Computer Engineering Department

Reinforcement Learning: Model Based RL

Mohammad Hossein Rohban, Ph.D.

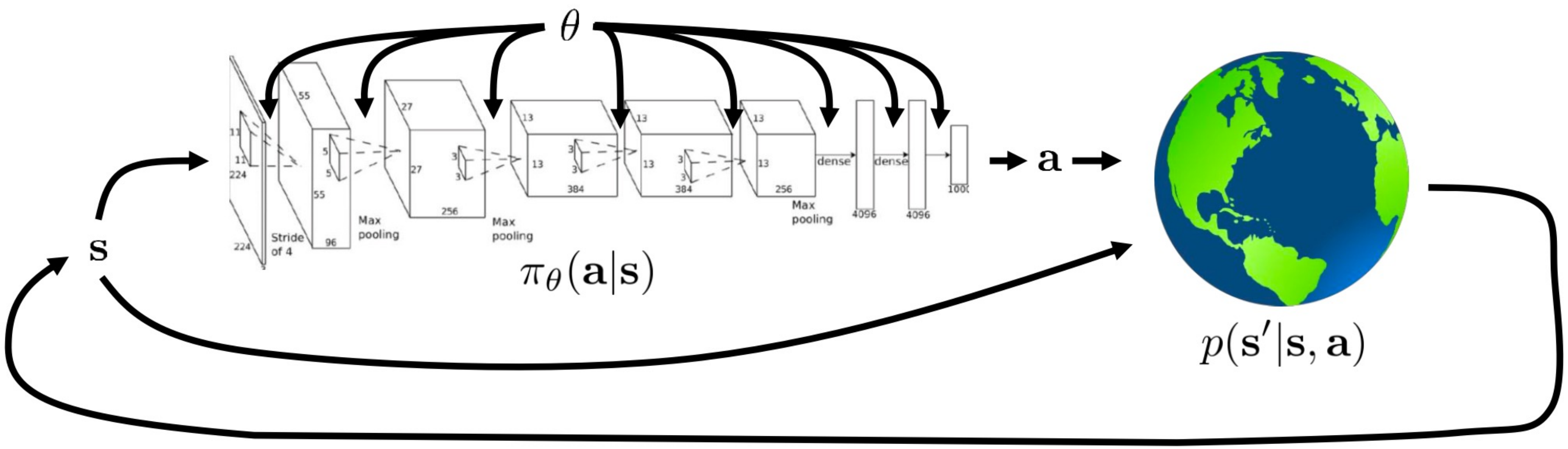
Spring 2025

Courtesy: Most of slides are adopted from CS 285 Berkeley.

Overview

- Introduction to model-based reinforcement learning
- What if we know the dynamics? How can we make decisions?
- Stochastic optimization methods
- Monte Carlo tree search (MCTS)
- Trajectory optimization
- Goal: Understand how we can perform planning with known dynamics models in discrete and continuous spaces

