A3Q2: Infinite MDP

Bodhi Nguyen

January 21st, 2021

1 Problem

Consider an array of $n+1$ lilypads on a pond, numbered 0 to n. A frog sits on a lilypad other than the lilypads numbered 0 or n. When on lilypad i ($1 \leq i \leq n-1$), the frog can croak one of two sounds A or B. If it croaks A when on lilypad i ($1 \leq i \leq n-1$), it is thrown to lilypad $i-1$ with probability $\frac{i}{n}$ and is thrown to lilypad $i+1$ with probability $\frac{n-i}{n}$. If it croaks B when on lilypad i ($1 \leq i \leq n-1$), it is thrown to one of the lilypads $0, ..., i-1, i+1, ...n$ with uniform probability $\frac{1}{n}$.

A snake, perched on lilypad 0, will eat the frog if the frog lands on lilypad 0. The frog can escape the pond (and hence, escape the snake!) if it lands on lilypad n. What should the frog croak when on each of the lilypads $1, 2, ..., n-1$, in order to maximize the probability of escaping the pond (i.e., reaching lilypad n before reaching lilypad 0)? Although there are more than one ways of solving this problem, we'd like to solve it by modeling it as an MDP and identifying the Optimal Policy.

2 Setup as MDP

The state space consists of $n+1$ states numbered 0 to n. The action space consists of two actions, A and B. The transition function is described above, but mathematically as follows:

$$P(S_{t+1} = i+k | S_t = i, A_t = A) = \begin{cases} \frac{i}{n} \text{ if k = +1 or -1} \\ 0 \text{ otherwise} \end{cases} \tag{1}$$

$$P(S_{t+1} = i+k | S_t = i, A_t = B) = \frac{1}{n} \text{ for } k \in \{-i, -i+1, ... -1, +1, ..., n-i-1, n-i\} \tag{2}$$

Additionally, states 0 and n are terminal states.

The reward function is not specifically defined in the above function. We know landing on pad 0 would be bad and pad n would be good, so we just set the reward for landing on pad 0 to be -1.0, for landing on pad n to be 1.0, and for any intermediate pad the reward is 0.