# Algorithms

## CISC 7404 - Decision Making

Steven Morad

University of Macau

# Quiz results on moodle

Quiz results on moodle

If you have no score, come see me

Quiz results on moodle

If you have no score, come see me

Mean score is $\frac{3.37}{4} \approx 84\%$

Quiz results on moodle

If you have no score, come see me

Mean score is $\frac{3.37}{4} \approx 84\%$

You did better than expected!

Quiz results on moodle

If you have no score, come see me

Mean score is $\frac{3.37}{4} \approx 84\%$

You did better than expected!

If mean course score is $> 80\%$ but you understand the material it is ok

Quiz results on moodle

If you have no score, come see me

Mean score is $\frac{3.37}{4} \approx 84\%$

You did better than expected!

If mean course score is $> 80\%$ but you understand the material it is ok

I will not decrease total score

Quiz results on moodle

If you have no score, come see me

Mean score is $\frac{3.37}{4} \approx 84\%$

You did better than expected!

If mean course score is $> 80\%$ but you understand the material it is ok

I will not decrease total score

Do not forget individual participation grade!

# Review

# Review

Diffusion models

# Review

Diffusion models

https://arxiv.org/pdf/2006.11239

# Algorithms

# Algorithms

# Algorithms

Our goal is to maximize the discounted return

# Algorithms

Our goal is to maximize the discounted return

Take actions in the MDP to maximize the discounted return

# Algorithms

Our goal is to maximize the discounted return

Take actions in the MDP to maximize the discounted return

We introduce a **policy** to select actions

# Algorithms

Our goal is to maximize the discounted return

Take actions in the MDP to maximize the discounted return

We introduce a **policy** to select actions

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

# Algorithms

Our goal is to maximize the discounted return

Take actions in the MDP to maximize the discounted return

We introduce a **policy** to select actions

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

The policy is the "brain" of the agent

# Algorithms

Our goal is to maximize the discounted return

Take actions in the MDP to maximize the discounted return

We introduce a **policy** to select actions

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

The policy is the "brain" of the agent

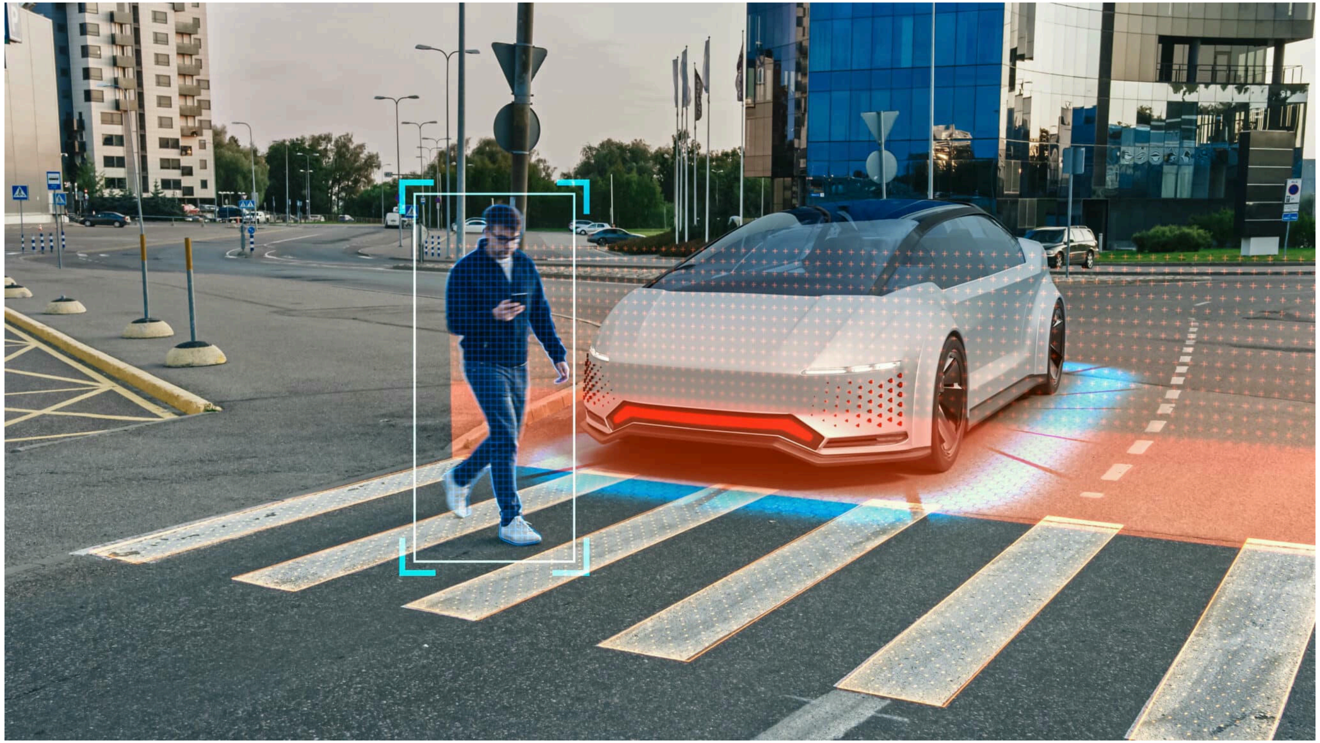
It makes decisions for the agent

# Algorithms

Policies can be good, bad, or even human!

# Algorithms

Policies can be good, bad, or even human!



