_n = 0.999) = (\frac{1}{2})^n$, $\mathrm{P}(Y_n = 100) = 1-(\frac{1}{2})^n$, 此时最大概率path是$Y_n$取100，这很容易改，只需把$Y_n$取100的概率平均分给$m$(可以针对每个$n$来取对应的$m$)个值: $100, 101, \ldots, 100+m-1$. 只要$m$够大，就使得$Y_n = 0.9$的概率最大for any n. 很容易看出，这个游戏以很大概率把资产翻100倍，极小概率变为0.999，不愿意玩的那只能说明风险厌恶太高，以至于抓不住任何机会.

2.2 赢的概率

我们现在关注的量是，玩n次以后，总资产比初始更多的概率，即$\mathrm{P}(V_n \geq V_0)$. 假设n足够大，我们用中心极限定理来估计即可.

$V_n \geq V_0$ 等价于$\ln V_n \geq \ln V_0$ 等价于$\sum_{i=1}^n \ln X_i \geq 0$.

记$Y_i = \ln X_i$, 由于$\ln(1/2) \approx -0.693$, $\ln(1.8) \approx 0.588$, 所以$\mathbb{E}[Y_i] < 0$.

由中心极限定理，$\frac{\sum_{i=1}^n (Y_i - \mathbb{E}[Y_i])}{\sqrt{n\mathrm{Var}(Y_i)}} \to_d N(0,1)$ in distribution. 近似而言，

$$
\begin{aligned}
\mathrm{P}(V_n \geq V_0) &= \mathrm{P}\Big(\sum_{i=1}^n Y_i \geq 0\Big) \\
&= \mathrm{P}\Big(\sum_{i=1}^n (Y_i - \mathbb{E}[Y]) \geq -n\mathbb{E}[Y]\Big) \\
&= \mathrm{P}\Big(\frac{\sum_{i=1}^n (Y_i - \mathbb{E}[Y])}{\sqrt{n\mathrm{Var}(Y)}} \geq \frac{-n\mathbb{E}[Y]}{\sqrt{n\mathrm{Var}(Y)}}\Big) \\
&\approx \mathrm{P}_{Z\sim N(0,1)}\Big(Z \geq \sqrt{n}\frac{\mathbb{E}[Y]}{std(Y)}\Big) \\
&\to 0
\end{aligned}
$$

实际含义就是，玩很多轮之后，以很大概率资产都会比初始时少。如何理解这个'很大概率'? 假设有100个人去玩这个游戏，都玩50轮，那么很可能其中1个人的资产数大于10000，其他都小于10000.

2.3 几乎处处收敛(almost surely convergence) / 依概率收敛(convergence in probability)

事实上，利用大数定律，能得到比以上更强的结论。在上面，我们已经得到：$\ln V_n = \ln V_0 + \sum_{i=1}^n Y_i$. 那么$\frac{\ln V_n}{n} = \frac{\ln V_0}{n} + \frac{\sum_{i=1}^n Y_i}{n}$, 根据强大数定律，$\frac{\sum_{i=1}^n Y_i}{n} \to_{a.s.} \mathbb{E}[Y]$ 等于一个-0.1左右的负数。（用弱大数定律能得到依概率收敛的结果，比几乎处处收敛要弱）

使用连续映射定理（continuous mapping theorem），

$$
\begin{aligned}
V_n &= e^{\ln V_n} \\
&= V_0 \cdot e^{\sum_{i=1}^n Y_i} \\
&= V_0 \cdot e^{n \cdot \frac{\sum_{i=1}^n Y_i}{n}} \\
&\to_{a.s.} V_0 \cdot e^{n \cdot \mathbb{E}[Y]} \\
&\to 0
\end{aligned}
$$

中间有点不严谨，几乎处处收敛的那一步，右边不应依赖于n，改变一下写法能够解决但这只是增加没有必要的复杂。

严格表达：

$$\mathrm{P}\{\omega : \limsup V_n(\omega) = 0\} = 1$$

意思为：任选一条path，这条path最终使人资产归零的概率是1.

3 每次下注全部，可选停止条件

随机过程这门课最早就是为了研究如何科学地赌博。从上面我们已经可以看到，如果一直玩下去，基本都会血本无归。但如果我们有策略地玩呢?

随机过程里有一个基本的定理，是说，对于一个鞅过程，是没有策略能使期望收益大于初始资金的。虽然期望收益不能代表全部，但一个风险中性的人是不会玩一个期望收益为负的游戏的，而数据表明，现实中大部分人都是风险厌恶者，也就是期望收益哪怕是正的，不够高也不玩。所以有些推论可据此做出。

好在这个游戏的期望是正的，且增长很快。我们唯一能操作的就是决定在什么时候停止继续玩，那有没有办法能从中获利呢?

