



Trajectory Optimization

CISC 7404 - Decision Making

Steven Morad

University of Macau

Review	2
Exam	3
Planning with a Model	4
Trajectory Optimization	6
Policies	33
Value Functions	39

Review

Exam

Planning with a Model

Planning with a Model

Model-based RL vs model-free RL Pros cons

Trajectory Optimization

Trajectory Optimization

<https://www.youtube.com/watch?v=3FNPSld6Lrg>

<https://www.youtube.com/watch?v=6qj3EfRTtkE>

Trajectory Optimization

The goal of this course is to learn to maximize the discounted return

Today, we will see some methods to do this

This is my favorite lecture, because things are still simple

Some of these ideas are very old, but in the last 1-2 years we revisit them with more compute

Dreamer video

Trajectory Optimization

Want to select best decisions/actions

Best means maximize the return

How can we write how our actions influence the return?

Start with the reward, move on to return

Trajectory Optimization

Given a state s_t and action a_t , what does the reward look like?

First, write the reward function

$$R(s_{t+1})$$

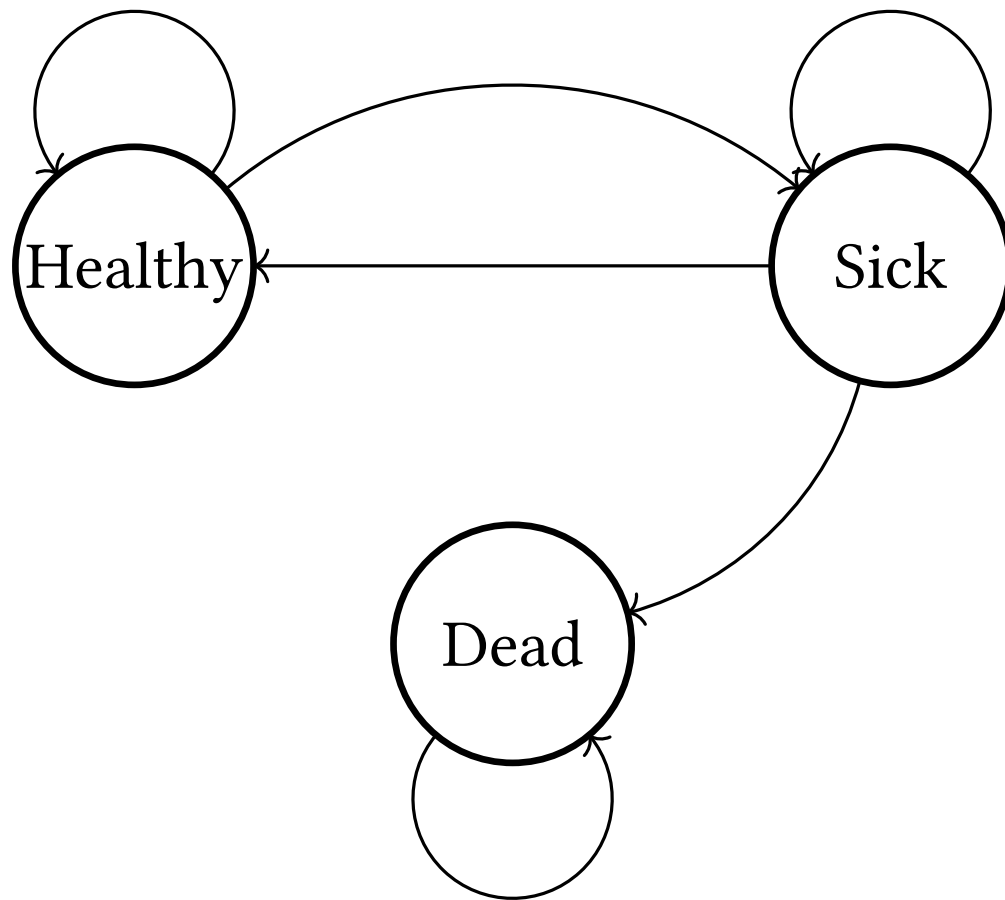
What are all possible rewards?

$$\{R(s_{t+1}) \mid s_{t+1} \in S\}$$

This is not true (example figure)

We do not consider our current state s_t or an action a_t !

Trajectory Optimization



$$\{R(s_{t+1}) \mid s \in S\} = \{R(\text{Healthy}), R(\text{Sick}), R(\text{Dead})\}$$

If $s_t = \text{Healthy}$? Can we still have a “Dead” reward?

