MDP and Reinforcemennt learning : exercises

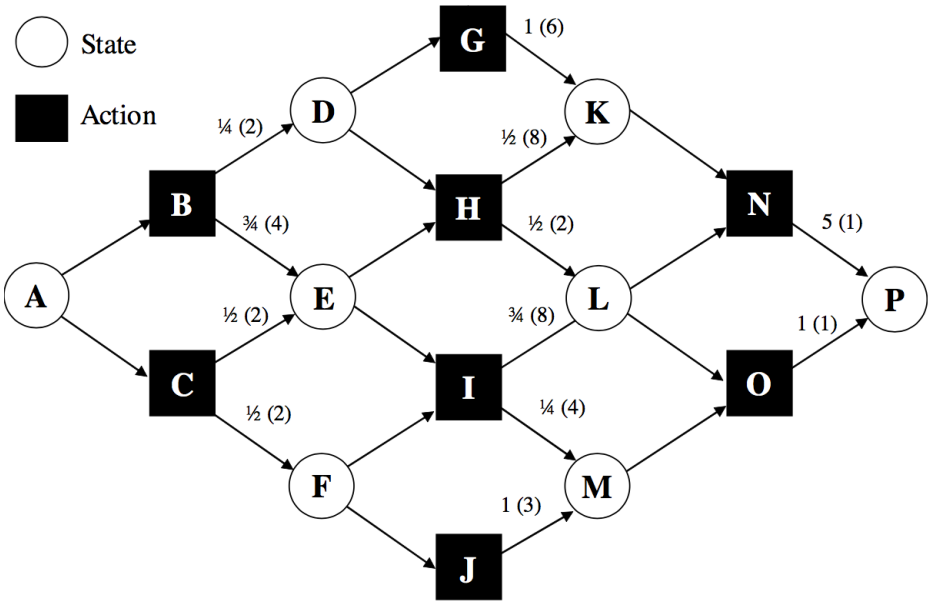
Nicolas Gast

Master MOSIG, 2023 – Course Mathematical Fundations of Machine Learning

1 Warm-up : Stochastic shortest path

The following figure represents a graph. Circle nodes correspond to decision nodes and square nodes to chance nodes. In the MDP language, a circle node is a state and square nodes corresponds to a state-action pair.

Decisions are made at circle nodes and work as follows: if you select action B when you are in state A, then you will go to state D with probability 1/4 (and distance 2) and to state E with probability 3/4 (and distance 4).



Find and evaluate the (deterministic) policy that minimizes the expected distance from A to P.

2 Example of a discounted MDP

Consider a MDP with state space $\{0, 1 \dots 10\}$ and action space $\{A, B\}$. In any given state s:

- Action A makes you go to 0 (with probability 1) and earns you s.
- Action B earns you 0 and makes you go to $s + 1 \mod 10$ with probability p or to 0 with probability $1 - p$.

You want to maximize your discounted reward for a discounted factor γ.

- What is the value function of the policy that takes action B for all states $s \geq 2$ and action A otherwise.
- Assume that $p = 0.8$ and $\gamma = 0$: what is the value function and the optimal policy?
- Assume that $p = 0.8$ and $\gamma = 0.9$: what is the value function and the optimal policy?

3 A game of dice: "stop or continue"

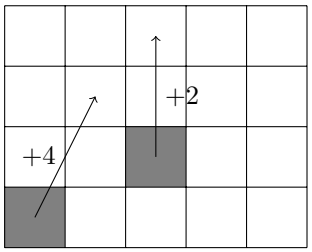
Consider an unbiased d face dice with faces numbered from 1 to d. You can throw the dice up to d times (each time with a random throw). You gain money as follows:

- After any throw, you can stop and earn the sum of the values that you obtained.
- If, after a throw, you obtain a value that you have already observed, you earn nothing.

For instance, if your sequence of throws is "1, 3", you can stop and earn 4 or throw again. If you throw again: if you obtain a "2", you can stop and earn 6 or throw again, but if you obtain "3", you have to stop and earn nothing.

1. Consider $d = 3$. Formulate the problem as a MDP, compute the optimal policy.
2. What is the difficulty for solving the problem for larger values of d?

4 Optimal grid movement and Q-learning



Consider a robot moving in the grid represented on the left.

- If you are in one of the white cases, you can move in either of the four directions. This costs 0. Actions that would push you out of the grid leave your position unchanged but costs -1.
- If you are in a gray case, there is no actions to take and you move directly to the cases indicated by the arrow. The reward that you obtain is the one written on the arrow.

You objective is to maximize the total reward with discount factor γ.

1. Compute the value of the RANDOM policy, that move randomly in any of the direction (with probability 1/4).
2. Write a program that uses **value iteration** and computes an optimal policy (consider $\gamma = 0.9$).
3. (*) Write a program that uses the **Q-learning** update to compute the optimal policy. How many iteration does it take?

5 Optimal Gambling

You enter a casino with a quantity of money $S_0 \in \{1, \dots, K-1\}$. Each time, you can gamble a quantity smaller or equal to S_t. If you bet u:

- with probability $p \in (0, 1)$, you win u : $S_{t+1} = S_t + u$
- with probability $1 - p$, you loose u: $S_{t+1} = S_t - u$.

