The Gambler's Ruin Problem and Block Chain

Ke(KEN)WANG

1. Problem Describe

Prove that the formula (4) is correct by using (1), (2) and (3) in [1].

2. The Gambler's Ruin Problem

In order to prove the formula (4) in [1], firstly we have to take a look at *the Gambler's Ruin Problem*. This Gambler problem is motivated by trying to determine the success or failure of a gambler who goes to a casino with some amount of money (e.g., \$100) initially and wants to leave with some larger amount of money (e.g., \$200) at the end of the evening [2].

Suppose the gambler start with a dollars and end up with c dollars and $0 \leq a \leq c$, then the probability of success for the gambler is

$$s_c(a) = ps_c(a+1) + qs_c(a-1) \tag{1}$$

where p is the probability of winning an individual play of the game, and $q = 1 - p$ is the probability of losing an individual play of the game. Obviously the equation (1) is a second order linear ordinary differential equations. Therefore, we first assume a solution form $s_c(a) = z^a$ for some unknown base value z, then substitute the form into (1), then it gives

$$z^a = pz^{a+1} + qz^{a-1} \tag{2}$$

It is noteworthy that we don't want to $z = 0$, so the equation (2) can factor out a common z^{a-1}. Then we have

$$pz^2 - z + q = 0 \tag{3}$$

So $z = 1$ and $z = \frac{1}{p} - 1 = \frac{q}{p}$, finally the solution of (1) is

$$s_c(a) = C_1(1)^a + C_2(\frac{q}{p})^a \tag{4}$$

The implementation of Block Chain is available at https://github.com/ken0225/Block-Chain-MATLAB

3. Prove the formula (4) in [1]

Now we focus on the formula (4) in [1]. Start when the attack transaction is included in the blockchain, the honest chain extend $z \in \mathbb{N}$ blocks and the attacker chain extend $k \in \mathbb{N}$ blocks. It is worthy noting that k can be $0, 1, \ldots, +\infty$.

If $k > z$, the attack is successful, otherwise $k \leq z$. But even $k \leq z$, the attacker still has chance to catch up from z blocks behind. Calculating the probability that the attacker will ever catch up from z blocks behind is similar to the Gambler's Ruin Problem. So we can obtain the equation (2) in [1]

$$q_z = \begin{cases} 1 & \text{if } p \leq q \\ (\frac{q}{p})^z & \text{if } p > q \end{cases}$$

where p denotes the probability an honest node finds the next block, q refers to the probability the attacker finds the next block and q_z is the probability the attacker will ever catch up from z blocks behind. Since k can be $0, 1, \ldots, +\infty$, there are **infinite** situations that the attack is successful. We therefore calculate $(1 - P_{\text{attack failure}})$ instead.

Poisson distribution is a discrete probability distribution that expresses the probability of a given number of events occurring in a fixed interval of time or space if these events occur with a known constant mean rate and independently of the time since the last event[3].

The time period that honest chain extend z blocks is a *fixed interval of time* and the extending one block by attacker chain is an *event*. Therefore, the probability that every different k appears is

$$P_{\text{every different k appears}} = \frac{\lambda^k e^{-\lambda}}{k!} \tag{5}$$

where λ is the expected value. Assume the honest chain wants to extend z blocks with the probability p, then the total times that the honest chain cost is z/p. During the same time period z/p, the attacker chain can create(i.e., the expected value) $\lambda = z/p \cdot q = \frac{zq}{p}$ blocks.

When $k > z$, the blocks that the attacker chain extends is more than the honest chain extends. So the attack is successful and $P_{\text{attack failure}} = 0$. When $k \leq z$, according the solution of the Gambler's Ruin Problem, the probability that the attacker chain still can catch up from $z - k$ blocks behind is $(\frac{q}{p})^{(z-k)}$, and the probability that the attacker chain can't catch up is

$$P_{\text{can't catch up}} = 1 - (\frac{q}{p})^{(z-k)} \tag{6}$$

Based on (5) and (6), we have

$$P_{\text{every different k attack failure}} = P_{\text{every different k appears}} \cdot P_{\text{can't catch up}}$$

$$= \frac{\lambda^k e^{-\lambda}}{k!} \cdot (1 - (\frac{q}{p})^{(z-k)}) \tag{7}$$

then

$$P_{\text{attack failure}} = \sum_{k=0}^{z} P_{\text{every different k attack failure}} = \sum_{k=0}^{z} \frac{\lambda^k e^{-\lambda}}{k!} \cdot (1 - (\frac{q}{p})^{(z-k)}) \tag{8}$$

finally we have the formula (4) in [1]

$$P_{\text{attack successful}} = 1 - P_{\text{attack failure}} = 1 - \sum_{k=0}^{z} \frac{\lambda^k e^{-\lambda}}{k!} \cdot (1 - (\frac{q}{p})^{(z-k)}) \tag{9}$$

Reference

[1]. Nakamoto S. Bitcoin: a peer-to-peer electronic cash system. 2008. www.bitcoin.org

[2]. Joe Koebbe. A Really Brief Review of the Solution of Linear Second Order Constant Coefficient Ordinary Differential Equations. http://www.math.usu.edu/koebbe

[3]. https://en.wikipedia.org

[4]. https://github.com/ken0225/Block-Chain-MATLAB