Classification of LQR problems

1. Discrete time $x_{k+1} = Ax_k + Bu_k$
   - Finite horizon (terminal step N)

     $$J = \sum_{k=0}^{N-1} (x_k^T Q x_k + u_k^T R u_k) + x_N^T Q x_N.$$

   - Infinite horizon

     $$J = \sum_{k=0}^{\infty} (x_k^T Q x_k + u_k^T R u_k)$$

2. Continuous time. $\dot{x} = Ax + Bu$
   - Finite horizon

     $$J = \int_0^{t_1} (x^T Q x + u^T R u) dt + x(t_1)^T Q x(t_1)$$

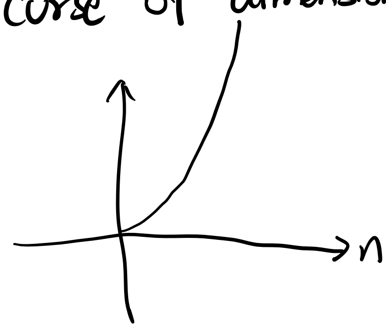
   - Infinite horizon

     $$J = \int_0^{\infty} (x^T Q x + u^T R u) dt$$

How to solve.

1) Dynamic Programming
2) Least square
3) Nonlinear programming

curse of dimension

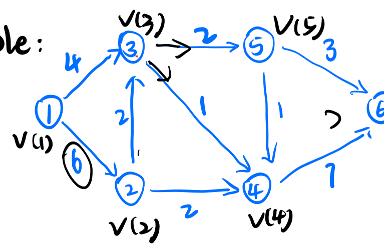


1. Dynamic Programming (intuition)

(mathematical optimization method & computer programming method)

simplify a complicated problem by breaking it down into simpler sub-problems in a recursive manner

Network example: to study the idea of DP



Goal: find min cost route (path) from 1 to node 6

All possible routes:	cost
$1 \underset{6}{-} 2 \underset{2}{-} 4 \underset{7}{-} 6$	15
1 − 2 − 3 − 4 − 6	16
1 − 2 − 3 − 5 − 6	13
1 − 3 − 4 − 6	12
1 − 3 − 5 − 6	9

$V(1) = \min_j (C_{1j} + V(j))$

j can be 2, 3.

Let $V(i)$ be the min cost from node i to node 6 over all possible paths

DP principle

$$\boxed{V(i) = \min_j (C_{ij} + V(j))}$$

compare to the LQR case:

$C_{ij} \sim L(k, x_k, u_k)$

$V(j) \sim V^*_{k+1}(x_{k+1})$

n nodes

n(n-1) links

n! routes.

1. $V(6) = 0$ (N)

2. $V(5) = \min_j (C_{5j} + V(j))$ $V(5) = 3.$

   $j = 4, 6.$ $C_{54} + V(4) = 1 + 7 = 8.$

   $C_{56} + V(6) = 3 + 0 = 3$

3. $V(4) = \min_j (C_{4j} + V(j))$ $V(4) = 7.$

   $j = 6$ $C_{46} + V(6) = 7.$

4. $V(3) = \min_j (C_{3j} + V(j))$ $V(3) = 5$

   $j = 4, 5$ $C_{34} + V(4) = 1 + 7 = 8$

   $C_{35} + V(5) = 2 + 3 = 5$

5. $V(2) = \min (C_{2j} + V(j))$ $V(2) = 7$

   $j = 3, 4$ $C_{23} + V(3) = 2 + 5 = 7$

$C_{24} + V(4) = 2+7 = 9$

6. $V(1) = \min_{j} \left(C_{1j} + V(j) \right)$ $V(1) = 9$

$j = 1, 2$ $C_{12} + V(2) = 6+7 = 13$

$C_{13} + V(3) = 4+5 = 9$

Bellman's principle

An optimality policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regarding to the state resulting from the first decision

The optimal control at time step k depends only on x_k.

$\boxed{\text{The key insight:}}$ work backward from final time to determine optimality.

Bellman's principle for general cases

Suppose we have a DT system: $x_{k+1} = f(x_k, u_k)$

$$x_{k+1} = Ax_k + Bu_k$$

$$V(x, u) = \sum_{k=0}^{N-1} \underbrace{\overset{x_k^T Q x_k + u_k^T R u_k}{L(k, x_k, u_k)}}_{\text{running cost}} + \underbrace{l(N, x_N)}_{\text{final cost.}}$$

k : time step

Let $V_k^*(x_k)$ denote the lowest cost from x_k to x_N.

$$V_k^*(x_k) = \min_{u_k} \left[L(k, x_k, u_k) + V_{k+1}^*(x_{k+1}) \right].$$

cost of x_k to x_N. = Cost of $x_k \to x_{k+1}$ + min cost of $x_{k+1} \to x_N$.

