Project 3

Application of Singular Value Decomposition to Image Compression

Diego Hernando Merchán Rueda S01429270
Carlos Roque Fontanez S01299757

Mark Lau, Ph.D.
ENGI – 360
CRN 300

12/04/2023

Contents

MATLAB Project

In this project we must analyze bets and strategies in the game of roulette. This will be done using a code provided by the teacher for use in MATLAB, to apply theory and practice about probability in this game.

However, we do not possess a MATLAB license, therefore we will be using GNU Octave instead to execute functions and display our results.

5.1 Pure Strategy

1. Download the MATLAB file roulette.m from the course web page. The program simulates bets on “even/odd”.

roulette.m	12/3/2023 11:35 PM	Octave.Document...	3 KB

Execute the program by entering the number of spins you wish to play, your initial capital, your standard bet, and your bet multiplying factor. For this part of the project enter 1 (one) as your bet multiplying factor.

```
Welcome to our ELECTRONIC CASINO!
This program simulates the exciting game of ROULETTE.

You can bet on either even or odd.
You can double your bet when you lose;
otherwise, you keep your bet constant.
If you wish to use the martingale strategy
simply enter a multiplying factor of 2
when prompted by the program;
otherwise, enter 1 as bet multiplying factor.

Enter the number of times you wish to play : 50
Enter your initial capital in dollars      : 100
Enter your standard bet in dollars         : 15
Enter your bet multiplying factor          : 1

You won 21 games out of 50 spins.
Estimated probability of winning the game : 0.42
Your final bankroll is $295.
Your net winnings or losses are $-120.
The average number of spins per game was 2.1905.
```

```
Welcome to our ELECTRONIC CASINO!
This program simulates the exciting game of ROULETTE.

You can bet on either even or odd.
You can double your bet when you lose;
otherwise, you keep your bet constant.
If you wish to use the martingale strategy
simply enter a multiplying factor of 2
when prompted by the program;
otherwise, enter 1 as bet multiplying factor.

Enter the number of times you wish to play : 100
Enter your initial capital in dollars      : 500
Enter your standard bet in dollars         : 50
Enter your bet multiplying factor          : 1

You won 52 games out of 100 spins.
Estimated probability of winning the game : 0.52
Your final bankroll is $3300.
Your net winnings or losses are $200.
The average number of spins per game was 1.9231.
```

```
Welcome to our ELECTRONIC CASINO!
This program simulates the exciting game of ROULETTE.

You can bet on either even or odd.
You can double your bet when you lose;
otherwise, you keep your bet constant.
If you wish to use the martingale strategy
simply enter a multiplying factor of 2
when prompted by the program;
otherwise, enter 1 as bet multiplying factor.

Enter the number of times you wish to play : 125
Enter your initial capital in dollars      : 750
Enter your standard bet in dollars         : 75
Enter your bet multiplying factor          : 1

You won 60 games out of 125 spins.
Estimated probability of winning the game : 0.48
Your final bankroll is $4875.
Your net winnings or losses are $-375.
The average number of spins per game was 2.0833.
```

```
Welcome to our ELECTRONIC CASINO!
This program simulates the exciting game of ROULETTE.

You can bet on either even or odd.
You can double your bet when you lose;
otherwise, you keep your bet constant.
If you wish to use the martingale strategy
simply enter a multiplying factor of 2
when prompted by the program;
otherwise, enter 1 as bet multiplying factor.

Enter the number of times you wish to play : 150
Enter your initial capital in dollars      : 1000
Enter your standard bet in dollars         : 100
Enter your bet multiplying factor          : 1

You won 71 games out of 150 spins.
Estimated probability of winning the game : 0.47333
Your final bankroll is $7300.
Your net winnings or losses are $-800.
The average number of spins per game was 2.1127.
```

Take note of your simulation results for various combinations of the input parameters. Present your results in a table with appropriate labels.

Considering that the multiplying factor will always be 1, we can obtain the following results.

