Average git users

Week 12

Objective: Understand the process of sequential decision making (stochastic environment) and the connection with reinforcement learning

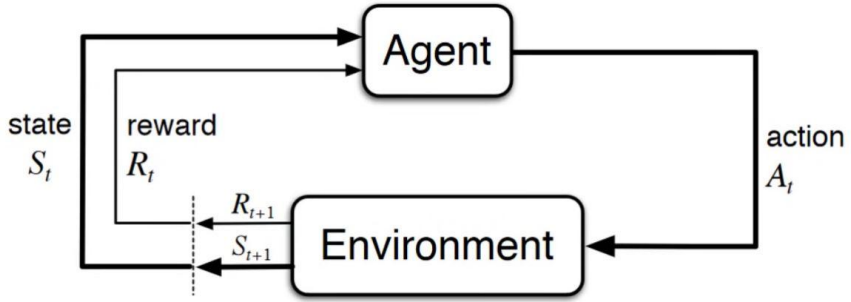
Markov Decision Process and Dynamic Programming

Markov Decision Process:
A Markov decision process (MDP) is defined as a stochastic decision-making process that uses a mathematical framework to model the decision-making of a dynamic system in scenarios where the results are either random or controlled by a decision maker, which makes sequential decisions over time.

Dynamic Programming:
The term dynamic programming (DP) refers to a collection of algorithms that can be used to compute optimal policies given a perfect model of the environment as a Markov decision process (MDP).

Markov Decision Process

- States: The state at any time is represented by the tuple (x, y), where x is the number of bikes at the first location, and y is the number of bikes at the second location.
- Actions: The action at any state is represented by the integer a, where a is the net number of bikes moved from the first location to the second location. If a is negative, it means that bikes are being moved from the second location to the first location. If a is positive, it means that bikes are being moved from the first location to the second location.
- Rewards: The reward for each action is calculated as follows:
    - If a is positive, the reward is 10 times the minimum of the number of bikes available at the first location and the absolute value of a, minus the cost of moving the bikes (2 times the absolute value of a).
    - If a is negative, the reward is 10 times the minimum of the number of bikes available at the second location and the absolute value of a, minus the cost of moving the bikes (2 times the absolute value of a).
    - If a is zero, the reward is 0.



Bellman equation and Poisson distribution

$$V(x) = \max_{a \in \Gamma(x)} \{F(x,a) + \beta V(T(x,a))\}.$$

$$P(X = k) = \frac{e^{-\lambda}\lambda^k}{k!}$$

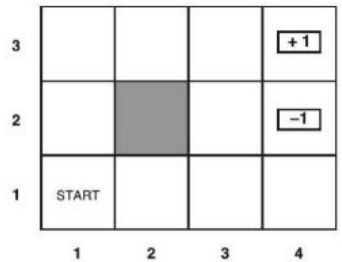
stochastic environment

stochastic environment is one in which the next state and reward are not solely determined by the current state and action taken by the agent. Instead, there is some level of randomness or uncertainty involved in the dynamics of the environment.

In order to deal with stochastic environments, reinforcement learning algorithms need to be able to handle uncertainty and randomness in the state transitions and rewards. This can be done by using probabilistic models to estimate the probability distributions over possible outcomes of state transitions and rewards. The agent can then use these distributions to make decisions that maximize expected rewards in the face of uncertainty.

Problem Statement:

(1) Suppose that an agent is situated in the 4x3 environment as shown in Figure 1. Beginning in the start state, it must choose an action at each time step. The interaction with the environment terminates when the agent reaches one of the goal states, marked +1 or -1. We assume that the environment is fully observable, so that the agent always knows where it is. You may decide to take the following four actions in every state: Up, Down, Left and Right. However, the environment is stochastic, that means the action that you take may not lead you to the desired state. Each action achieves the intended effect with probability 0.8, but the rest of the time, the action moves the agent at right angles to the intended direction with equal probabilities. Furthermore, if the agent bumps into a wall, it stays in the same square. The immediate reward for moving to any state (s) except for the terminal states S^+ is r(s)= -0.04. And the reward for moving to terminal states is +1 and -1 respectively. Find the value function corresponding to the optimal policy using value iteration.



