Weining Bao weiningb@usc.edu 8287723653

Project 7: Gambler's Ruin

Consider a simplified betting game based on coin tosses. The player picks heads (or tails) and places a bet of 1 dollar. A coin is tossed. If it comes up heads, the player get his bet back plus an additional amount equal to his bet. If it comes up tails, he loses his bet. Let us suppose that he starts the game with $\$D$ and plays N times (or until he runs out of money). We are interested in finding the following statistics. Run for $D = 30, N = 100$.

a) What is the distribution of the amount of money he has at the end of the game? Find this by simulation and by modelling the game as a birth-death Markov chain where the state corresponds to the amount of money that the player has.

b) He modifies his betting strategy as follows. Each time he loses, he doubles his bet (or bets all of his money if he cannot cover the doubled bet) and when he wins he reverts back to betting $1. Now what is the distribution of money he has at the end of the game?

Matlab Simulation

with code at last

a).

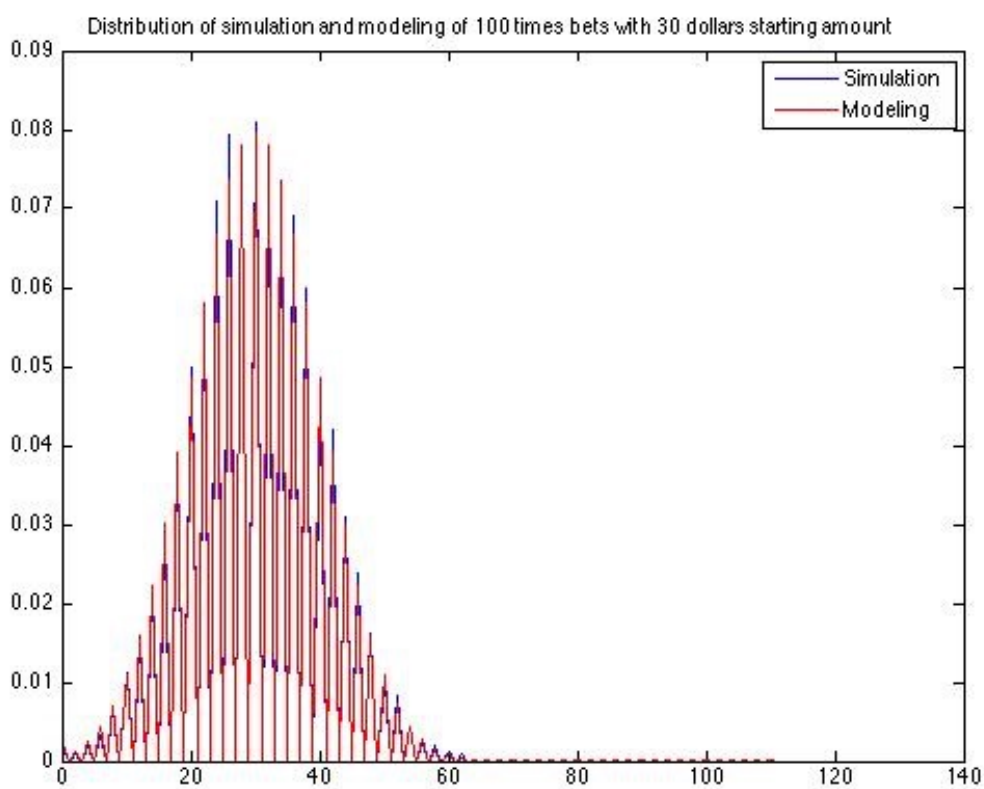
To model the game, the state corresponds to the amount of money that the player has is used. For example, at the beginning the player has D = 30 dollars, so the state the system in is represented by the number 30. If the player losses all his money, the state gets into the state 0. So after playing N = 100 times, the possible states are 0 - N+D = 130. Because the next state depends only on the current state, the system is a Markov Chain. Besides, because there are only transitions between neighboring states, the system can be modeled as a birth-death Markov Chain as following:

