PROJECT 1: MARTINGALE

Jing Yu

jyu497@gatech.edu

1 INTRODUCTION

In this paper, by applying the martingale strategy, I build a gambling simulator performing probabilistic experiments involving an American roulette wheel.

Roulette [1] is a casino game which main objective is to predict which pocket the roulette ball is going to settle into. Once the ball settles, the respective players who have placed a successful bet including that pocket get paid. Roulette betting is considered an "even money" bet. If you bet N chips and win, you keep your N chips and win another N chips. If you lose, those N chips are lost.

The American roulette wheel [2] has 38 divisions that are identically arranged on the wheel (Figure 1). Therefore, the roulette ball has an equal probability to settle in any of the divisions. The divisions are numbered with 1 to 36, 0 and 00. Numbers from 1 to 36 are alternately colored in red and black, while the single zero and the double zero are colored in green. A player may choose to bet on a single number, various grouping of numbers, color red or black, number odd or even, number high (19–36) or low (1–18). In the scope of this paper, I only consider the strategy that always betting on color black.

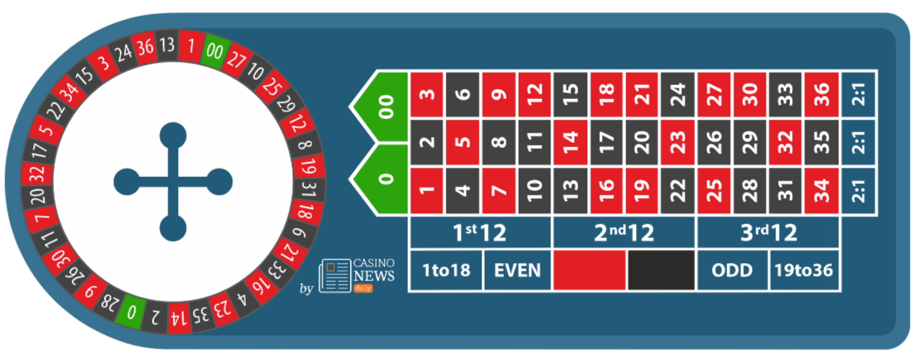

Figure 1— American roulette. Source: [2]

A martingale [3] is a class of betting strategies. It works best when the probability of the occurrence of an event is around 0.5. One typically application is tossing coin. The player using the strategy doubles the bet after every loss, so that the first win would recover all previous losses plus a winning equals to the original stake. No matter how bad luck the player is, he or she will surely eventually win. It seems to be a promising strategy that is certain to make money for the player.

However, this strategy based on the fact that players have infinite wealth and there is no limit on a single bet. It is not the case in reality. The casinos have a betting limit. The exponential growth of the bets can bankrupt unlucky players quickly before their first win.

The martingale strategy can also be applied to roulette, as the probability of hitting black is close to 50%. In this paper, I examine and evaluate this strategy by building a gambling simulator performing probabilistic experiments involving an American roulette wheel.

2 IMPLEMENTATION

I apply the Monte Carlo simulation [4] to experiment how well the betting strategy works. Monte Carlo method helps to explain the impact of uncertainty in prediction and forecasting models. It relies on repeated random sampling to obtain aggregated results.

As discussed in Chapter 1, The American roulette wheel has 38 numbers: 1 to 36, 0 and 00. Numbers 1 to 36 are alternately colored in red and black, numbers 0 and 00 are colored in green. And in assumption, the player always betting on color black. Therefore, the odd of winning of each spin is 18 / 38. On each bet, a random input between 0 and 1 is generated. If the random value is smaller than or equals to win probability, the player wins. Otherwise, he or she loses. The gambling simulator simulates multiple successive bets on the outcomes / spins. Each series of successive bets are referred to as "episode." The results of multiple episodes are averaged to obtain an estimate.

```
episode_winnings = $0
while episode_winnings < $80:
        won = False
        bet_amount = $1
        while not won
                wager bet_amount on black
                won = result of roulette wheel spin
                if won == True:
                        episode_winnings = episode_winnings + bet_amount
                else:
                        episode_winnings = episode_winnings - bet_amount
                        bet_amount = bet_amount * 2
```

Figure 2 — Pseudocode of Professor Balch's betting strategy

Professor Balch's betting strategy at roulette is based on the martingale. The pseudocode is in Figure 2. On each episode, start betting from $1. If you win, you keep your bet, win a same amount and start betting from $1 again. Otherwise you lose your bet and double the next bet. If the target of $80 winning is reached, stop betting and end current episode.

To make it more realistic, the player is limited to a certain amount of bankroll. If he or she goes bankrupt, stop betting and end current episode. If the next bet go beyond on that, bet with available balance.

3 EXPERIMENTS AND RESULTS

3.1 Simple gambling simulator

Two experiments are performed applying the Professor Balch's original betting strategy, allowing the player to use an unlimited bankroll. Each episode consists of 1000 spins. If the target of $80 winning is reached, stop betting.

