Ma 661 Dynamic Programming and Reinforcement Learning

Darinka Dentcheva darinka.dentcheva@stevens.edu

Homework 4
Solutions

Problem 1. Consider the equipment replacement problem of Assignment 2. Assume that we would like to identify the optimal replacement policy by solving an infinite-horizon discounted total reward problem.

(1.1) Formulate the infinite-horizon Markov decision problem.

Solution:

- State space: $\mathcal{X} = \{0, 1, 2, \dots\}$ - state of the equipment - $x = 0$ - new
- Control space: $\mathcal{U} = \{0, 1\}$ - 0 - wait, 1 - replace.
- Reward function for one period: $r(x,u) = \begin{cases} R - K(1-\gamma e^{-\mu x}) - c_0 & u = 1 \\ R - (c_0 + c_1 x) & u = 0 \end{cases}$.
- Dynamic programming equation:

$$v^*(x) = \max\Big\{ r(x,1) + \alpha \sum_{j=0}^{\infty} p_j v^*(j); \;\; r(x,0) + \alpha \sum_{j=0}^{\infty} p_j v^*(x+j) \Big\}$$

(1.2) If there is no salvage value, then show that the optimal value function is non-increasing function of the state.

Solution: We can show that the value function is nonincreasing by using the value iteration method.

We can define $v^1(x) \equiv 0$ or $v^1(x) = r(x, 0)$ for all $x \in \mathcal{X}$. In both cases $v^1(\cdot)$ is nonincreasing.

Assume that $v^k(\cdot)$ is nonincreasing at some iteration k. Then according to the value-iteration procedure, we have

$$v^{k+1}(x) = \max\Big\{ r(x,1) + \alpha \sum_{j=0}^{\infty} p_j v^k(j); \;\; r(x,0) + \alpha \sum_{j=0}^{\infty} p_j v^k(x+j) \Big\}$$

The functions $v^k(\cdot)$ and $v^k(\cdot + j)$ are decreasing, by induction assumption.

Denoting

$$h(x) = r(x,1) + \sum_{j=0}^{\infty} p_j v^k(j),$$

$$g(x) = r(x,0) + \sum_{j=0}^{\infty} p_j v^k(x+j),$$

we conclude that $h(\cdot)$ and $g(\cdot)$ are nonincreasing because they are sums of nonincreasing functions. Therefore, their maximum is also a nonincreasing function. Since v^k converges to v^* when $k \to \infty$ and weak inequalities get preserved in the limit, we conclude that $v^*(\cdot)$ is nonincreasing.

(1.3) Solve the infinite horizon problem (with salvage value present) for the following values of the parameters: $c_0 = 1$, $c_1 = 1$, $R = 5$, $K = 10$, $\gamma = 0.8$, $\mu = 0.2$, $\lambda = 1$ and discount factor $\alpha = 0.9$. Solve the problem in all three ways: value iteration method, policy iteration method and linear programming.

Solution: The optimal policy in this infinite-horizon problem is the same as in the finite-horizon for period $t = 1$.

Problem 2. We consider an inventory model as discussed in class. The stock at the beginning of period t denoted by x_t, orders at the beginning of period t by u_t, and random demand in period t (observed only *after* the orders are placed) by d_t. We assume ordering cost 5, selling price 10 and holding cost 2. The demands in successive periods are i.i.d. with values $(0, 1, 2, 3, 4)$ whose respective probabilities are $0.1, 0.2, 0.3, 0.2, 0.2$. The capacity of the inventory is 12.

(2.1) Formulate an infinite horizon problem with discount factor 0.8 to determine the best re-order policy.