# Algorithms

We use **algorithms** to find good policies

# Algorithms

We use **algorithms** to find good policies

**Question:** What makes a policy good?

# Algorithms

We use **algorithms** to find good policies

**Question:** What makes a policy good?

**Answer:** It achieves a large discounted return

# Algorithms

We use **algorithms** to find good policies

**Question:** What makes a policy good?

**Answer:** It achieves a large discounted return

Almost all the algorithms we learn in this course have guarantees

# Algorithms

We use **algorithms** to find good policies

**Question:** What makes a policy good?

**Answer:** It achieves a large discounted return

Almost all the algorithms we learn in this course have guarantees

That is, if you train long enough, your policy will become optimal

# Algorithms

We use **algorithms** to find good policies

**Question:** What makes a policy good?

**Answer:** It achieves a large discounted return

Almost all the algorithms we learn in this course have guarantees

That is, if you train long enough, your policy will become optimal

The policy is guaranteed to maximize the discounted return

# Algorithms

Today, we will derive the **trajectory optimization** algorithm

# Algorithms

Today, we will derive the **trajectory optimization** algorithm

This algorithm is old, and does not require deep learning

# Algorithms

Today, we will derive the **trajectory optimization** algorithm

This algorithm is old, and does not require deep learning

These ideas appear in classical robotics and control theory

# Algorithms

Today, we will derive the **trajectory optimization** algorithm

This algorithm is old, and does not require deep learning

These ideas appear in classical robotics and control theory

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

# Algorithms

There are two classes of algorithms

# Algorithms

There are two classes of algorithms

**Model-based**

# Algorithms

There are two classes of algorithms

## Model-based

We know $\mathrm{Tr}(s_{t+1} \mid s_t, a_t)$

# Algorithms

There are two classes of algorithms

**Model-based**

We know $\mathrm{Tr}(s_{t+1} \mid s_t, a_t)$

Cheap to train, expensive to use

# Algorithms

There are two classes of algorithms

**Model-based**

We know $\mathrm{Tr}(s_{t+1} \mid s_t, a_t)$

Cheap to train, expensive to use

Closer to traditional control theory

# Algorithms

There are two classes of algorithms

**Model-based**

We know $\mathrm{Tr}(s_{t+1} \mid s_t, a_t)$

Cheap to train, expensive to use

Closer to traditional control theory

**Model-free**

# Algorithms

There are two classes of algorithms

**Model-based**

We know $\mathrm{Tr}(s_{t+1} \mid s_t, a_t)$

Cheap to train, expensive to use

Closer to traditional control theory

**Model-free**

We do not know $\mathrm{Tr}(s_{t+1} \mid s_t, a_t)$

# Algorithms

There are two classes of algorithms

**Model-based**

We know $\mathrm{Tr}(s_{t+1} \mid s_t, a_t)$

Cheap to train, expensive to use

Closer to traditional control theory

**Model-free**

We do not know $\mathrm{Tr}(s_{t+1} \mid s_t, a_t)$

Expensive to train, cheap to use

# Algorithms

There are two classes of algorithms

**Model-based**

We know $\mathrm{Tr}(s_{t+1} \mid s_t, a_t)$

Cheap to train, expensive to use

Closer to traditional control theory

**Model-free**

We do not know $\mathrm{Tr}(s_{t+1} \mid s_t, a_t)$

Expensive to train, cheap to use

Closer to deep learning

# Algorithms

There are two classes of algorithms

| **Model-based** | **Model-free** |
|---|---|
| We know $\mathrm{Tr}(s_{t+1} \mid s_t, a_t)$ | We do not know $\mathrm{Tr}(s_{t+1} \mid s_t, a_t)$ |
| Cheap to train, expensive to use | Expensive to train, cheap to use |
| Closer to traditional control theory | Closer to deep learning |

Today, we will cover a model-based algorithm called trajectory optimization

# Algorithms

There are two classes of algorithms

**Model-based**

We know $\mathrm{Tr}(s_{t+1} \mid s_t, a_t)$

Cheap to train, expensive to use

Closer to traditional control theory

**Model-free**

We do not know $\mathrm{Tr}(s_{t+1} \mid s_t, a_t)$

Expensive to train, cheap to use

Closer to deep learning

Today, we will cover a model-based algorithm called trajectory optimization

Critical part of Alpha-* methods (AlphaGo, AlphaStar, AlphaZero)

# Algorithms

Recall the discounted return, our objective for the rest of this course

# Algorithms

Recall the discounted return, our objective for the rest of this course

$$G(\boldsymbol{\tau}) = \sum_{t=0}^{\infty} \gamma^t R(s_{t+1})$$

# Algorithms

Recall the discounted return, our objective for the rest of this course

$$G(\boldsymbol{\tau}) = \sum_{t=0}^{\infty} \gamma^t R(s_{t+1}) \qquad \tau = \begin{bmatrix} s_0 & a_0 \\ s_1 & a_1 \\ \vdots & \vdots \end{bmatrix}$$

# Algorithms

Recall the discounted return, our objective for the rest of this course

$$G(\boldsymbol{\tau}) = \sum_{t=0}^{\infty} \gamma^t R(s_{t+1}) \qquad \tau = \begin{bmatrix} s_0 & a_0 \\ s_1 & a_1 \\ \vdots & \vdots \end{bmatrix}$$

We want to maximize the discounted return

$$\arg \max_{\boldsymbol{\tau}} G(\boldsymbol{\tau}) = \arg \max_{s_1, s_2, \ldots \in S} \sum_{t=0}^{\infty} \gamma^t R(s_{t+1})$$

# Algorithms

Recall the discounted return, our objective for the rest of this course

$$G(\boldsymbol{\tau}) = \sum_{t=0}^{\infty} \gamma^t R(s_{t+1}) \qquad \tau = \begin{bmatrix} s_0 & a_0 \\ s_1 & a_1 \\ \vdots & \vdots \end{bmatrix}$$

We want to maximize the discounted return

$$\arg \max_{\boldsymbol{\tau}} G(\boldsymbol{\tau}) = \arg \max_{s_1, s_2, \ldots \in S} \sum_{t=0}^{\infty} \gamma^t R(s_{t+1})$$

We want to find $\tau$ that provides the greatest discounted return

# Algorithms

$$\arg\max_{\boldsymbol{\tau}} G(\boldsymbol{\tau}) = \arg\max_{s_1, s_2, \ldots \in S} \sum_{t=0}^{\infty} \gamma^t R(s_{t+1})$$

# Algorithms

$$\arg\max_{\boldsymbol{\tau}} G(\boldsymbol{\tau}) = \arg\max_{s_1, s_2, \ldots \in S} \sum_{t=0}^{\infty} \gamma^t R(s_{t+1})$$

This objective looks simple, but $R(s_{t+1})$ hides much of the process

# Algorithms

$$\arg\max_{\boldsymbol{\tau}} G(\boldsymbol{\tau}) = \arg\max_{s_1, s_2, \ldots \in S} \sum_{t=0}^{\infty} \gamma^t R(s_{t+1})$$

This objective looks simple, but $R(s_{t+1})$ hides much of the process

To understand what is hiding, let us examine the reward function

# The Mysterious Reward

# The Mysterious Reward

Consider the reward function

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

# The Mysterious Reward

Consider the reward function

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

Perhaps we want to maximize the reward

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

# The Mysterious Reward

Consider the reward function

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

Perhaps we want to maximize the reward

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

**Question:** In state $s_t$, take action $a_t$, what is $R(s_{t+1})$ ?

# The Mysterious Reward

Consider the reward function

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

Perhaps we want to maximize the reward

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

**Question:** In state $s_t$, take action $a_t$, what is $R(s_{t+1})$ ?

**Answer:** Not sure. $R(s_{t+1})$ depends on $\mathrm{Tr}(s_{t+1} \mid s_t, a_t)$

# The Mysterious Reward

Consider the reward function

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

Perhaps we want to maximize the reward

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

**Question:** In state $s_t$, take action $a_t$, what is $R(s_{t+1})$ ?

**Answer:** Not sure. $R(s_{t+1})$ depends on $\mathrm{Tr}(s_{t+1} \mid s_t, a_t)$

Cannot know $s_{t+1}$ with certainty, only know the distribution!