Bellman's principle for LQR.

$$V(x,u)=\sum_{k=0}^{N-1}(x_k^T Q x_k + u_k^T R u_k) + x_N^T Q x_N.$$

$$\boxed{V_k^*(x_k)=\min_{u_k}\left[x_k^T Q x_k + u_k^T R u_k + V_{k+1}^*(x_{k+1})\right]}$$

cost-to-go

1. The last time step $k=N$

$$V_N^*(x_N) = x_N^T Q x_N = x_N^T P_N x_N$$

$$x_N = A x_{N-1} + B u_{N-1} = (A-BK)x_{N-1}$$

2. $$V_{N-1}^*(x_N)=\min_{u_{N-1}}\left[x_{N-1}^T Q x_{N-1} + \underbrace{u_{N-1}^T}_{u_{N-1}=-Kx_{N-1}} R u_{N-1} + \underbrace{x_N^T P_N x_N}_{(Ax_{N-1}+Bu_{N-1})^T P_N (Ax_{N-1}+Bu_{N-1})}\right]$$

$\Rightarrow [(A-BK)x_{N-1}]^T P_N [(A-BK)x_{N-1}]$

$\Rightarrow x_{N-1}^T (A-BK)^T P_N (A-BK) x_{N-1}$

The quadratic form remains

$$\boxed{V_k^*(x_k) = x_k^T P_k x_k}$$

min cost from x_{k+1} to x_N.

write $V_k^*(x_k)$ as function of x_k, u_k, P_{k+1}. $V_{k+1}^*(x_{k+1})$

get rid of x_{k+1} :

$$V_k^*(x_k)=\min_{u_k}\left[x_k^T Q x_k + u_k^T R u_k + \underline{x_{k+1}^T P_{k+1} x_{k+1}}\right]$$

$x_{k+1} = Ax_k + Bu_k$

$$\min_{u_k}\left[x_k^T Q x_k + u_k^T R u_k + \underline{(Ax_k+Bu_k)^T} P_{k+1}(Ax_k+Bu_k)\right]$$

$$\Rightarrow \min_{u_k}\left[x_k^T Q x_k + u_k^T R u_k + (x_k^T A^T + u_k^T B^T) P_{k+1}(Ax_k+Bu_k)\right]$$

$$\Rightarrow \min_{u_k}\left[x_k^T Q x_k + u_k^T R u_k + x_k^T A^T P_{k+1} A x_k + u_k^T B^T P_{k+1} A x_k + x_k^T A^T P_{k+1} B u_k + u_k^T B^T P_{k+1} B u_k\right]$$

This is a minimization problem, we need to find u_k such that what's in the bracket is minimized find u_k to minimize $V_k^*(x_k)$ (Least square review in the last page)

$$\boxed{\min_{u_k}\left[u_k^T(R+B^TP_{k+1}B)u_k + 2u_k^TB^TP_{k+1}Ax_k\right]} ☆$$

take the derivative w.r.t. u_k.

$$2(R+B^TP_{k+1}B)u_k + 2B^TP_{k+1}Ax_k = 0$$

$$\boxed{\begin{aligned}\underset{\text{optimal}}{u_k^*} &= -(R+B^T\underline{P_{k+1}}B)^{-1}B^T\underline{P_{k+1}}A\underline{x_k} \quad ☆\\ &= -K_kx_k\end{aligned}}$$

obtained u_k^* as a function of P_{k+1} and x_k

Then plug this u_k^* into $V_k^*(x_k)$

$$\underbrace{V_k^*(x_k)}_{x_k^TP_kx_k} = \min_{u_k}\left[x_k^TQx_k + u_k^TRu_k + x_{k+1}^TP_{k+1}x_{k+1}\right]$$

$$= x_k^TQx_k + x_k^TA^TP_{k+1}B(R+B^TP_{k+1}B)^{-1}R(R+B^TP_{k+1}B)^{-1}B^TP_{k+1}Ax_k$$
$$+ [(A-BK)x_k]^TP_{k+1}[(A-BK)x_k]$$

$$= \overset{(1)}{x_k^TQx_k} + \underline{\overset{(1)}{x_k^TA^TP_{k+1}B(R+B^TP_{k+1}B)^{-1}}}\,\overset{(2)}{R}\,\underline{\overset{(2)}{(R+B^TP_{k+1}B)^{-1}B^TP_{k+1}Ax_k}}$$
$$+ x_k^T[\underbrace{A^T}_{(1)} - \underbrace{A^TP_{k+1}B(R+B^TP_{k+1}B)^{-1}B^T}_{(2)}]P_{k+1}[\underbrace{A}_{(3)} - \underbrace{B(R+B^TP_{k+1}B)^{-1}P_{k+1}A}_{(4)}]x_k$$

we call this last term the ☆ term

only focus on the ☆ term.

(1) $\cdot P_{k+1} \cdot$ (3) $\quad x_k^TA^TP_{k+1}Ax_k$ (3)

(2) P_{k+1} (3) $\quad -x_k^TA^TP_{k+1}B(R+B^TP_{k+1}B)^{-1}B^TP_{k+1}Ax_k$ (4)

(1) P_{k+1} (4) $\quad -x_k^TA^TP_{k+1}B(R+B^TP_{k+1}B)^{-1}B^TP_{k+1}Ax_k$ (5)

(2) P_{k+1} (4) $\quad \underbrace{x_k^TA^TP_{k+1}B(R+B^TP_{k+1}B)^{-1}}_{(1)}B^TP_{k+1}B\underbrace{(R+B^TP_{k+1}B)^{-1}B^TP_{k+1}Ax_k}_{(2)}$ (6)

Now $V_k^*(x_k)$ has 6 terms in total. I use green color to name them.