首先，决定什么时候停止必然依赖在那个时刻可以获得的信息，而这个游戏的所有信息就是当前手里有多少钱（由于题里没说，故不考虑对手破产的情况）。换言之，一个纯停止策略必然有如下的形式：'一旦钱数达到18000，就收手不玩'.

当然，复合停止策略也可以包括这种：'如果钱数已经低于5000，那么三次内如果没有回到10000，就停止游戏'。

我们在此考虑最简单的情况，假设策略是纯停止策略。停止时刻是一个随机变量，$T = \inf\{t : V_t \geq S\}$。

3.1 复习随机游走与停时

题中出现的是带漂移（drift）的随机游走，我们将使用一些技巧来处理它，先用均值为0的随机游走来熟悉一下。

标准的随机游走是指：

$$X_t = X_{t-1} + \varepsilon_t$$

其中$X_0 = 0$, ε_t 在时刻t才realise, 以1/2的概率取1或者-1. 可以看到随机过程$\{X_t\}$有可能取任何整数，且$\mathbb{E}_{t-1}[X_t] = X_{t-1} + \mathbb{E}_{t-1}[\varepsilon_t] = X_{t-1}$是鞅过程（martingale）

Proposition 1.
设定一个边界a和-b，其中a，b都是正整数，记$T = \inf\{t : X_t = a \ or\ X_t = -b\}$为$\{X_t\}$第一次到达$a$,$b$其中之一的时刻, 那么有以下结论：

1. $\mathrm{P}(T < \infty) = 1$;
2. $\mathrm{P}(X_T = a) = \frac{b}{a+b}$, $\mathrm{P}(X_T = b) = \frac{a}{a+b}$;
3. $\mathbb{E}[T] = ab$.

Proof. 1. $\{T > n\} \subset \{\max\limits_{1 \leq t \leq n} |X_t| \leq \max(a, b)\} \subset \{\sum\limits_{i=1}^{n} |\varepsilon_i| \leq \max(a, b)\}$ 最右边的集合的概率对于足够大的n肯定为0;

2. 使用optional stopping time theorem[1]，条件(c)满足，由于X_t是鞅，所以$X_{t\wedge T}$ 也是鞅。故$\mathbb{E}[X_{t\wedge T}]=X_0=0$. 设$p$为先取到$a$的概率，则先取到$-b$的概率为$(1-p)$,那么$\mathbb{E}[X_{t\wedge T}]=pa+(1-p)b$. 解得$p=\frac{b}{a+b}$.

3. 易验证X_t^2-t也是鞅。再次使用optional stopping time theorem，条件(c)满足。$\mathbb{E}[X_T^2-T]=\mathbb{E}[X_0^2-0]=0$. 所以：

$$\begin{aligned}\mathbb{E}[T] &= \mathbb{E}[X_T^2] \\ &= a^2\cdot\frac{b}{a+b}+b^2\cdot\frac{a}{a+b} \\ &= ab\end{aligned}$$

□

3.2 带偏移随机游走

由于原题的数字取对数后是非整数，会增加一些细节的不必要的困难，我们改为用整数。假设有这样一个随机游走：

$$X_t=X_{t-1}-0.5+\varepsilon_t$$

其中ε_t以1/2概率分别取1.5和-1.5，这样表达是为了保持$\mathbb{E}[\varepsilon]=0$的惯例。实际上，每次$X_t$比前一期的$X_{t-1}$增加1或者-2，各以1/2概率。

记$f(x)$为当前在x时，先到达a的概率。虽然X_t 不再是鞅了，但它还是Markov process，那么：

$$\begin{aligned}f(x) &= \mathrm{P}(\text{hits a first} \mid \text{starts at x}) \\ &= \mathrm{P}(\text{move to } x+1)\mathrm{P}(\text{hits } a \text{ first} \mid \text{starts at } x\text{, next move to } x+1) \\ &\quad +\mathrm{P}(\text{move to } x-2)\mathrm{P}(\text{hits } a \text{ first} \mid \text{starts at } x\text{, next move to } x-2) \\ &= \mathrm{P}(\text{move to } x+1)\mathrm{P}(\text{hits } a \text{ first} \mid \text{starts at } x+1) \\ &\quad +\mathrm{P}(\text{move to } x-2)\mathrm{P}(\text{hits } a \text{ first} \mid \text{starts at } x-2) \\ &= \frac{1}{2}f(x+1)+\frac{1}{2}f(x-2).\end{aligned}$$

