Darinka Dentcheva darinka.dentcheva@stevens.edu

Homework 7
Solutions

A machine may be in two states: good or bad. It produces an item at the end of each time period. If the machine is bad, the item is bad/defective as well. If the machine is good, then the item is good. A machine, which is good at stage t may become bad at stage $t+1$ with probability p. A bad machine remains bad, unless replaced. The state of the machine is not visible and can be identified only by inspecting the produced items. An item produced in period t may be inspected immediately at cost I. The inspection is perfect, that is, it distinguishes between good and defective items without mistakes. If the inspection finds the item bad, the machine may be replaced at a cost R. The cost of producing a bad item is C.

(a) Formulate a finite horizon dynamic programming problem to minimize the cost of operating the machine.

- State: Unobservable state of the machine is $\mathcal{Y} = \{G, B\}$. The belief state is a binary probability distribution $\xi = (\alpha, 1-\alpha)$, where α denotes the probability that the machine is good.
- Control set: $\mathcal{U} = \{0, 1\}$ do not inspect the produced item $u = 0$ or inspect the item $u = 1$ and replace the machine if it is brocken.
- Feasible control mapping: $U(x) = \{0, 1\}$ for all states.
- Transitions:
$$\alpha_{t+1} = \begin{cases} (1-p) * \alpha_t & \text{for } u = 0; \\ 1-p & \text{for } u = 1; \end{cases}$$
- Expected cost function
$$c(\alpha_t, u) = (1-\alpha_t)C + u(I + (1-\alpha_t)R).$$
- Dynamic Programming Equation:
$$\begin{aligned} v_t(\alpha) &= \min\left\{(1-\alpha)C + v_{t+1}((1-p)\alpha),\ I + \alpha(C+R) + v_{t+1}(1-p)\right\} \\ &\qquad t = T, \dots 1, \quad \alpha \in [0,1]. \\ v_{T+1}(\alpha) &= 0. \end{aligned}$$

(b) The initial state of the machine is good. Solve the problem for $p = 0.2$, $I = 1$, $R = 3$, $C = 2$, and $T = 18$.

We notice that α_t can only take values $(1-p)^k$ with $k = 1, 2, \dots t-1$. We simplify the dynamic programming equation further by identifying the belief state with k meaning that

$\alpha_t = (1-p)^k$. We obtain new simply dynamic programming equations:

$$
\begin{aligned}
v_t(k) &= \min\left\{(1-(1-p)^k)C + v_{t+1}(k+1),\ I + (1-p)^k(C+R) + v_{t+1}(1)\right\} \\
&\qquad t = 18, \dots 1, \quad k = 1, \dots t-1. \\
v_{19}(k) &= 0 \quad k = 1, \dots, 17.
\end{aligned}
$$

The optimal value at time 1 when the machine is good is 22.75315199999999.
The optimal policy is
('timestep:00 - ', [0])
('timestep:01 - ', [0])
('timestep:02 - ', [0, 0])
('timestep:03 - ', [0, 0, 0])
('timestep:04 - ', [0, 0, 0, 1])
('timestep:05 - ', [0, 0, 0, 1, 1])
('timestep:06 - ', [0, 0, 0, 1, 1, 1])
('timestep:07 - ', [0, 0, 0, 1, 1, 1, 1])
('timestep:08 - ', [0, 0, 1, 1, 1, 1, 1, 1])
('timestep:09 - ', [0, 0, 0, 1, 1, 1, 1, 1, 1])
('timestep:10 - ', [0, 0, 0, 1, 1, 1, 1, 1, 1, 1])
('timestep:11 - ', [0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1])
('timestep:12 - ', [0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1])
('timestep:13 - ', [0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1])
('timestep:14 - ', [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
('timestep:15 - ', [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
('timestep:16 - ', [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
('timestep:17 - ', [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0])