In class, we called ⑥ $*$, we called ② $**$

we want to simplify these six terms, first we compare $*$ and $**$. The $*$ term can be writen as

$$\left.\begin{aligned} *&: \ ①\, B^T P_{k+1} B\, ② \\ **&: \ ①\, R\, ② \end{aligned}\right\} \Rightarrow * + ** = ①\,(R + B^T P_{k+1} B)\,**$$

$$* + ** = x_k^T A^T P_{k+1} B (R + B^T P_{k+1} B)^{-1} B^T P_{k+1} A x_k \ ⑦.$$

Now we have ① ③ ④ ⑤ and the above term, call it ⑦.

Then we can see ④ = ⑤ = − ⑦, add them together, we have ④ + ⑤ + ⑦ = ④. hence for $V_k^*(x_k)$, we have ①, ③, ④ left.

$$V_k^*(x_k) = \overset{①}{x_k^T Q x_k} + \overset{③}{x_k^T A^T P_{k+1} A x_k}$$

$$\qquad - \overset{④}{x_k^T A^T P_{k+1} B (R + B^T P_{k+1} B)^{-1} B^T P_{k+1} A x_k}$$

$$= x_k^T \left[Q + A^T P_{k+1} A - A^T P_{k+1} B (R + B^T P_{k+1} B)^{-1} B^T P_{k+1} A \right] x_k$$

$$= x_k^T P_k x_k.$$

Lyapunov equation

Riccati equation $\quad Q + A^T P A = P$

$$\boxed{P_k = Q + A^T P_{k+1} A - A^T P_{k+1} B (R + B^T P_{k+1} B)^{-1} B^T P_{k+1} A} \ \star$$

$$V_{k+1}^*(x_{k+1}) = x_{k+1}^T \underline{P_{k+1}} x_{k+1}$$

$$V_k^*(x_k) = x_k^T \underline{P_k} x_k$$

$$J = \sum_{k=0}^{N-1} (x_k^T Q x_k + u_k^T R u_k) + \underline{x_N^T Q x_N}$$

$$P_N = Q \qquad V_N^*(x_N) = x_N^T Q x_N$$

$$P_{N-1} = Q + A^T P_N A - A^T P_N B (R + B^T P_N B)^{-1} B^T P_N A$$

$$P_{N-2} = Q + A^T P_{N-1} A - A^T P_{N-1} B (R + B^T P_{N-1} B)^{-1} B^T P_{N-1} A$$

$$\vdots$$

$$P_0.$$

$x_{k+1} = A x_k + B u_k \qquad x_0. \qquad Q, R \qquad$ given

Summarize of solving LQR by DP

step 1: set $P_N = Q$

step 2: for $k = N-1, N-2, \cdots 0$

$$P_k = Q + A^T P_{k+1} A - A^T P_{k+1} B (R + B^T P_{k+1} B)^{-1} B^T P_{k+1} A$$

step 3: for $k = 0, 1, 2, \cdots N-1$

$$K_k = (R + B^T P_{k+1} B)^{-1} B^T P_{k+1} A$$

$$u_k = -K_k x_k$$

Suppose you are given $x_{k+1} = A x_k + B u_k$, x_0 and Q, R.

you already computed all $P_N, P_{N-1}, P_{N-2}, \cdots P_2, P_1, P_0$

$$u_0 = -(R + B^T P_1 B)^{-1} B^T P_1 A x_0$$

$$x_1 = A x_0 + B u_1$$

$$u_1 = -(R + B^T P_2 B)^{-1} B^T P_2 A x_1$$

$$x_2 = A x_1 + B u_1$$

continue this process, you can have all controls and states: $u_0, u_1, \cdots u_{N-1}$ and $x_1, x_2, \cdots x_N$.

Review of least squares

$$\min_{x} \; \|y - Ax\|^2 = (y - Ax)^T (y - Ax) = J$$

$$\frac{\partial J}{\partial x} = \frac{\partial}{\partial x} (y - Ax)^T (y - Ax)$$

$$= \frac{\partial}{\partial x} \left(y^T y - x^T A^T y - y^T A x + x^T A^T A x \right)$$

$$\Delta J = -\Delta x^T A^T y - y^T A \Delta x + \Delta x^T A^T A x + x^T A^T A \Delta x$$

$$y^T A \Delta x \qquad\qquad x^T A^T A \Delta x$$

$$= \left(-2 y^T A + 2 x^T A^T A \right) \Delta x$$

$$\Rightarrow \frac{\partial J}{\partial x} = -2 y^T A + 2 x^T A^T A = 0.$$

$$\Rightarrow x^T = y^T A (A^T A)^{-1} \qquad x = (A^T A)^{-1} A^T y$$