In experiment 1, the simple simulator is run 10 episodes and the winnings of episodes are tracked. From the results in Figure 3, although the player may lose more than $200 sometimes, the next win could recover all previous losses plus $1. The winnings gradually increase and reach $80 on about spin 160. This results in accordance with expectation on the martingale in ideal situation.

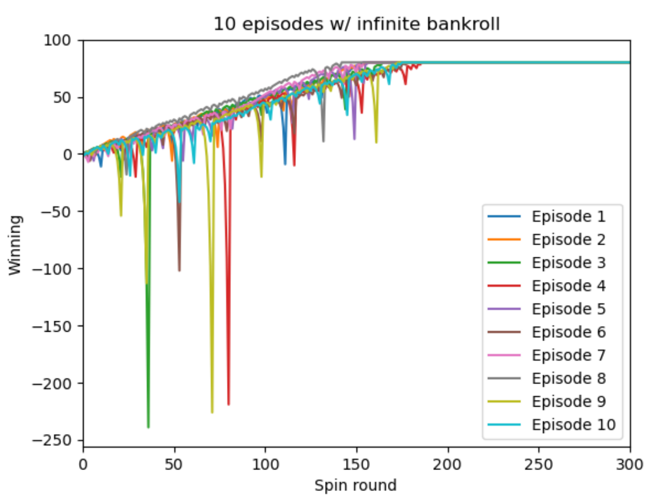

Figure 3 — The winnings of simple simulator after 10 episodes.

In experiment 2, the simple simulator is run 1000 episodes and track the winnings. Figure 4 shows the mean value of winnings and standard deviation for each spin round over 1000 episodes. The standard deviation is plotted as two lines, the upper standard deviation (mean + stdev) and the lower standard deviation (mean – stdev). Figure 5 shows the median value of winnings and standard deviation for each spin round over 1000 episodes. From the results, the original strategy indeed works well as expected in this scenario.

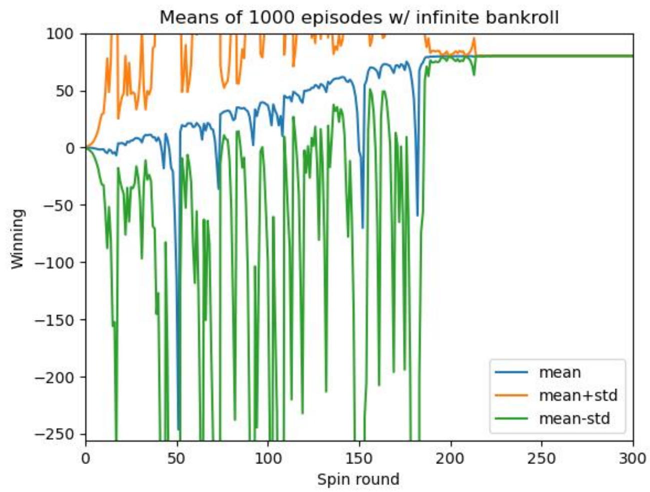

Figure 4 — The means and standard deviations for each spin round over 1000 episodes with infinite bankroll.

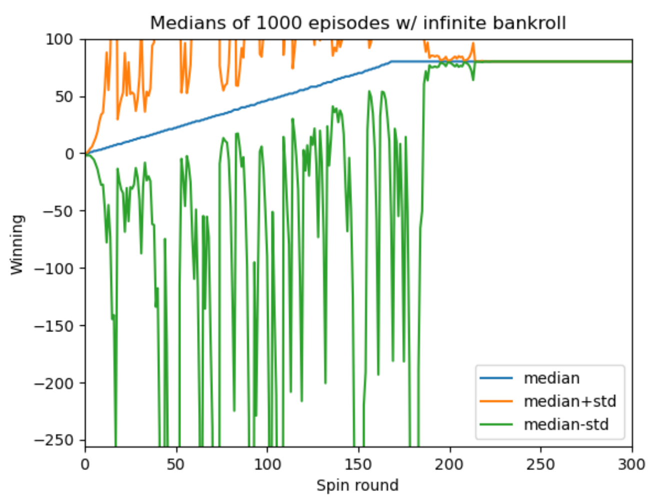

Figure 5 — The medians and standard deviations for each spin round over 1000 episodes with infinite bankroll.

Question 1: Based on the experiment results calculate and provide the estimated probability of winning \$80 within 1000 sequential bets.

With professor Balch's betting strategy, every win will increase episode winning by \$1. The probability of wining \$80 within 1000 sequential bets is equal to winning 80 times within 1000 sequential bets. For American roulette wheel, the probability of winning on each spin is p = 18 / 38. The probability of winning at least 80 times of 1000 spins is:

$$P_{80} = 1 - (C(1000,79) \times p^{79} \times (1-p)^{921} + C(1000,78) \times p^{78} \times (1-p)^{922} + \cdots + (1-p)^{1000}) \approx 1$$

Question 2: What is the estimated expected value of winnings after 1000 sequential bets?

Based on the result of Question 1:

$$Expection = P_{80} \times 80 + P_{79} \times 79 + \cdots \approx 80$$

Question 3: Do the upper standard deviation line (mean + stdev) and lower standard deviation line (mean – stdev) reach a maximum (or minimum) value and then stabilize? Do the standard deviation lines converge as the number of sequential bets increases?

