Stochastic Modeling and Optimization Problemset 1

Aimee Barciauskas, Andreas Lloyd, Francis Thomas Moynihan IV, and Seda Yilmaz

26 February 2016

1. Machine maintenance

Suppose that we have a machine that is either running or is broken down. If it runs throughout one week it makes a gross profit of $100. If it fails during the week, the gross profit is zero. If it is running and at the start of the week and we perform preventive maintenance, the probability that it will fail during the week is 0.4. If we do not perform such maintenance, the probability of failure is 0.7. However, maintenance will cost $20. When the machine is broken down at the start of the week, it may either be repaired at a cost of $40, in which case it will fail during the week with probability 0.4, or it may be replaced at a cost of $90 by a new machine that is guaranteed to work properly through its first week of operation. Reformulate the problem within the DP framework, and find the optimal repair-replacement-maintenance policy that maximizes total profit over four weeks, assuming a new machine at the start of the first week.

- x_k - (*state*) Machine is working or broken
- u_k - (*control*) Whether to perform maintenance
- w_k - (*uncertainty*) Machine breaks or doesn't break

There are no additional dynamics or constraints for this problem.

Expected profit is always $100 - g(u)$ where $g(u)$ is the exepected cost of decision u, (i.e.: $\mathbb{E}_w\{G(u, w)\}$). By minimizing expected cost we maximize expected profit.

At the beginning of each week, we much pick an optimal strategy given x_k: the machine is working or broken.

When the machine is working, the state at time 0, we have two strategies and their associated expected costs:

1. Perform Maintenence

If maintenance is performed, the expected cost is 20 (the cost of maintenance) plus the probability the machine breaks during the week times lost profit:

$J_{\mu_0} = 20 + 0.4 * 100 = 60$

2. Do nothing

If maintenance is not performed, the expected cost is the probability the machine breaks down times lost profit.

$J_{\mu_1} = 0.7 * 100 = 70$

When the machine is working, performing maintenance minimizes expected cost

When the machine is broken, there are three strategies:

1. Buy a new machine:

$J_{\mu_0} = 90$

2. Repair the machine:

$J_{\mu_1} = 40 + 0.4 * 100 = 80$

3. Do nothing:

$J_{\mu_2} = 100$

When the machine is broken, making a repair minimizes expected cost

The optimal strategy which minimizes expected cost (and maximizes expected profit) is to perform preventative maintenance when the machine is working and repair the machine when it is broken.

Alternately, we can state these strategies as the policies:

$\pi = \mu_k(x_k) = u_k$

$\mu_k(working) =$ perform preventative maintenance

$\mu_k(broken) =$ repair broken machine

DP Algorithm

1. $J^*(x_0) = min_\pi J_\pi(x_0) = \mu_k(working)$
2. From week $1, .., 4$, act according to policies $\pi = \mu(broken), \mu(working)$, as the policies are detailed above.

2. Discounted Cost

In the framework of the basic problem, consider the case where the cost is of the form:

$$\mathbb{E}_{w_k}\left\{\alpha^N g_N(x_N) + \sum_{k=0}^{N-1} \alpha^k g_k(x_k, u_k, w_k)\right\}$$

where $\alpha \in (0,1)$ is a discount factor. Develop a DP-like algorithm for this problem.

$$J_n(x_n) = \alpha^N g_N(x_N) <=> J_N(x_N)\alpha^{-N} = g_N(x_N)$$

$$J_k(x_k) = \min_{u_k} \mathbb{E}_{w_k}\left\{\alpha^k g_k(x_k, u_k, w_k) + J_{k+1}(f_k(x_k, u_k, w_k)\right\}$$

$$J_k(x_k)\alpha^{-k} = \min_{u_k} \mathbb{E}_{w_k}\left\{g_k(x_k, u_k, w_k) + \alpha^{-k} J_{k+1}(f_k(x_k, u_k, w_k)\right\}$$

$$J_k(x_k)\alpha^{-k} = \min_{u_k} \mathbb{E}_{w_k}\left\{g_k(x_k, u_k, w_k) + \alpha\alpha^{-k+1} J_{k+1}(f_k(x_k, u_k, w_k)\right\}$$

Let $\phi_k := J_k x_k \alpha^{-k}$, then for the DP algorithm:

Initial step: $\phi_N(x_N) = g_N(x_N$

Recursive function: $\phi_k(x_k) = \min_{u_k} \mathbb{E}_{w_k}\left\{g_k(x_k, u_k, w_k) + \alpha\phi_{k+1}(f_k(x_k, u_k, w_k))\right\}$