# The Mysterious Reward

$s_{t+1}$ is the **outcome** of a random process

# The Mysterious Reward

$s_{t+1}$ is the **outcome** of a random process

$$s_{t+1} \sim \mathrm{Tr}(\cdot \mid s_t, a_t), \quad s_t, s_{t+1} \in S$$

# The Mysterious Reward

$s_{t+1}$ is the **outcome** of a random process

$$s_{t+1} \sim \mathrm{Tr}(\cdot \mid s_t, a_t), \quad s_t, s_{t+1} \in S$$

**Question:** What is $S$?

# The Mysterious Reward

$s_{t+1}$ is the **outcome** of a random process

$$s_{t+1} \sim \mathrm{Tr}(\cdot \mid s_t, a_t), \quad s_t, s_{t+1} \in S$$

**Question:** What is $S$?

**Answer:** State space, also the outcome space $\Omega$ of $\mathrm{Tr}$

# The Mysterious Reward

$s_{t+1}$ is the **outcome** of a random process

$$s_{t+1} \sim \mathrm{Tr}(\cdot \mid s_t, a_t), \quad s_t, s_{t+1} \in S$$

**Question:** What is $S$?

**Answer:** State space, also the outcome space $\Omega$ of Tr

$$s_{t+1} \in S \equiv \omega \in \Omega$$

# The Mysterious Reward

$s_{t+1}$ is the **outcome** of a random process

$$s_{t+1} \sim \mathrm{Tr}(\cdot \mid s_t, a_t), \quad s_t, s_{t+1} \in S$$

**Question:** What is $S$?

**Answer:** State space, also the outcome space $\Omega$ of Tr

$$s_{t+1} \in S \equiv \omega \in \Omega$$

**Question:** Ok, now what is the definition of $R$?

**Answer:**

$$R : S \mapsto \mathbb{R}$$

# The Mysterious Reward

$$s_{t+1} \sim \mathrm{Tr}(\cdot \mid s_t, a_t), \quad s_t, s_{t+1} \in S$$

# The Mysterious Reward

$$s_{t+1} \sim \mathrm{Tr}(\cdot \mid s_t, a_t), \quad s_t, s_{t+1} \in S$$

$$R : S \mapsto \mathbb{R}$$

# The Mysterious Reward

$$s_{t+1} \sim \mathrm{Tr}(\cdot \mid s_t, a_t), \quad s_t, s_{t+1} \in S$$

$$R : S \mapsto \mathbb{R}$$

If you can answer the following question, you understand the course

# The Mysterious Reward

$$s_{t+1} \sim \mathrm{Tr}(\cdot \mid s_t, a_t), \quad s_t, s_{t+1} \in S$$

$$R : S \mapsto \mathbb{R}$$

If you can answer the following question, you understand the course

**Question:** $R$ is a special kind of function, what is it?

# The Mysterious Reward

$$s_{t+1} \sim \mathrm{Tr}(\cdot \mid s_t, a_t), \quad s_t, s_{t+1} \in S$$

$$R : S \mapsto \mathbb{R}$$

If you can answer the following question, you understand the course

**Question:** $R$ is a special kind of function, what is it?

**Answer:** $R$ is a random variable!

# The Mysterious Reward

$$s_{t+1} \sim \mathrm{Tr}(\cdot \mid s_t, a_t), \quad s_t, s_{t+1} \in S$$

$$R : S \mapsto \mathbb{R}$$

If you can answer the following question, you understand the course

**Question:** $R$ is a special kind of function, what is it?

**Answer:** $R$ is a random variable!

$R : S \mapsto \mathbb{R}$

# The Mysterious Reward

$$s_{t+1} \sim \mathrm{Tr}(\cdot \mid s_t, a_t), \quad s_t, s_{t+1} \in S$$

$$R : S \mapsto \mathbb{R}$$

If you can answer the following question, you understand the course

**Question:** $R$ is a special kind of function, what is it?

**Answer:** $R$ is a random variable!

$R : S \mapsto \mathbb{R}$ $\qquad\qquad$ $S = \Omega$

# The Mysterious Reward

$$s_{t+1} \sim \mathrm{Tr}(\cdot \mid s_t, a_t), \quad s_t, s_{t+1} \in S$$

$$R : S \mapsto \mathbb{R}$$

If you can answer the following question, you understand the course

**Question:** $R$ is a special kind of function, what is it?

**Answer:** $R$ is a random variable!

$R : S \mapsto \mathbb{R}$ $\qquad S = \Omega$ $\qquad R : \Omega \mapsto \mathbb{R}$

# The Mysterious Reward

$$s_{t+1} \sim \mathrm{Tr}(\cdot \mid s_t, a_t), \quad s_t, s_{t+1} \in S$$

$$R : S \mapsto \mathbb{R}$$

If you can answer the following question, you understand the course

**Question:** $R$ is a special kind of function, what is it?

**Answer:** $R$ is a random variable!

$R : S \mapsto \mathbb{R}$ $\qquad$ $S = \Omega$ $\qquad$ $R : \Omega \mapsto \mathbb{R}$

We should write it as $\mathcal{R} : S \mapsto \mathbb{R}$

# The Mysterious Reward

$$\mathcal{R} : S \mapsto \mathbb{R}$$

# The Mysterious Reward

$$\mathcal{R} : S \mapsto \mathbb{R}$$

**Question:** What do we like to do with random variables?

# The Mysterious Reward

$$\mathcal{R} : S \mapsto \mathbb{R}$$

**Question:** What do we like to do with random variables?

**Answer:** Take the expectation!

# The Mysterious Reward

$$\mathcal{R} : S \mapsto \mathbb{R}$$

**Question:** What do we like to do with random variables?

**Answer:** Take the expectation!

**Question:** Why do we like to take the expectation of random variables?

# The Mysterious Reward

$$\mathcal{R} : S \mapsto \mathbb{R}$$

**Question:** What do we like to do with random variables?

**Answer:** Take the expectation!

**Question:** Why do we like to take the expectation of random variables?

**Answer:** It maps complex random processes to a single value, which is much easier to work with

# The Mysterious Reward

$$\mathcal{R}(s_{t+1}), \quad s_{t+1} \sim \mathrm{Tr}(\cdot \mid s_t, a_t)$$

# The Mysterious Reward

$$\mathcal{R}(s_{t+1}), \quad s_{t+1} \sim \mathrm{Tr}(\cdot \mid s_t, a_t)$$

We cannot know for certain which reward we get in the future

# The Mysterious Reward

$$\mathcal{R}(s_{t+1}), \quad s_{t+1} \sim \mathrm{Tr}(\cdot \mid s_t, a_t)$$

We cannot know for certain which reward we get in the future

But we can know the **average** future reward using the expectation

# The Mysterious Reward

$$\mathcal{R}(s_{t+1}), \quad s_{t+1} \sim \mathrm{Tr}(\cdot \mid s_t, a_t)$$

We cannot know for certain which reward we get in the future

But we can know the **average** future reward using the expectation

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

# The Mysterious Reward

$$\mathcal{R}(s_{t+1}), \quad s_{t+1} \sim \mathrm{Tr}(\cdot \mid s_t, a_t)$$

We cannot know for certain which reward we get in the future

But we can know the **average** future reward using the expectation

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

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

# The Mysterious Reward

$$\mathcal{R}(s_{t+1}), \quad s_{t+1} \sim \mathrm{Tr}(\cdot \mid s_t, a_t)$$

We cannot know for certain which reward we get in the future

But we can know the **average** future reward using the expectation

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

$$\mathbb{E}[\underbrace{\mathcal{R}(s_{t+1}) \mid s_t, a_t}_{\text{Random variable conditioned on } s_t, a_t}] = \sum_{s_{t+1} \in S} \mathcal{R}(s_{t+1}) \cdot \mathrm{Tr}(s_{t+1} \mid s_t, a_t)$$

# The Mysterious Reward

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

# The Mysterious Reward

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

As an agent, we have partial control of the future reward

# The Mysterious Reward

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

As an agent, we have partial control of the future reward

We cannot directly control the world ($s_{t+1}$)

# The Mysterious Reward

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

As an agent, we have partial control of the future reward

We cannot directly control the world ($s_{t+1}$)

But we can choose an action $a_t$ that maximizes the expected reward

# The Mysterious Reward

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

As an agent, we have partial control of the future reward

We cannot directly control the world ($s_{t+1}$)

But we can choose an action $a_t$ that maximizes the expected reward

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

# The Mysterious Reward

$$\underset{a_t \in A}{\arg\max} \, \mathbb{E}[\mathcal{R}(s_{t+1}) \mid s_t, a_t] = \underset{a_t \in A}{\arg\max} \sum_{s_{t+1} \in S} \mathcal{R}(s_{t+1}) \cdot \mathrm{Tr}(s_{t+1} \mid s_t, a_t)$$

# The Mysterious Reward

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

What does this mean in English:

# The Mysterious Reward

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

What does this mean in English:

1. Compute the probability for each outcome $s_{t+1} \in S$, for each $a_t \in A$
2. 
3. 
4. 

# The Mysterious Reward

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

What does this mean in English:

1. Compute the probability for each outcome $s_{t+1} \in S$, for each $a_t \in A$
2. Compute the reward for each possible outcome $s_{t+1} \in S$
3. 
4. 