You goal is to maximize the probability of attaining K knowing that you can play at much T times. How do you play?

1. Formulate the problem has an MDP
2. Write down Bellman's equation
3. We assume that $p = 1/2$, $K = 8$ and $T = 10$. What is your strategy?
4. Restart the question with $p = 1/3$ and $p = 2/3$.

   Suppose now that $T = +\infty$.
5. Can you describe the optimal strategy as a function of p?

6 An abstract MDP

Consider a Markov decision processes with two states : $\{0, 1\}$ and two actions : $\{A, B\}$. The transitions and rewards are as follows:

$$\forall i, j : p(i|j, A) = 0.5 \text{ and } r(i, A) = i$$
$$p(0|0, B) = 0.8 \text{ and } p(0, |1, B) = 0.4 \text{ and } r(i, A) = 1 - i.$$

1. You are allowed to make 5 steps and you want to maximize the expected total reward over this 5 steps. What is the optimal strategy? What is its value?
2. You now consider a discount factor $\gamma > 0$ and you want to maximize the discounted reward over an infinite time horizon.
   (a) Let $\gamma = 0$. What is the optimal strategy? What is its value function?
   (b) Consider $\gamma = 0.9$. What is the optimal strategy? What is its value function?

7 Inventory control

In this exercise, we present a problem of inventory control. You own a warehouse that sells only one type of product. This warehouse can store up to M items and S_t denotes the stock level (*i.e.* the number of items that the warehouse has at time t).

The system evolves as follows :

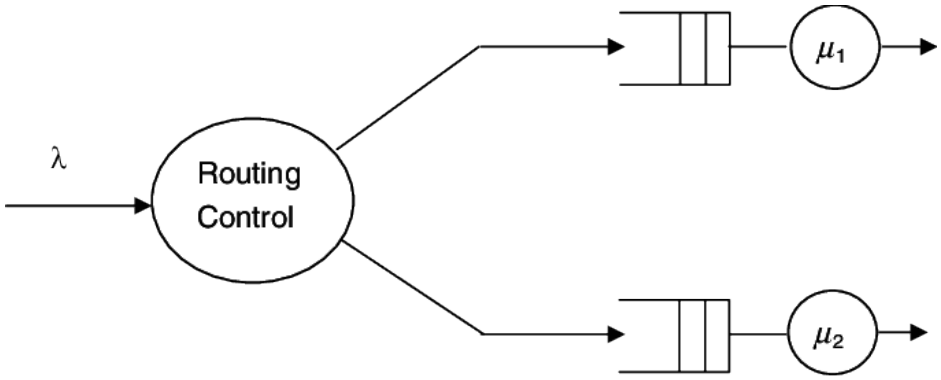
- D_t is the demand during period t. We assume that the D_ts are *i.i.d* random variables. A demand that is not immediately satisfied is lost. We denote by $p_i := \mathbb{P}(D_t = i)$ the probability that the depend is i.
- O_t is the number of order.
- The sequence of event is : at the beginning of the period t, S_t is observed, then a quantity O_t is ordered that is immediately delivered. Finally, the demand D_t is satisfied with the available stock $S_t + O_t$.

The cost is:

- h : cost of holding one unit of inventory from period t to period $t + 1$.
- $cq + k\mathbf{1}_{k>0}$: cost of order q unit (equals $cq + k$ is $q \geq 1$, 0 otherwise).
- s : price of each sold object.

1. Express S_{t+1} as a function of S_t, O_t and D_t.
2. Formulate the problem as a Markov decision process, by detailing the state space, the action space, the costs and the transition probabilities.
3. We consider a discounted problem.
   - Give the optimally equations.
   - (using a computer) : Program value iteration to compute an optimal policy for the following instance:
     - $M = 50$, $h = 0.1$, $s = 5$, $k = 2$, $c = 1$, $p_i = 2^{-(i+1)}$. And the discount factor is $\gamma = 0.95$.
   - (on a computer) : Program policy iteration to compute an optimal policy for the same values. How many iterations did the algorithm take?
4. Redo the same question with the average cost.

8 (*) Optimal Routing



We consider a discrete-time queuing network that works as follows : Jobs arrive in the system and are routed to one of the two queues (according to a policy that will be specified later).

At each time step, one of the three following events occurs :

- With probability $\lambda/(\lambda + \mu_1 + \mu_2)$ a new job arrive in the system. It is allocated to one of the two servers.
- With probability $\mu_i/(\lambda + \mu_1 + \mu_2)$ a job is served by the server $i \in \{1, 2\}$ if this server has a job in its queue (otherwise nothing happens)

The instantaneous cost of holding a job in the any of the two queues is 1. Your goal is to implement an algorithm that finds the policy that minimizes the expected discounted cost $\gamma = 0.99$.

1. Model the problem as a MDP (*e.g.*, give the state space and action space).
2. Implement an algorithm that uses value iteration and computes an optimal policy. To ensure a finite state-space, we can assume that a queue cannot contain more than 50 jobs.
3. We assume that $\mu_1 = 1$ and $\mu_2 = 2$. Compare the optimal policies for $\lambda = 0.1$ and $\lambda = 2.5$. Are they the same?