Justine Serdoncillo

IE 5571 Dynamic Programming

HW1

September 25, 2023

Problem 1 Political campaign entering final stage. Candidate can fund ads for 5 commercials to get possible extra votes.

Commercials	A1	A2	A3	A4
0	0	0	0	0
1	4	6	5	3
2	7	8	9	7
3	9	10	11	12
4	12	11	10	14
5	15	12	9	16

stage - # of commercials purchased overall

state - # of commercials per station

controls - choose any of {A1, A2, A3, A4}

dynamical system - follow the chart

cost function - extra votes by choosing new areas

$$\text{maximize} \sum_{n=1}^{4} A_n(x_n) \text{ such that } \sum_{n=1}^{4} x_n = 5 = w$$

The goal of the system is to maximize votes. This is a general resource allocation problem

work backwards

Define $V_j(w) = \sum_{n=j}^{4} A_n(x_n)$

$w = \sum_{n=j}^{4} x_n$

$V_n(w) = \max[r_n(x) + V_{n+1}(w-x)]$

$V_5(0) = 0$

$V_4(0) = 0, V_4(1) = 3, V_4(2) = 7, V_4(3) = 12, V_4(4) = 14, V_4(5) = 1$

try $V_3(w)$

$V_3(0) = 0$

$V_3(1) = \max[5, 3] = 5$

$V_3(2) = \max[9, 5+3, 7] = 9$

$V_3(3) = \max[11, 9+3, 5+7, 12] = 12$

$V_4(4) = \max[10, 11+3, 9+7, 5+12, 14] = 17$

$V_4(5) = \max[9, 10+3, 11+7, 9+12, 5+14, 16] = 2$

try $V_2(w)$

$V_2(0) = 0$

$V_2(1) = \max[6, 5] = 6$

$V_2(2) = \max[8, 6+5, 9] = 11$

$V_2(3) = \max[10, 8+5, 6+9, 12] = 15$

$V_2(4) = \max[11, 10+5, 8+9, 6+12, 16] = 18$

$V_2(5) = \max[12, 11+5, 10+9, 8+12, 6+17, 21] = 23$

lastly try $V_1(w)$

$V_1(0) = 0$

$V_1(1) = \max[4, 6] = 6$

$V_1(2) = \max[7, 4+6, 11] = 11$

$V_1(3) = \max[9, 7+6, 4+11] = 15$

$V_1(4) = \max[12, 9+6, 7+11, 4+15] = 19$

$V_1(5) = \max[15, 12+6, 9+11, 7+15, 4+18, 23]$

$= 23$

$\boxed{\text{Upon funding Area 2 - 1 commercial, Area 3 1 commercial \& Area 4 3 commercials we get a total of 23k additional votes}}$

Problem 2) Farmer J has \$5000 & 10 tons of wheat

Rules
1) can't fall on debt
2) can't lie to buyer
3) $m \le 10$

3)

stage - j

state - # of \$ C_j (cash) & wheat W_j & P_j (Price of wheat per month)

controls - sell up to M, S_j & Buy up to $10 - M_j$, B_j

dynamical system -

cost function - $g(x, u) =$

dynamic programming to maximize N

Using Dynamic Programming we get the following descriptions from above

$C_0 = 5000$
$W_0 = 10$

Rules equate to ① $B_j P_j \le C_j$ ③ $W_j + B_j \le 10$

② $S_j \le W_j$

The dynamical system is $C_{j+1} = C_j - B_j P_j + S_j P_j = C_j + (S_j - B_j) P_j$

& $W_{j+1} = W_j + B_j - S_j = W_j - (S_j - B_j)$

Want to maximize C_j at $j = 3$ where $j = 0, 1, 2, 3$

from month 2 to 3

$C_3 = C_2 + (S_2 - B_2) P_2$ & $W_3 = W_2 - (S_2 - B_2)$

to maximize $C_3 \Rightarrow (S_2 - B_2)_{max} \Rightarrow S_2 = W_2$ & $\underline{B_2 = 0}$ or essentially sell all

from month 1 to 3

$C_2 = C_1 + (S_1 - B_1) P_1$ & $W_2 = W_1 - (S_1 - B_1)$ let $D_1 = S_1 - B_1$

$C_3 = C_2 + S_j P_2$ & $W_3 = W_2 - (S_2)$

to maximize C_3

$C_3 = C_1 + (S_1 - B_1) P_1 + S_2 P_2$ & $W_3 = W_1 - (S_1 - B_1) - S_2 = W_1 + B_1 - (S_1 + S_2) = 0$

$W_1 + B_1 = S_1 + S_2$

if $P_1 = P_2$ then best do nothing

or $P_1 > P_2$, $P_2 > P_1$: $C_3 = C_1 + W_1 P_1 + S_2 (P_2 - P_1)$ $\rightarrow$ $C_3 = C_1 + B_1 P_1 + S_2 P_2$ if $P_1 > P_2$ then maximize $W_1 - S_2$

if $P_2 > P_1$ then minimize $W_1 - S_2$

from month 0 to 3

$C_1 = C_0 + (S_0 - B_0) P_0$ & $W_1 = W_0 - (S_0 - B_0)$

$C_3 = 5000 + (S_0 - B_0) P_0 + (S_1 - B_1) P_1 + S_2 P_2$ $W_3 = 10 - (S_0 - B_0) - (S_1 - B_1) - S_2 = 0$

$(10 + B_0 + B_1) = S_0 + S_1 + S_2$

using DP,

$$\underset{D_j \in}{\text{maximize}} \left\{ C_3 \;\middle|\; C_0 = 5000,\ W_0 = 10 \right\}$$

low debt
to buyer

3) stage - De...