# The Mysterious Reward

$$\underset{a_t \in A}{\arg\max}\, \mathbb{E}[\mathcal{R}(s_{t+1}) \mid s_t, a_t] = \underset{a_t \in A}{\arg\max} \sum_{s_{t+1} \in S} \mathcal{R}(s_{t+1}) \cdot \mathrm{Tr}(s_{t+1} \mid s_t, a_t)$$

What does this mean in English:

1. Compute the probability for each outcome $s_{t+1} \in S$, for each $a_t \in A$
2. Compute the reward for each possible outcome $s_{t+1} \in S$
3. Compute expected reward for $s_{t+1} \in S$, probability times reward
4.

# The Mysterious Reward

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

What does this mean in English:

1. Compute the probability for each outcome $s_{t+1} \in S$, for each $a_t \in A$
2. Compute the reward for each possible outcome $s_{t+1} \in S$
3. Compute expected reward for $s_{t+1} \in S$, probability times reward
4. Take the action $a_t \in A$ that produces the largest the expected reward

# The Mysterious Reward

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

What does this mean in English:

1. Compute the probability for each outcome $s_{t+1} \in S$, for each $a_t \in A$
2. Compute the reward for each possible outcome $s_{t+1} \in S$
3. Compute expected reward for $s_{t+1} \in S$, probability times reward
4. Take the action $a_t \in A$ that produces the largest the expected reward

**Question:** Have we seen this before?

# The Mysterious Reward

$$\underset{a_t \in A}{\arg\max}\, \mathbb{E}[\mathcal{R}(s_{t+1}) \mid s_t, a_t] = \underset{a_t \in A}{\arg\max} \sum_{s_{t+1} \in S} \mathcal{R}(s_{t+1}) \cdot \mathrm{Tr}(s_{t+1} \mid s_t, a_t)$$

What does this mean in English:

1. Compute the probability for each outcome $s_{t+1} \in S$, for each $a_t \in A$
2. Compute the reward for each possible outcome $s_{t+1} \in S$
3. Compute expected reward for $s_{t+1} \in S$, probability times reward
4. Take the action $a_t \in A$ that produces the largest the expected reward

**Question:** Have we seen this before?

**Answer:** Bandits!

# The Mysterious Reward

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

What does this mean in English:

1. Compute the probability for each outcome $s_{t+1} \in S$, for each $a_t \in A$
2. Compute the reward for each possible outcome $s_{t+1} \in S$
3. Compute expected reward for $s_{t+1} \in S$, probability times reward
4. Take the action $a_t \in A$ that produces the largest the expected reward

**Question:** Have we seen this before?

**Answer:** Bandits!

$$\arg\max_{a \in \{1 \dots k\}} \mathbb{E}[\mathcal{X}_a]$$

# The Mysterious Reward

$$\underset{a_t \in A}{\arg\max} \, \mathbb{E}[\mathcal{R}(s_{t+1}) \mid s_t, a_t] = \underset{a_t \in A}{\arg\max} \sum_{s_{t+1} \in S} \mathcal{R}(s_{t+1}) \cdot \mathrm{Tr}(s_{t+1} \mid s_t, a_t)$$

But earlier, we said that algorithms provide a policy $\pi$

# The Mysterious Reward

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

But earlier, we said that algorithms provide a policy $\pi$

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

# The Mysterious Reward

$$\underset{a_t \in A}{\arg\max}\, \mathbb{E}[\mathcal{R}(s_{t+1}) \mid s_t, a_t] = \underset{a_t \in A}{\arg\max} \sum_{s_{t+1} \in S} \mathcal{R}(s_{t+1}) \cdot \mathrm{Tr}(s_{t+1} \mid s_t, a_t)$$

But earlier, we said that algorithms provide a policy $\pi$

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

So we can turn this equation into a policy

# The Mysterious Reward

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

But earlier, we said that algorithms provide a policy $\pi$

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

So we can turn this equation into a policy

$$\pi(a_t \mid s_t; \theta) = \Pr(a_t \mid s_t; \theta) = \begin{cases} 1 \text{ if } a_t = \arg\max_{a_t \in A} \mathbb{E}[\mathcal{R}(s_{t+1}) \mid s_t, a_t, \theta] \\ 0 \text{ otherwise} \end{cases}$$

# The Mysterious Reward

$$\underset{a_t \in A}{\arg\max} \, \mathbb{E}[\mathcal{R}(s_{t+1}) \mid s_t, a_t] = \underset{a_t \in A}{\arg\max} \sum_{s_{t+1} \in S} \mathcal{R}(s_{t+1}) \cdot \mathrm{Tr}(s_{t+1} \mid s_t, a_t)$$

But earlier, we said that algorithms provide a policy $\pi$

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

So we can turn this equation into a policy

$$\pi(a_t \mid s_t; \theta) = \Pr(a_t \mid s_t; \theta) = \begin{cases} 1 \text{ if } a_t = \arg\max_{a_t \in A} \mathbb{E}[\mathcal{R}(s_{t+1}) \mid s_t, a_t, \theta] \\ 0 \text{ otherwise} \end{cases}$$

This policy will always act to maximize the expected reward!

# The Mysterious Reward

We figured out the mystery the reward function was hiding

# The Mysterious Reward

We figured out the mystery the reward function was hiding

We found a policy that is optimal with respect to the reward

# The Mysterious Reward

We figured out the mystery the reward function was hiding

We found a policy that is optimal with respect to the reward

**Question:** Are we done? Why or why not?

**Answer:** No, we want to maximize the discounted return, not the reward!

# The Mysterious Reward

We figured out the mystery the reward function was hiding

We found a policy that is optimal with respect to the reward

**Question:** Are we done? Why or why not?

**Answer:** No, we want to maximize the discounted return, not the reward!

We have one more thing to do

# Trajectory Optimization

# Trajectory Optimization

What we have:

# Trajectory Optimization

What we have:

Expected reward, as a function of state and action

# Trajectory Optimization

What we have:

Expected reward, as a function of state and action

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

# Trajectory Optimization

What we have:

Expected reward, as a function of state and action

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

What we want:

Expected return, as a function of initial state and actions

# Trajectory Optimization

What we have:

Expected reward, as a function of state and action

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

What we want:

Expected return, as a function of initial state and actions

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

# Trajectory Optimization

What we have:

Expected reward, as a function of state and action

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

What we want:

