Stanford CME 241 (Winter 2021) - Assignment 3

Frog on Lilypad (code part)

Chih-Hsuan 'Carolyn' Kao (chkao831@stanford.edu)

Feb 28th, 2021

In [1]:

```
import sys
sys.path.append('/Users/chih-hsuankao/Desktop/CME241/RL-book/')

from rl.distribution import Categorical, Constant
from rl.dynamic_programming import (
    evaluate_mrp_result,
    policy_iteration_result,
    value_iteration_result
)
from rl.markov_decision_process import (
    FiniteMarkovDecisionProcess,
    FinitePolicy,
    StateActionMapping,
)
from rl.markov_process import (
    Transition,
    RewardTransition,
    FiniteMarkovProcess,
    Optional,
    FiniteMarkovRewardProcess,
)
```

```
/Users/chih-hsuankao/.pyenv/versions/anaconda3-2019.03/lib/py
thon3.7/site-packages/scipy/__init__.py:137: UserWarning: Num
Py 1.16.5 or above is required for this version of SciPy (det
ected version 1.16.2)
  UserWarning)
```

In [2]:

```
from dataclasses import dataclass
import itertools
import matplotlib.pyplot as plt
from typing import Mapping, Dict, Tuple, List
```

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.

Write code to model this MDP as an instance of the FiniteMarkovDecisionProcess class. We have learnt that there exists an optimal deterministic policy, and there are $2^{n−1}$ possible deterministic policies for this problem. Write code to create each of these $2^{n−1}$ deterministic policies (as instances of FinitePolicy class), create a policy-implied Finite MRP for each of these deterministic policies (using the apply finite policy method of FiniteMarkovDecisionProcess class), and evaluate the Value Function for each of those implied Finite MRPs. This should gives you the Optimal Value Function and the Optimal Deterministic Policy.

In [3]:

```
@dataclass(frozen=True)
class FrogState:
    position: int
```

In [4]:

```
FrogJumpMap = StateActionMapping[FrogState, int]
```

In [5]:

```
class FrogMDP(FiniteMarkovDecisionProcess[FrogState, str]):

    def __init__(
        self,
        num_pad: int = 10,
    ):
        self.num_pad = num_pad

        super().__init__(self.get_action_transition_reward_map())

    def get_action_transition_reward_map(self) -> StateActionMapping[FrogS
tate, str]:

        d: Dict[FrogState, Dict[str, Categorical[Tuple[FrogState, float
]]]] = {}

        # ref: https://github.com/coverdrive/MDP-DP-RL/blob/master/src/exa
mples/exam_problems/frog_lilypad.py
        for i in range(1, self.num_pad):

            d1: Dict[str, Categorical[Tuple[FrogState, float]]] = {}

            # Croak A
            d1["A"] = Categorical({(FrogState(i - 1), 0.):
                                   i / self.num_pad,
                               (FrogState(i + 1), 1. if i == self.num_
pad-1 else 0.):
```

```
                                                (self.num_pad - i) /self.num_pad})
            # Croak B
            d1["B"] = Categorical({(FrogState(j), 1. if j == self.num_pad
else 0.):
                                                1/self.num_pad for j in range(self.
num_pad + 1) if j != i})

            d[FrogState(i)] = d1

        d[FrogState(self.num_pad)] = None
        d[FrogState(0)] = None

        return d

    def rewardf(
        self,
        current_pad: int,
        num_pad: int
    ):
        if current_pad == num_pad:
            return 1.

        elif current_pad == 0:
            return -1.

        else:
            return 0.
```