Recap: Model-Free RL

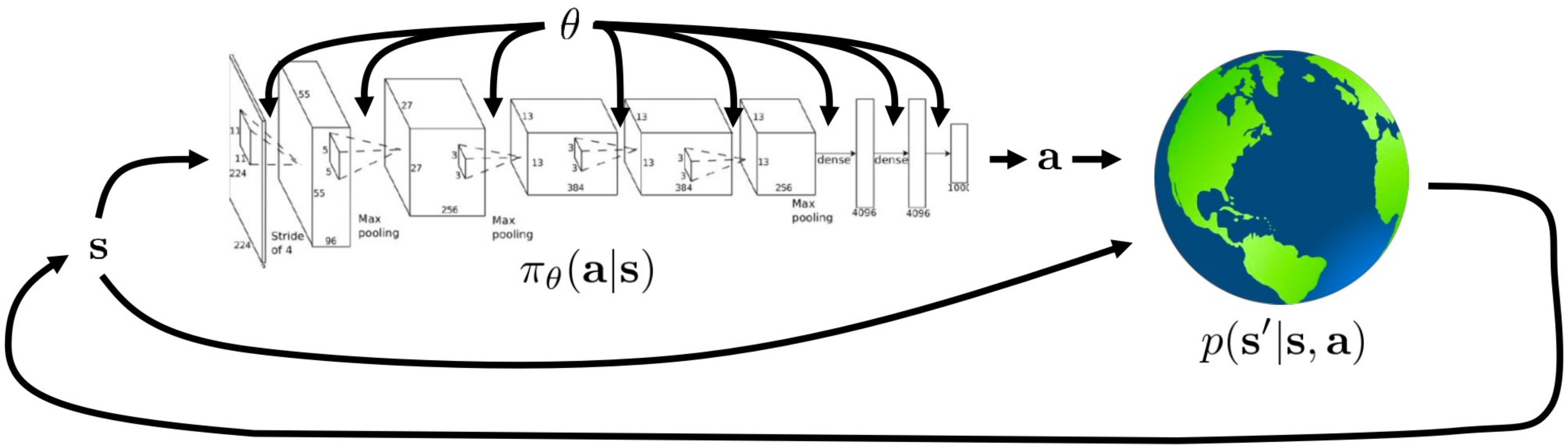


$$p_\theta(\mathbf{s}_1, \mathbf{a}_1, \ldots, \mathbf{s}_T, \mathbf{a}_T) = p(\mathbf{s}_1) \prod_{t=1}^{T} \pi_\theta(\mathbf{a}_t|\mathbf{s}_t) p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$$

$$\underbrace{p_\theta(\mathbf{s}_1, \mathbf{a}_1, \ldots, \mathbf{s}_T, \mathbf{a}_T)}_{\pi_\theta(\tau)}$$

$$\theta^\star = \arg\max_\theta E_{\tau \sim p_\theta(\tau)} \left[\sum_t r(\mathbf{s}_t, \mathbf{a}_t) \right]$$

Recap: Model-Free RL



$$\underbrace{p_\theta(\mathbf{s}_1,\mathbf{a}_1,\ldots,\mathbf{s}_T,\mathbf{a}_T)}_{\pi_\theta(\tau)} = p(\mathbf{s}_1)\prod_{t=1}^{T}\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)p(\mathbf{s}_{t+1}|\mathbf{s}_t,\mathbf{a}_t)$$

assume this is unknown

don't even attempt to learn it

$$\theta^\star = \arg\max_\theta E_{\tau\sim p_\theta(\tau)}\left[\sum_t r(\mathbf{s}_t,\mathbf{a}_t)\right]$$

What if we knew the transition dynamics?

- Often we do know the dynamics
  - Games (e.g., Atari games, chess, Go)
  - Easily modeled systems (e.g., navigating a car)
  - Simulated environments (e.g., simulated robots, video games)
- Often we can learn the dynamics
  - System identification – fit unknown parameters of a known model
  - Learning – fit a general-purpose model to observed transition data

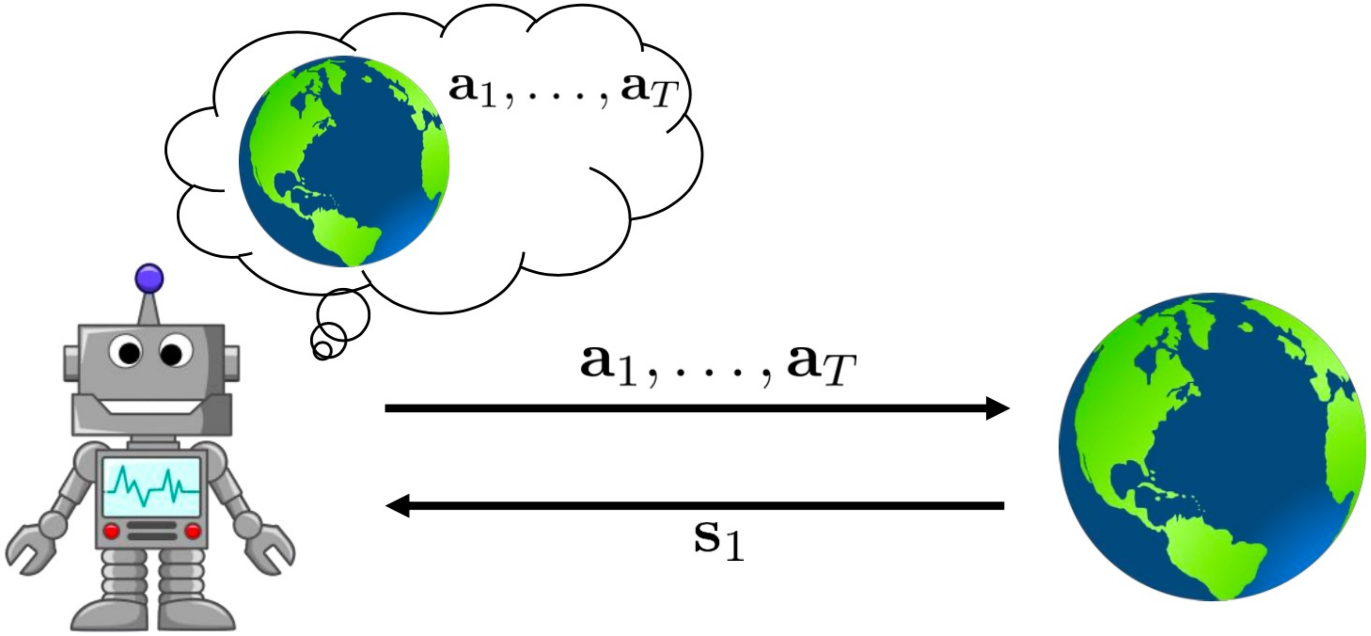
Does knowing the dynamics make things easier?