Num. of Turns	Initial Capital	Standard Bet	Estimated Probability of Winning the Game	Final Bankroll	Winnings	Losses	Net Winnings /Losses	Average Number of Spins per Game
50	$100	$15	0.42	$295	21	29	$-120	2.1905
100	$500	$50	0.52	$3,300	52	48	$200	1.9231
125	$750	$75	0.48	$4875	60	65	$-375	2.0833
150	$1000	$100	0.47333	$7300	71	79	$-800	2.1127

2. Are the estimated probabilities reasonably close to the value predicted by theory?

```
>> even_odd = 18/38
even_odd = 0.4737
```

Previously, we ran 4 simulations using the provided code. In the 4 simulations, 2 were a little far away, these being the first two. Then we had one that was close, this being 0.48. However, there is one that is almost exact, this being the last one in which we obtained 0.47333.

3. Observe that the same program can simulate bets on "red/black" or "high/low". Why?

These games have a similar structure in terms of probability, that's because the probability of winning is 18/38. What does this mean? Well, this mean that each option covers 18 numbers out of 38 in total, The unique difference between these bets it's:

a. For even/odd we can win depending on the number we choose, even or odd.
b. For red/black we can win depending on the color we choose, red or black.
c. For high/low we can win depending on the number we choose, if is greater than 18 or less than 19.

4. Modify the roulette.m program so that it can simulate"split" bets. Save the modified program under a different file name.

d. First, we changed the 32 line for this:

```
p = 1/19;        % Probability of winning on split
```

This line was changed because in a split bet, only two numbers are chosen to win.

e. Second, we changed the 33 line for this:

```
payoff = 17;      % Payoff for split bet
```

This line was changed because in a split bet, the bet is multiplied by 17 and the result of said multiplication is the amount of money the person wins.

Enter various combinations of input parameters and present your simulation results in a table. Comment on the results you obtained. Include a listing of the modified MATLAB program you used to generate your results.

```
Welcome to our ELECTRONIC CASINO!
This program simulates the exciting game of ROULETTE.

You can bet on a split between two numbers.
You can double your bet when you lose;
otherwise, you keep your bet constant.
If you wish to use the martingale strategy
simply enter a multiplying factor of 2
when prompted by the program;
otherwise, enter 1 as bet multiplying factor.

Enter the number of times you wish to play : 25
Enter your initial capital in dollars      : 250
Enter your standard bet in dollars         : 5
Enter your bet multiplying factor          : 1

You won 2 games out of 25 spins.
Estimated probability of winning the game : 0.08
Your final bankroll is $315.
Your net winnings or losses are $55.
The average number of spins per game was 9.5.
```

```
Welcome to our ELECTRONIC CASINO!
This program simulates the exciting game of ROULETTE.

You can bet on a split between two numbers.
You can double your bet when you lose;
otherwise, you keep your bet constant.
If you wish to use the martingale strategy
simply enter a multiplying factor of 2
when prompted by the program;
otherwise, enter 1 as bet multiplying factor.

Enter the number of times you wish to play : 50
Enter your initial capital in dollars      : 500
Enter your standard bet in dollars         : 10
Enter your bet multiplying factor          : 1

You won 1 games out of 50 spins.
Estimated probability of winning the game : 0.02
Your final bankroll is $190.
Your net winnings or losses are $-320.
The average number of spins per game was 18.
```

```
Welcome to our ELECTRONIC CASINO!
This program simulates the exciting game of ROULETTE.

You can bet on a split between two numbers.
You can double your bet when you lose;
otherwise, you keep your bet constant.
If you wish to use the martingale strategy
simply enter a multiplying factor of 2
when prompted by the program;
otherwise, enter 1 as bet multiplying factor.

Enter the number of times you wish to play : 75
Enter your initial capital in dollars      : 750
Enter your standard bet in dollars         : 15
Enter your bet multiplying factor          : 1

You won 4 games out of 75 spins.
Estimated probability of winning the game : 0.053333
Your final bankroll is $765.
Your net winnings or losses are $-45.
The average number of spins per game was 15.5.
```

```
Welcome to our ELECTRONIC CASINO!
This program simulates the exciting game of ROULETTE.

You can bet on a split between two numbers.
You can double your bet when you lose;
otherwise, you keep your bet constant.
If you wish to use the martingale strategy
simply enter a multiplying factor of 2
when prompted by the program;
otherwise, enter 1 as bet multiplying factor.

Enter the number of times you wish to play : 100
Enter your initial capital in dollars      : 1000
Enter your standard bet in dollars         : 20
Enter your bet multiplying factor          : 1

You won 3 games out of 100 spins.
Estimated probability of winning the game : 0.03
Your final bankroll is $140.
Your net winnings or losses are $-920.
The average number of spins per game was 20.
```

