MTL106 Assignment 1

Puran Mayur
2023MT10366

Due Date: 4 November 2024

1 Question 1

1.1 Part A - Classic Gambler's Ruin

Let P_i represent the probability that the gambler wins, starting with an initial wealth of i. To determine P_i, we can develop a recurrence relation based on her wealth at state i during the gamble.

If she advances to the $(i+1)$-th state of the Markov chain (with probability p), then her probability of winning becomes P_{i+1}. Similarly, if her wealth decreases by 1 (moving to the $(i-1)$-th state) with probability $q = 1 - p$, then her probability of winning becomes P_{i-1}.

This gives us the following recurrence relation:

$$P_i = p \cdot P_{i+1} + q \cdot P_{i-1}$$

Now we just need to solve this recurrence, using the initial conditions that $P_0 = 0$ (since 0 is an absorbing state, and hence no way to win when the gambler reaches 0).

We will use the method of characteristic equations to solve this recursion. The characteristic equation is:

$$pr^2 - r + q = 0$$

Case 1: $p \neq q$
For $p \neq q$, the characteristic equation has roots:

$$r = 1 \text{ and } r = \frac{q}{p}$$

Therefore, the general solution is:

$$P_k = A + B(\frac{q}{p})^k$$

Using boundary conditions:

$$P_0 = A + B = 0$$
$$P_N = A + B(\frac{q}{p})^N = 1$$

Solving these equations:

$$A + B = 0$$
$$A + B(\frac{q}{p})^N = 1$$

This gives:

$$A = \frac{1}{1 - (\frac{q}{p})^N}$$
$$B = -\frac{1}{1 - (\frac{q}{p})^N}$$

Therefore:

$$\boxed{P_k = \frac{1 - (\frac{q}{p})^k}{1 - (\frac{q}{p})^N}}$$

Case 2: $p = q = \frac{1}{2}$

When $p = q$, the characteristic equation becomes:

$$\frac{1}{2}r^2 - r + \frac{1}{2} = 0$$

This has a repeated root $r = 1$. The general solution is:

$$P_k = A + Bk$$

Using boundary conditions:

$$P_0 = A = 0$$
$$P_N = A + BN = 1$$

This gives $B = \frac{1}{N}$, therefore:

$$\boxed{P_k = \frac{k}{N}}$$

1.2 Part B - Infinite Wealth Probability

We can take the limit for both the cases in part A as N tends to ∞.

If $p > 0.5$, then $\frac{q}{p} < 1$,

$$P_i = \frac{1 - (\frac{q}{p})^i}{1 - (\frac{q}{p})^N}$$

and thus,

$$\lim_{N \to \infty} P_i = \boxed{1 - \left(\frac{q}{p}\right)^i} > 0, \quad p > 0.5.$$

If $p \leq 0.5$, then $\frac{q}{p} \geq 1$,

$$\boxed{P_k = \frac{k}{N}}$$

and thus,

$$\lim_{N \to \infty} P_i = 0, \quad p \leq 0.5.$$

1.3 Part C - Expected Duration

Let T_k be the expected number of rounds starting from k dollars. Using first step analysis:

$$\begin{aligned}
T_k &= 1 + pT_{k+1} + qT_{k-1}, \quad 1 \leq k \leq N-1 \\
T_0 &= 0 \quad \text{(boundary condition)} \\
T_N &= 0 \quad \text{(boundary condition)}
\end{aligned}$$

Rearranging:

$$pT_{k+1} - T_k + qT_{k-1} = -1$$

This is a non-homogeneous difference equation. The characteristic equation of the homogeneous part is:

$$pr^2 - r + q = 0$$

Case 1: $p \neq q$
The general solution is the sum of:

- Complementary function: $c_1 + c_2(\frac{q}{p})^k$
- Particular integral: Try $T_k = Ak + B$

Substituting the particular form:

$$p(A(k+1) + B) - (Ak + B) + q(A(k-1) + B) = -1$$

This gives:

$$A = \frac{1}{p-q}, \quad B = \frac{N}{p-q}$$

Therefore:

$$T_k = c_1 + c_2(\frac{q}{p})^k + \frac{k-N}{p-q}$$

Using boundary conditions:

$$\begin{aligned}
T_0 &= c_1 + c_2 - \frac{N}{p-q} = 0 \\
T_N &= c_1 + c_2(\frac{q}{p})^N = 0
\end{aligned}$$