No, because the state transition function does not allow this

$$P(\text{Dead} \mid \text{Healthy}) = 0$$

$$P(\text{Dead} \mid \text{Sick}) > 0$$

Trajectory Optimization

The reward depends on the state transition function

$$T(s_t, a_t) = \Pr(s_{t+1} \mid s_t, a_t)$$

$$s_{t+1} \sim T(s_t, a_t)$$

$$R(s_{t+1})$$

$$\{R(s_{t+1}) \mid s_{t+1} \sim T(s_t, a_t)\}$$

Trajectory Optimization

But we care about the rewards we can get with the current state/action

s_t, a_t

How can we write this? Use the state transition function

Trajectory Optimization

$$\Pr(s_{t+1} \mid s_t, a_t) \qquad \{R(s_{t+1}) \mid s_{t+1} \sim T(s_t, a_t)\}$$

How can we combine these in a meaningful way?

What if we compute the mean reward for a given action?

Question: How can we do this?

Answer: Take the expectation

$$\mathbb{E} : \underbrace{(\Omega \mapsto \mathbb{R})}_{\text{random variable}} \mapsto \mathbb{R}$$

Trajectory Optimization

$$\Pr(s_{t+1} \mid s_t, a_t) \qquad \{R(s_{t+1}) \mid s_{t+1} \sim T(s_t, a_t)\}$$

How can we combine these in a meaningful way?

What if we compute the mean reward for a given action?

Question: How can we do this?

Answer: Take the expectation

$$\mathbb{E} : \underbrace{(\Omega \mapsto \mathbb{R})}_{\text{random variable}} \mapsto \mathbb{R}$$

$$\mathbb{E}[\mathcal{X}] = \sum_{\omega \in \Omega} \mathcal{X}(\omega) \cdot \Pr(\omega)$$

Trajectory Optimization

$$\Pr(s_{t+1} \mid s_t, a_t) \qquad \{R(s_{t+1}) \mid s_{t+1} \sim T(s_t, a_t)\}$$

How can we combine these in a meaningful way?

What if we compute the mean reward for a given action?

Question: How can we do this?

Answer: Take the expectation

$$\mathbb{E} : \underbrace{(\Omega \mapsto \mathbb{R})}_{\text{random variable}} \mapsto \mathbb{R}$$

$$\mathbb{E}[\mathcal{X}] = \sum_{\omega \in \Omega} \mathcal{X}(\omega) \cdot \Pr(\omega)$$

Trajectory Optimization

$$\Pr(s_{t+1} \mid s_t, a_t) \qquad \{R(s_{t+1}) \mid s_{t+1} \sim T(s_t, a_t)\}$$

$$\mathbb{E}[\mathcal{X}] = \sum_{\omega \in \Omega} \mathcal{X}(\omega) \cdot \Pr(\omega)$$

We can write the expected reward because R is a random variable

The outcome space $\Omega = S$

$$\mathbb{E}[R(s_{t+1}) \mid s_t, a_t] = \sum_{s_{t+1} \in S} R(s_{t+1}) \cdot \Pr(s_{t+1} \mid s_t, a_t)$$

The expectation of the random variable R **conditioned** on s_t, a_t

Trajectory Optimization

$$\mathbb{E}[R(s_{t+1}) \mid s_t, a_t] = \sum_{s_{t+1} \in S} R(s_{t+1}) \cdot \Pr(s_{t+1} \mid s_t, a_t)$$

Question: How can we use this to find a good action?

Answer: We can find the action that gives us the best reward (on average)

$$\arg \max_{a_t \in A} \mathbb{E}[R(s_{t+1}) \mid s_t, a_t] = \arg \max_{a_t \in A} \sum_{s_{t+1} \in S} R(s_{t+1}) \cdot \Pr(s_{t+1} \mid s_t, a_t)$$

But in RL, we do not maximize the reward

Question: What do we maximize? **Answer:** The discounted return

Trajectory Optimization

What we have

$$\mathbb{E}[R(s_{t+1}) \mid s_t, a_t] = \sum_{s_{t+1} \in S} R(s_{t+1}) \cdot \Pr(s_{t+1} \mid s_t, a_t)$$

What we want

$$\mathbb{E}[G(\tau_n) \mid s_0, a_0, a_1, \dots, a_n] = ?$$

What we need

$$\Pr(s_t \mid s_0, a_0, a_1, a_2, \dots, a_{t-1})$$

Trajectory Optimization

$$\Pr(s_1 \mid s_0, a_0)$$

$$\Pr(s_2 \mid s_0, a_0, a_1) = \Pr(s_2 \mid s_1, a_1) \Pr(s_1 \mid s_0, a_0)$$

$$\begin{aligned} \Pr(s_t \mid s_0, a_0, a_1, \dots, a_t) &= \Pr(s_t \mid s_{t-1}) \dots \Pr(s_2 \mid s_1, a_1) \Pr(s_1 \mid s_0, a_0) \\ &= \end{aligned}$$

Trajectory Optimization

$$\mathbb{E}[G(\boldsymbol{\tau}_n) \mid s_0, a_0, a_1, \dots, a_n] = ?$$

Plug in definition of discounted return

$$\mathbb{E}[G(\boldsymbol{\tau}) \mid s_0, a_0, a_1, \dots] = \sum_{t=0}^{\infty} \gamma^t \mathbb{E}[R(s_{t+1})]$$