Num. of Turns	Initial Capital	Standard Bet	Estimated Probability of Winning the Game	Final Bankroll	Winnings	Losses	Net Winnings /Losses	Average Number of Spins per Game
25	**$250**	**$5**	**0.08**	**$316**	**2**	**23**	**$55**	**9.5**
50	**$500**	**$10**	**0.02**	**$190**	**1**	**49**	**$-320**	**10**
75	**$750**	**$15**	**0.053333**	**$765**	**4**	**71**	**$-46**	**15.5**
100	**$1000**	**$20**	**0.03**	**$140**	**3**	**97**	**$-920**	**20**

```
>> split = 1/19
split = 0.052632
```

Before discussing the results, we could use the theoretical models for numbers of games, with these we could go into more depth with the commenting of results for a row. Using the first row of the previous table we have:

a. Expected Number of Spins in a Game:

$$E[S] = \frac{1-(1-p)^n}{p}$$

```
>> E_S = (1-((1-(1/19))^25))/(1/19)
E_S = 14.083
```

b. Expected Number of Games Won:

$$E[G] = \frac{1}{(1-p)^n}.$$

```
>> E_G = 1/(1-((1/19)^25))
E_G = 1
```

c. Total Expected Number of Spins

$$E[T] = n + \frac{[1-(1-p)^n]^2}{p(1-p)^n}.$$

```
>> E_T = 25 + ((1 - (1 - (1/19))^25)^2) / ((1/19) * (1 - (1/19))^25)
E_T = 65.332
```

According to the results obtained, from what we can see of the standard probabilities of winning, there is only one that is close to the theoretical value, and this is the third, being 0.053333. Furthermore, we can see that according to the first row, most of the procedures carried out theoretically are far from the results obtained in the table. The closest is the number of times it was possible to win, and the difference was one, in the end twice as many as were won.

5.2 Martingale Strategy

1. Load the roulette.m file into MATLAB. You are going to simulate "even" or "odd" bets again. For this part of the project you are asked to test the martingale strategy with a multiplying factor of 2 (two).

```
Welcome to our ELECTRONIC CASINO!
This program simulates the exciting game of ROULETTE.

You can bet on a even/odd.
You can double your bet when you lose;
otherwise, you keep your bet constant.
If you wish to use the martingale strategy
simply enter a multiplying factor of 2
when prompted by the program;
otherwise, enter 1 as bet multiplying factor.

Enter the number of times you wish to play : 50
Enter your initial capital in dollars      : 100
Enter your standard bet in dollars         : 15
Enter your bet multiplying factor          : 2

You won 14 games out of 25 spins.
Estimated probability of winning the game : 0.56
Your final bankroll is $175.
Your net winnings or losses are $-255.
The average number of spins per game was 1.4286.
```

```
Welcome to our ELECTRONIC CASINO!
This program simulates the exciting game of ROULETTE.

You can bet on a even/odd.
You can double your bet when you lose;
otherwise, you keep your bet constant.
If you wish to use the martingale strategy
simply enter a multiplying factor of 2
when prompted by the program;
otherwise, enter 1 as bet multiplying factor.

Enter the number of times you wish to play : 100
Enter your initial capital in dollars      : 500
Enter your standard bet in dollars         : 50
Enter your bet multiplying factor          : 2

You won 14 games out of 27 spins.
Estimated probability of winning the game : 0.51852
Your final bankroll is $1050.
Your net winnings or losses are $-850.
The average number of spins per game was 1.5714.
```

In an appropriately labeled table present your simulation results for various combinations of input parameters. Comment on the results you obtained by following the martingale strategy.