Often, yes!

Model-based RL

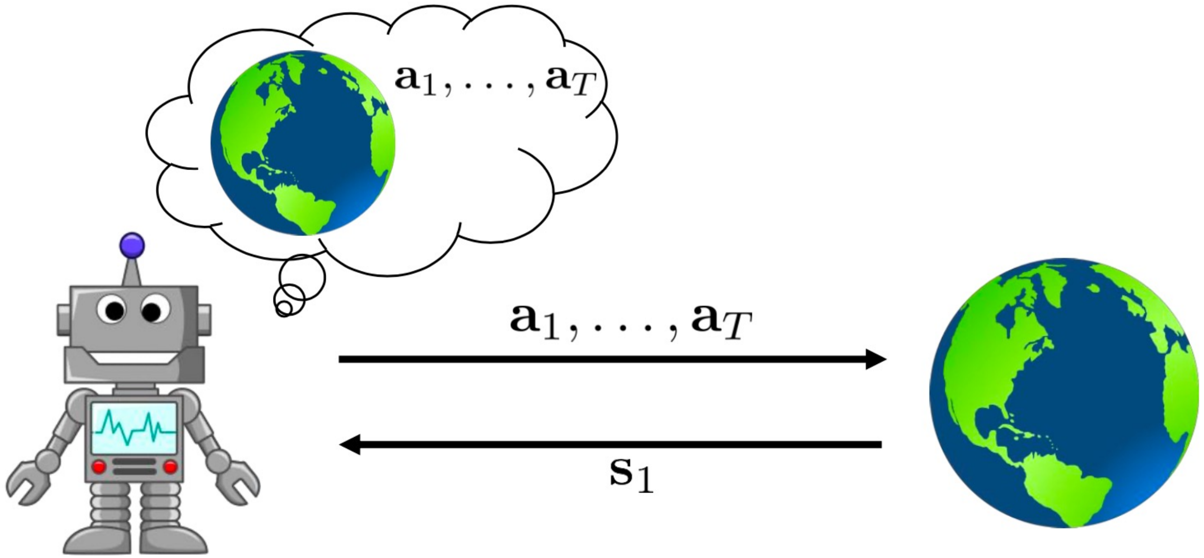
- Model-based reinforcement learning: learn the transition dynamics, then figure out how to choose actions.
- Today: how can we make decisions if we know the dynamics?
  - a. How can we choose actions under perfect knowledge of the system dynamics?
  - b. Optimal control, trajectory optimization, planning

The deterministic case



$$\mathbf{a}_1, \ldots, \mathbf{a}_T = \arg \max_{\mathbf{a}_1, \ldots, \mathbf{a}_T} \sum_{t=1}^{T} r(\mathbf{s}_t, \mathbf{a}_t) \text{ s.t. } \mathbf{a}_{t+1} = f(\mathbf{s}_t, \mathbf{a}_t)$$

The stochastic open-loop case



$$p_\theta(\mathbf{s}_1, \ldots, \mathbf{s}_T | \mathbf{a}_1, \ldots, \mathbf{a}_T) = p(\mathbf{s}_1) \prod_{t=1}^{T} p(\mathbf{s}_{t+1} | \mathbf{s}_t, \mathbf{a}_t)$$

$$\mathbf{a}_1, \ldots, \mathbf{a}_T = \arg \max_{\mathbf{a}_1, \ldots, \mathbf{a}_T} E\left[\sum_t r(\mathbf{s}_t, \mathbf{a}_t) | \mathbf{a}_1, \ldots, \mathbf{a}_T\right]$$

why is this suboptimal?

