Markov Chains and Chutes and Ladders

Liam Koehler Kara Black Yu Cao

April 2019

Abstract

Chutes and ladders is an easily explained and popular children's game. Due to the nature of the grid-based gameplay and simple rules, the game is easily converted to matrix form. We use this converted matrix to explore various aspects of probability vectors and markov chains.

1 Introduction

Chutes and ladders is a simple and popular track-based board game. Players take turns spinning a 6-option, equal probability spinner, and moving their pieces an appropriate number of steps along the track [2]. A **track** is defined as a finite number of **squares**, from 0 to n, where each square represents a valid position for the player character, and is denoted $square_i$, for i 1 to n. The **start square** is defined as $square_0$, and the **end square** is defined as $square_n$. A player is declared the winner when they reach the end square. A track can also be defined with shortcuts, which allows the track to form loops, and other nonlinear structures. This is a topic we will re-visit in more detail in the following sections.

To reason about the game in any reasonable way, we must first convert these rules into a mathematical model of some kind. In our case, we have chosen to represent a chutes and ladders game as a Markov Chain. A **Markov chain**, also known as a discrete-time Markov chain, was discovered by a Russian mathematician, Andrei Markov [3]. Markov chains are a stochastic model describing a sequence of possible events in which the probability of the next event depends only on the current state, and is not affected by past states. This property is refereed to as the **Markov property**, or **Memorylessness** [5].

At each step of the Markov chain, the system can change from one state to another or remain in the current state according to the probability distribution. A change in state is called a "Transition", and the probabilities associated with different state changes are called transition probabilities.Markov chains are widely used in areas, such as physics, chemistry, statistics, economics, and web applications, such as Google's PageRank.

Due to the memorylessness of each game state, and the consistent randomness of the spinner, chutes and ladders is conveniently represented via probability vectors and markov chains. Although the actual game only supports a single

board configuration, for the sake of mathematical interest, we chose to abstract this configuration, and define our algorithms for any valid chutes and ladders board. This abstraction allows us to explore many concepts and patterns.

2 Markov Chains

2.1 Representing Markov chains mathematically

Suppose we have an ordered state:

$$\vec{x}_0, \vec{x}_1, \vec{x}_2 \cdots \vec{x}_{t-2}, \vec{x}_{t-1}, \vec{x}_t, \vec{x}_{t+1}, \vec{x}_{t+2}, \cdots$$

So our conditional probability at $\vec{x}_{t+1}$ of moving to the next state is only influenced by $\vec{x}_t$. Which is written as:

$$P(\vec{x}_{t+1} \mid \cdots \vec{x}_t{-}2, \vec{x}_{t-1}, \vec{x}_t) = P(\vec{x}_{t+1} \mid \vec{x}_t)$$

We can now define:

1. A **probability vector** is a vector with non-negative entries that add up to 1
2. A **stochastic matrix** is a square matrix whose columns are all probability vectors

Thus, a Markov chain is a sequence of probability vectors $\vec{x}_0, \vec{x}_1, \vec{x}_2 \cdots \vec{x}_n$ in combination with a stochastic matrix A such that

$$\vec{x}_1 = A * \vec{x}_0, \vec{x}_2 = A * \vec{x}_1 \cdots, \vec{x}_{n+1} = P * \vec{x}_n$$

2.2 A Markov chain example

Let us take the example of a stock market, which has three states: bull, bear, and stagnant.

Each state has a certain probability to transform to the next state. For example, Bull market has 90% chance to remains itself to be a bull market, but it has 7.5% chance to transform into a bear market and has 2.5% chance to transform into a stagnant market. Similarly, a bear market has 15% chance to transform into a bull market, 80% chance to remains itself to be a bear market and has 5% chance to transform into a stagnant market. And if the stock is a stagnant market, it has 25% chance to transform into a bull market, 25% chance to transform into a bear market, and 50% chance to remains itself to be a stagnant market. Since we have has 3 states, and they will become one of the three after transforming , thus matrix A is a 3*3 matrix, If we define a matrix A that the value of a position A(i,j) it's the value of the probability transform from state i to state j, in this case we have i=1,2,3 and j=1,2,3. if we use 1 represents bull market, 2 represents bear market, and 3 represents

stagnant market, each row in the matrix represents corresponding state. we can get the following matrix:

$$\begin{bmatrix} 0.9 & 0.15 & 0.25 \\ 0.075 & 0.8 & 0.25 \\ 0.025 & 0.05 & 0.5 \end{bmatrix}$$

To be more specific, $A(1 \mid 1)$ has i=1, j=1 means the probability of a bull market would transform into a bull market which equals to 0.9 (90%) , $A(1 \mid 3) = 0.025$ means the probability of a bull market would transform into a stagnant market, etc.

