Lecture 9: Policy Gradient Methods

Dr. Wen Fuxi

Introduction

In this lecture, we will move

- From value-based methods to policy-based methods
- From value function methods to policy function methods (or called policy gradient methods)

1. Indirect and Direct RL

2. Basic idea of policy gradient

3. Metrics to define optimal policies
   - 3.0 Influence of Initial State Distribution
   - 3.1 Metric 1: Average value
   - 3.2 Metric 2: Average reward
   - 3.3 Summary of the two metrics

4. Gradients of the metrics

5. Gradient-Ascent algorithm

6. Summary

1. Indirect and Direct RL

2. Basic idea of policy gradient

3. Metrics to define optimal policies
   - 3.0 Influence of Initial State Distribution
   - 3.1 Metric 1: Average value
   - 3.2 Metric 2: Average reward
   - 3.3 Summary of the two metrics

4. Gradients of the metrics

5. Gradient-Ascent algorithm

6. Summary

Basis of RL problems

- To find an optimal policy to maximize a weighted sum of expected return
- Subject to

  (1) analytical environment model (i.e., model-based) or

  (2) data samples from environment interaction (i.e., model-free)

Basis of RL problems

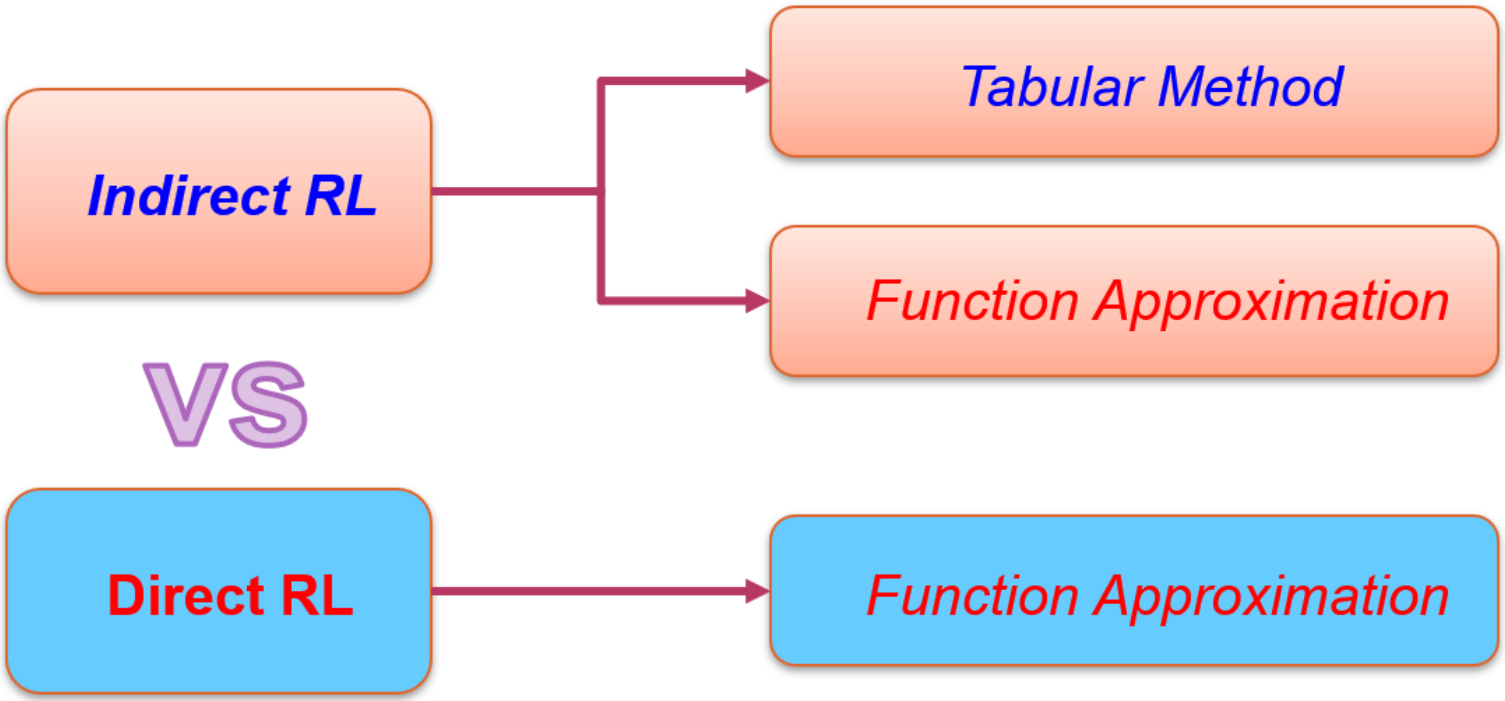
$$\max_{\pi} \mathbb{E}_{s\sim d(s)} \left\{ v^{\pi}(s) \right\}$$