Trajectory Optimization

Goal: Given an initial state and some actions, predict the expected discounted return

Trajectory Optimization

Goal: Given an initial state and some actions, predict the expected discounted return

$$\mathbb{E}[R(s_1) \mid s_0, a_0] = \sum_{s_1 \in S} R(s_1) \Pr(s_1 \mid s_0, a_0)$$

$$\mathbb{E}[R(s_2) \mid s_0, a_0, a_1] = \sum_{s_2 \in S} R(s_2) \sum_{s_1 \in S} \Pr(s_2 \mid s_1, a_1) \Pr(s_1 \mid s_0, a_0)$$

$$\mathbb{E}[R(s_{n+1}) \mid s_0, a_0, a_1, \dots, a_n] = \sum_{s_{n+1} \in S} R(s_{n+1}) \sum_{s_1, \dots, s_n \in S} \prod_{t=0}^n \Pr(s_{t+1} \mid s_t, a_t)$$

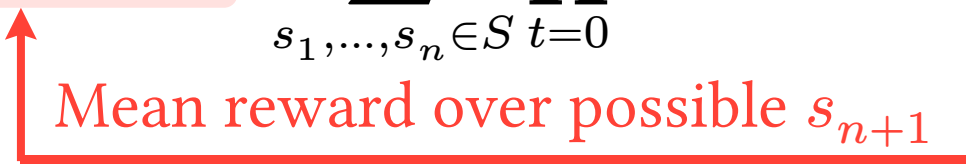
Trajectory Optimization

$$\mathbb{E}[R(s_{n+1}) \mid s_0, a_0, a_1, \dots, a_n] = \sum_{s_{n+1} \in S} R(s_{n+1}) \sum_{s_1, \dots, s_n \in S} \prod_{t=0}^n \Pr(s_{t+1} \mid s_t, a_t)$$

Trajectory Optimization

$$\mathbb{E}[R(s_{n+1}) \mid s_0, a_0, a_1, \dots, a_n] = \sum_{s_{n+1} \in S} R(s_{n+1}) \sum_{s_1, \dots, s_n \in S} \prod_{t=0}^n \Pr(s_{t+1} \mid s_t, a_t)$$

Mean reward over possible s_{n+1}

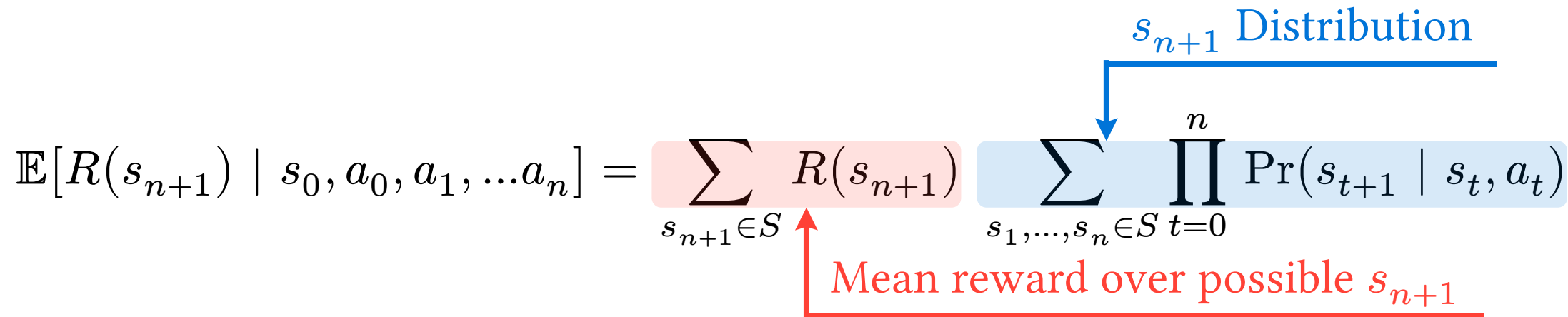


Trajectory Optimization

$$\mathbb{E}[R(s_{n+1}) \mid s_0, a_0, a_1, \dots, a_n] = \sum_{s_{n+1} \in S} R(s_{n+1}) \sum_{s_1, \dots, s_n \in S} \prod_{t=0}^n \Pr(s_{t+1} \mid s_t, a_t)$$