The stochastic open-loop case

کری می خواست به عیادت بیماری برود.اندیشید که هنگام احوال پرسی ممکن است صدای اورانشنوم وپاسخی ناشایسته بدهم.ازاین رودرپی چاره برآمدوبالاخره باخود گفت:بهتراست پرسشهارا پیش ازرفتن بسنجم وپاسخ رانیزبرآورد کنم تادچاراشتباه نشوم.

بنابراین پرسشهای خودراچنین پیش بینی کرد:

-ابتداازاومی پرسم حالت بهتراست؟ اوخواهد گفت "آری" من درجواب می گویم:خدا را شکر

-بعدازاومی پرسم چه خورده ای؟ لابد نام غذایی راخواهد آورد.من می گویم گوارا باد.

-درپایان می پرسم پزشکت کیست؟ نام پزشکی رامی گویدومن پاسخ می دهم:مقدمش مبارک باد.

....

چون به خانه ی بیماررسید همان گونه که ازپیش آماده شده بودبه احوال پرسی پرداخت:

-کر گفت:"چگونه ای؟"

بیمارگفت: مُردم

کر گفت: خدارا شکر

بیمارازاین سخن بیجا برآشفت.

-بعدازآن پرسید:"چه خورده ای؟"

بیمارگفت: زهر

کر گفت: گواراباد.داروی خوبی است.

بیمار ازاین پاسخ نیزبیشتربه خود پیچید.

-بعد ازآن کر گفت:" ازطبیبان کیست او کاوهمی آید به چاره پیش تو؟"

بیمار که آشفتگی وناراحتی اش به نهایت رسیده بود در پاسخ گفت:

عزرائیل می آید, برو.

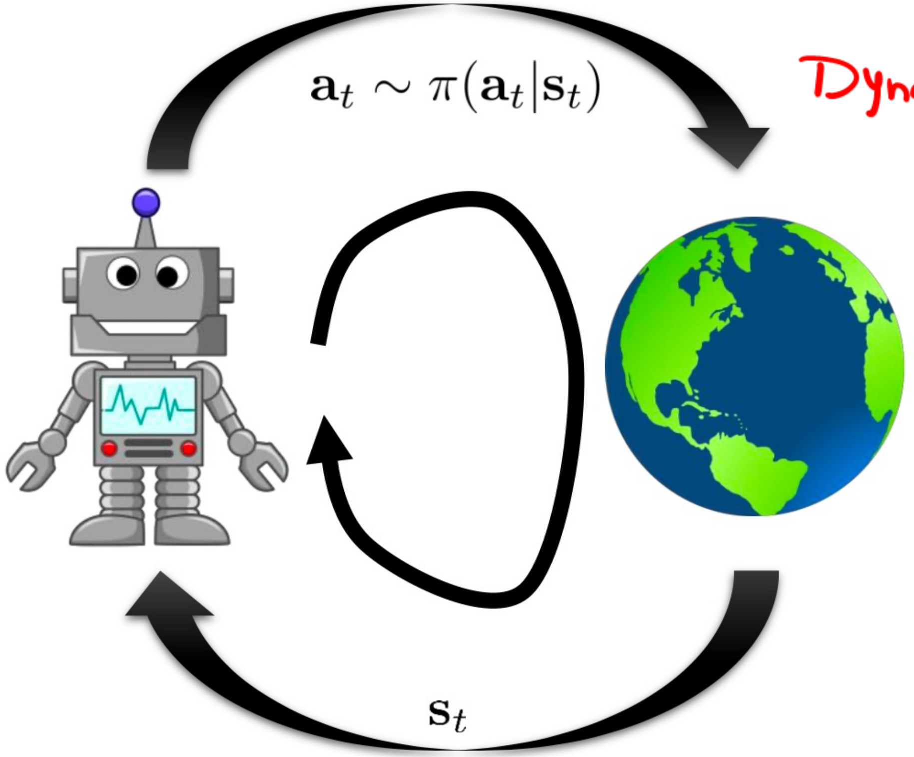
کر گفت: پایش بس مبارک.شاد شو!

