PROJECT 1 REPORT
RANDOM SIGNAL THEORY

FEBRUARY 3, 2020

HARSHITH CHITTAJALLU
ASU ID- 1218707243

Table of Contents

Introduction

In this Project, the task was to estimate the probabilities of certain hands from a unique poker deck consisting of 40 cards, with each suite only having a rank from 1 to 10. The Poker cards are drawn into a deck of 5 cards at random and the task was to estimate the probabilities of certain hands such as Full House, Two Pairs, etc.

Then, the next task was to compare the probabilities obtained by theoretical calculation with that of the results obtained through Monte Carlo Simulations in MATLAB (A technique to simulate multiple instances and determine the number of unique scenarios through repeating for a large number of times) [1]

Finally, the task was to find the relationship between the number of repetitions required by the Monte Carlo Simulation to obtain a favorable result matching upto 5 decimal places with the theoretical result and then compare the probabilities of the cases where a poker deck contains 52 cards with that of the unique 40 card deck.

Theoretical calculation

Let us start by estimating the probabilities of the 40 card poker deck.

The number of ways to choose 5 cards out of 40 is

$$\binom{40}{5} = 40! \div (35! \times 5!) = 658008$$

The hands taken into consideration are

1. Full house: 3 cards of the same rank and 2 cards of same rank with the set of 3 equally ranked cards NOT being equal to the set of 2 equally ranked cards.[2]

   The number of ways a Full house can be obtained is:

   $$\binom{10}{1} \times \binom{4}{3} \times \binom{9}{1} \times \binom{4}{2} = 10 \times 4 \times 9 \times 3 \times 2 = 2160$$

   Explanation: The rank of the first 3 cards can be chosen in (10 choose 1) ways and their suites can be chosen in (4 choose 3) ways. This is then multiplied with the number of ways to choose the remaining 2 cards' rank out of the remaining 9 ranks in (9 choose 1) ways (due to the cards being equal in rank) and their suites chosen in (4 choose 2) ways.

   The probability of getting a Full house is

   $$2160 \div 658008 = 0.0032825$$

2. Three of a kind: 3 cards of the same rank and 2 cards of different ranks. The 2 cards must not be equal as well.

   The number of ways to obtain three of a kind is:

   $$\binom{10}{1} \times \binom{4}{3} \times \binom{9}{2} \times \binom{4}{1} \times \binom{4}{1} = 10 \times 4 \times 9 \times 4 \times 4 \times 4 = 23040$$

   Explanation: The rank of the first 3 cards can be chosen in (10 choose 1) ways and their suites can be chosen in (4 choose 3) ways. This is then multiplied with the number of ways to choose the other 2 cards' rank out of the remaining 9 ranks in (9 choose 2) ways (due to the cards being unequal in rank) and their suites chosen in (4 choose 1) and (4 choose 1) ways respectively.

The probability of getting three of a kind is:

$$23040 \div 658008 = 0.0350138$$

3. Two Pairs: 2 cards of the same rank and 2 cards of a same but different rank along with the final card with a different rank to the pairs of equally ranked cards obtained.

   The number of ways two pairs can be obtained is:

$$\binom{10}{2} \times \binom{4}{2} \times \binom{4}{2} \times \binom{8}{1} \times \binom{4}{1} = 10 \times 9 \times 3 \times 2 \times 3 \times 2 \times 7 \times 3 = 68040$$

   Explanation: The ranks of the first 2 pairs of cards can be chosen in (10 choose 2) ways and their suites can be chosen in (4 choose 2) ^2 ways respectively. This is then multiplied with the number of ways to choose the other card's rank out of the remaining 8 ranks in (8 choose 1) ways and its suite chosen in (4 choose 1) ways.

   The probability of obtaining two pairs is:

$$68040 \div 658008 = 0.787832$$

4. Two of a kind: 2 cards of the same rank and the remaining 3 cards have different ranks each

   The number of ways to obtain two of a kind is:

$$\binom{10}{1} \times \binom{4}{2} \times \binom{9}{3} \times \binom{4}{1} \times \binom{4}{1} \times \binom{4}{1} = 10 \times 6 \times 7 \times 12 \times 4 \times 4 \times 4$$
$$= 322560$$

   Explanation: The ranks of the first pair of cards can be chosen in (10 choose 1) ways and their suites can be chosen in (4 choose 2) ways. This is then multiplied with the number of ways to choose the other 3 different cards' rank out of the remaining 9 ranks in (9 choose 3) ways and their suites chosen in (4 choose 1) ^3 ways.