Expected return, as a function of initial state and actions

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

**Question:** Why depend on future actions?

# Trajectory Optimization

What we have:

Expected reward, as a function of state and action

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

What we want:

Expected return, as a function of initial state and actions

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

**Question:** Why depend on future actions?

**Answer:** Agent picks actions, optimize over actions to maximize $G$

# Trajectory Optimization

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

# Trajectory Optimization

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

**Note:** $G$ is also a random variable

# Trajectory Optimization

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

**Note:** $G$ is also a random variable

$$G : \underbrace{S^n \times A^n}_{\text{Outcome of stochastic Tr}, \pi} \mapsto \mathbb{R}$$

# Trajectory Optimization

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

**Note:** $G$ is also a random variable

$$G : \underbrace{S^n \times A^n}_{\text{Outcome of stochastic } \mathrm{Tr}, \pi} \mapsto \mathbb{R}$$

We can rewrite it curly since it is a random variable

# Trajectory Optimization

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

**Note:** $G$ is also a random variable

$$G : \underbrace{S^n \times A^n}_{\text{Outcome of stochastic Tr}, \pi} \mapsto \mathbb{R}$$

We can rewrite it curly since it is a random variable

$$\mathcal{G} : \underbrace{S^n \times A^n}_{\text{Outcome of stochastic Tr}, \pi} \mapsto \mathbb{R}$$

# Trajectory Optimization

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

**Note:** $G$ is also a random variable

$$G : \underbrace{S^n \times A^n}_{\text{Outcome of stochastic Tr}, \pi} \mapsto \mathbb{R}$$

We can rewrite it curly since it is a random variable

$$\mathcal{G} : \underbrace{S^n \times A^n}_{\text{Outcome of stochastic Tr}, \pi} \mapsto \mathbb{R}$$

Back to the problem...

# Trajectory Optimization

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

**Note:** $G$ is also a random variable

$$G : \underbrace{S^n \times A^n}_{\text{Outcome of stochastic Tr}, \pi} \mapsto \mathbb{R}$$

We can rewrite it curly since it is a random variable

$$\mathcal{G} : \underbrace{S^n \times A^n}_{\text{Outcome of stochastic Tr}, \pi} \mapsto \mathbb{R}$$

Back to the problem...

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

# Trajectory Optimization

First step, write out the return

# Trajectory Optimization

First step, write out the return

$$\mathcal{G}(\boldsymbol{\tau}) = \sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s_{t+1})$$

# Trajectory Optimization

First step, write out the return

$$\mathcal{G}(\boldsymbol{\tau}) = \sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s_{t+1})$$

Remember, we can only maximize the expectation

# Trajectory Optimization

First step, write out the return

$$\mathcal{G}(\boldsymbol{\tau}) = \sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s_{t+1})$$

Remember, we can only maximize the expectation

Take the expected value of both sides

# Trajectory Optimization

First step, write out the return

$$\mathcal{G}(\boldsymbol{\tau}) = \sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s_{t+1})$$

Remember, we can only maximize the expectation

Take the expected value of both sides

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

# Trajectory Optimization

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

# Trajectory Optimization

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

The expectation is a linear function, we can move it inside the sum

# Trajectory Optimization

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

The expectation is a linear function, we can move it inside the sum

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

# Trajectory Optimization

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

The expectation is a linear function, we can move it inside the sum

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

Expectation is linear, can factor out $\gamma$

# Trajectory Optimization

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

The expectation is a linear function, we can move it inside the sum

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

Expectation is linear, can factor out $\gamma$

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

# Trajectory Optimization

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

Write out the sum

# Trajectory Optimization

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

Write out the sum

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

$$\gamma^0 \mathbb{E}[\mathcal{R}(s_{t+1}) \mid s_0, a_0, a_1, ..., a_t] + \gamma^1 \mathbb{E}[\mathcal{R}(s_{t+1}) \mid s_0, a_0, a_1, ..., a_t] + ...$$

# Trajectory Optimization

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

Write out the sum

$$\mathbb{E}[\mathcal{G}(\boldsymbol{\tau}_n) \mid s_0, a_0, a_1, ..., a_n] =$$
$$\gamma^0 \mathbb{E}[\mathcal{R}(s_{t+1}) \mid s_0, a_0, a_1, ..., a_t] + \gamma^1 \mathbb{E}[\mathcal{R}(s_{t+1}) \mid s_0, a_0, a_1, ..., a_t] + ...$$

Rewards do not depend on future actions

# Trajectory Optimization

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

Write out the sum

$$\mathbb{E}[\mathcal{G}(\boldsymbol{\tau}_n) \mid s_0, a_0, a_1, ..., a_n] =$$
$$\gamma^0 \mathbb{E}[\mathcal{R}(s_{t+1}) \mid s_0, a_0, a_1, ..., a_t] + \gamma^1 \mathbb{E}[\mathcal{R}(s_{t+1}) \mid s_0, a_0, a_1, ..., a_t] + ...$$

Rewards do not depend on future actions

$$\mathbb{E}[\mathcal{G}(\boldsymbol{\tau}_n) \mid s_0, a_0, a_1, ..., a_n] =$$
$$\gamma^0 \mathbb{E}[\mathcal{R}(s_{t+1}) \mid s_0, a_0] + \gamma^1 \mathbb{E}[\mathcal{R}(s_{t+1}) \mid s_0, a_0, a_1] + ...$$

# Trajectory Optimization

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

$$\gamma^0 \; \mathbb{E}[\mathcal{R}(s_{t+1}) \mid s_0, a_0] + \gamma^1 \mathbb{E}[\mathcal{R}(s_{t+1}) \mid s_0, a_0, a_1] + ...$$

# Trajectory Optimization

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

$$\gamma^0 \; \mathbb{E}[\mathcal{R}(s_{t+1}) \mid s_0, a_0] + \gamma^1 \mathbb{E}[\mathcal{R}(s_{t+1}) \mid s_0, a_0, a_1] + ...$$

**Question:** Do any terms look familiar?

# Trajectory Optimization

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

$$\gamma^0 \; \mathbb{E}[\mathcal{R}(s_{t+1}) \mid s_0, a_0] + \gamma^1 \mathbb{E}[\mathcal{R}(s_{t+1}) \mid s_0, a_0, a_1] + ...$$