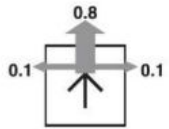
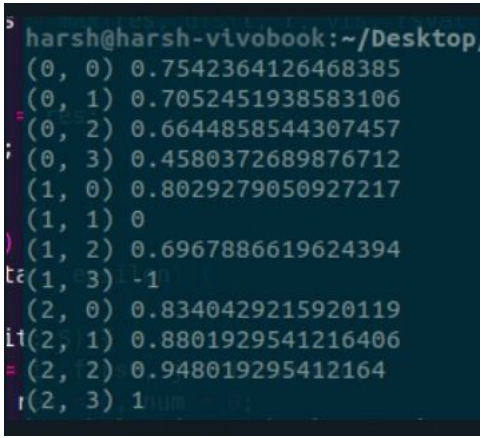


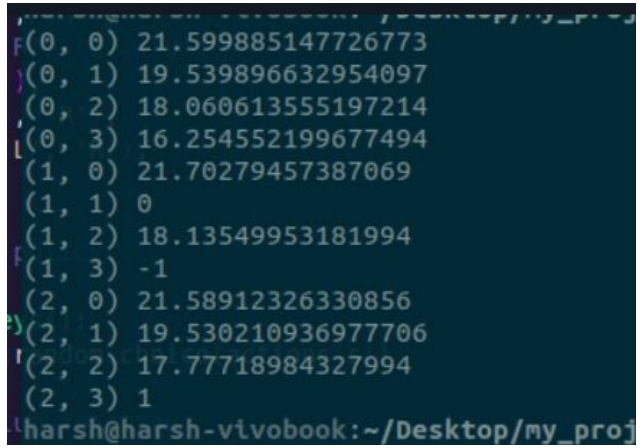
Figure 1: 4×3 grid world with uncertain state change.

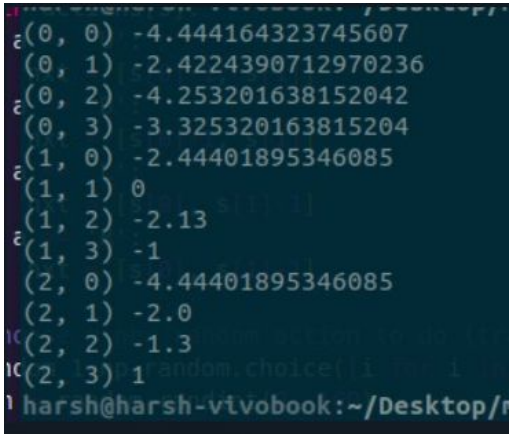
Find the value functions corresponding optimal policy for the following:

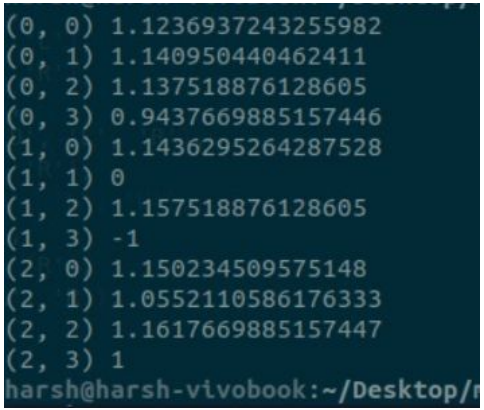
(a) r(s)=-2
(b) r(s)=0.1
(c) r(s)=0.02
(d) r(s)=1

Outputs


harsh@harsh-vivobook:~/Desktop
(0, 0) 0.7542364126468385
(0, 1) 0.7052451938583106
(0, 2) 0.6644858544307457
(0, 3) 0.4580372689876712
(1, 0) 0.8029279050927217
(1, 1) 0
(1, 2) 0.6967886619624394
(1, 3) -1
(2, 0) 0.8340429215920119
(2, 1) 0.8801929541216406
(2, 2) 0.948019295412164
(2, 3) 1


(0, 0) 21.599885147726773
(0, 1) 19.539896632954097
(0, 2) 18.060613555197214
(0, 3) 16.254552199677494
(1, 0) 21.70279457387069
(1, 1) 0
(1, 2) 18.13549953181994
(1, 3) -1
(2, 0) 21.58912326330856
(2, 1) 19.530210936977706
(2, 2) 17.77718984327994
(2, 3) 1
harsh@harsh-vivobook:~/Desktop/my_proj


(0, 0) -4.444164323745607
(0, 1) -2.4224390712970236
(0, 2) -4.253201638152042
(0, 3) -3.325320163815204
(1, 0) -2.44401895346085
(1, 1) 0
(1, 2) -2.13
(1, 3) -1
(2, 0) -4.44401895346085
(2, 1) -2.0
(2, 2) -1.3
(2, 3) 1
harsh@harsh-vivobook:~/Desktop/


(0, 0) 1.1236937243255982
(0, 1) 1.140950440462411
(0, 2) 1.137518876128605
(0, 3) 0.9437669885157446
(1, 0) 1.1436295264287528
(1, 1) 0
(1, 2) 1.157518876128605
(1, 3) -1
(2, 0) 1.150234509575148
(2, 1) 1.0552110586176333
(2, 2) 1.1617669885157447
(2, 3) 1
harsh@harsh-vivobook:~/Desktop/

(2) [Gbike bicycle rental] You are managing two locations for Gbike. Each day, some number of customers arrive at each location to rent bicycles. If you have a bike available, you rent it out and earn INR 10 from Gbike. If you are out of bikes at that location, then the business is lost. Bikes become available for renting the day after they are returned. To help ensure that bicycles are available where they are needed, you can move them between the two locations overnight, at a cost of INR 2 per bike moved.