Considering that the multiplying factor will always be 2 for the Martingale strategy, we can obtain the following results.

Num. of Turns	Initial Capital	Standard Bet	Estimated Probability of Winning the Game	Final Bankroll	Winnings	Losses	Net Winnings /Losses	Average Number of Spins per Game
50	**$100**	**$15**	**0.56**	**$175**	**14**	**11**	**$-255**	**1.4286**
100	**$500**	**$50**	**0.51852**	**$1050**	**14**	**13**	**$-850**	**1.5714**

The Martingale strategy allows you to recover your losses after constantly losing, if you manage to win a game. In the image below we can see how this works. However, it may be that players do not have enough money to continue betting, casinos also set limits on betting, so this can prevent the strategy we just used from happening. We can see that according to the theoretical probability of winning, with this strategy we obtain a greater probability of winning (looking at the first two attempts).

```
You won 18 games out of 31 spins.
Estimated probability of winning the game : 0.58065
Your final bankroll is $835.
Your net winnings or losses are $255.
You won 18 games out of 32 spins.
Estimated probability of winning the game : 0.5625
Your final bankroll is $805.
Your net winnings or losses are $225.
You won 18 games out of 33 spins.
Estimated probability of winning the game : 0.54545
Your final bankroll is $745.
Your net winnings or losses are $165.
You won 19 games out of 34 spins.
Estimated probability of winning the game : 0.55882
Your final bankroll is $985.
Your net winnings or losses are $285.
You won 20 games out of 35 spins.
Estimated probability of winning the game : 0.57143
Your final bankroll is $1015.
Your net winnings or losses are $300.
You won 20 games out of 36 spins.
```

2. How do the number of spins per game, number of games, and total number of spins compare with their corresponding theoretical values? Present your comparisons in a neatly labeled table.

a. Expected Number of Spins in a Game:

$$E[S] = \frac{1-(1-p)^n}{p}$$

```
>> E_S = (1-((1-(18/38))^50))/(18/38)
E_S = 2.1111
```

b. Expected Number of Games Won:

$$E[G] = \frac{1}{(1-p)^n}.$$

```
>> E_G = 1/(1-((18/38)^50))
E_G = 1.0000
```

c. Total Expected Number of Spins

$$E[T] = n + \frac{[1-(1-p)^n]^2}{p(1-p)^n}.$$

```
>> E_T = 50 + ((1 - (1 - (18/38))^50)^2) / ((18/38) * (1 - (18/38))^50)
E_T = 1.8289e+14
```

Data with Code

Num. of Turns	Initial Capital	Standard Bet	Estimated Probability of Winning the Game	Final Bankroll	Winnings	Losses	Net Winnings /Losses	Average Number of Spins per Game
50	$100	$15	0.56	$175	14	11	$-255	1.4286
100	$500	$50	0.51852	$1050	14	13	$-850	1.5714

Data with Equations

Num. of Turns	Initial Capital	Standard Bet	Estimated Probability of Winning the Game	Winnings	Losses	Average Number of Spins per Game	Average Number of Spins per Game in Total
50	$100	$15	1	14	11	2.1111	1.8289*10^14
100	$500	$50	1	14	13	2.1111	1.5844*10^28

Once we have obtained these data, when we compare them, we can see that they are not very far from each other. So, they are more concise and accurate on both sides. However, when using the Martingale strategy, it is denoted that this strategy makes many more victories certain than what this bet could have been when made on a regular basis. The only number that is further from each other is the average number of spins in total for the first row. Then would follow the chances of winning where both are one in the equations and more than 0.5 in practice. However, this can change depending on the margin of error one wants.

4.3 Annexes

1.