**Question:** Do any terms look familiar?

**Answer:** We know the expected reward from before!

# Trajectory Optimization

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

We previously found

$$\mathbb{E}[\mathcal{R}(s_{t+1}) \mid s_t, a_t]$$

# Trajectory Optimization

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

We previously found

$$\mathbb{E}[\mathcal{R}(s_{t+1}) \mid s_t, a_t]$$

Since we have $s_0, a_0$, we can compute the term for $t = 0$

# Trajectory Optimization

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

We previously found

$$\mathbb{E}[\mathcal{R}(s_{t+1}) \mid s_t, a_t]$$

Since we have $s_0, a_0$, we can compute the term for $t = 0$

$$\mathbb{E}[\mathcal{R}(s_1) \mid s_0, a_0]$$

# Trajectory Optimization

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

We previously found

$$\mathbb{E}[\mathcal{R}(s_{t+1}) \mid s_t, a_t]$$

Since we have $s_0, a_0$, we can compute the term for $t = 0$

$$\mathbb{E}[\mathcal{R}(s_1) \mid s_0, a_0]$$

Now, let's try to find $\mathcal{R}(s_2)$

# Trajectory Optimization

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

We previously found

$$\mathbb{E}[\mathcal{R}(s_{t+1}) \mid s_t, a_t]$$

Since we have $s_0, a_0$, we can compute the term for $t = 0$

$$\mathbb{E}[\mathcal{R}(s_1) \mid s_0, a_0]$$

Now, let's try to find $\mathcal{R}(s_2)$

$$\mathbb{E}[\mathcal{R}(s_2) \mid s_0, a_0, a_1]$$

# Trajectory Optimization

$$\mathbb{E}[\mathcal{R}(s_2) \mid s_0, a_0, a_1]$$

**Question:** Any problems?

# Trajectory Optimization

$$\mathbb{E}[\mathcal{R}(s_2) \mid s_0, a_0, a_1]$$

**Question:** Any problems?

**Answer:** $\mathcal{R}$ needs $s_2$, but we only have $s_0$!

# Trajectory Optimization

$$\mathbb{E}[\mathcal{R}(s_2) \mid s_0, a_0, a_1]$$

**Question:** Any problems?

**Answer:** $\mathcal{R}$ needs $s_2$, but we only have $s_0$!

For $t = 1$, the reward relies on the distribution $\mathrm{Tr}(s_1 \mid s_0, a_0)$

# Trajectory Optimization

$$\mathbb{E}[\mathcal{R}(s_2) \mid s_0, a_0, a_1]$$

**Question:** Any problems?

**Answer:** $\mathcal{R}$ needs $s_2$, but we only have $s_0$!

For $t = 1$, the reward relies on the distribution $\mathrm{Tr}(s_1 \mid s_0, a_0)$

For $t = 2$, the reward relies on $\mathrm{Tr}(s_2 \mid s_1, a_1)$ and $\mathrm{Tr}(s_1 \mid s_0, a_0)$

# Trajectory Optimization

$$\mathbb{E}[\mathcal{R}(s_2) \mid s_0, a_0, a_1]$$

**Question:** Any problems?

**Answer:** $\mathcal{R}$ needs $s_2$, but we only have $s_0$!

For $t = 1$, the reward relies on the distribution $\mathrm{Tr}(s_1 \mid s_0, a_0)$

For $t = 2$, the reward relies on $\mathrm{Tr}(s_2 \mid s_1, a_1)$ and $\mathrm{Tr}(s_1 \mid s_0, a_0)$

For $\mathcal{R}(s_{n+1})$ we need an expression for $\Pr(s_{n+1} \mid s_0, a_0, a_1, ...)$

# Trajectory Optimization

**Question:** How do we find $\Pr(s_{n+1} \mid s_0, a_0, a_1, \ldots)$?

# Trajectory Optimization

**Question:** How do we find $\Pr(s_{n+1} \mid s_0, a_0, a_1, ...)$?

**Answer:** In lecture 3 we computed the probability of a future state

# Trajectory Optimization

**Question:** How do we find $\Pr(s_{n+1} \mid s_0, a_0, a_1, ...)$?

**Answer:** In lecture 3 we computed the probability of a future state

$$\Pr(s_{n+1} \mid s_0) = \sum_{s_1, s_2, \dots s_n \in S} \prod_{t=0}^{n} \Pr(s_{t+1} \mid s_t)$$

# Trajectory Optimization

**Question:** How do we find $\Pr(s_{n+1} \mid s_0, a_0, a_1, ...)$?

**Answer:** In lecture 3 we computed the probability of a future state

$$\Pr(s_{n+1} \mid s_0) = \sum_{s_1, s_2, \ldots s_n \in S} \prod_{t=0}^{n} \Pr(s_{t+1} \mid s_t)$$

We just need to include the actions!

# Trajectory Optimization

**Question:** How do we find $\Pr(s_{n+1} \mid s_0, a_0, a_1, ...)$?

**Answer:** In lecture 3 we computed the probability of a future state

$$\Pr(s_{n+1} \mid s_0) = \sum_{s_1, s_2, ... s_n \in S} \prod_{t=0}^{n} \Pr(s_{t+1} \mid s_t)$$

We just need to include the actions!

$$\Pr(s_{n+1} \mid s_0, a_0, a_1, ..., a_{n-1}) = \sum_{s_1, s_2, ... s_n \in S} \prod_{t=0}^{n} \Pr(s_{t+1} \mid s_t, a_t)$$

# Trajectory Optimization

**Question:** How do we find $\Pr(s_{n+1} \mid s_0, a_0, a_1, \ldots)$?

**Answer:** In lecture 3 we computed the probability of a future state

$$\Pr(s_{n+1} \mid s_0) = \sum_{s_1, s_2, \ldots s_n \in S} \prod_{t=0}^{n} \Pr(s_{t+1} \mid s_t)$$

We just need to include the actions!

$$\Pr(s_{n+1} \mid s_0, a_0, a_1, \ldots, a_{n-1}) = \sum_{s_1, s_2, \ldots s_n \in S} \prod_{t=0}^{n} \Pr(s_{t+1} \mid s_t, a_t)$$

We are predicting the future state of an MDP

# Trajectory Optimization

$$\Pr(s_n \mid s_0, a_0, a_1, ..., a_{n-1}) = \sum_{s_1, s_2, ... s_{n-1} \in S} \prod_{t=0}^{n-1} \Pr(s_{t+1} \mid s_t, a_t)$$

TODO write out expectation so we can plug in $R(s_t) \Pr(s_t \mid s_0, a_0, ...)$

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

# 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

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

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

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

# Trajectory Optimization

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

# Trajectory Optimization

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

# Trajectory Optimization

$s_{n+1}$ Distribution

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

Mean reward over possible $s_{n+1}$

# Trajectory Optimization

$s_{n+1}$ Distribution

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

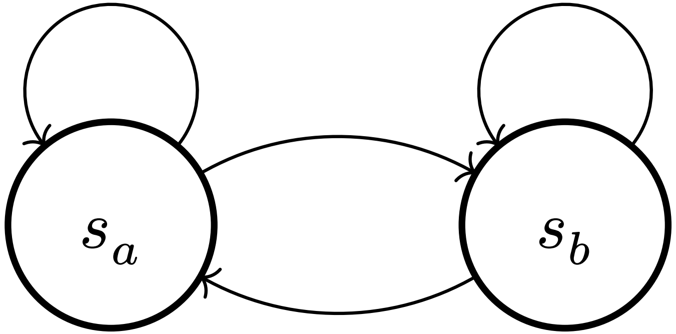
Mean reward over possible $s_{n+1}$

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

# Trajectory Optimization

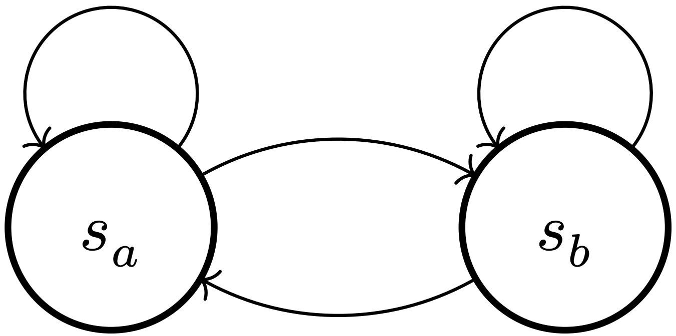
$$
\begin{aligned}
\mathbb{E}[G \mid s_0, a_0, a_1, ...] = & \quad \mathbb{E}[R(s_1) \mid s_0, a_0] \\
+\gamma & \quad \mathbb{E}[R(s_2) \mid s_0, a_0, a_1] \\
+\gamma^2 & \quad \mathbb{E}[R(s_3) \mid s_0, a_0, a_1, a_2] \\
+ & \quad ... \\
= & \quad \sum_{s_1 \in S} R(s_1) \Pr(s_1 \mid s_0, a_0) \\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