open-loop vs. closed-loop case

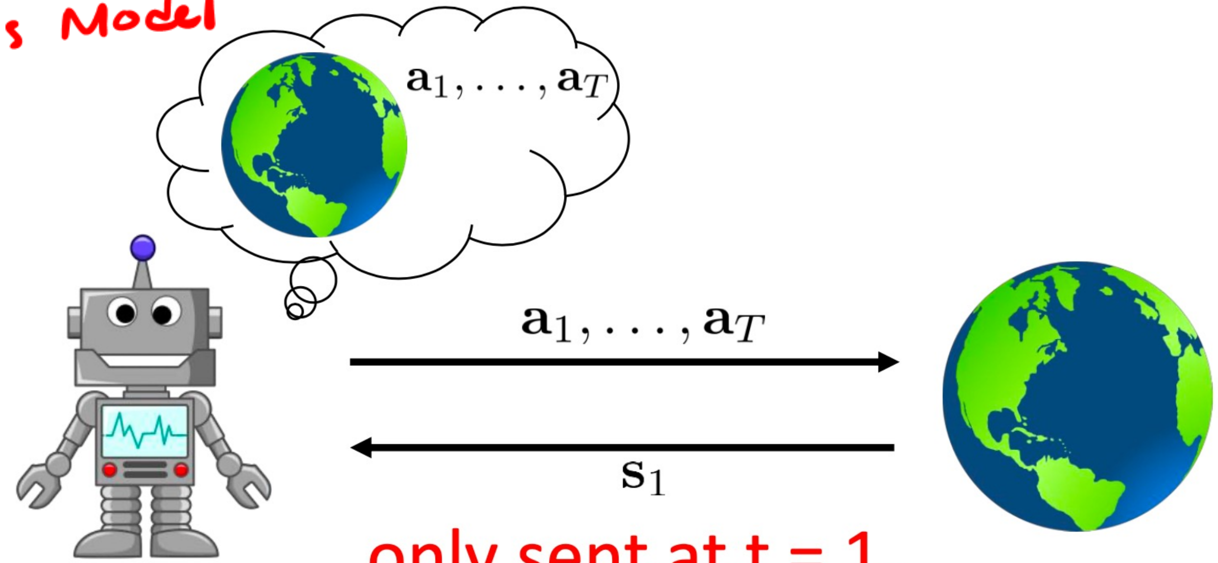
$s_{t+1} = f(s_t, a_t)$

Dynamics Model

closed-loop

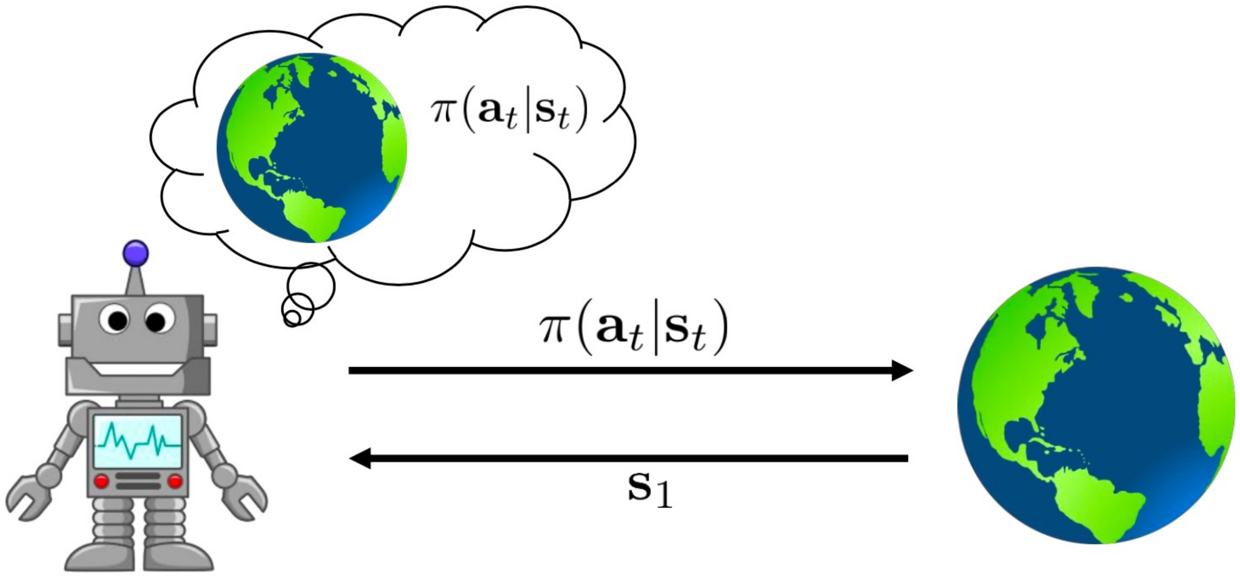


open-loop



only sent at t = 1,
then it's one-way!