The matrix A we got here is called a stochastic matrix in a Markov chain.

Also, before transforming mechanism happens, our stock market also has probabilities to be one of those three markets, in this example, we assume that the initial probabilities to be 30% bull market, 40% bear market, 30% stagnant market. Thus we have a 3*1 matrix that represents three probabilities that the initial market tend to be. Mathematically, it can be written as

$$\begin{bmatrix} 0.3 \\ 0.4 \\ 0.3 \end{bmatrix}$$

Then if we use our stochastic matrix multiplied by our probability vector, we can get a new probability vector, that is after one round of transition, the the market has a 40.5% to be a bull market, a 41.75% to be a bear market, and a 17.75% to be a stagnant market. We can represent this whole process mathematically, that is :

$$A * \vec{x}_0 = \vec{x}_1 : \begin{bmatrix} 0.9 & 0.15 & 0.25 \\ 0.075 & 0.8 & 0.25 \\ 0.025 & 0.05 & 0.5 \end{bmatrix} \begin{bmatrix} 0.3 \\ 0.4 \\ 0.3 \end{bmatrix} = \begin{bmatrix} 0.405 \\ 0.4175 \\ 0.1775 \end{bmatrix}$$

- A is the stochastic matrix
- $\vec{x}_0$ is the initial probability vector
- $\vec{x}_1$ is the new probability vector

This is an example of one step of Markov chains. In general, it can be represented by

$$A * \vec{x}_{n-1} = \vec{x}_n$$

2.3 Steady state vectors

After a sufficient number of steps from $\vec{x}_0$, say $\vec{x}_n$, we notice that the "final" probabilities stabilized to:

$$\begin{bmatrix} 0.625 \\ 0.3125 \\ 0.0625 \end{bmatrix}$$

If we change our initial $\vec{x}_0$ to an arbitrary probability vector, while allowing the stochastic matrix remains the same, we find that $\vec{x}_n$ for very large n, will always converge to:

$$\begin{bmatrix} 0.625 \\ 0.3125 \\ 0.0625 \end{bmatrix}$$

This is one of the properties that the Markov chain has, which means at this point, once we get the probability vector equals to the vector above, if we keep going through the Markov process, we get the exact same probability vector. Also, if we got the stable probability distribution of the corresponding state transition matrix of Markov chain model, then we can start with arbitrary probability distribution of the sample, into state transition matrix of Markov chain model, so that after some sequence transformation, we finally can get stable conforms to the corresponding probability distribution of the sample, or we can say, without knowing the initial probability vector, once we know the stochastic matrix, we can get the stable vector, and that vector is what we called steady-state vector for Markov chain, having said that, we can use this property to estimate the final state we will have and use that result to do the decision making, just like the stock market example, if you are an investor, you can use the steady-state to decide whether you should invest in this stock market or not. But how can we solve the steady-state for a Markov chain? We know that the Markov chain will converge to the same result, and our stochastic matrix remains the same, thus we can set the equation :

$$A * \vec{x}_i = \vec{x}_i$$

, to find out the steady-state vector and then we can monitor the probability of bull market is large or not and decide whether we should invest in this stock market or not. To be specifically in this example, we can assume the steady-state to be

$$\vec{x}_i = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix}$$

And our stochastic matrix is given, so if we solve $A * \vec{x}_i = \vec{x}_i$, we can get the answer we want.

