CS 4/5789 14 Feb 2022 Prof Sarah Dean

1) Nonlinear Control

$$\min \mathbb{E}_{s_0 \sim \mu_0}\left[\sum_{t=0}^{H-1} c(s_t, a_t) \;\middle|\; s_{t+1} = f(s_t, a_t) \right]$$

In its full generality, this problem is hard to solve! Today we will learn an approach based on approximations.

In LQR (last lecture) we saw that the optimal policy did not depend on the disturbance w_t. Since we are going to use an approximation based on LQR for the nonlinear problem, we consider deterministic dynamics.

Assumption: dynamics $f: \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ are differentiable and cost $c: \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is twice differentiable.

$\nabla_s f(s,a)$ $\quad$ $\nabla_s c(s,a)$ $\quad$ $\nabla_s^2 c(s,a)$ $\quad$ $\nabla_{sa}^2 c(s,a)$

$\nabla_u f(s,a)$ $\quad$ $\nabla_a c(s,a)$ $\quad$ $\nabla_a^2 c(s,a)$

Assumption: Either

1) we know the analytical form of f & c, or
2) we have black-box access to f & c, i.e. for any s, a we can observe $s' = f(s,a)$ and $c = c(s,a)$.

IDEA: when dynamics are linear and costs are quadratic, we know how to find the optimal policy. So why not try

$\left.\begin{array}{l}\text{1) linearizing } f \\ \text{2) quadraticizing } c\end{array}\right\}$ linear/quadratic approximation

2) Linear / Quadratic Approximation

How can we find a good linear or quadratic approximation?

Recall Taylor Expansions: (in 1D)

$$g(x) = g(x_0) + g'(x_0)(x-x_0) + \frac{1}{2}g''(x_0)(x-x_0)^2 + \dots$$

when x is close to x_0, the higher order terms get vanishingly small

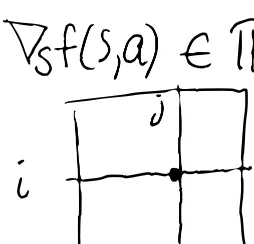
$$\varepsilon^p \to 0 \text{ as } p \to \infty \text{ for } \varepsilon < 1$$

e.g. $\underset{1e-3}{(0.001)}^3 = \underset{1e-9}{0.000000001}$

Linear Approximation

$$f(s,a) \approx f(s_0,a_0) + \nabla_s f(s_0,a_0)^T (s - s_0) + \nabla_a f(s_0,a_0)^T (a - a_0)$$

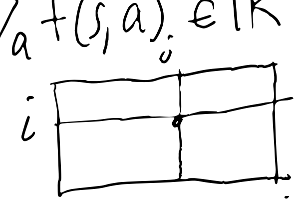
$\nabla_s f(s,a) \in \mathbb{R}^{n_s \times n_s}$



$\frac{\partial f_j}{\partial s_i}(s,a)$

entry row i col j corresponds to f_j and s_i.

$\nabla_a f(s,a) \in \mathbb{R}^{n_a \times n_s}$



$\frac{\partial f_j}{\partial a_i}(s,a)$

entry row i col j corresponds to f_j and a_i.

$$f(s,a) \approx f(s_0,a_0) + \nabla_s f(s_0,a_0)^T (s - s_0) + \nabla_a f(s_0,a_0)^T (a - a_0)$$