s_{n+1} Distribution

Mean reward over possible s_{n+1}

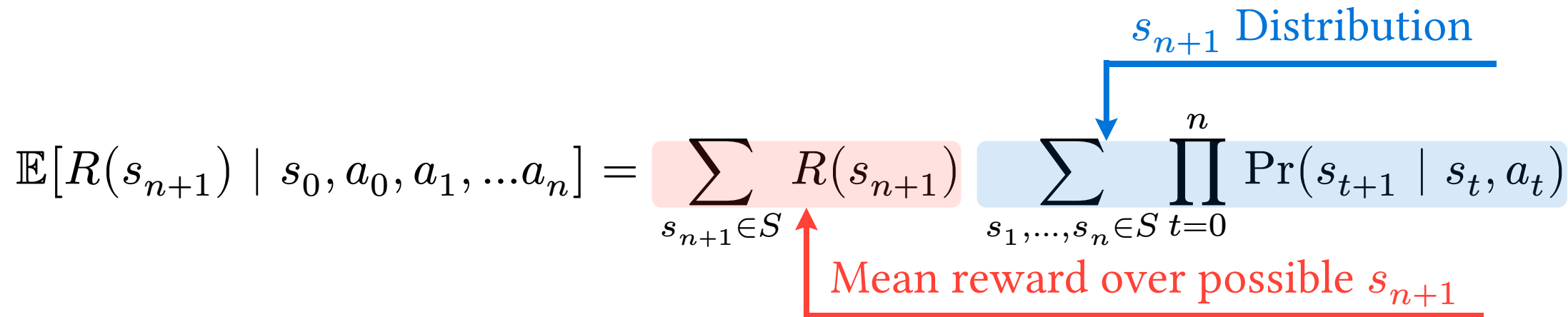


Trajectory Optimization

$$\mathbb{E}[R(s_{n+1}) \mid s_0, a_0, a_1, \dots, a_n] = \sum_{s_{n+1} \in S} R(s_{n+1}) \sum_{s_1, \dots, s_n \in S} \prod_{t=0}^n \Pr(s_{t+1} \mid s_t, a_t)$$

s_{n+1} Distribution

Mean reward over possible s_{n+1}



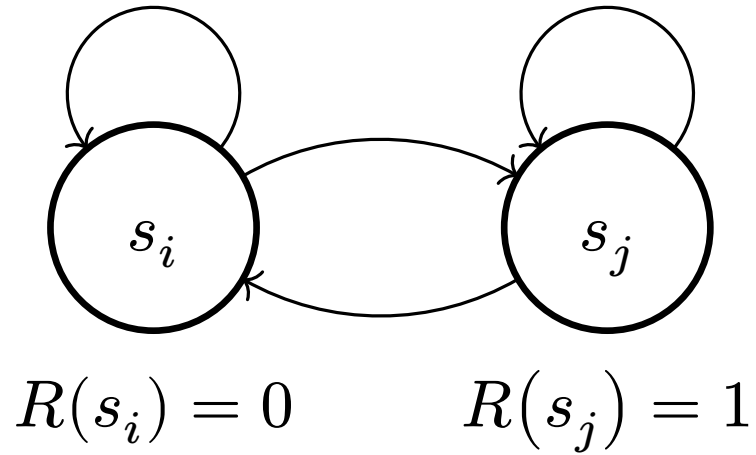
$$\mathbb{E}[R(s_{n+1}) \mid s_0, a_0, a_1, \dots, a_n] = \sum_{s_1, \dots, s_{n+1} \in S} R(s_{n+1}) \prod_{t=0}^n \Pr(s_{t+1} \mid s_t, a_t)$$

Trajectory Optimization

$$\begin{aligned}\mathbb{E}[G \mid s_0, a_0, a_1, \dots] &= \mathbb{E}[R(s_1) \mid s_0, a_0] \\ &+ \gamma \mathbb{E}[R(s_2) \mid s_0, a_0, a_1] \\ &+ \gamma^2 \mathbb{E}[R(s_3) \mid s_0, a_0, a_1, a_2] \\ &+ \dots \\ &= \sum_{s_1 \in S} R(s_1) \Pr(s_1 \mid s_0, a_0) \\ &+ \gamma \sum_{s_2 \in S} R(s_2) \sum_{s_1 \in S} \Pr(s_2 \mid s_1, a_1) \Pr(s_1 \mid s_0, a_0) \\ &+ \gamma^2 \sum_{s_3 \in S} R(s_3) \sum_{s_2 \in S} \Pr(s_3 \mid s_2, a_2) \sum_{s_1 \in S} \Pr(s_2 \mid s_1, a_1) \dots \\ &+ \dots\end{aligned}$$