Solving gives:

$$\boxed{T_k = \frac{p+q}{q-p}(k - N\frac{1-(\frac{q}{p})^k}{1-(\frac{q}{p})^N})}$$

Case 2: $p = q = \frac{1}{2}$

For the symmetric case, try particular integral $T_k = Ak^2 + Bk$. This leads to:

$$\boxed{T_k = k(N - k)}$$

These solutions match our implementation in the code:

```
def game_duration(p, q, k, N):
    if p == q:
        return k*(N-k)
    else:
        return ((q+p)/(q-p))*(k-N*((1-(q/p)**k)/(1-(q/p)**N)))
```

2 Question 2: Aggressive Betting Strategy

In this problem, the gambler follows a more aggressive betting strategy compared to the classical gambler's ruin problem in Question 1.

2.1 Part A - Probability of Winning

Let W_k denote the probability of winning the game when the gambler starts with an initial wealth of k dollars.

The aggressive betting strategy can be described as follows:

- If $k < \frac{N}{2}$, the gambler bets the entire amount k. If she wins, her wealth becomes $2k$; if she loses, she is ruined.
- If $k \geq \frac{N}{2}$, the gambler bets the difference $N - k$. If she wins, her wealth becomes N; if she loses, her wealth becomes $k - (N - k) = 2k - N$.

Using this strategy, we can derive the following recurrence relations:

Case 1: $k < \frac{N}{2}$

$$\boxed{W_k = pW_{2k}}$$

This is because if the gambler starts with $k < \frac{N}{2}$, she bets the entire amount k. If she wins, her wealth becomes $2k$, and the probability of winning from there is W_{2k}.

Case 2: $k \geq \frac{N}{2}$

$$\boxed{W_k = p + qW_{2k-N}}$$

In this case, the gambler bets the difference $N - k$. If she wins, her wealth becomes N (the maximum), and she wins the game with probability 1. If she loses, her wealth becomes $2k - N$, and the probability of winning from there is W_{2k-N}.

These re+ currence relations can be solved recursively, leveraging the binary representation of $\frac{k}{N}$ as mentioned in the problem statement.

2.2 Part B - Expected Duration

Similarly, we can derive the recurrence relations for the expected duration of the game under the aggressive betting strategy.

Case 1: $k < \frac{N}{2}$

$$\boxed{T_k = 1 + pT_{2k}}$$

This is because if the gambler starts with $k < \frac{N}{2}$, she bets the entire amount k. If she wins, her wealth becomes $2k$, and the expected duration from there is T_{2k}. The total expected duration is 1 (the current round) plus the expected duration from the next state.

Case 2: $k \geq \frac{N}{2}$

$$\boxed{T_k = 1 + qT_{2k-N}}$$

In this case, the gambler bets the difference $N - k$. If she loses, her wealth becomes $2k - N$, and the expected duration from there is T_{2k-N}. The total expected duration is 1 (the current round) plus the expected duration from the next state.

These recurrence relations can also be solved recursively, with the base cases being $T_0 = T_N = 0$ (the game ends when the gambler is ruined or wins).

The provided code implementation directly follows these recurrence relations.

3 Question 3 - Bad Luck

We need to find the expected number of rounds required for the wealth to reach a target value t, starting from an initial wealth k. Let this expected number of rounds be denoted as τ.

Since each step from k to $k-1$ is independent of the previous steps, we can decompose the expected number of steps using the linearity of expectation:

$$\tau = D_k + D_{k-1} + \cdots + D_{t+1},$$

where D_i is the expected number of rounds needed to go from wealth i to $i-1$ for the first time.

Claim: $D_i = D_j$ for all $i, j > t$.

Proof:

- **Lower Step:** $i - 1$
- **Upper Step:** $i + W$

Each case represents a random walk of width $W+1$, starting from the second step. Thus, all cases exhibit similar calculations.

Consequently, we have:

$$\begin{aligned} T &= \sum D_i = (k-t)D_i \\ &= (k-t)D_1. \end{aligned}$$

To calculate D_1, representing the expected number of rounds needed to reach zero wealth for the first time, define:

$$E[x] = \text{Expected number of rounds to reach 0 from } x.$$

We are particularly interested in $E(1)$, which represents the expected number of rounds to reach zero starting from a wealth of one.