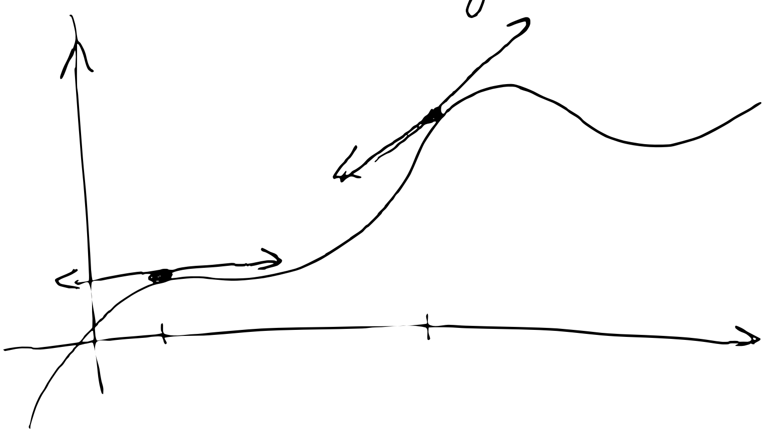
$$= As + Ba + V$$

$$A = \nabla_s f(s_0, a_0)^T$$

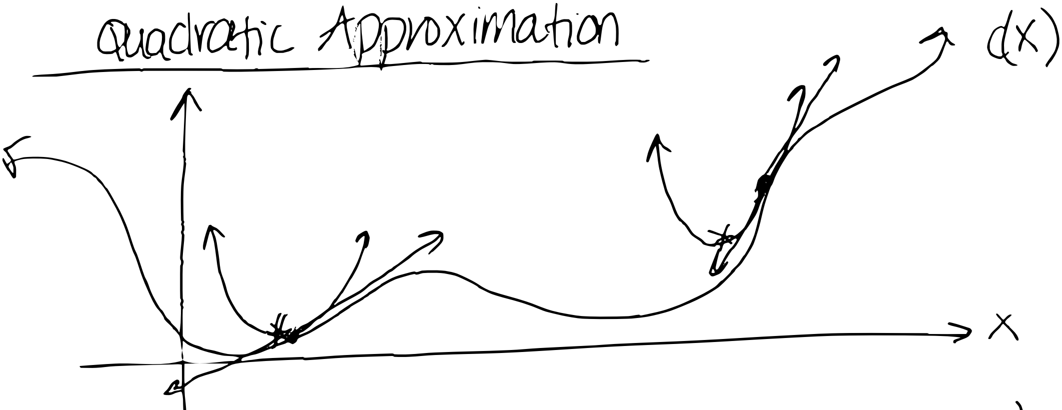
$$B = \nabla_a f(s_0, a_0)^T$$

$$V = f(s_0, a_0) + \nabla_s f(s_0,a_0)^T s_0 + \nabla_a f(s_0,a_0)^T a_0$$

depend only on the point we are linearizing around.



Quadratic Approximation



linear approximation doesn't encode (local) optima, so we use a quadratic (second-order) approximation for the cost function.

$$c(s,a) \approx c(s_0,a_0) + \nabla_s c(s_0,a_0)^T (s-s_0) + \nabla_a c(s_0,a_0)^T (a-a_0)$$
$$+ \frac{1}{2}(s-s_0)^T \nabla_s^2 c(s_0,a_0)(s-s_0)$$
$$+ \frac{1}{2}(a-a_0)^T \nabla_a^2 c(s_0,a_0)(a-a_0)$$
$$+ (a-a_0)^T \nabla_{as}^2 c(s_0,a_0)(s-s_0)$$

$\nabla_s c(s,a) \in \mathbb{R}^{n_s}$
@ index i $\frac{\partial c}{\partial s_i}(s,a)$

$\nabla_a c(s,a) \in \mathbb{R}^{n_a}$
@ index i $\frac{\partial c}{\partial a_i}(s,a)$

$\left.\begin{array}{l} \nabla_s^2 c(s,a) \in \mathbb{R}^{n_s \times n_s} \\ @\ i,j \quad \frac{\partial^2 c}{\partial s_i \partial s_j}(s,a) \\ \nabla_a^2 c(s,a) \in \mathbb{R}^{n_a \times n_a} \\ @\ i,j \quad \frac{\partial^2 c}{\partial a_i \partial a_j}(s,a) \end{array}\right\}$ symmetric

$\nabla_{as}^2 c(s,a) \in \mathbb{R}^{n_a \times n_s}$
@ i,j $\frac{\partial^2 c}{\partial a_i \partial s_j}(s,a)$

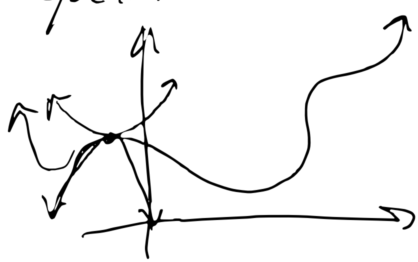
$$c(s,a) \approx c(s_0,a_0) + \nabla_s c(s_0,a_0)^T (s-s_0) + \nabla_a c(s_0,a_0)^T (a-a_0)$$
$$+ \frac{1}{2}(s-s_0)^T \nabla_s^2 c(s_0,a_0)(s-s_0)$$
$$+ \frac{1}{2}(a-a_0)^T \nabla_a^2 c(s_0,a_0)(a-a_0)$$
$$+ (a-a_0)^T \nabla_{as}^2 c(s_0,a_0)(s-s_0)$$

$$= s^T \bar{Q} s + a^T \bar{R} a + a^T M s$$
$$+ q^T s + r^T a + c$$

Practical consideration:

It doesn't make sense to minimize a downward facing quadratic.

Recall we had $Q, R \succ 0$, positive definite.



Therefore we
1) put all negative eigenvalues to 0
2) add regularization λI

If $\bar{Q} = \sum_{i=1}^{n_s} \sigma_i v_i v_i^T$, $Q = \sum_{i=1}^{n_s} \max(\sigma_i, 0) v_i v_i^T + \lambda I$

same for $\bar{R} \to R$

Black Box Access

What if we don't know the analytical forms of f & c and can only observe

$s' = f(s,a)$, $c = c(s,a)$ for s, a that we query?