The stochastic open-loop case



$$p(\mathbf{s}_1, \mathbf{a}_1, \ldots, \mathbf{s}_T, \mathbf{a}_T) = p(\mathbf{s}_1) \prod_{t=1}^{T} \pi(\mathbf{a}_t|\mathbf{s}_t) p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$$

$$\pi = \arg\max_{\pi} E_{\tau \sim p(\tau)} \left[\sum_t r(\mathbf{s}_t, \mathbf{a}_t) \right]$$

form of π?

neural net — global

time-varying linear

$\mathbf{K}_t \mathbf{s}_t + \mathbf{k}_t$ — local

Stochastic optimization

abstract away optimal control/planning:

$$\sum_{t=1}^{T} r(s_t, a_t)$$

$$s_t = f(s_{t-1}, a_{t-1})$$

$$\mathbf{a}_1, \ldots, \mathbf{a}_T = \arg\max_{\mathbf{a}_1, \ldots, \mathbf{a}_T} \underbrace{J(\mathbf{a}_1, \ldots, \mathbf{a}_T)}_{\text{don't care what this is}}$$

$$\mathbf{A} = \arg\max_{\mathbf{A}} J(\mathbf{A})$$

$$A = (a_1 \ldots a_T)$$

simplest method: guess & check "random shooting method"

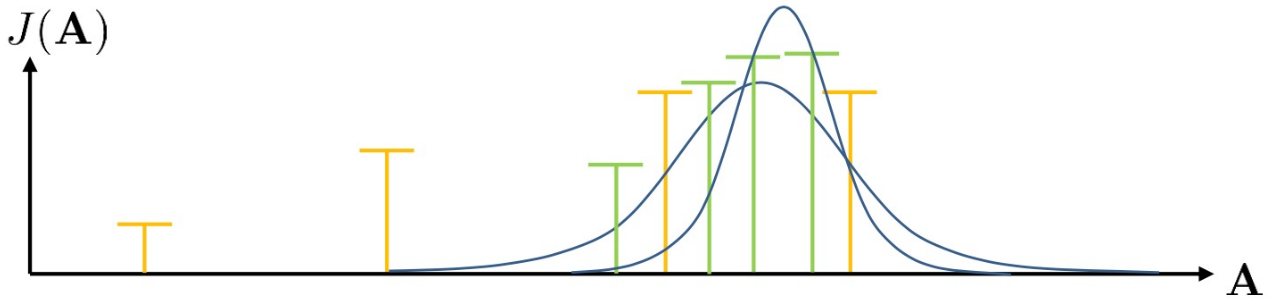
1. pick $\mathbf{A}_1, \ldots, \mathbf{A}_N$ from some distribution (e.g., uniform)
2. choose $\mathbf{A}_i$ based on $\arg\max_i J(\mathbf{A}_i)$

Cross-entropy Method (CEM)

1. pick $\mathbf{A}_1, \dots, \mathbf{A}_N$ from some distribution (e.g., uniform)

2. choose $\mathbf{A}_i$ based on $\arg\max_i J(\mathbf{A}_i)$

can we do better?



$A_1 = (a_1^{(1)} \dots a_T^{(1)})$

$A_2 = (a_1^{(2)} \dots a_T^{(2)})$

cross-entropy method with continuous-valued inputs:

1. sample $\mathbf{A}_1, \dots, \mathbf{A}_N$ from $p(\mathbf{A})$ → $\mathcal{N}(\mu, \Sigma)$
2. evaluate $J(\mathbf{A}_1), \dots, J(\mathbf{A}_N)$
3. pick the *elites* $\mathbf{A}_{i_1}, \dots, \mathbf{A}_{i_M}$ with the highest value, where $M < N$
4. refit $p(\mathbf{A})$ to the elites $\mathbf{A}_{i_1}, \dots, \mathbf{A}_{i_M}$

Pros and Cons

- Pros
  - Could be very fast (Parallelizable)
  - Extremely simple
- Cons
  - Very harsh dimensionality limit
  - Only open-loop planning

MPC

$(a_1^*, \dots a_T^*) \leftarrow \text{CEM}(f, s_1)$

$s_2 \leftarrow \text{Execute}(a_1^*)$

$(b_1^*, \dots b_T^*) \leftarrow \text{CEM}(f, s_2)$