This system actually has infinite states if the player keeps playing the game. The solution provided in the lecture note is not applicable here because both α and β equals 0.5, which makes $\rho = \frac{\alpha}{\beta} = 1$ gives $\sum_{i=1}^{\infty} \rho^i$ infinite. So probability transition matrix is made to multiply initial state probability vector N = 100 times to get the analytical distribution of different outcome. To model the infinite states system, we can just make sure that the dimension of square matrix and vector is larger than N+D+1. So here 150 is used as dimension. The initial state probability vector is $\begin{bmatrix} 0 & ... & 1 & ... & 0 \end{bmatrix}$ where only the 31th element is 1. The probability transition matrix is $\begin{bmatrix} 1 & 0 & . & . & . & 0 \\ 0.5 & 0 & 0.5 & . & . & 0 \\ 0 & 0.5 & 0 & 0.5 & ... & 0 \\ & & ... & & & \\ 0 & . & . & 0.5 & 0 & 0.5 \\ 0 & . & . & . & 0.5 & 0.5 \end{bmatrix}$. There is no transition out of the state 0 where there is half-half probabilities to get into nearby states for other states. The last row vector has different transition probability but it does not matter because we care only about the final distribution of state 0 - 130 and states with number larger than 130 will have only zero values.

Simulation of playing game N = 100 times is also run 10000 times in the Matlab simulation. The amounts of money left are recorded and then normalized by being divided by 10000 to give the distribution. Means of both simulation and modeling are calculated and distributions are plotted. Here are the results:

Part a:

Simulating 100 bets with starting $30 10000 times;
Modeling with birth-death Markov chain;
The mean of simulation distribution is 30.151200.
The mean of modeling distribution is 30.000000.

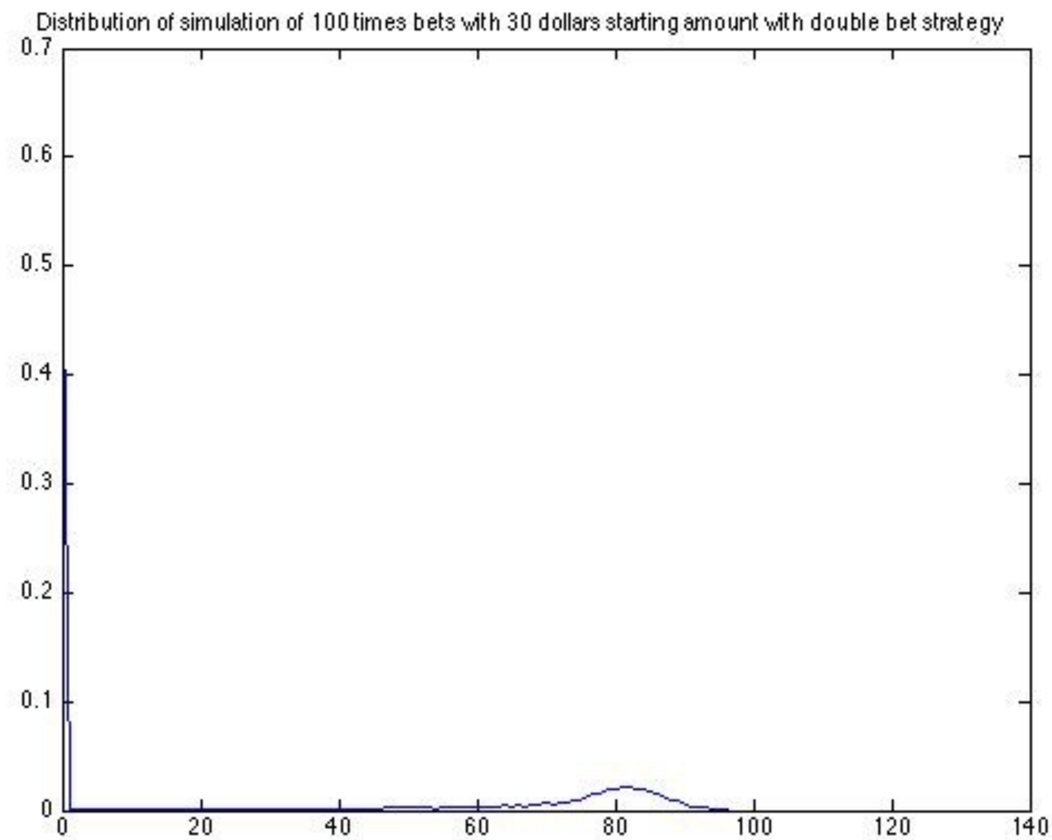


As we can see, the means of simulation and modeling are close. It is reasonable to get 30 as the mean because the probability is half-half to get into or out the neighboring states. Two plots of distribution are also close. It is interesting that the probabilities of states represented by odd numbers are all zero after 100 times of plays.