Trajectory Optimization

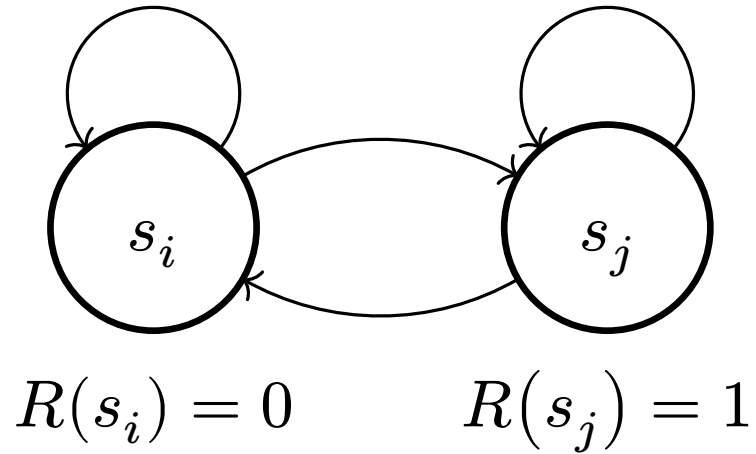
$$S = \{s_i, s_j\} \quad A = \{a_i, a_j\}$$



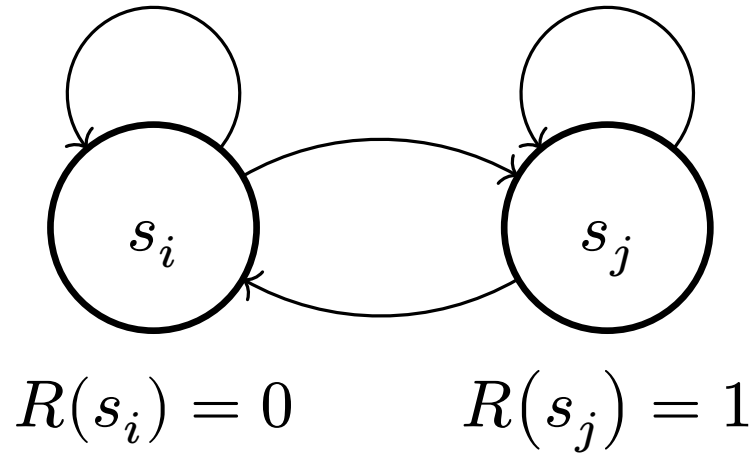
Trajectory Optimization

$$S = \{s_i, s_j\} \quad A = \{a_i, a_j\}$$

$$\Pr(s_i \mid s_i, a_i) = 0.8; \quad \Pr(s_j \mid s_i, a_i) = 0.2$$



Trajectory Optimization

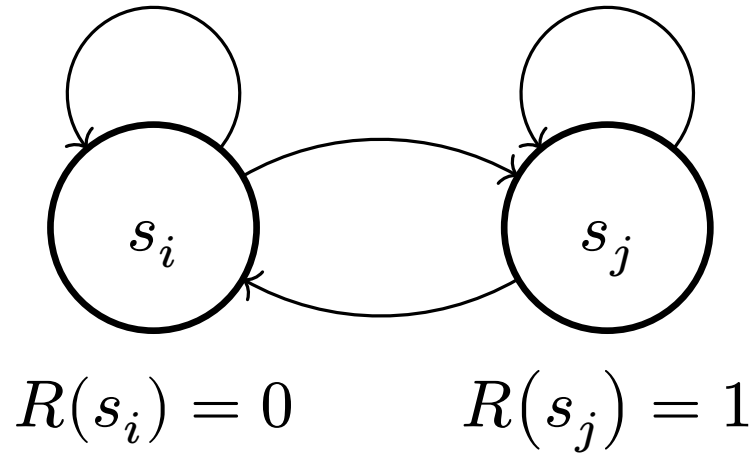


$$S = \{s_i, s_j\} \quad A = \{a_i, a_j\}$$

$$\Pr(s_i \mid s_i, a_i) = 0.8; \quad \Pr(s_j \mid s_i, a_i) = 0.2$$

$$\Pr(s_i \mid s_i, a_j) = 0.7; \quad \Pr(s_j \mid s_i, a_j) = 0.3$$

Trajectory Optimization



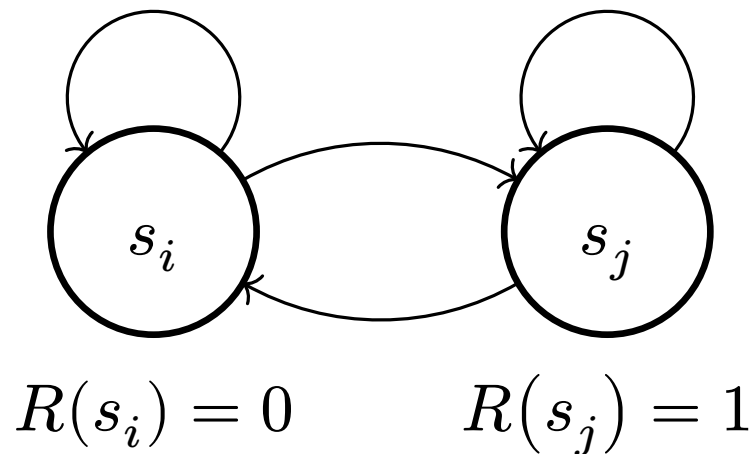
$$S = \{s_i, s_j\} \quad A = \{a_i, a_j\}$$