也就是说，这些概率是一个有边界条件的差分方程组的解。其中边界条件是：$f(a)=1$, $f(-b)=0$. 该差分方程的通解是$f(x)=Ae^{kx}$, 为了确定k, 我们把通解代入：

$$\begin{aligned}\frac{1}{2}(f(x-2)+f(x+1)) &= \frac{1}{2}Ae^{k(x+1)}+\frac{1}{2}Ae^{k(x-2)} \\ &= A(\frac{e^k+e^{-2k}}{2})e^{kx} \\ &= Ae^{kx}=f(x),\end{aligned}$$

故而k需要满足$e^k+e^{-2k}=2$. 代换$y=e^k$, 原式成为$y+y^{-2}=2$, 等价于$\frac{(y-1)(y^2-y-1)}{y^2}=0$, $y_{1,2,3}=1,\frac{1+\sqrt{5}}{2},\frac{1-\sqrt{5}}{2}$. 由于边界条件只有两个，选取其中两根即可。（解的唯一性可以

[1]wikipedia上没有此定理的中文名，当年学的时候也用的英文教材，故使用原名

用线性方程组的系数矩阵满秩来证明，用以说明选哪两个根并不影响最后的结果）选取$y_1 = \frac{1+\sqrt{5}}{2}$, $y_2 = 1$, 对应的k为：$k_1 = \ln\frac{1+\sqrt{5}}{2}$, $k_2 = 0$.

现在待定$f(x) = Ae^{k_1 x} + Be^{k_2 x} = Ae^{k_1 x} + B$, 代入边界条件可解出$A$和$B$. 最终：

$$f(x) = \frac{e^{k_1 x} - e^{-k_1 b}}{e^{k_1 a} - e^{-k_1 b}}, \quad x = -b, -b+1, \ldots, a-1, a.$$

由于原问题中，资产可以缩水至无限小（假设无限可分），但任何有限期内都不会归0，故对应的$-b$是$-\infty$. 我们取极限观察一下：

$$\lim_{b\to\infty} f(x) = \lim_{b\to\infty} \frac{e^{k_1 x} - e^{-k_1 b}}{e^{k_1 a} - e^{-k_1 b}} = \frac{\lim\limits_{b\to\infty}(e^{k_1 x} - e^{-k_1 b})}{\lim\limits_{b\to\infty}(e^{k_1 a} - e^{-k_1 b})} = \frac{e^{k_1 x} - 0}{e^{k_1 a} - 0} = e^{k_1(x-a)},$$

由于我们感兴趣从0开始，故再evaluate at 0，得到：$\mathrm{P}(\{\omega : \text{无法达到} a\}) = e^{-ak_1}$.

对于无偏移的随机游走，这个概率是1，即那些无法到达a的path构成零测集（例如一条在每一期ε_t都取-1的path）。而现在由于负偏移的存在，无法到达a的path的集合的概率（测度）大于0了。

简单的增减性分析：a越大，能达到a的概率越小。符合直观。

此外，最终的表达式里，除了a是要达到的点以外，k_1是由题里的数字确定的方程的特征根，也影响着概率衰减的程度。实际上，如果允许X_t取值不仅在整数上，结论几乎是类似的，把X_t换成带漂移的布朗运动(Brownian motion)，例如$dX(t) = mdt + dW(t)$，结论也是几乎一样的。

再说回原题意，用一些case来验证我们的计算是对的/增加直观。a代表资产增加至的比例的对数（$a = \ln\frac{V_n}{V_0}$）。例如，如果要达到1倍就收手，这实际上代表着完全不玩这个游戏，此时$a = \ln\frac{1}{1} = 0$，代入$a = 0$得到概率为1，符合意义。

按照我们设置的简单停止法，对于任何$x \geq 1$，这个游戏能以$x^(-k)$的概率让资产翻x倍. k是方程$e^{k\ln(1.8)} + e^{k\ln(0.5)} = 2$的除了1以外的根。

实际上还可以有更复杂的停止法，给定一个优化目标（例如期望收益与方差的加权和，权重自取），也可以找到对应的最优停止策略。但感觉很难求解析解，在此仅抛砖引玉。

```python
[2]: import numpy as np
     from scipy.optimize import fsolve

     def equation(x):
         return np.exp(x * np.log(1.8)) + np.exp(x * np.log(0.5)) - 2

     initial_guess = 0.5

     solution = fsolve(equation, initial_guess)

     solution

[2]: array([0.25801877])
```