3 Chutes and Ladders

3.1 Introduction

The rules of Chutes and Ladders are very simple. Like many board games, a turn consists of generating a random number, moving the current player's pawn that number of squares along the board, and then, if applicable, taking some action indicated by the square they land on. For Chutes and Ladders

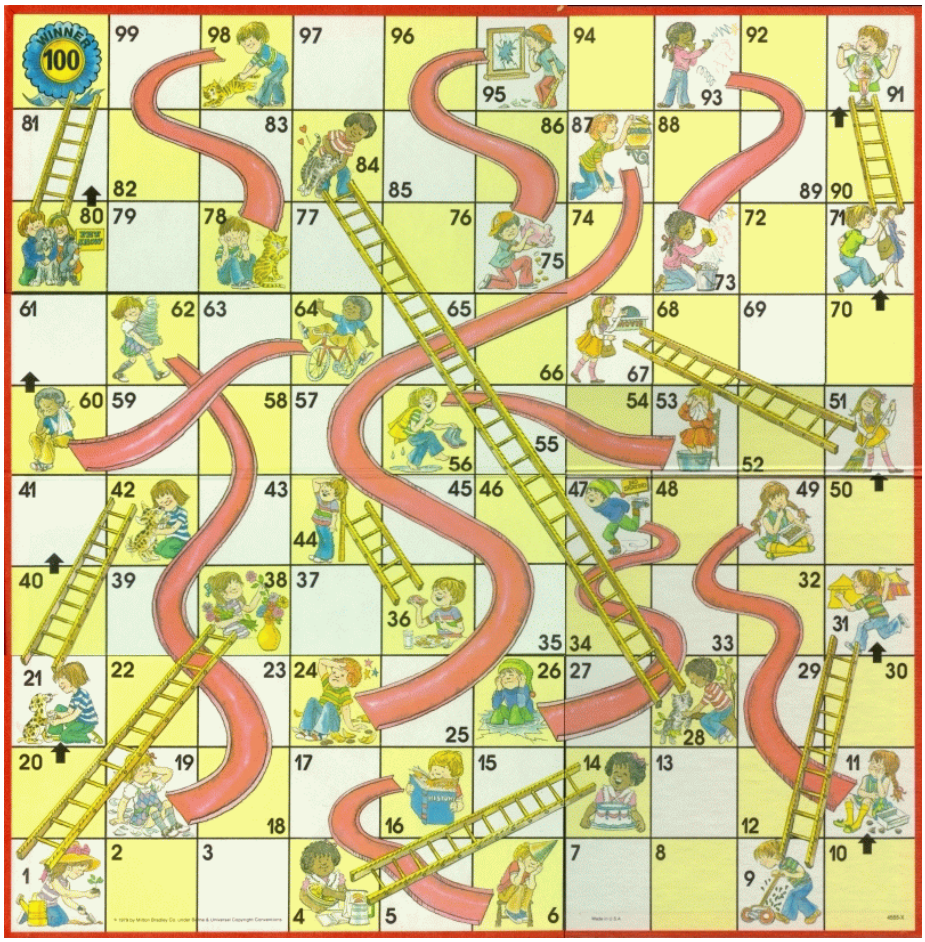

Figure 1: The classic Hasbro Chutes and Ladders board with 100 squares, 10 chutes, and 9 ladders

specifically, the random number is generated by spinning an arrow attached to a spinner board and then reading the number the arrow points to when it stops. If a pawn ends its movement on a square that is at the top of a chute or the bottom of a ladder, it is immediately moved to the other end of the chute or ladder. Then the turn ends, and the next player takes their turn. A player wins the game by landing on the last square (square number 100 in the classic board) before any other players. The squares on the Chutes and Ladders board are numbered to indicate the order in which they are meant to be travelled. It is worth noting that the only meaningful difference between a chute and a ladder is that a ladder causes the player's pawn to move from lower-number square to a higher-number square, and a chute does the opposite.

From these rules, we can see that the game Chutes and Ladders can easily be

represented with probability vectors, stochastic matrices, and Markov chains. We can see this is true because the probability of moving to any particular square on the board is determined entirely by two things: The design of the board itself - the number of squares, the positions of the chutes and ladders, and the possible numbers on the spinner - and the position of the current player's pawn on the board. This means the the game is memoryless. In other words, the history of the game, the past positions of the current player's pawn, and the positions of other player's pawns have no influence on where the current pawn will end up next. It also means that the movement of the pawns is determined entirely by chance, rather than by player choice or strategy. This makes it much easier to find and describe the relevant probabilities and organize them into vectors and matrices.