$$\Pr(s_i \mid s_i, a_i) = 0.8; \quad \Pr(s_j \mid s_i, a_i) = 0.2$$

$$\Pr(s_i \mid s_i, a_j) = 0.7; \quad \Pr(s_j \mid s_i, a_j) = 0.3$$

$$\Pr(s_i \mid s_j, a_i) = 0.6; \quad \Pr(s_j \mid s_j, a_i) = 0.4$$

Trajectory Optimization



$$S = \{s_i, s_j\} \quad A = \{a_i, a_j\}$$

$$\Pr(s_i \mid s_i, a_i) = 0.8; \quad \Pr(s_j \mid s_i, a_i) = 0.2$$

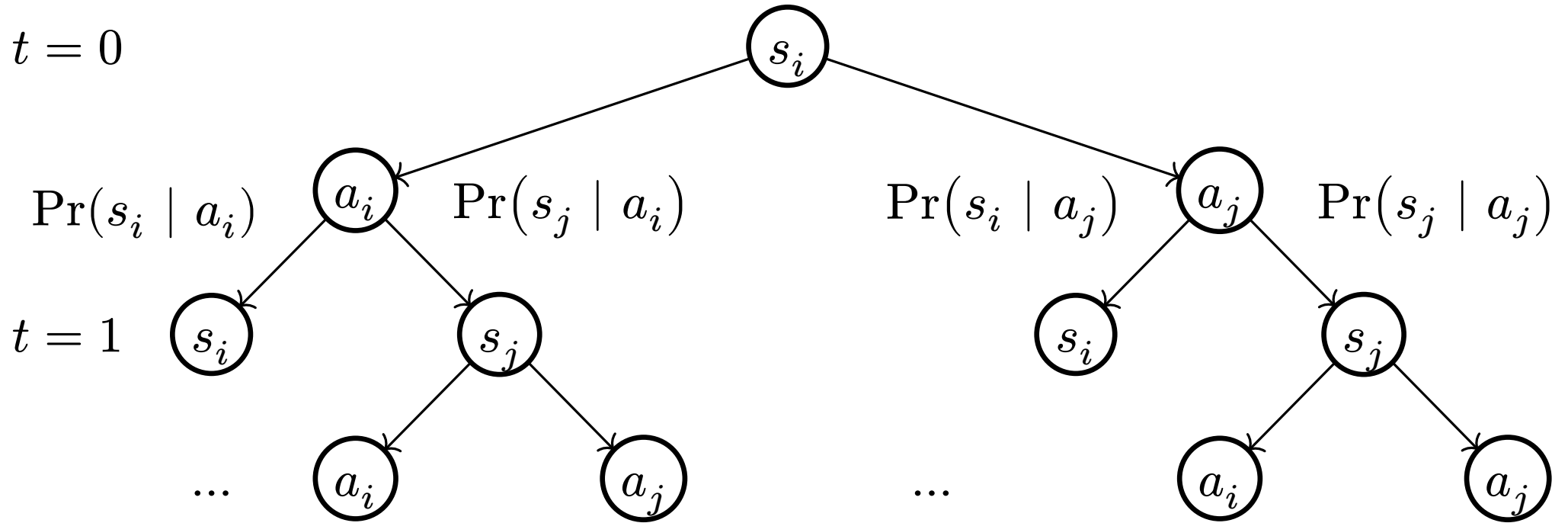
$$\Pr(s_i \mid s_i, a_j) = 0.7; \quad \Pr(s_j \mid s_i, a_j) = 0.3$$