   The probability of getting two of a kind is:

$$322560 \div 658008 = 0.4902068$$

Now we can easily estimate the probabilities of the deck with 52 cards having 4 suites as the total number of cards in each suite just increases by +3.

By replacing the ranks of the cards from the previous equations where 10 is used with 13, where 9 is used with 12…etc, we can obtain the probabilities of the hands directly which is given by:

1. Probability of getting a full house : 0.00144057
2. Probability of getting Three of a kind : 0.021128451
3. Probability of getting Two pairs : 0.04753901561
4. Probability of getting Two of a kind: 0.4225690276

Monte Carlo Simulation and Algorithm

Monte Carlo simulations are used to model the probability of different outcomes in a process that cannot easily be predicted due to the intervention of random variables. It is a technique used to understand the impact of risk and uncertainty in prediction and forecasting models.

Monte Carlo simulation can be used to tackle a range of problems in virtually every field such as finance, engineering, supply chain, and science.

Monte Carlo simulation is also referred to as multiple probability simulation [3].

Here, the algorithm makes the machine do many iterations of obtaining a random deck of 5 cards and then calculating the probability of obtaining certain hands within that many iterations.

The algorithm goes as follows in 10 simple steps:

1. Define a variable to define the number of iterations required by the monte carlo simulation

2. Define a variable to count the number of times a favorable outcome is generated in each iteration

3. Create a loop with number of iterations defined previously

4. Define an array with a single row vector using the function randperm() to generate 40 numbers randomly in columns

5. Define a variable for the number of cards drawn and equate it to 5 (since all hands have 5 cards each)

6. Define an array such that it takes the first 5 elements from the previous array of 40 numbers

7. Define another array to calculate the ranks of the cards obtained from the array with 5 elements which can consist of numbers 1 to 40 by using the mod() function and dividing with 10 such that the ranks become from 0-9 with 0 being considered as rank 10

8. Define a final array to calculate the number of instances where the ranks repeat and add them to obtain to get their frequency to satisfy certain probability conditions for repetitions

9. Use a conditional if statement to determine whether certain repetitions satisfied the condition of the hand such as having 3 equally ranking cards and 2 equally ranking cards different from the other 3 cards and make the counting variable increase by 1 every time a favorable hand is drawn.

10. End the loop and calculate the probability by dividing the variable that keeps count of the favorable conditions obtained with the total number of possibilities, which is the total number of iterations and then print the answer.

Comparison of results & Conclusion

The results obtained are in the table below:

Probability results(upto 7 decimal places):	Theoretical with deck of 40 cards	Monte carlo n=10e^6 Simulation time: Instantaneous	Monte carlo n=10e^7 Simulation time: 100 seconds	Monte carlo n=10e^8 Simulation time: 10 minutes	Monte carlo n=10e^9 Simulation time: 4 hours	Monte carlo n=10e^10 Simulation time: 80 hours (expected)	Theoretical with deck of 52 cards
Full house	0.0032825	0.003067	0.003214	0.0032616	0.0032819	N/A	0.0014405
Three of a kind	0.0350138	0.029633	0.033274	0.0350021	0.0350112	N/A	0.0211284
Two Pairs	0.0787832	0.071492	0.075391	0.0787315	0.0787831	N/A	0.0475390
Two of a kind	0.4902068	0.416489	0.464190	0.4901981	0.4902038	N/A	0.4225690

Table 1: Probability chart from simulation results

- As we can see from the table, the probability obtained from the Monte carlo simulations becomes more and more accurate to the probability obtained by the theoretical calculations when the number of iterations are increased by a factor of 10. Though this requires high computational power as the hardware used for simulating this project results was not capable of repeating the loop 10e^10 times.

  The relationship between N (Which is basically the number of simulations) and the accuracy of obtaining the required probability is directly proportional. The higher the value of N, the more accurate the obtained probability is, compared with the theoretical probability.