# Trajectory Optimization




$R(s_a) = 0$ $\quad\quad$ $R(s_b) = 1$

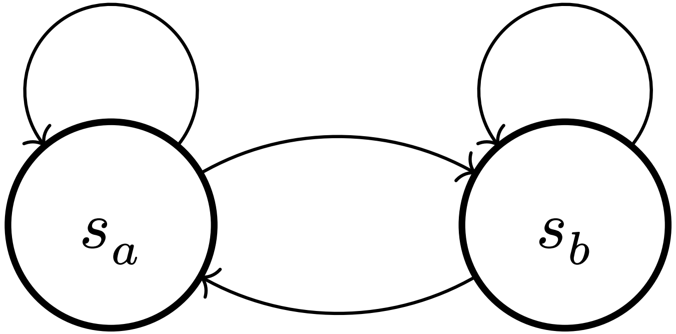
# Trajectory Optimization




$R(s_a) = 0 \qquad R(s_b) = 1$

$$S = \{s_a, s_b\} \quad A = \{a_a, a_b\}$$

# Trajectory Optimization

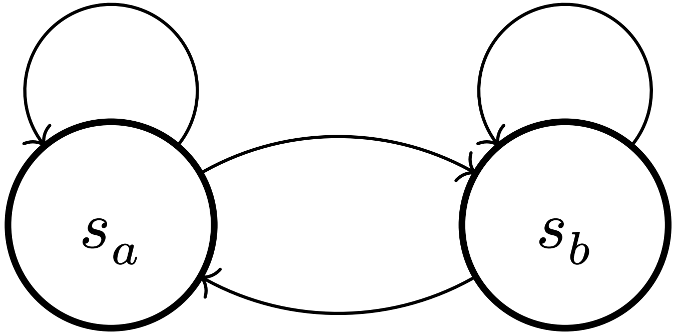



$R(s_a) = 0$ $\quad$ $R(s_b) = 1$

$$S = \{s_a, s_b\} \quad A = \{a_a, a_b\}$$

$$\Pr(s_a \mid s_a, a_a) = 0.8; \; \Pr(s_b \mid s_a, a_a) = 0.2$$

# Trajectory Optimization



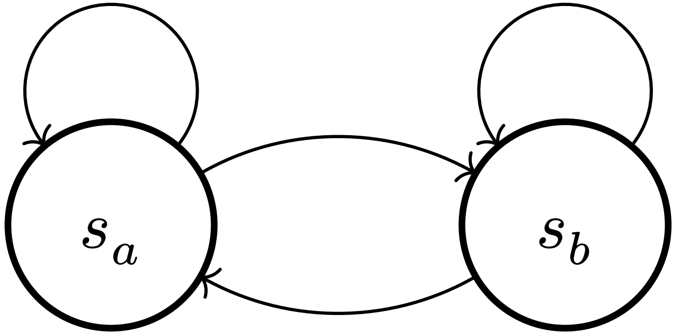

$R(s_a) = 0 \qquad R(s_b) = 1$

$$S = \{s_a, s_b\} \quad A = \{a_a, a_b\}$$

$$\Pr(s_a \mid s_a, a_a) = 0.8; \;\; \Pr(s_b \mid s_a, a_a) = 0.2$$

$$\Pr(s_a \mid s_a, a_b) = 0.7; \;\; \Pr(s_b \mid s_a, a_b) = 0.3$$

# Trajectory Optimization




$R(s_a) = 0$ $\quad$ $R(s_b) = 1$

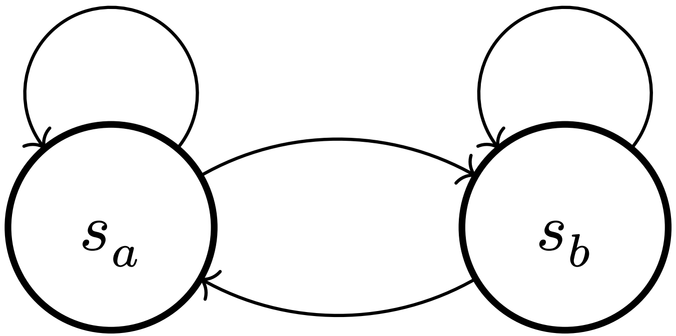
$$S = \{s_a, s_b\} \quad A = \{a_a, a_b\}$$