In [6]:

```
if __name__ == '__main__':

    gamma = 0.8
    pad = 10

    si_mdp: FiniteMarkovDecisionProcess[FrogState, int] =\
        FrogMDP(
            num_pad = pad
        )

    print("MDP Transition Map")
    print("------------------")
    print(si_mdp)

    policies = list(itertools.product([0, 1], repeat = pad - 1))
    #print(policies)

    # For each deterministic policy
    for policy in policies:
        # print("A Deterministic Policy:")
        fdp: FinitePolicy[FrogState, int] =\
            FinitePolicy(
                {FrogState(padnum):
                    Constant(policy[padnum - 1]) for padnum in range(1, pa
d)}
            )
        # commented out to avoid long output; uncomment the line below as
 needed
        # print(fdp)
```

```python
    print("Optimal Value Function and Optimal Policy")
    print("--------------------------------------------")

    opt_vf_vi, opt_policy_vi = value_iteration_result(si_mdp, gamma=gamma)
    print(opt_vf_vi)
    print(opt_policy_vi)
```

```
MDP Transition Map
------------------
From State FrogState(position=1):
  With Action A:
    To [State FrogState(position=0) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=2) and Reward 0.000] with Pr
obability 0.900
  With Action B:
    To [State FrogState(position=0) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=2) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=3) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=4) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=5) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=6) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=7) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=8) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=9) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=10) and Reward 1.000] with P
robability 0.100
From State FrogState(position=2):
  With Action A:
    To [State FrogState(position=1) and Reward 0.000] with Pr
obability 0.200
    To [State FrogState(position=3) and Reward 0.000] with Pr
obability 0.800
  With Action B:
    To [State FrogState(position=0) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=1) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=3) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=4) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=5) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=6) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=7) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=8) and Reward 0.000] with Pr
```

```
obability 0.100
    To [State FrogState(position=9) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=10) and Reward 1.000] with P
robability 0.100
From State FrogState(position=3):
  With Action A:
    To [State FrogState(position=2) and Reward 0.000] with Pr
obability 0.300
    To [State FrogState(position=4) and Reward 0.000] with Pr
obability 0.700
  With Action B:
    To [State FrogState(position=0) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=1) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=2) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=4) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=5) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=6) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=7) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=8) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=9) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=10) and Reward 1.000] with P
robability 0.100
From State FrogState(position=4):
  With Action A:
    To [State FrogState(position=3) and Reward 0.000] with Pr
obability 0.400
    To [State FrogState(position=5) and Reward 0.000] with Pr
obability 0.600
  With Action B:
    To [State FrogState(position=0) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=1) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=2) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=3) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=5) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=6) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=7) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=8) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=9) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=10) and Reward 1.000] with P
robability 0.100
From State FrogState(position=5):
```

```
  With Action A:
    To [State FrogState(position=4) and Reward 0.000] with Pr
obability 0.500
    To [State FrogState(position=6) and Reward 0.000] with Pr
obability 0.500
  With Action B:
    To [State FrogState(position=0) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=1) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=2) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=3) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=4) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=6) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=7) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=8) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=9) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=10) and Reward 1.000] with P
robability 0.100
From State FrogState(position=6):
  With Action A:
    To [State FrogState(position=5) and Reward 0.000] with Pr
obability 0.600
    To [State FrogState(position=7) and Reward 0.000] with Pr
obability 0.400
  With Action B:
    To [State FrogState(position=0) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=1) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=2) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=3) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=4) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=5) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=7) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=8) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=9) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=10) and Reward 1.000] with P
robability 0.100
From State FrogState(position=7):
  With Action A:
    To [State FrogState(position=6) and Reward 0.000] with Pr
obability 0.700
    To [State FrogState(position=8) and Reward 0.000] with Pr
obability 0.300
  With Action B:
    To [State FrogState(position=0) and Reward 0.000] with Pr
```

```
    To [State FrogState(position=0) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=1) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=2) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=3) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=4) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=5) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=6) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=8) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=9) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=10) and Reward 1.000] with P
robability 0.100
From State FrogState(position=8):
  With Action A:
    To [State FrogState(position=7) and Reward 0.000] with Pr
obability 0.800
    To [State FrogState(position=9) and Reward 0.000] with Pr
obability 0.200
  With Action B:
    To [State FrogState(position=0) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=1) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=2) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=3) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=4) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=5) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=6) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=7) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=9) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=10) and Reward 1.000] with P
robability 0.100
From State FrogState(position=9):
  With Action A:
    To [State FrogState(position=8) and Reward 0.000] with Pr
obability 0.900
    To [State FrogState(position=10) and Reward 1.000] with P
robability 0.100
  With Action B:
    To [State FrogState(position=0) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=1) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=2) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=3) and Reward 0.000] with Pr
```

```
    To [State FrogState(position=3) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=4) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=5) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=6) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=7) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=8) and Reward 0.000] with Pr
obability 0.100
    To [State FrogState(position=10) and Reward 1.000] with P
robability 0.100
FrogState(position=10) is a Terminal State
FrogState(position=0) is a Terminal State

Optimal Value Function and Optimal Policy
-----------------------------------------
{FrogState(position=1): 0.2824061058293976, FrogState(positio
n=2): 0.2824061058293976, FrogState(position=3): 0.2824061058
293976, FrogState(position=4): 0.2824061058293976, FrogState
(position=5): 0.2824061058293976, FrogState(position=6): 0.28
24061058293976, FrogState(position=7): 0.2824061058293976, Fr
ogState(position=8): 0.2824061058293976, FrogState(position=
9): 0.30332433866457054}
For State FrogState(position=1):
  Do Action B with Probability 1.000
For State FrogState(position=2):
  Do Action B with Probability 1.000
For State FrogState(position=3):
  Do Action B with Probability 1.000
For State FrogState(position=4):
  Do Action B with Probability 1.000
For State FrogState(position=5):
  Do Action B with Probability 1.000
For State FrogState(position=6):
  Do Action B with Probability 1.000
For State FrogState(position=7):
  Do Action B with Probability 1.000
For State FrogState(position=8):
  Do Action B with Probability 1.000
For State FrogState(position=9):
  Do Action A with Probability 1.000
```