Figure 1: 特征方程数值解

从数值结果可以看到，$k \approx 0.26$, 那么，假如设定玩到2倍就收手，

$$\begin{aligned}
& \mathbb{E}[V_n|\text{stop when exceed } 2V_0] \\
= \; & \mathrm{P}(\text{can reach } 2V_0)\mathbb{E}[V_n|\text{stop when exceed } 2V_0, \text{ can reach } 2V_0] \\
& + \mathrm{P}(\text{cannot reach } 2V_0)\mathbb{E}[V_n|\text{stop when exceed } 2V_0, \text{ cannot reach } 2V_0] \\
\geq \; & (2^{-k}) \cdot (2V_0) + 0 \\
\approx \; & 0.836 * 2V_0.
\end{aligned}$$

能有0.836的概率把资产翻倍，另一边的条件期望我直接当作0了，但显然还可以有更精细的策略来止损。

3.3 数值模拟

30个人去玩，我循环10000次，记录并打印所有达到过10倍/2倍的人的编号。有的人达到之后继续玩财产就下去了，这也计入在内，因为我们的假设他使用的策略是，到达某个倍数就停止不玩了。

```
[82]: def simulate(num_of_people=30, max_ite=10000, multiples=2):
          X = torch.ones(num_of_people)
          ite = 0
          has_reach = set()

          while ite <= max_ite:
              ite += 1
              r = (torch.rand(num_of_people) >= 0.5).double()
              factor = 1.8*r + 0.5*(1-r)
              X *= factor
              has_reach.update((X >= multiples).nonzero().flatten().tolist())


              # if len(has_reach_10) >= 25:
              #     break

          # print(has_reach)
          return len(has_reach)
      # print(ite)
```

Figure 2: num_of_people个人，分别模拟10000次，达到原资产2倍就停，return其中达到2倍的人数

```
[89]: nums_reach = []
      for _ in range(50):
          nums_reach.append(simulate(30, 10000, 2))

      print(nums_reach)

      [21, 22, 23, 26, 25, 24, 22, 26, 23, 22, 25, 21, 24, 28, 20, 25, 25, 24, 25, 23, 23, 22, 23, 20, 24, 26, 2
      3, 25, 24, 24, 23, 24, 25, 20, 22, 20, 26, 19, 24, 23, 26, 25, 28, 20, 20, 24, 19, 22, 20, 23]

[90]: np.mean(nums_reach)/30

[90]: 0.7739999999999999
```

Figure 3: 重复50次，看多少人达到过2倍

```
[91]: nums_reach = []
      for _ in range(50):
          nums_reach.append(simulate(30, 10000, 10))

      print(nums_reach)

      [13, 11, 19, 15, 10, 13, 16, 15, 19, 15, 19, 12, 18, 12, 16, 12, 14, 16, 11, 13, 17, 18, 16, 13, 15, 18, 1
      8, 12, 13, 12, 12, 15, 15, 19, 13, 16, 14, 20, 9, 18, 19, 14, 13, 21, 19, 19, 19, 15, 11, 15]

[92]: np.mean(nums_reach)/30

[92]: 0.5046666666666667
```

Figure 4: 重复50次，看多少人达到过10倍

可以看到，玩10000次，有超过0.77的人曾到达过2倍初始资产，有超过0.5的人到达过10倍初始资产。为什么说‘超过’呢，因为10000次之后还有人可能达到这个倍数，而已经达到的人就停止游戏了，所以不会减少。

4 可以下注，可以停止

如果可以选择下注，是否能够停止就不重要了。

凯利公式是说，初始资金设为1，每次下注资产比例f^*, 如果p概率以赔率b赢钱（变为$1+f^*b$），以$(1-p)$概率全亏(变为$1-f^*$)，那么可以用最大化对数收益算出最优的f^*。

先把原题等效为每次下注5000，有0.5概率全亏，有0.5概率赢到18000（赔率3.6）。

带进去算，求导，得最优$f^*=2/9$，这个是等效下注比例，实操时需要再还原为两倍，也就是$4/9$。

4.1 数值模拟

```
[115…  def simulate_Kelly(num_of_people=30, max_ite=500):
           X = torch.ones(num_of_people)
           ite = 0
           while ite <= max_ite:
               ite += 1
               rand = (torch.rand(num_of_people) >= 0.5).double()
               factor = 1.8*rand + 0.5*(1-rand)
               X += 4/9 * X * (factor-1)

           return X#.tolist()

[129…  nums_people = 100000
       (simulate_Kelly(nums_people, 3000) < 1).sum()/nums_people

[129…  tensor(0.)
```

Figure 5: Kelly策略结果

100000个人参与实验，3000次后没有一个人是亏的。