- First time of the code:

```
% roulette.m

% This program simulates the game of roulette.
% The roulette has numbers 00,0,1,2,...,36.
% The rules of the bet are:
%   - The player bets on either even or odd.
%   - The player can only win with numbers 1-36.
%   - The numbers 00 and 0 are house wins.
%   - The player can choose to double the bet
%     when a loss occurs and keep it
%     constant when he/she wins.

clc;  clear all;  close all;
disp(' ')
```

```
disp('Welcome to our ELECTRONIC CASINO!')
disp('This program simulates the exciting game of ROULETTE.')
disp(' ')
disp('You can bet on either even or odd.')
disp('You can double your bet when you lose;')
disp('otherwise, you keep your bet constant.')
disp('If you wish to use the martingale strategy')
disp('simply enter a multiplying factor of 2')
disp('when prompted by the program;')
disp('otherwise, enter 1 as bet multiplying factor.')
disp(' ')

n = input('Enter the number of times you wish to play : ');
cap = input('Enter your initial capital in dollars      : ');
std_bet = input('Enter your standard bet in dollars        : ');
r = input('Enter your bet multiplying factor          : ');

p = 18/38;      % Probability of winning on even/odd
payoff = 1;     % Payoff for even/odd bet
win = 0;        % Count the number of wins or games
lose = 0;       % Count the number of losses
nsg = 0;        % Number of spins per game
net = 0;        % Net winnings or losses in dollars
s = [];         % Number of spins in each game
bet = std_bet;  % Initial bet / bet at each spin
k = 1;          % Number of spin

while k <= n
   nsg = nsg + 1;
   if rand() < p
      win = win + 1;
      cap = cap + (payoff + 1)*bet;
      net = net + payoff*bet;
      bet = std_bet;
      s = [s nsg];    % Update with new entry nsg
      nsg = 0;        % Reset nsg after a win
   else
      lose = lose + 1;
      cap = cap - bet;
      net = net - bet;
      bet = r*bet;
   end;
   if (cap >= bet) && (k < n)
      k = k + 1;
   elseif (cap >= bet) && (k == n)
      k = n;
```

```
      break;
   else
      break;
   end;
end;
avgnsg = mean(s);

disp(' ')
disp(['You won ',num2str(win),' games out of ',num2str(k),' spins.'])
disp(['Estimated probability of winning the game : ',num2str(win/k)])
disp(['Your final bankroll is $',num2str(cap),'.'])
disp(['Your net winnings or losses are $',num2str(net),'.'])
if win > 0
   disp(['The average number of spins per game was ',num2str(avgnsg),'.'])
end;
```

- Second time of the code:

```
% roulette.m

% This program simulates the game of roulette.
% The roulette has numbers 00,0,1,2,...,36.
% The rules of the bet are:
%   - The player bets on either even or odd.
%   - The player can only win with numbers 1-36.
%   - The numbers 00 and 0 are house wins.
%   - The player can choose to double the bet
%     when a loss occurs and keep it
%     constant when he/she wins.

clc;  clear all;  close all;
disp(' ')
disp('Welcome to our ELECTRONIC CASINO!')
disp('This program simulates the exciting game of ROULETTE.')
disp(' ')
disp('You can bet on a split between two numbers.')
disp('You can double your bet when you lose;')
disp('otherwise, you keep your bet constant.')
disp('If you wish to use the martingale strategy')
disp('simply enter a multiplying factor of 2')
disp('when prompted by the program;')
disp('otherwise, enter 1 as bet multiplying factor.')
disp(' ')

n = input('Enter the number of times you wish to play : ');
cap = input('Enter your initial capital in dollars      : ');
std_bet = input('Enter your standard bet in dollars         : ');
```

```
r = input('Enter your bet multiplying factor        : ');

%p = 18/38;      Probability of winning on even/odd
%payoff = 1;     % Payoff for even/odd bet
p = 1/19;      % Probability of winning on split
payoff = 17;     % Payoff for split bet
win = 0;       % Count the number of wins or games
lose = 0;      % Count the number of losses
nsg = 0;       % Number of spins per game
net = 0;       % Net winnings or losses in dollars
s = [];        % Number of spins in each game
bet = std_bet;  % Initial bet / bet at each spin
k = 1;         % Number of spin

while k <= n
   nsg = nsg + 1;
   if rand() < p
      win = win + 1;
      cap = cap + (payoff + 1)*bet;
      net = net + payoff*bet;
      bet = std_bet;
      s = [s nsg];    % Update with new entry nsg
      nsg = 0;        % Reset nsg after a win
   else
      lose = lose + 1;
      cap = cap - bet;
      net = net - bet;
      bet = r*bet;
   end;
   if (cap >= bet) && (k < n)
      k = k + 1;
   elseif (cap >= bet) && (k == n)
      k = n;
      break;
   else
      break;
   end;
end;
avgnsg = mean(s);

disp(' ')
disp(['You won ',num2str(win),' games out of ',num2str(k),' spins.'])
disp(['Estimated probability of winning the game : ',num2str(win/k)])
disp(['Your final bankroll is $',num2str(cap),'.'])
disp(['Your net winnings or losses are $',num2str(net),'.'])
if win > 0
```

```
   disp(['The average number of spins per game was ',num2str(avgnsg),'.'])
end;
```