$$\Pr(s_a \mid s_a, a_a) = 0.8; \;\; \Pr(s_b \mid s_a, a_a) = 0.2$$

$$\Pr(s_a \mid s_a, a_b) = 0.7; \;\; \Pr(s_b \mid s_a, a_b) = 0.3$$

$$\Pr(s_a \mid s_b, a_a) = 0.6; \;\; \Pr(s_b \mid s_a, a_a) = 0.4$$

# Trajectory Optimization



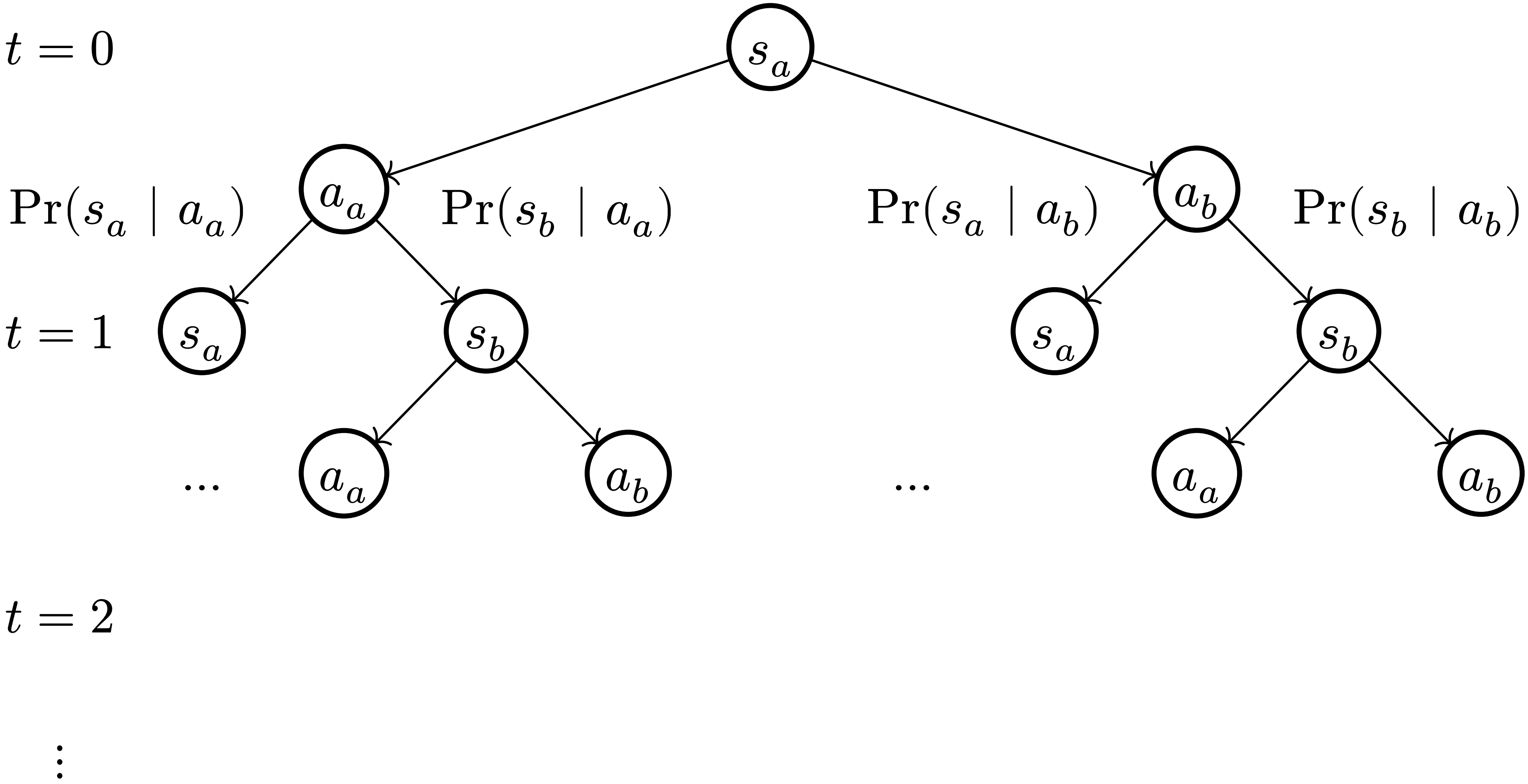

$R(s_a) = 0$ $\quad$ $R(s_b) = 1$

$$S = \{s_a, s_b\} \quad A = \{a_a, a_b\}$$

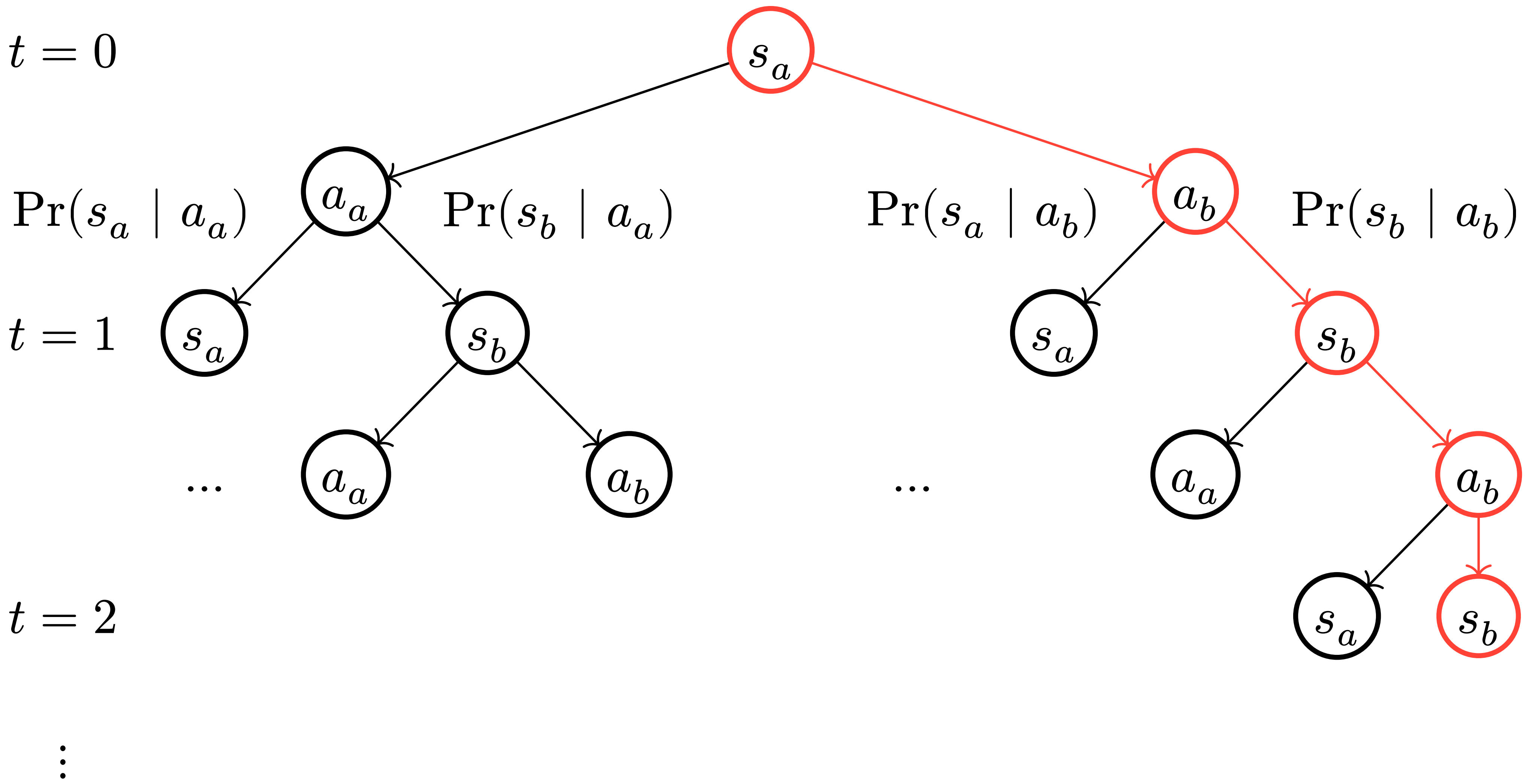
$$\Pr(s_a \mid s_a, a_a) = 0.8; \ \Pr(s_b \mid s_a, a_a) = 0.2$$
$$\Pr(s_a \mid s_a, a_b) = 0.7; \ \Pr(s_b \mid s_a, a_b) = 0.3$$

$$\Pr(s_a \mid s_b, a_a) = 0.6; \ \Pr(s_b \mid s_a, a_a) = 0.4$$
$$\Pr(s_a \mid s_b, a_b) = 0.1; \ \Pr(s_b \mid s_a, a_b) = 0.9$$

# Trajectory Optimization



# Trajectory Optimization



# Trajectory Optimization

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

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

**Question:** How can we maximize $J$?

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

# Trajectory Optimization

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

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

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

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

**Question:** Any problem?

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

We can't

# Trajectory Optimization

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

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

$$\underset{a_0, a_1, \ldots, a_n \in A}{\arg\max}\, J(a_0, a_1, \ldots, a_n) =$$

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

Now, we can perform a search/optimization

# Trajectory Optimization

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

**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

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

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

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

How do we optimize $J$ in practice?

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

# Algorithms and Policies

# Algorithms and Policies

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

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

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

# Algorithms and Policies

To simplify, we introduce the **policy** $\pi$ 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”

# Algorithms and Policies

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

We can rewrite the expected return using the policy $\pi$ and parameters $\theta$

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

# Algorithms and Policies

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

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

# Algorithms and 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