A Game-Related Markov Chain

1. Terminology

The random variable X represents damage values that can be inflicted by an enemy or boss on a player character, and the random variable Y represents recovery values that the player character can heal themselves by. When $X = 0$ a dodge or miss takes place, and when $Y = 0$ the player character is stunned or unable to heal for that round. The player character is assumed to have H health points for presentation purposes. Mathematically, H can be replaced with 1 without losing any structure. We'll use H_i to represent health after i time units, and T to represent the amount of time units after which zero health is reached.

Let X and Y be two independent random variables both with finite non-negative probability distributions. Given a positive integer H, we define a sequence of random variables $H_1, H_2, H_3, ...$ where:

(1) $H_1 = H$,

(2) $H_{i+1} = 0$ if $H_i = 0$,

(3) $H_{2i} = \max\{0, H_{2i-1} - X\}$, and

(4) $H_{2i+1} = \min\{H, H_{2i} + Y\}$ if $H_{2i} > 0$.

We define a random variable $T_{X,Y,H}$ with $T_{X,Y,H} \equiv \frac{1}{2}\min_{i\in\mathbb{N}}\{i : H_i = 0\}$. If the meaning of $T_{X,Y,H}$ is clear in context, then we sometimes write T.

2. Markov Chains

The random variables H_i are all bounded below by 0, above by H, and since there is some smallest possible change δ in $\{y - x : x \in X, y \in Y\}$,it follows that $\{H_i\}_{i=1}^{\infty}$ has finite range. The state of H_i also depends only on the state of H_{i-1}, so $\{H_i\}_{i=1}^{\infty}$ is a markov chain. The expected value of T is the expected number of attacks it takes to reach a health value of zero from a health value of H. In terms of the markov chain, this means that we can calculate $E(T)$ by finding the mean time from $H_1 = H$ to the first value i where $H_i = 0$.

3. An Example

Let $X \sim P_X$ where:

$$P_X(X = x) = \begin{cases} 1/3 & x = 20 \\ 1/3 & x = 10 \\ 1/3 & x = 0 \end{cases}, \text{ and}$$

let $Y \sim P_Y$ where:

$$P_Y(Y = y) = \begin{cases} 2/3 & y = 30 \\ 1/3 & y = 0 \end{cases}.$$

Furthermore let $H = 30$. This information can be encoded as:

```
Dvals = [20, 10, 0]
Dprobs = [1/3, 1/3, 1/3]
Rvals = [30, 0]
Rprobs = [2/3, 1/3]
H = 30
```

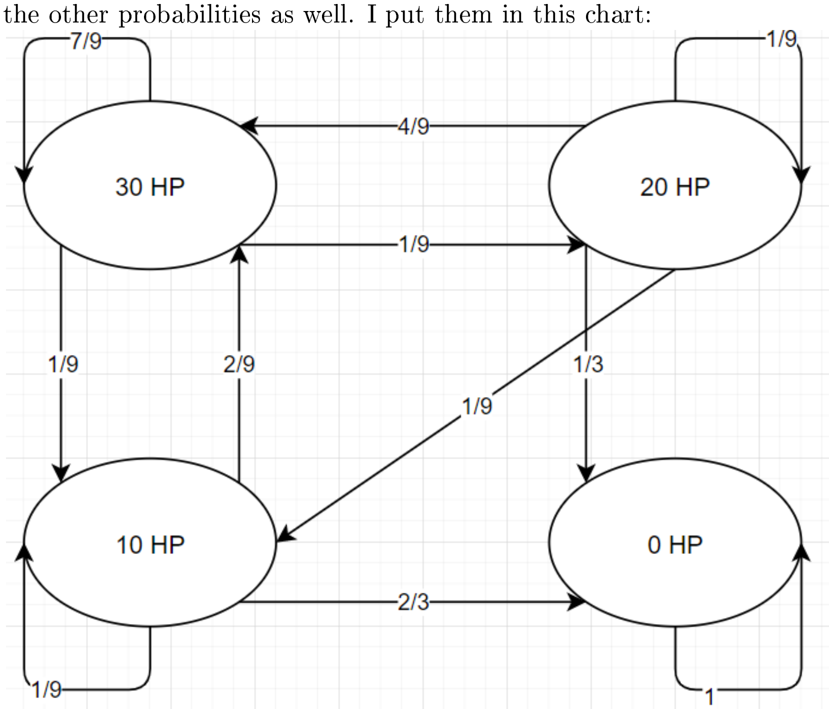
(See: MarkovCode.py from https://github.com/grassyseas/Markov-Chains)

In order to find all the possible states of our system, we need to find all the possible health values that can appear. In this case we start with 30 hp, can take 20, 10, or 0 damage, and can recover 30 or 0 hp. Our possible health values are: 30, 20, 10, 0.