3.2 Representing a Chutes and Ladders game with vectors and matrices

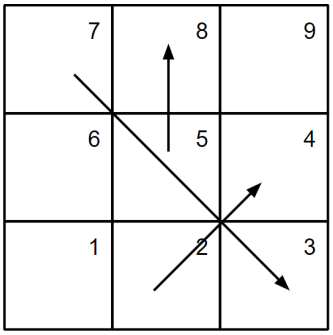

Figure 2: An example Chutes and Ladders board with 9 squares. The board has one chute, represented by an arrow pointing from square 7 to square 3. The board also has two ladders, represented by an arrow pointing from square 2 to square 4 and an arrow pointing from square 5 to square 8.

In order to explain the process of constructing a matrix, and eventually a Markov chain, for a given Chutes and Ladders board, we will be using this relatively simple 3x3 board as an example. The chutes and ladders are both represented by arrows, because the only meaningful difference between a chute and a ladder is what direction it points. In this section we will refer to the square that contains the top of a chute or the bottom of a ladder as the "base" of an arrow, and the square that contains the bottom of a chute or the top of a ladder as the "target" of the arrow. In the diagram of the board, the arrow points from the base and toward the target.

The first thing to notice is that this board is essentially just a linear path that has been folded to into a square to make it easier to use as a game board. By stretching it back out, we can see more easily how to represent this board with vectors.

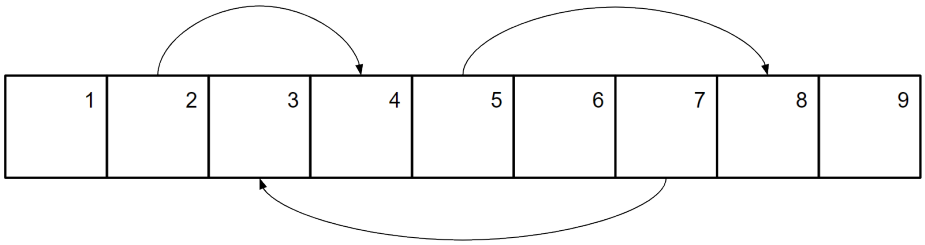

Figure 3: The same example board of 9 squares, now arranged in a line. There is still one chute represented by an arrow pointing from square 7 to square 3. There are also still two ladders represented by arrows pointing from square 2 to square 4 and from square 5 to square 8

Essentially, this is a set of 9 numbered squares, some of which are the base or target of an arrow. We will also need to decide what numbers the spinner associated with this board has. In this case, we will use a spinner numbered from 1 to 4 with an equal chance of landing on any of the four options. From this new linear board, and the spinner we just defined, we can start constructing probability vectors that correspond to particular positions on the board.

The probability vector we will construct will be a 1 x n vector corresponding to a particular square j on the board, where n is the number of squares on the board. In this vector the i^{th} entry will represent the probability of moving to square i from square j. To construct the probability vector, follow the steps described below. As an example, we will show the construction of the probability vector for square 4 of our example board, along with the description of the general steps.

- Setup: Let n be the number of squares on the board. Let s be the number of spaces on the spinner. Let j be the square the vector corresponds to. For our example, $n = 9$, $s = 4$, and $j = 4$
- Step 1: If square j is the base of an arrow pointing toward square t, create a 1 x n vector where entry t is 1 and all other entries are 0
- Step 2: Otherwise, create a 1 x n vector where each entry $i \leq j$ is 0, and each entry $j < i < j + s$ is $\frac{1}{s}$. For our example, this gives us the following vector:

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0.25 \\ 0.25 \\ 0.25 \\ 0.25 \\ 0 \end{bmatrix}$$

- Step 3: If $j + s > n$, replace the value in entry j with $\frac{j+s-n}{s}$. This should result in a vector where the sum of the entries equals 1, i.e. a probability vector.

- Step 4: For each non-zero entry i, check if square i is the base of an arrow. If it is, let the t be the target square of the arrow. Then, add the value in entry i to entry t. Then, replace the value in entry i with 0. This results in the probability vector corresponding to space j. For our example, the resulting vector is:

$$\begin{bmatrix} 0 \\ 0 \\ 0.25 \\ 0 \\ 0 \\ 0.25 \\ 0 \\ 0.5 \\ 0 \end{bmatrix}$$

This 9x1 vector is a probability vector where the i^{th} entry represents the probability of moving to square i from square 4 on the next turn.