s.t. $p\left(s' \mid s, a\right) = \mathcal{P}_{ss'}^{a}$ or $\left\{ s_0, a_0, r_0, s_1, a_1, r_1, s_2, a_2, r_2, s_3, a_3, r_3, \cdots \right\}$

How to find $\pi^*(a|s)$?

1. Indirect RL
2. Direct RL

Indirect and Direct RL



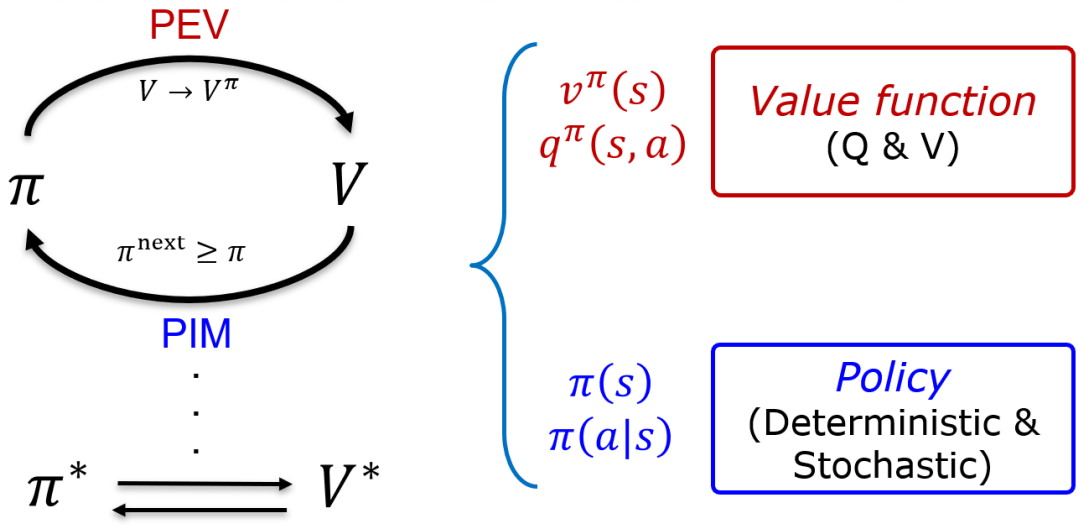
Basis of Indirect Reinforcement Learning

Indirect RL seeks to find the solution to the Bellman Optimal equation

$$v^*(s) = \max_{\pi} \left\{ \mathbb{E}_{\pi} \left\{ r + \gamma v^* \left(s' \right) \right\} \right\}$$