$$\Pr(s_i \mid s_j, a_i) = 0.6; \quad \Pr(s_j \mid s_j, a_i) = 0.4$$

$$\Pr(s_i \mid s_j, a_j) = 0.1; \quad \Pr(s_j \mid s_j, a_j) = 0.9$$

Trajectory Optimization

$t = 0$

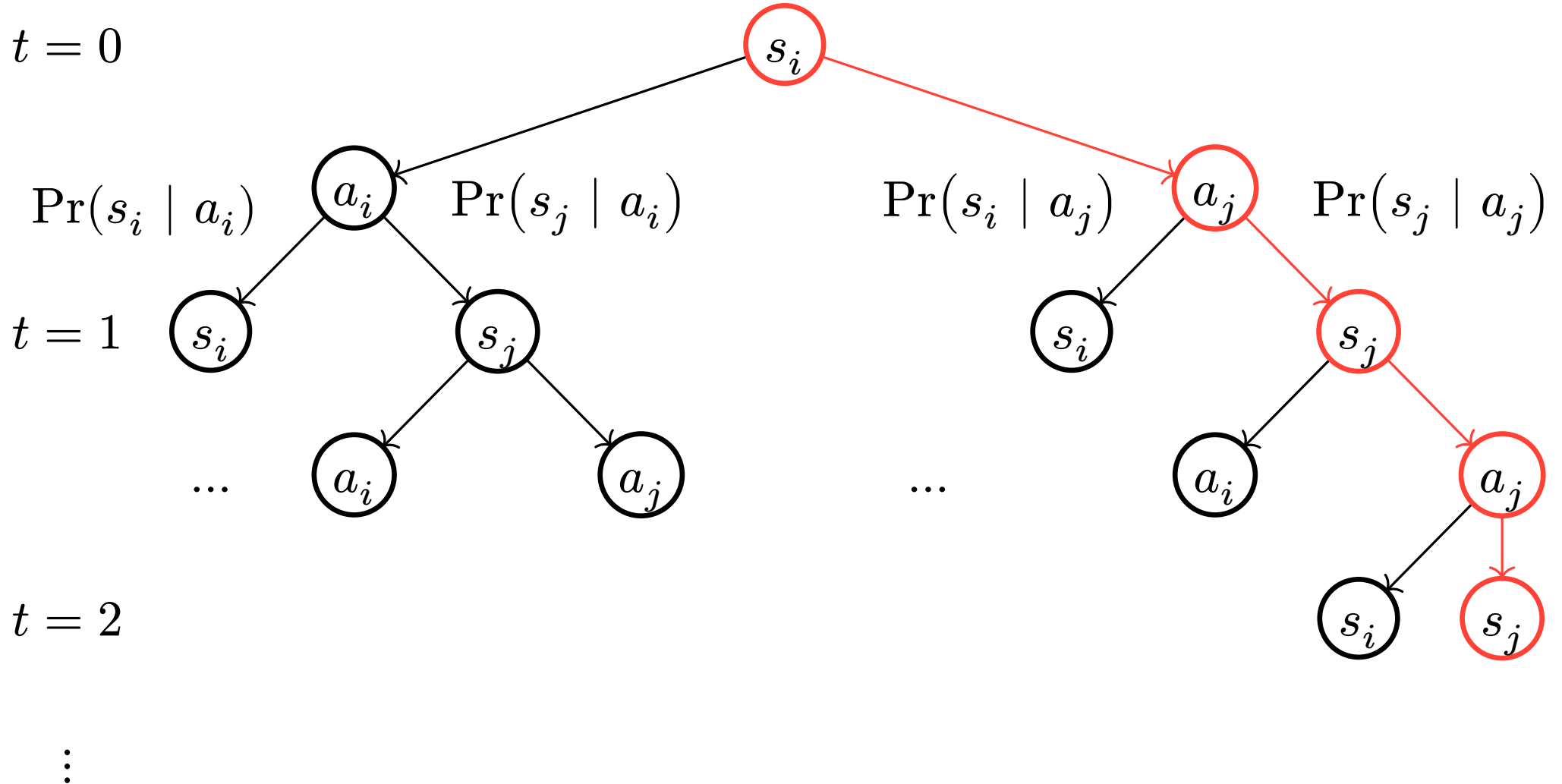


$t = 2$

\vdots

Trajectory Optimization

$t = 0$



Trajectory Optimization

$$J(a_0, a_1, \dots) = \mathbb{E}[G \mid s_0, a_0, a_1, \dots] = \sum_{t=0}^{\infty} \gamma^t \sum_{s_{t+1} \in S} R(s_{t+1}) \cdot \Pr(s_{t+1} \mid s_t, a_t)$$

This expression gives us the **expected discounted return** J

Question: How can we maximize J ?

$$\arg \max_{a_0, a_1, \dots \in A} J(a_0, a_1, \dots) = \arg \max_{a_0, a_1, \dots \in A} \sum_{t=0}^{\infty} \gamma^t \sum_{s_{t+1} \in S} R(s_{t+1}) \cdot \Pr(s_{t+1} \mid s_t, a_t)$$

Trajectory Optimization

$$\arg \max_{a_0, a_1, \dots \in A} J(a_0, a_1, \dots) = \arg \max_{a_0, a_1, \dots \in A} \sum_{t=0}^{\infty} \gamma^t \sum_{s_{t+1} \in S} R(s_{t+1}) \cdot \Pr(s_{t+1} \mid s_t, a_t)$$

In RL, we call this **trajectory optimization**

Question: What do we need to know about the problem to use trajectory optimization?

Answer:

- Must know the reward function R
- Must know the state transition function $T = \Pr(s_{t+1} \mid s_t, a_t)$

Trajectory Optimization

$$\arg \max_{a_0, a_1, \dots \in A} J(a_0, a_1, \dots) = \arg \max_{a_0, a_1, \dots \in A} \sum_{t=0}^{\infty} \gamma^t \sum_{s_{t+1} \in S} R(s_{t+1}) \cdot \Pr(s_{t+1} \mid s_t, a_t)$$

Approach: Try all possible actions sequences and pick the one with the best return

Question: Any problem?