Solution:

- State space: $\mathcal{X} = \{0, 1, 2, \dots, 12\}$ - items on stock.
- Control space: $\mathcal{U} = \{0, 1, 2, \dots, 12\}$ - number of items to order.
- Feasible controls: $U(x) = \{0, 1, \dots, 12 - x\}$, $x \in \mathcal{X}$.
- Expected reward function for one period:

$$r(x, u) = \mathbb{E}\big[10 \min(d, x + u) - 5u - h(x + u)\big].$$

  Here d is the random variable representing the demand.
- Transition kernel:

$$p(y|x, u) = \begin{cases} 0.1 & \text{for } y = x + u, \\ 0.3 & \text{for } y = \max(0, x + u - 2) \\ 0.2 & \text{for } y = \max(0, x + u - i),\ i = 1, 3, 4, \\ 0 & \text{otherwise.} \end{cases}$$

- Dynamic programming equation:

$$v^*(x) = \max_{u \in U(x)} \Big\{ -5u - h(x + u) + \sum_{j=1}^{12} p(y|x, u)\big(10 \min(d, x + u) + \alpha v^*(y)\big) \Big\}$$

(2.2) Solve the problem in (2.1) by value and policy iteration methods.

Solution:

With numerical accuracy of 10^{-10}, the value iteration method converges at in 116 iterations. Policy iteration method converges stops after two iterations.

```
SOLUTION
  Columns 1 through 11

   18.0000    23.0000    28.0000    32.3478    35.2779    36.4869    36.9133    36.3947    34.9616    32.6876    29.7284

  Columns 12 through 13

   26.1071    21.8868

     2     1     0     0     0     0     0     0     0     0     0     0     0
```

The optimal decision is to order 2 at state 0, order 1 at state 1, and not to order at other states.

Problem 3. Fisher boat is sent to the waters of three connected lakes during one fishing season. Let x_i $i = 1, 2, 3$ be the (estimated) current amounts of fish in lake i. If we fish in lake i, then we harvest $r_i x_i$ fish, provided the fishing conditions are good. The weather may change abruptly with probability p so that we end the fishing season. We assume that $0 < r_i < 1$ for all $i = 1, 2, 3$. Identify the lake-selection policy that maximizes the amount of fish before the end of the season.

Solution:

- State: the amount of remaining fish in the lakes $(x_1, x_2, x_3) \in \mathbb{R}^3_+$.
- Control set: $\mathcal{U} = \{1, 2, 3\}$ on which lake to fish.
- Transition Kernel:

$$P\Big[(x_1', x_2', x_3') = (x_1(1-r_1), x_2, x_3) \mid (x_1, x_2, x_3), u = 1\Big] = 1 - p,$$
$$P\Big[(x_1', x_2', x_3') = (x_1, x_2(1-r_2), x_3) \mid (x_1, x_2, x_3), u = 2\Big] = 1 - p,$$
$$P\Big[(x_1', x_2', x_3') = (x_1, x_2, x_3(1-r_2)) \mid (x_1, x_2, x_3), u = 3\Big] = 1 - p,$$
$$P\Big[(x_1', x_2', x_3') = (0, 0, 0) \mid (x_1, x_2, x_3), u\Big] = p.$$

  As we can only fish a portion of the avalable fish, the amounts of remaining fish are never 0. Here $(x_1, x_2, x_3) = 0$ only indicates the state when the weather has become adverse.
- Dynamic Programming Equation:

$$v^*(x_1, x_2, x_3) = \max\left\{(1-p) r_i x_i + (1-p) v^*(x_i(1-r_i), x^{-i})\right\}.$$

  The constant $1 - p$ can be interpreted as a discount factor, and the whole problem is a 3-armed bandit problem with $M = 0$.

- We observe that the problem belongs to the class of "deteriorating" models, that is, the index of the state of a project decreases, after we act on this project. Indeed, when $M = m_i(x_i)$, acting on project i (which is equally good as retiring) changes its state to $y_i = (1-r_i)x_i < x_i$. At this state, retiring is at least as good as continuation, because $v_i^*((1-r_i)x_i; M) \leq v_i^*(x_i; M)$. Thus, $m_i(y_i) \leq m_i(x_i)$.

  We apply the result of class with retirement reward $M = 0$. The optimal policy is: *whenever the weather permits, fish at lake i with highest $r_i x_i$.*