Assumptions: Assume that the number of bikes requested and returned at each location are Poisson random variables. Expected numbers of rental requests are 3 and 4 and returns are 3 and 2 at the first and second locations respectively. No more than 20 bikes can be parked at either of the locations. You may move a maximum of 5 bikes from one location to the other in one night. Consider the discount rate to be 0.9.

Formulate the continuing finite MDP, where time steps are days, the state is the number of bikes at each location at the end of the day, and the actions are the net number of bikes moved between the two locations overnight.

Download and extract files from gbike.zip. Try to compare your formulation with the code. Before proceeding further, ensure that you understand the policy iteration clearly.

OUTPUTS

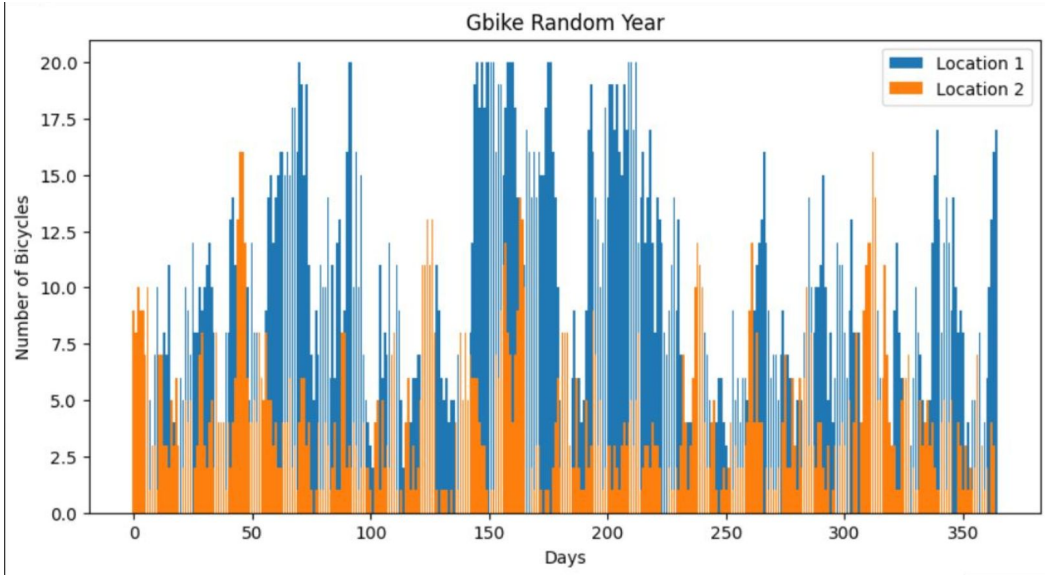
```
2 13 7.794445387892944
2 14 7.794445387892944
2 15 7.794445387892944
2 16 7.794445387892944
2 17 7.794445387892944
2 18 7.794445387892944
2 19 7.794445387892944
2 20 7.794445387892944
3 0 3.496517644970252
3 1 5.227137457824362
3 2 6.75008289313598
3 3 7.8576795733626135
3 4 8.41147791347593
3 5 8.41147791347593
3 6 8.41147791347593
3 7 8.41147791347593
3 8 8.41147791347593
3 9 8.41147791347593
```

```
8.41147791347593
```

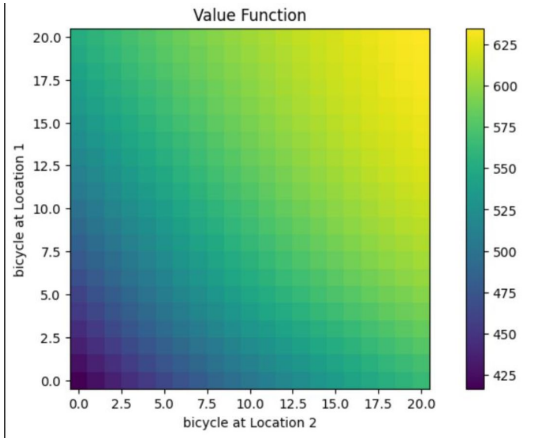
(3) Write a program for policy iteration and resolve the Gbike bicycle rental problem with the following changes. One of your employees at the first location rides a bus home each night and lives near the second location. She is happy to shuttle one bike to the second location for free. Each additional bike still costs INR 2, as do all bikes moved in the other direction. In addition, you have limited parking space at each location. If more than 10 bikes are kept overnight at a location (after any moving of cars), then an additional cost of INR 4 must be incurred to use a second parking lot (independent of how many cars are kept there).