- Third time of the code changed for see each changes in the bet:

```
% roulette.m

% This program simulates the game of roulette.
% The roulette has numbers 00,0,1,2,...,36.
% The rules of the bet are:
%   - The player bets on either even or odd.
%   - The player can only win with numbers 1-36.
%   - The numbers 00 and 0 are house wins.
%   - The player can choose to double the bet
%     when a loss occurs and keep it
%     constant when he/she wins.

clc;  clear all;  close all;
disp(' ')
disp('Welcome to our ELECTRONIC CASINO!')
disp('This program simulates the exciting game of ROULETTE.')
disp(' ')
disp('You can bet on a even/odd.')
disp('You can double your bet when you lose;')
disp('otherwise, you keep your bet constant.')
disp('If you wish to use the martingale strategy')
disp('simply enter a multiplying factor of 2')
disp('when prompted by the program;')
disp('otherwise, enter 1 as bet multiplying factor.')
disp(' ')

n = input('Enter the number of times you wish to play : ');
cap = input('Enter your initial capital in dollars      : ');
std_bet = input('Enter your standard bet in dollars        : ');
r = input('Enter your bet multiplying factor          : ');

p = 18/38;      %Probability of winning on even/odd
payoff = 1;     % Payoff for even/odd bet
%p = 1/19;       % Probability of winning on split
%payoff = 17;     % Payoff for split bet
win = 0;        % Count the number of wins or games
lose = 0;       % Count the number of losses
nsg = 0;        % Number of spins per game
net = 0;        % Net winnings or losses in dollars
s = [];         % Number of spins in each game
bet = std_bet;  % Initial bet / bet at each spin
k = 1;          % Number of spin
```

```
while k <= n
   nsg = nsg + 1;
   if rand() < p
      win = win + 1;
      cap = cap + (payoff + 1)*bet;
      net = net + payoff*bet;
      bet = std_bet;
      s = [s nsg];    % Update with new entry nsg
      nsg = 0;        % Reset nsg after a win
   else
      lose = lose + 1;
      cap = cap - bet;
      net = net - bet;
      bet = r*bet;
   end;
   if (cap >= bet) && (k < n)
      k = k + 1;
      disp(['You won ',num2str(win),' games out of ',num2str(k),' spins.'])
disp(['Estimated probability of winning the game : ',num2str(win/k)])
disp(['Your final bankroll is $',num2str(cap),'.'])
disp(['Your net winnings or losses are $',num2str(net),'.'])
   elseif (cap >= bet) && (k == n)
   disp(['You won ',num2str(win),' games out of ',num2str(k),' spins.'])
disp(['Estimated probability of winning the game : ',num2str(win/k)])
disp(['Your final bankroll is $',num2str(cap),'.'])
disp(['Your net winnings or losses are $',num2str(net),'.'])
      k = n;
      break;
   else
      break;
   end;
end;
avgnsg = mean(s);

disp(' ')
disp(['You won ',num2str(win),' games out of ',num2str(k),' spins.'])
disp(['Estimated probability of winning the game : ',num2str(win/k)])
disp(['Your final bankroll is $',num2str(cap),'.'])
disp(['Your net winnings or losses are $',num2str(net),'.'])
if win > 0
   disp(['The average number of spins per game was ',num2str(avgnsg),'.'])
end;
```