2).
The modified betting strategy disqualifies the system as a Markov Chain because the future system state depends on all the past history of system states. So only simulation is made here and here are the results:

Part b:

Simulating 100 bets with starting $30 10000 times and modified strategy;
The mean of simulation distribution is 29.517100.



The mean is still about 30 but according to the distribution plot, it is about half probability that the player will lose all the money at last. To keep the mean still at about 30, there are higher possibilities in states with larger number representing more money at last. So with this strategy, players may get more money at last, but they also take more risk.

Code:

```
% part a
fprintf('\nPart a: \n\n');

dim = 150; % dimension of the state and transition matrix
d = 30; % starting amount
iter = 100; % number of iteration

run = 10000; % times of simulations
fprintf('Simulating %d bets with starting $%d %d times;\n', iter, d, run);
left = zeros(1, iter+d+1); % the possible amount at the end is 0 to N+D
for r = 1:run % simulate 10000 times
   amount = d;
   i = 1;
   while (amount > 0) && (i < iter+1) % do iter times bets till no money
      if rand() > 0.5 % half-half to win or loss
         amount = amount + 1;
      else
         amount = amount - 1;
      end
      i = i + 1;
   end
   left(amount+1) = left(amount+1) + 1;
end
left = left ./ run; % normalize the times to probability
m = 0; % mean
for i = 1:(d+iter)
   m = m + left(i+1)*i;
end

fprintf('Modeling with birth-death Markov chain;\n');
pt = zeros(dim, dim); % prob transition matrix
ps = zeros(1, dim); % state probability
ps(d+1) = 1; % state starts from d where subscript 1 represents state 0
pt(1, 1) = 1; % if loss to no money, end
for i = 2:(dim-1) % possibility transition from all states follow birth-death Markov chain
   pt(i, i-1) = 0.5;
   pt(i, i+1) = 0.5;
end
pt(dim, dim-1) = 0.5;
pt(dim, dim) = 0.5; % possibility transition from last state, does not matter as long as dim-1 > N
+D

for i = 1:iter % run bet iter times
   ps = ps * pt;
end
ps = ps(1, 1:(d+iter+1)); % only need non-zero possiblities
sum = 0; % mean
```

```
for i = 1:(d+iter)
   sum = sum + ps(i+1)*i;
end

fprintf('The mean of simulation distribution is %f.\n', m);
fprintf('The mean of modeling distribution is %f.\n', sum);
figure;
plot(0:(iter+d), left); hold on; % plot distribution
plot(0:(iter+d), ps, 'r'); hold off; % plot distribution
title(['Distribution of simulation and modeling of ', num2str(iter), ' times bets with ',...
     num2str(d), ' dollars starting amount']);
legend('Simulation', 'Modeling');

% part b
fprintf('\nPart b: \n\n');
fprintf('Simulating %d bets with starting $%d %d times and modified strategy;\n', iter, d, run);
left = zeros(1, iter+d+1); % the possible amount at the end is 0 to N+D
for r = 1:run % simulate 10000 times
   amount = d;
   i = 1;
   bet = 1; % original bet is 1
   while (amount > 0) && (i < iter+1) % do iter times bets till no money
      if rand() > 0.5 % half-half to win or loss
         amount = amount + bet; % win bet
         bet = 1; % revert bet to 1
      else
         amount = amount - bet;
         if amount < (2*bet) % if amount can not cover double bet
            bet = amount;
         else % double bet
            bet = 2 * bet;
         end
      end
      i = i + 1;
   end
   left(amount+1) = left(amount+1) + 1;
end
left = left ./ run; % normalize the times to probability
m = 0; % mean
for i = 1:(d+iter)
   m = m + left(i+1)*i;
end

fprintf('The mean of simulation distribution is %f.\n', m);
figure;
plot(0:(iter+d), left); % plot distribution
title(['Distribution of simulation of ', num2str(iter), ' times bets with ',...
     num2str(d), ' dollars starting amount with double bet strategy']);
```