3. Multiplicative cost

In the framework of the basic problem, consider the case where the cost has the multiplicative form:

$$\mathbb{E}_{w_k}\left\{g_N(x_N)g_{N-1}(x_{N-1},u_{N-1},w_{N-1})...g_0(x_0,u_0,w_0)\right\}$$

The primitives of the problem are the same as for the generic DP framework:

- x_k - state at time k
- u_k - control at time k (a strategy)
- w_k - uncertainty at time k, the realization of a random variable

The DP algorithm proceeds as follows:

1. Start with:

$J_N(x_N) = g_N(x_N)$

2. Work backwards in time:

At time i, select the u_k which minimizes the expected cost

$$J_k(x_k) = min_{u_k \in U_k}\mathbb{E}_{w_k}g_N(x_N) + \prod_{k=i}^{N-1} g_k(x_k, \mu_k(x_k), w_k)$$

And the optimal policy becomes the $\mu_i^*, ..., \mu_{N-1}^*$ which result from this optimizes.

$$J_k^*(x_k) = J_k(x_k)$$

(as shown in slide 7 of Lecture 2 Slides*)

The policy which minimizes the expected cost is equivalent to a shortest path problem and thus satisfies the principal of optimality: the policies which optimize the tail subproblems are necessarily also a part of the optimal policy. Minimizing cost working backwards solves for the minimal cost.

4. Knapsack Problem

Assume that we have a vessel whose maximum weight capacity is Z and whose cargo is to consist of different quantities of N different items. Let v_i denote the value of the ith type of item, w_i the weight of the i^{th} type of item, and x_i the number of items of type i that are loaded in the vessel. The problem is to find the most valuable cargo, i.e., to maximize $\Sigma_{i=1}^{N} x_i v_i$ subject to the constraints $\Sigma_{i=1}^{N} x_i w_i \leq Z$ and $x_i \in N$. Reformulate the problem within the DP framework.

- *states:* z_i weight left after adding item i
- *controls:* u_i adding a weight $x_i w_i$

$$u_i(x_i) \in \{0 \leq u_i \geq z_i, z_i \in [0, Z]\}$$

- *uncertainty:* None
- *dynamics:* $f_i(x_i, u_i) = z_{i+1} = z_i - x_i w_i$
- *cost:*
$$g_i(x_i, u_i) = \frac{(z_i - x_i w_i)}{w_i} v_i$$
The cost of not filling up the remaining space with that item.

DP algorithm

$$g_N(x_N) = J_N(x_N)$$
$$J_i(x_i) = \min_{u_i} \Big\{ g_i(x_i, u_i) + J_{i+1}(f_i(x_i, u_i)) \Big\}$$

5. Traveling Repairman Problem...

A repairman must service N sites, which are located along a line and are sequentially numbered $1, 2, ..., N$. The repairman starts at a given site s with $1 < s < N$, and is constrained to service only sites that are adjacent to the ones serviced so far, i.e., if he has already serviced sites $i, i+1, ..., j$, then he may service only site $i-1$ (assuming $1 < i$) or site $j+1$ (assuming $j < N$). There is a waiting cost of ci for each time period that site i has remained unserviced and there is a travel cost t_{ij} for servicing site j right after site i. Reformulate the problem within the DP framework.

- *states:* x_k Sites served so var
- *controls:* u_k Site to serve next

$$u_k(x_k) \in \big\{s_{i-1}, s_{j+1}\big\}, x_k = \big\{s_i, s_{i+1}, ..., s_j\big\}$$

- *uncertainty:* w_k n/a
- *dynamics:* $f(x_k, u_k) = x_k \cup \big\{u_k(x_k)\big\} = x_{k+1}$
- *cost:*

$$g_k(x_k, u_k) = t_k + \sum_{i=0}^{U_k} c_i$$

where t_k is travelling cost from $u_{k-1}(x_{k-1})$ to $u_k(x_k)$ and c_i is the waiting cost of each site not yet visited. S is the set of all sites and $U_k = S - \big\{x_k\big\}$, e.g. we can chose from all sites not yet visited.

- *constraints:* Next site to serve must be adjacient to the sites served so far.

DP algorithm

$$g_N(x_N) = J_N(x_N)$$

$$J_k(x_k) = \min_{u_k} \Big\{ g_k(x_k, u_k) + J_{k+1}(f_k(x_k, u_k)) \Big\}$$