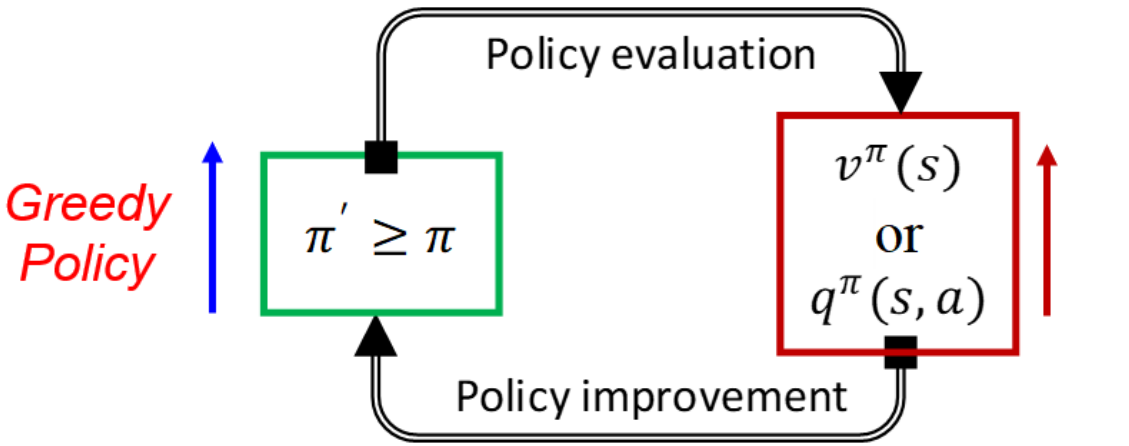
- **Generalized Policy Iteration (GPI)** framework

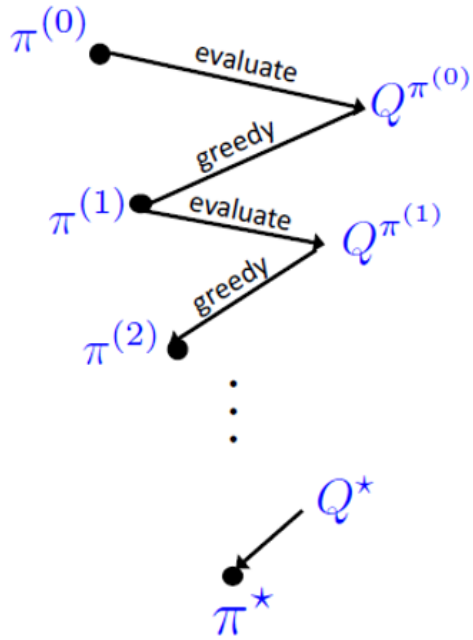
  Policy EValuation (PEV) + Policy IMprovement (PIM)



Basis of Indirect Reinforcement Learning

Take the solution of the Bellman optimality equation as the optimal policy.





Indirect RL vs Direct RL

Indirect RL

- Sufficient & necessary condition of optimality

  - Bellman equation (discrete-time)

$$\pi^*(a \mid s) = \text{ Solution of Bellman equation or BOE}$$

- Convergence: *Bellman operator is γ-contractive*

Direct RL

- Search for a parameterized policy $\pi(a \mid s; \theta)$ that maximizes the overall objective function $J(\theta)$