Problem 2) Attempt #2
Redoing #2

let stage: months finished ; $j = 0, 1, 2, 3$

state: cash, wheat, price: C_j, W_j, P_j

controls: # of sold - # bought : D_j s.t. $W_j - D_j \le 10$ $D_j \le W_j$

$-D_j P_j \le C_j$

Dynamical system: $C_{j+1} = C_j + D_j P_j$ $W_{j+1} = W_j - D_j$

Cost function $g(x) = D_j P_j$

Dynamic programming equation

$$\boxed{V_j(x) = \underset{D < W}{\text{maximize}} \left[C_j + V_{j+1}\left(C_j + D_j P_j \right) \right] \quad \text{at } j = 3 \quad V_{j+1} = 0}$$

$V_3(x) = \underset{D_3}{\text{maximize}} \left[C_3 + \right.$

3) stage - period; n up to N

state variables - stocklevel; x_n

control - quantity ordered; $u_n \geq 0$

dynamics - $x_{n+1} = x_n + u_n - Y_{n+1}$ — demand at n

cost function - operation cost $c(x_n, u_n; Y_{n+1}) = pu_n + h(x_n + u_n - Y_{n+1})^+ + b(x_n + u_n - Y_{n+1})^-$

production cost

unit holding cost

backlogging cost

$b > p$

minimize $E\left[\sum_{n=0}^{n-1} c(x_n, u_n; Y_{n+1})\right]$

$x^+ = \max\{x, 0\}$ & $x^- = -\min\{x, 0\}$

a) dynamic programming equation

b) value function & optimal policy at $n = N-1$

---

a) The dynamic programming equation

the recursive equation to solve this system is

$$\boxed{\begin{aligned} V_n(x) &= \underset{0 \leq u_n \leq 1}{\text{minimize}} \left\{ C_n(x, u, y) + E\left[V_{n+1}(x_n + u_n - y_{n+1})\right]\right. \\ V_n(x) &= \underset{-1\ 0 \leq u_n \leq 1}{\text{minimize}} \left\{ pu_n + h(x_n + u_n - y_{n+1})^+ + b(x_n + u_n - y_{n+1})^- \right. \\ &\qquad + E\left[V_{n+1}(x_n + u_n - y_{n+1})\right] \Big\} \end{aligned}}$$

b) at $n = N-1$

no need to think about future stage

$$\boxed{V_{N-1}(x_{n-1}) = \underset{0 \leq u_{N-1} \leq 1}{\text{minimize}} \left\{ C_{n-1}(x, u, y) + E[0] \right\}}$$

To get the optimal policy need to minimize C_{n-1} w/respect to u_{n-1}

so take $\frac{dC_{n-1}}{du_{n-1}} = 0 \Rightarrow C(x_{n-1}, u_{n-1}; y_n) = pu_{n-1} + h(x_{n-1} + u_{n-1} - y_n)^+ + b(x_{n-1} + u_{n-1} - y_n)^+$

$$\frac{dC_{n-1}}{du_{n-1}} = p + h^+ + b^+ = 0 \rightarrow \boxed{p + \max(h, 0) - \min(b, 0) = 0} \quad ?$$

4) initial wealth x_0 - state each time point $n \in \mathbb{N}_{N-1}$ - stage

get proportion $u_n \in [0,1]$ to consume x_n & put the rest in the bank

interest $r > 0$ $\quad R = 1+r$

$x_{n+1} = R x_n (1-u_n) \quad n \in \mathbb{N}_{N-1}$ get utility $g(u_n x_n)$

at N $\quad g_N(x_N)$

choose controls $\{u_n\}$ to maximize

$$v(x_0) = \max_{\{u_n\}} \left\{ g_N(x_N) + \sum_{n=0}^{N-1} g(u_n x_n) \right\}$$

a) write DPE

b) assume $f(x) = x^\alpha$ & $g(x) = x^\alpha$

$\alpha \in (0,1)$

Show $V_n(x) = C_n x^\alpha$

C_n defined recursively & independent of current wealth

a) The dynamic programming equation here is

$$\boxed{V_n(x) = \max_{u_n} \left[g_N(x_N) + \sum_{n=0}^{N-1} g(u_n x_n) + V_{n+1}\left(R x_n (1-u_n)\right) \right]}$$

b) assume $g(x) = x^\alpha$ simply $g(x)$ by x^α

$$V_n(x) = \max_{u_n} \left[x^\alpha + \sum_{n=0}^{N-1} (u_n x_n)^\alpha + V_{n+1}\left(R x_n (1-u_n)\right) \right]$$