If we construct a probability vector in this way for each square on the board and then put these probability vectors together in order, we get the following 9x9 stochastic matrix:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.25 & 0 & 0.25 & 0.25 & 0 & 0.25 & 1 & 0 & 0 \\ 0.5 & 1 & 0.25 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.25 & 0.25 & 0 & 0.25 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.25 & 0 & 0.25 & 0.5 & 1 & 0.25 & 0 & 0.75 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.25 & 0 & 0.25 & 1 \end{bmatrix}$$

This matrix has entries $[p_{i,j}]$ which correspond to the probability of moving to square i from square j. Note that for squares that contain the base of an arrow, the corresponding column contains entries that are all 0 except for the one that the chute or ladder points to, which is 1. Because there is no way to end a turn on one of these squares, there is then also no way to start a turn on one of them, therefore so far there no reason to care at all what values these columns contain. We choose to construct them in the way shown in order to make our eventual Markov chain behave more reasonably. We do something similar with the column corresponding to the last square on the board. A pawn that reaches the last square has won the game, and therefore has nowhere left to go. For this reason the last column of the matrix has a 0 in every row but the last, indicating that the pawn has no chance of doing anything other than staying where it is.

There are just a few more steps left, in order to account for two special situations in the game. The first concerns the question of where the pawns are placed at the very beginning of the game, before anyone has taken a turn. The rules of Chutes and Ladders state that on a player's first turn, they move their pawn the number of squares indicated by the spinner, starting with square 1. This rule means that before anyone has taken a turn, all pawns are off the board. Effectively, there is a "square 0" that all pawns start at. We can account for this square 0 easily by just adding an additional row and column to our stochastic matrix. The added row has entries that are all zero, and the added column is constructed in the same way as the probability vectors for all other squares.

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.25 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.25 & 0.25 & 0 & 0.25 & 0.25 & 0 & 0.25 & 1 & 0 & 0 \\ 0.5 & 0.5 & 1 & 0.25 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.25 & 0.25 & 0 & 0.25 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.25 & 0 & 0.25 & 0.5 & 1 & 0.25 & 0 & 0.75 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.25 & 0 & 0.25 & 1 \end{bmatrix}$$

This results in a 10x10 matrix, with entries $[p_{i,j}]$ corresponding to the probability of moving to square $i-1$ from square $j-1$.

The second special situation concerns what should be done when moving the number of squares indicated by the spinner would take a pawn past the last square on the board. This situation is a bit more complicated, as there are several variant rules depending on the version of Chutes and Ladders one is playing. The rule used in the process described above states that if the number indicated by the spinner would take the pawn past the last square on the board, the pawn instead stays in the square it is and does not move at all (we refer to this rule variant as the "default" rule or the "sticky" rule). Other possible rules for this situation, along with the associated alternate Step 3 include:

- **The "brick wall" rule:** move the pawn to the last square and stop there. The player wins the game. In this case, Step 3 is: If $j+s>n$, add $\frac{j+s-n}{s}$ to the value in entry n.
- **The "reverse" rule:** move the pawn backward the number of squares indicated by the spinner, rather than forward. In this case, Step 3 is: If $j+s>n$, replace all entries with new values such that each entry $i \geq j$ is 0, and each entry $j>i>j-s$ is $\frac{1}{s}$.

Each of these rule variants results in a slightly different matrix. In our example, the only square that can have a spin result that would take it off the board is square 8, so we can look at just the column vector for square 8 under each of the possible rule conditions.

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0.75 \\ 0.25 \end{bmatrix}$$ Sticky (default)

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$ Brick Wall

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0.25 \\ 0 \\ 0.25 \\ 0 \\ 0.25 \\ 0.25 \end{bmatrix}$$ Reverse

To summarize, we now know how to create a stochastic matrix that represents the length of the game board, the position of all the chutes and ladders, the size of the spinner, and the choice of variant rules for the end of the board. Now we can move on to using Markov chains to analyze and understand how a game will play out using a particular board.