From the results, the upper and lower standard deviation line do converge and stabilize at around spin 220. It indicates that all episodes reach \$80 limit at around 220th spin. This could be approved by the estimated probability of winning \$80 within 220 sequential bets also approximately equals to 1.

3.2 Realistic gambling simulator

To make the Professor Balch's original betting strategy more realistic, a \$256 limit on the bankroll is set. The realistic simulator is run 1000 episodes. Each episode consists of 1000 spins. If the target of \$80 winning is reached, stop betting.

Figure 6 shows the mean value of winnings and standard deviation for each spin round over 1000 episodes. Figure 7 shows the median value of winnings and standard deviation for each spin round over 1000 episodes.

Question 4: Based on the experiment results calculate and provide the estimated probability of winning \$80 within 1000 sequential bets.

Even from \$79 episode winning, it takes 9 successive lose to lose the whole episode. The probability it happens is $(1-p)^9 \approx 0.003$. From my results, 634 of 1000 episodes achieve \$80. The estimated probability P_{win} of winning \$80 within 1000 sequential bets is about 0.634.

Question 5: What is the estimated expected value of winnings after 1000 sequential bets?

$$Expectation = -256 \times P_{lose} + 80 \times P_{win} \approx -43$$

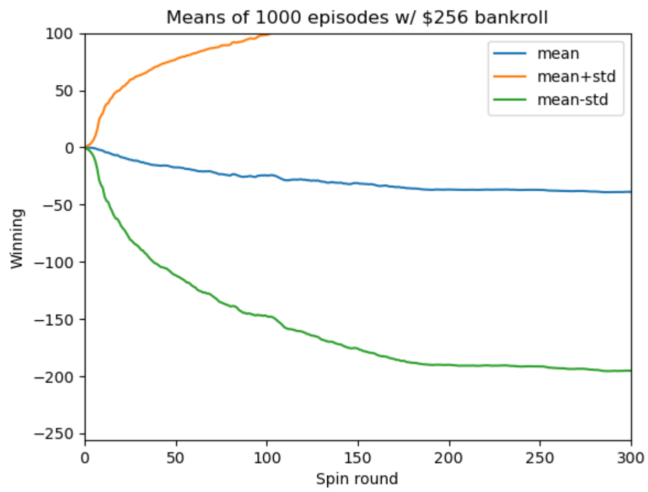

Figure 6— The means and standard deviations for each spin round over 1000 episodes with \$256 bankroll.

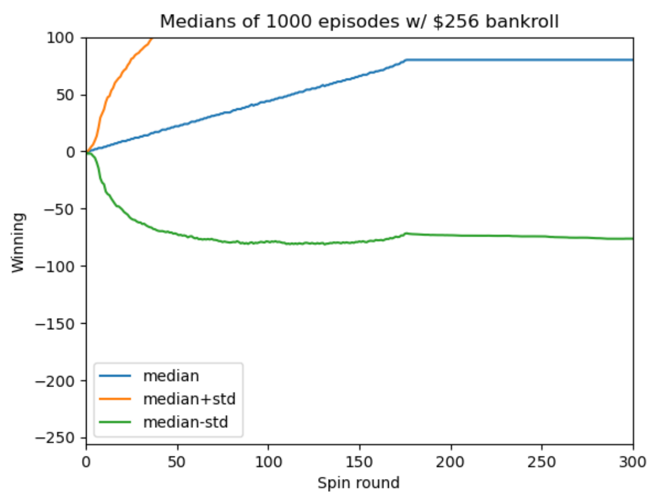

Figure 7— The medians and standard deviations for each spin round over 1000 episodes with \$256 bankroll.

This could explain the mean and median results. Although more than half of episodes achieve $80, the mean winnings converge at around -$40. The probability of a consecutive lose seems small intuitively, it could have a catastrophic effect. Despite the fact that this strategy usually wins a small net reward, it bankrupts a player quickly in reality.

Question 6: Do the upper standard deviation line (mean + stdev) and lower standard deviation line (mean – stdev) reach a maximum (or minimum) value and then stabilize? Do the standard deviation lines converge as the number of sequential bets increases?

From the results, the upper and lower standard deviation line do stabilize at around spin 175. It indicates that all episodes reach either $80 or -$256, the upper and lower limit of winnings. As the number of spins increases, the winnings will converge to one of those two values eventually.

Question 7: What are some of the benefits of using expected values when conducting experiments instead of simply using the result of one specific random episode?

It takes all possibilities into account and gives a risk adjusted result. It could alleviate the impact of randomness and uncertainty in prediction. Therefore, it is fairer than one specific random episode.

4 REFERENCES

1. Roulette. https://en.wikipedia.org/wiki/Roulette
2. American Roulette. https://www.casinonewsdaily.com/roulette-guide/american-roulette/
3. Martingale (betting system). https://en.wikipedia.org/wiki/Martingale_(betting_system)
4. Monte Carlo method. https://en.wikipedia.org/wiki/Monte_Carlo_method