to get max need to take derivative of $V_n(x)$ w.r.t. u_n & $= 0$

$$\frac{dV_n(x)}{du_n} = \sum_{n=0}^{N-1} \alpha u_n^{\alpha-1} x_n^\alpha + V_{n+1}'\left(R x_n (1-u_n)\right) - R x_n u_n V_{n+1} = 0$$

let $C_n^\alpha = \alpha (u_n x_n)^\alpha$ & not sure but

this can be defined recursively &

C_n shouldn't be a $f(x^n)$

if $V_n(x) = C_n x^\alpha$

$$\text{then } \frac{dV_n(x)}{du_n} = \sum_{n=0}^{N-1} \alpha u_n^{\alpha-1} x_n^\alpha + C_{n+1} x^\alpha \left(R x_n (1-u_n)\right) - R x_n u_n C_{n+1} x^\alpha -$$

becomes 0

then $\boxed{V_n(x) = C_n x^\alpha}$

Sutton & Barto *Tracking a Nonstationary problem* < *Multi-armed Bandits*

2.4) if $\alpha_n \neq C$ then estimate Q_n is a weighted average of previously received rewards with a weighting different from (2.6) $Q_{n+1} = (1-\alpha)^n Q_1 + \sum_{i=1}^{n} \alpha(1-\alpha)^{n-i} R_i$

what is the weighting on each prior reward for the general case, analogous to (2.6) in terms of the sequence of step-size parameters

---

from 2.6 $$Q_{n+1} = (1-\alpha)^n Q_1 + \sum_{i=1}^{n} \alpha(1-\alpha)^{n-i} R_i$$

from $Q_{n+1} = Q_n + \alpha[R_n - Q_n]$

now $Q_{n+1} = Q_n + \alpha_n[R_n - Q_n] = \alpha_n R_n + Q_n(1-\alpha_n)$

$$Q_{n+1} = \alpha_n R_n + (1-\alpha_n)\left[Q_{n-1} + \alpha_{n-1}\left[R_{n-1} - Q_{n-1}\right]\right]$$

$$Q_{n+1} = \alpha_n R_n + \alpha_{n-1} R_{n-1}(1-\alpha_n) + (1-\alpha_n)(1-\alpha_{n-1}) Q_{n-1}$$

$$Q_{n+1} = \alpha_n R_n + \alpha_{n-1}(1-\alpha_n) R_{n-1} + (1-\alpha_n)(1-\alpha_{n-1})(1-\alpha_{n-2}) Q_{n-2} + (1-\alpha_n)(1-\alpha_{n-1})\alpha_{n-2} R_{n-2}$$

$$\boxed{Q_{n+1} = Q_1 \prod_{i=1}^{n} (1-\alpha_i) + \sum_{i=1}^{n} \alpha_i R_i \prod_{j=i+1}^{n} (1-\alpha_j)}$$

pattern form

2.5) Design & conduct an experiment to demonstrate difficulties that sample-average methods have for nonstationary problems: Modified version of 10-armed testbed $q_*(a)$ all =. independent random walks ($\mu = 0$, $\sigma = 0.01$)

$\alpha = 0.1$, $\varepsilon = 0.1$ $k = 10{,}000$ steps

2:7) *Optimistic Initial Values*

on nonstationary problems

use step size of $\beta_n \doteq \alpha / \bar{o}_n$ — $\alpha > 0$ conventional constant, $\bar{o}_n$ trace of one that starts at 0:

$$\bar{o}_n \doteq \bar{o}_{n-1} + \alpha(1 - \bar{o}_{n-1}) \quad \text{for } n > 0 \text{ w/ } \bar{o}_0 \doteq 0$$

show that Q_n is an exponential recency-weighted average w/o initial bias

---

$$\bar{o}_{n+1} = \bar{o}_n + \alpha(1 - \bar{o}_n) = \alpha + (1-\alpha)\bar{o}_n \,, \quad \bar{o}_n = \alpha + (1-\alpha)\bar{o}_{n-1}$$

following this

$$\bar{o}_{n+1} = \alpha + (1-\alpha)\alpha + (1-\alpha)^2 \bar{o}_{n-1}$$

$$= \alpha + (1-\alpha)\alpha + (1-\alpha)^2\alpha + (1-\alpha)^3 \bar{o}_{n-2}$$

$$= \text{''} \qquad \text{''} \qquad + \dots + \cancel{(1-\alpha)^{n+1}\bar{o}_0} + (1-\alpha)^n \bar{o}_1$$

$$= \alpha \sum_{i=0}^{n} (1-\alpha)^i$$

$$\beta_n = \frac{\alpha}{\bar{o}_n} = \frac{\alpha}{\alpha \sum_{i=0}^{n-1} (1-\alpha)^i} = \frac{1}{\sum_{i=0}^{n-1} (1-\alpha)^i}$$

$$\boxed{\beta_n = \frac{1}{\sum_{i=0}^{n-1} (1-\alpha)^i}}$$