Now that we know our possible states, we can create a diagram showing how they are related to one another. We can go from 30 hp to 30 hp either by taking 0 damage and healing any amount, or by taking 10 or 20 damage and healing by an amount 30. We have to add together the probabilities of each of these events to find the probability of going from 30 hp to 30 hp in a single round.

Event #	Event Description	Probability
Event 1	Take 0 dmg, heal any amount	$1/3 * 3/3$
Event 2	Take 10 dmg, heal 30 hp	$1/3 * 2/3$
Event 3	Take 20 dmg, heal 30 hp	$1/3 * 2/3$
Total	Go from 30 hp to 30 hp	$7/9$

This gives us the probability of going from 30 hp to 30 hp, so now we need all the other probabilities as well. I put them in this chart:



We can represent the transition matrix, T, as follows:

$$T = \begin{bmatrix} 7/9 & 1/9 & 1/9 & 0 \\ 4/9 & 1/9 & 1/9 & 1/3 \\ 2/9 & 0 & 1/9 & 2/3 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The powers of this information give us more information. For instance:

$$T^5 \approxeq \begin{bmatrix} 0.405 & 0.059 & 0.067 & 0.469 \\ 0.251 & 0.036 & 0.042 & 0.670 \\ 0.117 & 0.017 & 0.019 & 0.846 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The rows still add to 1, but now the entry in the i-th row and j-th column represent the probability that state i leads to state j on the fifth step (in correspondence with the fifth power of the matrix T).

To find the average number of steps it takes to go from 30 hp to 0 hp we want to solve a system of equations. Let m_{30}, m_{20}, m_{10}, and m_0 be the average number of steps it takes to reach 0 hp from 30 hp, 20 hp, 10 hp, and 0 hp, respectively. We can immediately see that $m_0 = 0$.

If we start at 30 hp then in one step we reach 20 hp with 1/9 probability, at which point logically it would take m_{20} steps on average to reach 0 hp. Similarly with 1/9 probability we'll reach 10 hp, at which point it will take m_{10} steps on average to reach 0 hp. We can also stay at 30 hp with 7/9 probability. Any of these three steps involves taking 1 step, so we now have an equation:

$m_{30} = 1 + 7/9m_{30} + 1/9m_{20} + 1/9m_{10}$

We can get the other equations similarly.

$m_{20} = 1 + 4/9m_{30} + 1/9m_{20} + 1/9m_{10}$

$m_{10} = 1 + 2/9m_{30} + 1/9m_{10}$

Notice that the coefficient of the j-th variable in the i-th equation is the (i, j)-th entry in the transition matrix T.

This is a system of three equations in three variables, which can be solved by hand:

$(m_{30}, m_{20}, m_{10}) = (243/26, 81/13, 45/13)$.

For larger transition matrices it helps to execute an algorithm on a computer which exactly solves or estimates the solutions.

4. Conclusion

We started with a model inspired by a video game scenario involving a player that loses and gains health in a known probabilistic manner. This might describe an action game in which occasionally a player takes a hit from a boss and then attempts to recover health with a potion. They might fail to recover and take a second hit in a row from the boss. In some cases they might dodge an attack from the boss and then recover. The number of rounds that the player experiences of this risk can be used to determine their probability of surviving. More complex models could be built that involve factors such as: the player also inflicting damage to the boss, a finite number of potions of varying potency, and the boss changing phases. In any case, the player behavior would need to be predictable enough to capture in a simple stochastic model. For an action game there would be major model assumptions involved, but they might still be useful in exploring the solution to a problem (equipment options vs. a boss) without being especially precise at predicting real gameplay outcomes. In an RPG game it'll often be possible to make less assumptions.