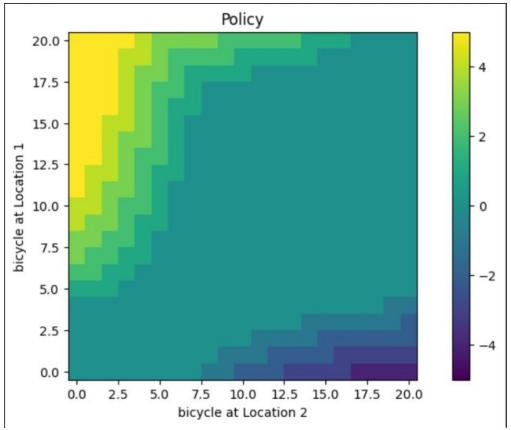
OUTPUTS

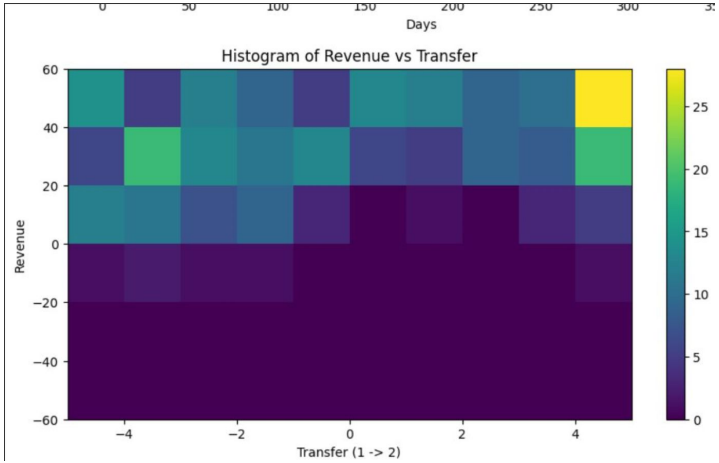
https://colab.research.google.com/drive/1z2IuIq48ANsRWkkqJcmI8ULKjwMK054o#scrollTo=6Jbtn1rgNz3-

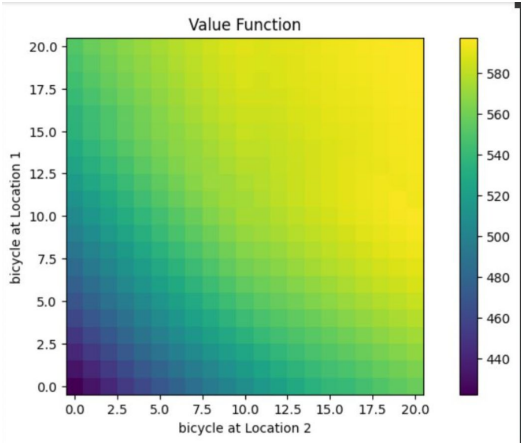


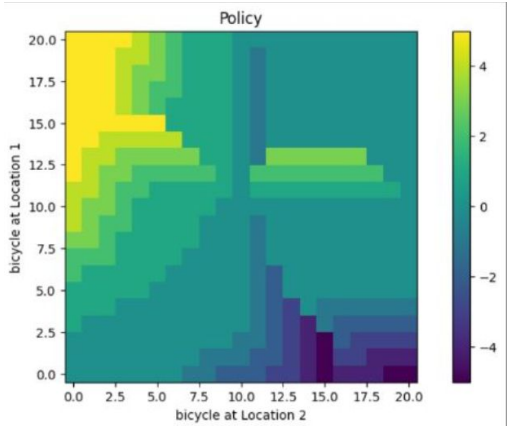
OUTPUTS


Value Function
bicycle at Location 1
bicycle at Location 2


Policy
bicycle at Location 1
bicycle at Location 2


Days
Histogram of Revenue vs Transfer
Revenue
Transfer (1 -> 2)


Value Function
bicycle at Location 1
bicycle at Location 2


Policy
bicycle at Location 1
bicycle at Location 2