3.3 Generating and Interpreting Markov chains for a small board

To generate a Markov Chain for a Chutes and Ladders game, recall that a Markov chain starts with two parts: a stochastic matrix P, and a probability vector $\vec{x_0}$. From these starting pieces, we can generate $\vec{x_1}$, $\vec{x_2}$, ... , $\vec{x_{m-1}}$, $\vec{x_m}$, $\vec{x_{m+1}}$, ... where $\vec{x_m} = P\vec{x_{m-1}}$

For our purposes, P is the stochastic matrix representing the structure of the board and other rules decisions, created using the method described in the previous section. The $\vec{x_0}$ we use represents the state of the board we want to

start our analysis from, or potentially a set of possible board states instead. Thus, for a board with n squares, we have:

- P : an $n+1 \times n+1$ stochastic matrix with elements $[p_{i,j}]$ representing the probability of the pawn moving to square $i-1$ from square $j-1$ on the next turn
- $\vec{x_0}$: an $n+1 \times 1$ probability vector that represents the possible initial states of the game board
- $\vec{x_m}$: an $n+1 \times 1$ probability vector equal to $P\vec{x_{m-1}}$ with elements $\vec{x_{mk}}$ representing the probability of the pawn being on square $k-1$ after m turns, from the initial state $\vec{x_0}$

To illustrate, we will examine our example stochastic matrix from the previous section using different $\vec{x_0}$. First, consider an $\vec{x_0}$ with a 1 in one of the entries and 0 in all others:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.25 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.25 & 0.25 & 0 & 0.25 & 0.25 & 0 & 0.25 & 1 & 0 & 0 \\ 0.5 & 0.5 & 1 & 0.25 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.25 & 0.25 & 0 & 0.25 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.25 & 0 & 0.25 & 0.5 & 1 & 0.25 & 0 & 0.75 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.25 & 0 & 0.25 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0.25 \\ 0.25 \\ 0 \\ 0.25 \\ 0 \\ 0.25 \\ 0 \end{bmatrix}$$

In this example, we can see that using an $\vec{x_0}$ with a 1 in the fourth entry and 0 in all others results in an $\vec{x_1}$ that is equivalent to the fourth column of P. This $\vec{x_0}$ represents starting a pawn on square 3, and the resulting $\vec{x_1}$ represents the probability of being at a particular square after taking one turn from that starting point. We can extend this to see that an $\vec{x_0}$ with a 1 in one of its entries and a 0 in all others represents starting the pawn at the corresponding square.

If we instead use an $\vec{x_0}$ with values in several entries, we get a more complex result:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.25 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.25 & 0.25 & 0 & 0.25 & 0.25 & 0 & 0.25 & 1 & 0 & 0 \\ 0.5 & 0.5 & 1 & 0.25 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.25 & 0.25 & 0 & 0.25 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.25 & 0 & 0.25 & 0.5 & 1 & 0.25 & 0 & 0.75 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.25 & 0 & 0.25 & 1 \end{bmatrix} \begin{bmatrix} 0.333 \\ 0 \\ 0.125 \\ 0 \\ 0 \\ 0 \\ 0.375 \\ 0 \\ 0.167 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0.083 \\ 0 \\ 0.177 \\ 0.292 \\ 0 \\ 0.094 \\ 0 \\ 0.219 \\ 0.135 \end{bmatrix}$$

This $\vec{x_0}$ represents a randomly determined initial condition, with a $\frac{1}{6}$ probability of the pawn starting on square 8, a $\frac{3}{8}$ probability of the pawn starting on square 6, a $\frac{1}{8}$ probability of starting on square 2, and a $\frac{1}{3}$ probability of starting off the board (square 0) as in the normal start of the game. Note that this $\vec{x_0}$ gives a possible starting initial position of the pawn as square 2, even though it is normally impossible for a pawn to start a turn on a square that has the bottom of a ladder or the top of a chute. This means that on the first turn, instead of moving spinning the spinner and moving normally, the pawn will just be moved to the top of the ladder (square 4) and stop.

Now that we have a solid understanding of how to construct and interpret Markov chains representing turns taken on a small example Chutes and Ladders board, we are ready to extend these concepts to the much larger board of classic Chutes and Ladders.