$$\theta^* = \arg\max_{\theta} J\big(\pi(a \mid s; \theta)\big)$$

  - Search θ^* by using a numerical optimization technique

- Convergence: *Same as optimization algorithms*

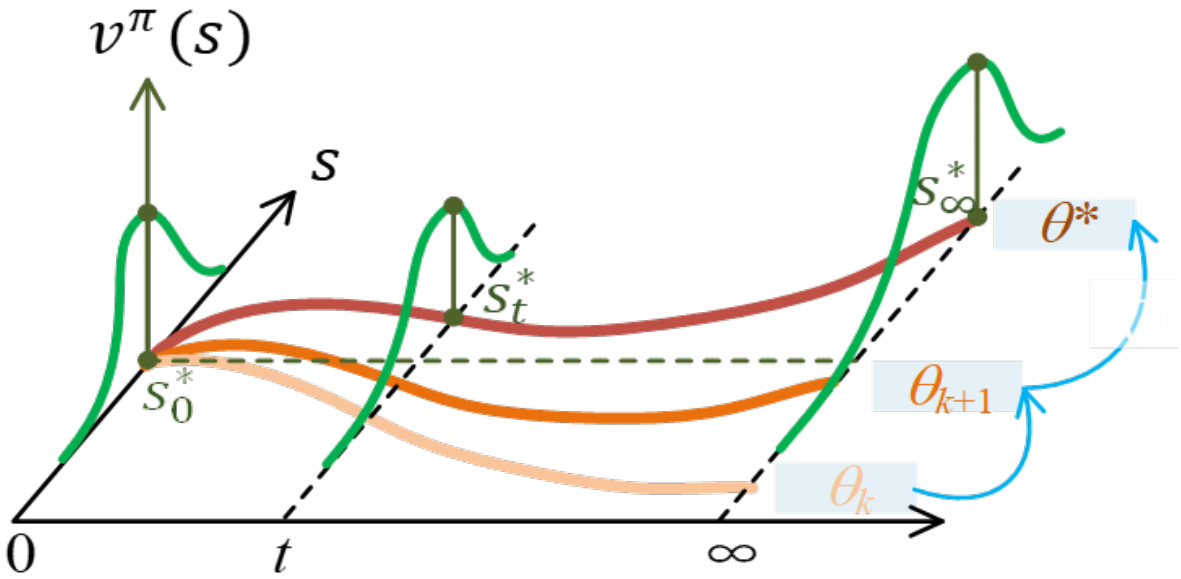
Basis of Direct Reinforcement Learning

View RL as an optimization problem and calculate its optimum

Search for a parameterized policy to the scalar performance index $J(\theta)$

$$\theta^* = \arg\max_\theta J(\theta) = \arg\max_\theta \mathbb{E}_{s\sim d_{\text{init}}(s)}\{v_{\pi_\theta}(s)\}$$

$$\theta \leftarrow \theta + \alpha \cdot \nabla_\theta J(\theta)$$

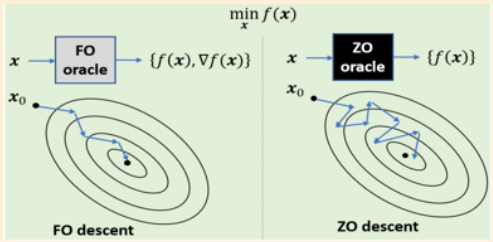


Direct RL

Mainstream direct RL methods

Zero-order optimization

- Evolutionary algorithm (e.g., finite difference)
- Bayesian optimization



First-order optimization

- Likelihood ratio gradient
- Natural policy gradient
- Deterministic policy gradient

Second-order optimization

- Newton method
- Quasi-Newton method

$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}$$

1. Indirect and Direct RL

2. Basic idea of policy gradient

3. Metrics to define optimal policies
   - 3.0 Influence of Initial State Distribution
   - 3.1 Metric 1: Average value
   - 3.2 Metric 2: Average reward
   - 3.3 Summary of the two metrics

4. Gradients of the metrics

5. Gradient-Ascent algorithm

6. Summary

Basic idea of policy gradient

Previously, policies have been represented by tables:

- The action probabilities of all states are stored in a table $\pi(a \mid s)$. Each entry of the table is indexed by a state s_t and an action a_t.

	a_1	a_2	a_3	a_4	a_5
s_1	$\pi(a_1 \mid s_1)$	$\pi(a_2 \mid s_1)$	$\pi(a_3 \mid s_1)$	$\pi(a_4 \mid s_1)$	$\pi(a_5 \mid s_1)$
s_2	$\pi(a_1 \mid s_2)$	$\pi(a_2 \mid s_2)$	$\pi(a_3 \mid s_2)$	$\pi(a_4 \mid s_2)$	$\pi(a_5 \mid s_2)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
s_9	$\pi(a_1 \mid s_9)$	$\pi(a_2 \mid s_9)$	$\pi(a_3 \mid s_9)$	$\pi(a_4 \mid s_9)$	$\pi(a_5 \mid s_9)$

Basic idea of policy gradient

Now, policies can be represented by parameterized functions:

$$\pi(a \mid s, \theta)$$

where $\theta \in \mathbb{R}^m$ is a parameter vector. The function representation is also sometimes written as $\pi(a, s, \theta)$, $\pi_\theta(a \mid s)$, or $\pi_\theta(a, s)$.

