赌博游戏与数学期望

北京理工大学
梁耀波

September 7, 2013

Abstract

基本问题：平均需要抛多少次硬币，才会首次出现连续的 n 个正面？这个问题可以使用赌博的方法来解决，方案来自 matrix67。对于这种方法，我们可以加以推广，用以解决一类的相似问题。

1 连续 n 次相同的期望

1.1 抛硬币出现连续正面

抛硬币，当出现连续的 n 个正面的时候停止。问出现这种情况的期望步数是多少。

算法分析：

抛硬币共有 2 种结果，正面和反面，出现的概率是均等的。

假设有这样子一个赌场，赌博的方式是抛硬币，猜正反面。如果赌徒压 1 块钱，猜中的话赌徒赚 1 块钱（加上返还的本金，现在共有 2 块钱）；如果猜不中的话赌徒就会输掉 1 块钱。

赌场每天都会来一个人，身上带着一块钱的基本赌资。每次赌博赌徒总是将身上钱全部压正面。当输光之后，赌徒就会离开赌场。赌场有个规定，如果存在一个赌徒连续 n 次都赢钱的话，赌场就会关门。

首先我们可以发现，这个赌博是公平的，赌徒有 $\frac{1}{2}$ 的概率赢 1 块钱，$\frac{1}{2}$ 的概率输 1 块钱。那么对于赌场来说，在关门的时候，从期望上来讲是收支平衡的。

关门的时候赌场收入的钱来自于每天来的一个赌徒，因为一个赌徒只有一块钱，所以赌场收入的钱就等于赌场关门时的天数。

关门的时候赌场支出的钱都在最后带着钱离开的赌徒手里。因为赌徒每次总是将所有的钱都压上，所以在结束的时候赌徒要么连赢，要么输光。从后向前数，第 n+1 个人及之前的人一定都输光了，否则他们连赢的次数就超过了 n。而第 1 个到第 n 个人手里分别有 $2^1, 2^2, \ldots 2^n$ 块钱，和为 $2^{n+1}-2$

这个方法来自matrix67，我们注意到将这个数学问题转化为赌博问题的一个关键是赌场的收支平衡，也就是赌博的公平性。只要我们能保证公平，就可以将很多问题进行类似的转化。

转化有几个关键点：

1. 赌徒压注的策略，和对应的赔率。必须保证整个赌博是公平的。设获胜概率为 p，因为赌徒会输光，所以获胜的收益应为 $\frac{1-p}{p}$ 倍的本金，加上返还的本金，获胜后的钱是原来的 $\frac{1}{p}$ 倍。赌博一次之后，赌徒手里的钱的期望为 $E = \frac{1}{p} * p + 0 * (1-p) = 1$，也就是不变。

2. 赌场关门的条件

当关门的时候，所有赌徒手里目前的钱就是赌场关门时天数的期望。

1.2 掷骰子出现连续 1

有一个有 p 个面的骰子，上面分别写着数字 1,2,...,p。问出现连续 n 个 1 期望需要掷多少次?

和硬币类似，赌徒每次都压 1，这时候赌徒有 $\frac{1}{p}$ 的概率赢钱，$\frac{p-1}{p}$ 的概率输钱。所以如果压一元，赢则赢 p-1 元 (钱变为 p 元)，输则输 1 元。

当某个赌徒连赢 n 次，使得赌场关门的时候，从后向前数，第 1 个到第 n 个人手里分别有 $p^1, p^2, \ldots p^n$ 块钱，和为 $\frac{p^{n+1}-p}{p-1}$

这个问题是硬币问题的推广版本。

1.3 掷骰子出现连续任意相同数字

有一个有 p 个面的骰子，上面分别写着数字 1,2,...,p。问出现连续 n 个相同数字期望需要掷多少次?

赌徒赌博策略：一个人刚来的时候，首先观望一轮，看掷出了什么，然后从第二天开始压在第一天看到的数字，赔率的规则和前一个问题相同。

当某个赌徒连赢 n-1 次，使得赌场关门的时候，从后向前数，第 1 个到第 n 个人手里分别有 $p^0, p^1, \dots p^{n-1}$ 块钱，和为 $\frac{p^n-1}{p-1}$

这个问题可以转化成上一个问题。假设赌场第一天不赌，从第二天开始赌。赌徒压的是骰子和前一天一样。和前一天相同的概率为 $\frac{1}{6}$，掷出 1 的概率也是 $\frac{1}{6}$，所以连续 n-1 次相邻两天点数相同的期望，和连续 n-1 次 1 的期望相同。最后再加上第一天的 1 即可。

$$\frac{p^n - p}{p-1} + 1 = \frac{p^n - 1}{p-1}$$

2 出现某种模式的期望

设字符集的大小为 k，分别对应键盘上的一颗键。猴子以均匀的概率随机敲击键盘。输入一个字符串，问期望需要敲击多少次可以出现这个字符串。URAL 1667 CTSC2006

算法分析：

设字符串串长为 l，赌场每天都会随机生成 1 到 k 之间的某个字符。每个赌徒进来的第一天，压字符串的第一个字符，如果获胜，第二天压第二个字符，以此类推。直到某个赌徒获胜 l 次，此时就相当于出现了要求的字符串，赌场关门。因为每天赌徒总是压一个字符，所以，获胜的概率是 $\frac{1}{k}$，所以获得的回报就是 k-1。

现在考虑赌徒。一个赌徒根据自己历次压注的字母可以形成一个字符串，如果连续 i 次都压中了，那么这个字符串必然是目标串一个长度为 i 的前缀。从后向前数，l 之前的赌徒必然输光了。l 这个赌徒连赢了 l 次，使得赌场关门。从他这里，我们可以知道赌场最后 l 次产生的字符都是什么。从后向前 1 到 l 的赌徒，第 i 个赌徒的字符串只有是目标串的长度为 i 的后缀的时候，最后才能有钱，而同时又必须是目标串长度为 i 的前缀，这样他后手里会有 k^i 的钱。换言之，如果目标串长度为 i 的后缀等于长度为 i 的前缀，那么第 i 个人就会有 k^i 的钱，否则为 0。最后的公式为：

$$\sum_{i=1}^{l} \begin{cases} k^i, & suffix(i) = prefix(i) \\ 0, & suffix(i) \neq prefix(i) \end{cases}$$

出现 n 个相同 1 的问题可以认为是这个问题的一个特殊版本。即任意长度的前缀都等于后缀。

在判定哪些 i 是合法的时候，不需要暴力的判定。可以发现 KMP 算法的 next 数组含义和此相同。$next[l]$ 就是这些 i 中，除 l 之外最大的 i。

程序如下：

```
for (i=l; i>0; i=next[i])
    ans += pow(k, i)
```

3 出现连续不相同的数字

设字符集为 k，出现 n$(1 \leq n \leq k)$ 个连续的不相同的字符的期望是多少？

赌场规则为：每次从 k 个字符中随机产生一个，关门条件是某个赌徒压中 n 次。

每个赌徒都有一个字符的集合，初始为空。每次都会压新出现的字符和集合中的字符不一样。如果获胜，新的字符就加入到集合中。当集合大小变成 n 的时候，就对应了赌场的关门。假设集合中现在有 i 个字符，那么赌徒获胜的概率就是 $\frac{k-i}{k}$，为了公平，对应的回报就是 $\frac{i}{k-i}$，也就是获胜后的钱变为原来的 $\frac{k}{k-i}$ 倍。所以，最后赌徒手里的钱共有：

$$\sum_{i=1}^{n} \prod_{j=0}^{i-1} \frac{k}{k-j}$$