4 Extrapolating to the full board

Thus far, we haven't made any calculations based on the actual chutes and ladders board. The reason for this is the magnitude of the stochastic matrix. The original chutes and ladders board is 10x10, or 100 squares long. The stochastic matrix for such a board is a very sparse, 101x101 Matrix. Based on the size of such a matrix, any written calculations quickly become unwieldy. For that reason, we only include results in this section, with the understanding that all calculations are performed to the same specifications as the examples we have included above.

A useful thing to know about chutes and ladders is how long, roughly, a game can expect to last. Recall that a game is over when a player reaches the end square, or $square_n$ for an n-length track. Let us say that S represents the stochastic-matrix for the standard chutes and ladders board. We can generate the first entry of a Markov chain M by calculating $S\vec{x}_0 = \vec{x}_1$, where:

$$\vec{x}_0 = \begin{bmatrix} 0 \\ 1/6 \\ 1/6 \\ 1/6 \\ 1/6 \\ 1/6 \\ 1/6 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

And each row i of $\vec{x}_0$ represents the probability of the player moving to space i. The probability in rows for $1 <= i <= 6$ is a natural consequence of the spinner having 6 options, all equally probably.

We can now calculate $S\vec{x}_1 = \vec{x}_2$. If we recursively repeat this pattern n

times, we will have a n-length markoc chain. Let us write write the j_{th} element of the Markov chain as:

$$\vec{x}_j = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_{99} \\ x_{100} \\ x_{101} \end{bmatrix}$$

Recall that each row_i represents the chance to move to $square_i$, and thus row_{101} represents the chance to move to $square_{101}$, which is the end square, and therefor the chance of the game ending on turn j+1.

Let us calculate the chance of ending the game within the first couple of moves, which we have represented in this table:

Turn	Chance	Percent
1	0	0%
2	0	0%
3	0	0%
4	0	0%
5	49/23328	0.2%
6	4727/559872	0.8%
7	5269/279936	1%
8	1979365/60466176	3%
9	9188797/181398528	5%
10	156823933/2176782336	7%

In this table, **Turn** represents which element of the Markov chain we are looking at. **Chance** represents the value in the last row of the current element. **Percent** represents the Chance calculated as a percent. From the table, we can make the following claims:

- It is impossible to win the game before move 5
- The chance of winning within 10 moves is fairly small
- The chance of winning is growing, which suggests the game is winnable within a reasonable amount of moves

In fact, if we calculate a longer Markov chain, we find that the chance of winning after 50 iterations is 80%. This is consistent with the fact that Chutes and Ladders is a children game, and should be winnable by all players within a reasonable time.

5 Conclusions

In conclusion, Markov chains provide a very convenient method for reasoning about the Chutes and Ladders game. The probabilities of moving from each space to every other space is very naturally represented as a stochastic matrix. By mathematically defining a chutes and ladders track, we are able to make claims about the nature of the Chutes and Ladders rules themselves, without becoming distracted by the specifics of any given board. This allowed us to make claims and calculations about end-of-game rules, as well as statistics for how long games can be expected to last.

References

[1] Bernard Kolman, David R. Hill. *Elementary Linear Algebra with Applications, 9th addition*. Pearson Modern Classics, 2009

[2] *Chutes and Ladders Instructions*, Milton Bradley Company, East Longmeadow, MA, 1997. Accessed on: May. 2, 2019. [Online]. Available: https://www.hasbro.com/common/instruct/ChutesandLadders.PDF

[3] Gagniuc, Paul A. (2017). *Markov Chains: From Theory to Implementation and Experimentation*. USA, NJ: John Wiley Sons. pp. 1–235. ISBN 978-1-119-38755-8.

[4] *Regenerative Analysis and Steady State Distributions for Markov Chains* Winfried K. Grassmann, Michael I. Taksar, Daniel P. Heyman

[5] S. P. Meyn and R.L. Tweedie, 2005. *Markov Chains and Stochastic Stability Archived* 2013-09-03 at the Wayback Machine

Regenerative Analysis and Steady State Distributions for Markov Chains Winfried K. Grassmann, Michael I. Taksar, Daniel P. Heyman