- When the theoretical probabilities of the two sets of cards, one having 40 and the other having 52 are compared, the results are interesting:

- The probability of getting a full house from a 52 card deck is 56.11% lower than the probability of getting a full house from a 40 card deck
- The probability of getting Three of a kind from a 52 card deck is 39.65% lower than the probability of getting Three of a kind from a 40 card deck
- The probability of getting Two Pairs from a 52 card deck is 39.65% lower than the probability of getting Two Pairs from a 40 card deck
- The probability of getting Two of a kind from a 52 card deck is 13.79 % lower than the probability of getting Two of a kind from a 40 card deck

This led me to conclude that in conditions where card repetitions are higher, such as with full house, the probability of obtaining it from a deck of 52 cards will be significantly lower than from a deck of 40 cards due to the required number of combinations being lesser compared to the overall total number of combinations.

Code used for simulations

-Copy pasted directly from MATLAB, copy directly onto MATLAB to simulate

-Comments apply to all codes and are generalized

1. Full House:

```
No_of_trials = 10000000;

% Number of iterations for this Monte Carlo simulation

Full_house = 0;

 % Variable to count the number of favorable cases set to zero initially

for n = 1:No_of_trials

% The loop is iterated 10000000 times
   deck = randperm(40);

% Using an array to generate and randomly place 40 numbers in different indexes
   No_of_cards_drawn = 5;

% variable to define the number of cards drawn
   draw = deck(1:No_of_cards_drawn);

% array to store the first 5 elements from the array of randomly placed 40 numbers which is like drawing 5 cards
from a deck of 40
   rank = mod(draw,10);

% a 5x10 matrix to store the ranks of the 5 numbers or cards drawn from the 40 numbers or cards by dividing them
with 10 and keeping the remainders. The remainder is from 0-9 which represents the 10 ranks with 0 being rank 10.
   instance= sum(rank==[0:9]',2);

% A 10x5 boolean matrix is generated where if a card has a certain rank, the row corresponding to that rank would
become 1 and the remaining rows would be zero. This matrix then has its columns added and converted to a 10x1
matrix where each value represents the frequency of the rank.
   if sum(instance==2)==1 && sum(instance==3)==1

% A conditional statement is used to check when certain frequencies match the required condition such as having 3
cards with same rank and 2 cards with a same but different rank. When this is the case, for example if the no of
times a card with rank 4 is repeated 2 times and not more, the condition would be sum(instance==2)==1
       Full_house = Full_house + 1;

%variable for counting
   end
end
Probability = Full_house/No_of_trials;
```

```
% probability formula
fprintf("%1.10f\n",Probability)

% for printing probability upto 10 decimal places
```

2. Three of a kind:

```
No_of_trials = 10000000;
Three_of_a_kind = 0;

for n = 1:No_of_trials
    deck = randperm(40);
    No_of_cards_drawn = 5;
    draw = deck(1:No_of_cards_drawn);
    rank = mod(draw,10);
    instance= sum(rank==[0:9]',2);
    if sum(instance==3)==1 && sum(instance==1)==2
        Three_of_a_kind = Three_of_a_kind + 1;
    end
end
Probability = Three_of_a_kind/No_of_trials;
fprintf("%1.10f\n",Probability)
```

3. Two Pairs:

```
No_of_trials = 10000000;
Two_pairs = 0;

for n = 1:No_of_trials
    deck = randperm(40);
    No_of_cards_drawn = 5;
    draw = deck(1:No_of_cards_drawn);
    rank = mod(draw,10);
    instance= sum(rank==[0:9]',2);
    if sum(instance==2)==2 && sum(instance==1)==1
        Two_pairs = Two_pairs + 1;
    end
end
Probability = Two_pairs/No_of_trials;
fprintf("%1.10f\n",Probability)
```

4. Two of a Kind:

```
No_of_trials = 10000000;
One_Pair = 0;

for n = 1:No_of_trials
    deck = randperm(40);
    No_of_cards_drawn = 5;
    draw = deck(1:No_of_cards_drawn);
    rank = mod(draw,10);
```

```
    instance= sum(rank==[0:9]',2);
    if sum(instance==2)==1 && sum(instance==1)==3
        One_Pair = One_Pair + 1;
    end
end
Probability = One_Pair/No_of_trials;
fprintf("%1.10f\n",Probability)
```

.

Bibliography

[1] Mathworks, "mathworks.com," 4 2 2020. [Online]. Available: https://www.mathworks.com/videos/working-with-arrays-in-matlab-69022.html.

[2] S. M, Intuitive Probability and Random processes using MATLAB, Rhode Island: Springer, 2006.

[3] J. Chen, "Investopedia," Investopedia, 1999. [Online]. Available: https://www.investopedia.com/terms/m/montecarlosimulation.asp. [Accessed 5 2 2020].