This gives us two base cases:

$$\begin{aligned} E(0) &= 0, \\ E(W+1) &= E(W) + 1. \end{aligned}$$

The recurrence relation is given by:

$$E(i) = 1 + pE(i+1) + qE(i-1).$$

For $i = W$:

$$\begin{aligned} E(W) &= 1 + pE(W+1) + qE(W-1), \\ E(W) &= 1 + p + pE(W) + qE(W-1), \\ E(W) - E(W-1) &= \frac{1+p}{q}. \end{aligned}$$

Similarly,

$$E(W-1) = 1 + pE(W) + qE(W-2).$$

Solving, we find:

$$\begin{aligned} E(W-1) - E(W-2) &= \frac{q+p+p^2}{q^2}, \\ E(W-2) - E(W-3) &= \frac{q^2+pq+p^2+p^3}{q^3}. \end{aligned}$$

In general, we have:

$$E(W-i+1) - E(W-i) = \left(\frac{p}{q}\right)^i + \frac{1}{q}\left[1 + \frac{p}{q} + \cdots + \left(\frac{p}{q}\right)^{i-1}\right].$$

Putting $i = W$,

$$\begin{aligned} E(1) - E(0) &= \left(\frac{p}{q}\right)^W + \frac{1}{q}\left[1 + \frac{p}{q} + \left(\frac{p}{q}\right)^2 + \cdots + \left(\frac{p}{q}\right)^{W-1}\right] \\ &= \left(\frac{p}{q}\right)^W + \frac{1}{q} \cdot \frac{1 - \left(\frac{p}{q}\right)^W}{1 - \frac{p}{q}} \\ &= \boxed{\frac{1 - 2p\left(\frac{p}{q}\right)^W}{q-p}}. \end{aligned}$$

When $p = q = 0.5$, the result will be slightly different.

For this case, we try $b \cdot i$ for a particular solution to get $b = 2$.

Now, we have $E_i = a + 2i$. By satisfying the boundary condition, we find the answer:

$$E_i = 2i + 1.$$

4 Question 4: Stock Price Markov Chain

4.1 Part A - Stationary Distribution

To determine the stationary distribution $\pi = (\pi_0, \pi_1, \ldots, \pi_N)$ of the Markov chain, we need to satisfy the equation:

$$\pi P = \pi$$

where P represents the transition matrix.

For the interior states where $0 < k < N$, the stationary distribution fulfills the condition:

$$\pi_k \cdot p_k = \pi_{k+1} \cdot q_{k+1}$$

This equality indicates that the probability of moving from state k to state $k+1$ must equal the probability of transitioning back from state $k+1$ to state k.

Moreover, we must include the normalization constraint:

$$\sum_{k=0}^{N} \pi_k = 1$$

To find the stationary distribution, we can apply linear algebra techniques. The code we implement utilizes NumPy's linear algebra solver to compute the stationary distribution π as follows:

We first construct the transition matrix P based on the specified transition probabilities $\{(p_k, r_k, q_k)\}_{k=0}^{N}$. Next, we account for the boundary conditions related to the edge states (0 and N). Subsequently, we establish the linear system of equations $\pi P = \pi$ by creating the matrix $A = P^T - I$ and defining the vector b to incorporate the normalization condition. Finally, we employ NumPy's 'np.linalg.solve' function to resolve the linear system and derive the stationary distribution π.

For the second part where we have to find the expected price of the stock in steady state, it given by the weighted sum:

$$E[\text{price}] = \sum_{k=0}^{N} k \cdot \pi_k$$

where each possible price k is weighted by its stationary probability π_k. This value represents the long-term average price of the stock in steady state.

4.2 Part B - Expected Hitting Time

Let $t_{a,s}$ denote the expected time it takes for the price to reach a specific state s starting from state a. We can formulate a recurrence relation for $t_{a,s}$ using the principle of first step analysis:

$$t_{a,s} = 1 + p_a \cdot t_{a+1,s} + r_a \cdot t_{a,s} + q_a \cdot t_{a-1,s}$$

with the boundary condition set as $t_{s,s} = 0$.

This leads to a system of linear equations that can be addressed using linear algebra methods. In our code, we utilize NumPy's linear algebra solver to calculate the expected hitting times $t_{a,s}$. To do this, we construct the coefficient matrix corresponding to the linear equations based on the recurrence relation and apply the boundary condition $t_{s,s} = 0$. This approach allows us to solve the system of linear equations and obtain the expected hitting times $t_{a,s}$.