Answer: a_0, a_1, \dots is infinite, how can we try infinitely many actions?

We can't

Trajectory Optimization

$$\arg \max_{a_0, a_1, \dots \in A} J(a_0, a_1, \dots) = \arg \max_{a_0, a_1, \dots \in A} \sum_{t=0}^{\infty} \gamma^t \sum_{s_{t+1} \in S} R(s_{t+1}) \cdot \Pr(s_{t+1} \mid s_t, a_t)$$

In trajectory optimization, we must introduce a **horizon** n

$$\begin{aligned} \arg \max_{a_0, a_1, \dots, a_n \in A} J(a_0, a_1, \dots, a_n) = \\ \arg \max_{a_0, a_1, \dots, a_n \in A} \sum_{t=0}^n \gamma^t \sum_{s_{t+1} \in S} R(s_{t+1}) \cdot \Pr(s_{t+1} \mid s_t, a_t) \end{aligned}$$

Now, we can perform a search/optimization

Trajectory Optimization

$$\arg \max_{a_0, \dots, a_n \in A} J(a_0, \dots, a_n) = \arg \max_{a_0, \dots, a_n \in A} \sum_{t=0}^n \gamma^t \sum_{s_{t+1} \in S} R(s_{t+1}) \cdot \Pr(s_{t+1} \mid s_t, a_t)$$

Question: What are the consequences of using a finite horizon n ?

Answer:

- Our model can only consider rewards n steps into the future
- Actions will **not** be optimal

In certain cases, we do not care much about the distant future

Trajectory Optimization

$$\arg \max_{a_0, \dots, a_n \in A} J(a_0, \dots, a_n) = \arg \max_{a_0, \dots, a_n \in A} \sum_{t=0}^n \gamma^t \sum_{s_{t+1} \in S} R(s_{t+1}) \cdot \Pr(s_{t+1} \mid s_t, a_t)$$

For example, we often use trajectory optimization to avoid crashes

If we can avoid any crash in 10 actions, then $n = 10$ is enough for us

One application of trajectory optimization:

<https://www.youtube.com/watch?v=6qj3EfRTtkE>

Trajectory Optimization

$$\arg \max_{a_0, \dots, a_n \in A} J(a_0, \dots, a_n) = \arg \max_{a_0, \dots, a_n \in A} \sum_{t=0}^n \gamma^t \sum_{s_{t+1} \in S} R(s_{t+1}) \cdot \Pr(s_{t+1} \mid s_t, a_t)$$

How do we optimize J in practice?

- Try all possible sequences a_0, \dots, a_n , pick the best one
- Randomly pick some sequences, pick the best one
- Use gradient descent to find a_0, \dots, a_n
 - **Note:** The state transition function and reward function must be differentiable

Policies

Policies

With trajectory optimization, we plan all of our actions at once

$$\arg \max_{a_0, a_1, \dots \in A} J(a_0, a_1, \dots) = \arg \max_{a_0, a_1, \dots, a_n \in A} \sum_{t=0}^n \gamma^t \sum_{s_{t+1} \in S} R(s_{t+1}) \cdot \Pr(s_{t+1} \mid s_t, a_t)$$

It is difficult to think about many actions and states at once

Policies

To simplify, we introduce the **policy** π with parameters $\theta \in \Theta$

$$\pi : S \times \Theta \mapsto \Delta A$$

$$\Pr(a \mid s; \theta)$$

It maps a current state to a distribution of actions

The policy determines the behavior of our agent, it is the “brain”

Policies

$$J(a_0, a_1, \dots) = \sum_{t=0}^n \gamma^t \sum_{s_{t+1} \in S} R(s_{t+1}) \cdot \Pr(s_{t+1} \mid s_t, a_t)$$

We can rewrite the expected return using the policy π and parameters θ

$$J(\theta) = \sum_{t=0}^n \gamma^t \sum_{s_{t+1} \in S} R(s_{t+1}) \cdot \Pr(s_{t+1} \mid s_t, a_t) \cdot \pi(a_t \mid s_t; \theta)$$

Policies

$$\arg \max_{a_0, a_1, \dots \in A} J(a_0, a_1, \dots) = \arg \max_{a_0, a_1, \dots, a_n \in A} \sum_{t=0}^n \gamma^t \sum_{s_{t+1} \in S} R(s_{t+1}) \cdot \Pr(s_{t+1} \mid s_t, a_t)$$

In controls and robotics, we call this **model-predictive control** (MPC)

Where do we use trajectory optimization/MPC?

<https://www.youtube.com/watch?v=Kf9WDqYKYQQ>

Policies

Trajectory optimization is expensive

The optimization process requires us to simulate thousands/millions of possible trajectories

However, as GPUs get faster these methods become more interesting

TODO: Visualization

TODO: What is the state transition function

Value Functions