Finite Differencing:

for scalar $g'(x) \approx \dfrac{g(x+\delta) - g(x-\delta)}{2\delta}$

For multivariate:

$$\frac{\partial f_i}{\partial s_j} \approx \frac{f(s+\delta e_j, a) - f(s-\delta e_j, a)}{2\delta}$$

$$f = \begin{bmatrix} f_1 \\ \vdots \\ f_{ns} \end{bmatrix} \qquad e_j = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \leftarrow j\text{th entry}$$

similar for $\dfrac{\partial c}{\partial s_j}$, $\dfrac{\partial f_i}{\partial a_j}$, $\dfrac{\partial c}{\partial a_j}$.

For second derivatives:

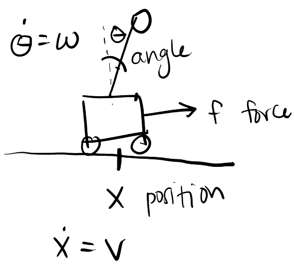
$$\frac{\partial c}{\partial a_i \partial s_j} = \frac{\partial}{\partial a_i}\left[\frac{\partial c}{\partial s_j}\right]$$

first estimate $\partial c / \partial s_i$ and then <u>another</u> with respect to a_i

3. Local LQR Control

Setting: minimize distance from a goal state/ action s_*, a_* $c(s,u) = d(s,s_*) + d(a,a_*)$

e.g. cartpole (HW 1)



$s = \begin{bmatrix} \theta \\ \omega \\ x \\ v \end{bmatrix}$ $a = f$

$\underbrace{s_* = 0, \ a_* = 0}_{\text{objective: balance upright}}$

Approach: Locally linearize f around (s_*, a_*) and 2nd order approximation of c around (s_*, a_*)

1) use finite differencing to compute approximate

a) $[\nabla_s f(s_*, a_*), \ \nabla_a f(s_*, a_*)$,

b) $[\nabla_s c(s_*, a_*), \ \nabla_a c(s_*, a_*), \ \nabla^2_{sa} c(s_*, a_*), \ \nabla^2_s c(s_*, a_*), \ \nabla^2_a c(s_*, a_*)$

2) use formulas above to compute

a) A, B, v

b) Q, R, M, q, r, c

we will call this procedure

$A, B, v, Q, R, M, q, r, c = \text{APPROX}(f, c, (s_*, a_*))$

$$\min_{} \mathbb{E}_{s_0 \sim \mu_0}\left[\sum_{t=0}^{H-1} s_t^T Q s_t + a_t^T R a_t + a_t^T M s_t + q^T s_t + r^T a_t + c \,\middle|\, s_{t+1} = A s_t + B a_t + v \right]$$

Generalization of the LQR problem we discussed last lecture (HW 1)

Results in quadratic V^* and affine π^*

$$\pi_t^*(s) = K_t^* s + k_t^*$$

$$\begin{matrix} K_0^*, \ldots, K_{H-1}^* \\ k_0^*, \ldots, k_{H-1}^* \end{matrix} = \text{LQR}(A, B, v, Q, R, M, q, r, c)$$

Today we abstract this computation.

In HW1 you will see that this works quite well for balancing the cartpole.

Problem: The approximations fail when s, a are far from s_*, a_*

4) Iterative LQR for trajectory optimization

IDEA: given a trajectory $\{(\bar{s}_t, \bar{a}_t)\}_{t=0}^{H-1}$ we can approximate around $(\bar{s}_t, \bar{a}_t)$ at time t.

This leads to a time-varying LQR problem with A_t, B_t, V_t and $Q_t, R_t, M_t, q_t, r_t, c_t$ still results in $a^*_t = K^*_t s_t + k^*_t$

However, which trajectory should we approximate around? Iterate!

Alg: iLQR

initialize $\bar{u}^0_0, \dots, \bar{u}^0_{H-1}$ and $\bar{s}^0_0 \sim \mu_0$

generate nominal trajectory $\tau_0 = \{(\bar{s}^0_t, \bar{a}^0_t)\}_{t=0}^{H-1}$ by $\bar{s}^0_{t+1} = f(\bar{s}^0_t, \bar{a}^0_t)$

for $i = 0, 1, \dots$

$$\{(A_t, B_t, V_t, Q_t, R_t, q_t, r_t, c_t)\}_{t=0}^{H-1} = \text{APPROX}(f, c, \tau_i)$$

$$\{K^*_t, k^*_t\}_{t=0}^{H-1} = \text{LQR}(\{A_t, B_t, V_t, Q_t, R_t, q_t, r_t, c_t\})$$

generate $\tau_{i+1} = \{\bar{s}^{i+1}_t, \pi^*_t(\bar{s}^{i+1}_t)\}$, $\bar{s}^{i+1}_{t+1} = f(\bar{s}^i_t, \pi^*_t(\bar{s}^i_t))$