- The function can be a neural network
- *Advantage*: when the state space is large, the tabular representation will be of low efficiency in terms of storage and generalization.

Basic idea of policy gradient

Differences between tabular and function representations:

- First, how to define optimal policies?
  - In the tabular case, a policy π is optimal if it can maximize every state value.
  - In the function case, a policy π is optimal if it can maximize certain scalar metrics.

Basic idea of policy gradient

Differences between tabular and function representations:

- Second, how to access the probability of an action?
  - In the tabular case, the probability of taking a at s can be directly accessed by looking up the tabular policy.
  - In the function case, we need to calculate the value of $\pi(a \mid s, \theta)$ given the function structure and the parameter θ.

Basic idea of policy gradient

Differences between tabular and function representations:

- Third, how to update policies?
  - In the tabular case, a policy π can be updated by directly changing the entries in the table.
  - In the function case, a policy π cannot be updated in this way anymore. Instead, it can only be updated by changing the parameter θ.

Basic idea of policy gradient

The basic idea of the policy gradient is simple:

- First, metrics (or objective functions) to define optimal policies $J(\theta)$, which can define optimal policies.
- Second, gradient-based optimization algorithms to search for optimal policies:

$$\theta_{t+1} = \theta_t + \alpha \nabla_\theta J(\theta_t)$$

Although the idea is simple, the complication emerges when we try to answer the following questions.

1. What appropriate metrics should be used?
2. How to calculate the gradients of the metrics?

1. Indirect and Direct RL

2. Basic idea of policy gradient

3. Metrics to define optimal policies
   - 3.0 Influence of Initial State Distribution
   - 3.1 Metric 1: Average value
   - 3.2 Metric 2: Average reward
   - 3.3 Summary of the two metrics

4. Gradients of the metrics

5. Gradient-Ascent algorithm

6. Summary

Overall RL objective function

$$\begin{aligned}\max_{\theta} J(\theta) &= \mathbb{E}_{s_t \sim d(s_t)} \left\{ v_{\pi} \left(s_t \right) \right\} \\ &= \int d\left(s_t \right) v_{\pi_\theta} \left(s_t \right) \mathrm{d}s_t \\ &= \mathbb{E}_{s_t, a_t, s_{t+1}, \ldots \sim \rho_{\pi_\theta}} \left\{ \sum_{\tau=t}^{\infty} \gamma^{\tau - t} r_{\tau} \right\}\end{aligned}$$

ρ_{π_θ} is the joint probability of states and actions in the trajectory.

Revisit the value function in terms of the trajectory concept

$$v^{\pi}(s) = \mathbb{E}_{a_t, s_{t+1}, a_{t+1}, s_{t+2}, a_{t+2}, \ldots \sim \rho_{\pi_\theta}} \left\{ \sum_{\tau=t}^{\infty} \gamma^{\tau - t} r_{\tau} \mid s_t = s \right\}$$

$$q^{\pi}(s, a) = \mathbb{E}_{s_{t+1}, a_{t+1}, s_{t+2}, a_{t+2}, \ldots \sim \rho_{\pi_\theta}} \left\{ \sum_{\tau=t}^{\infty} \gamma^{\tau - t} r_{\tau} \mid s_t = s, a_t = a \right\}$$

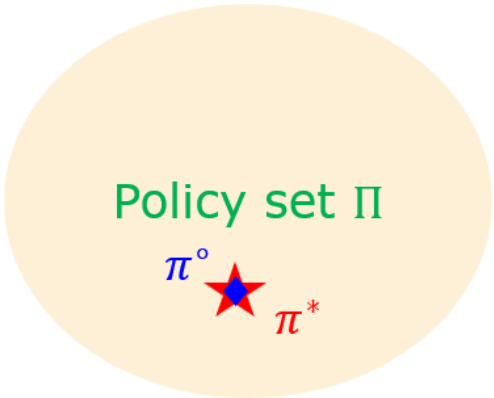
Overall RL objective function

Define two kinds of "optimal" policies.

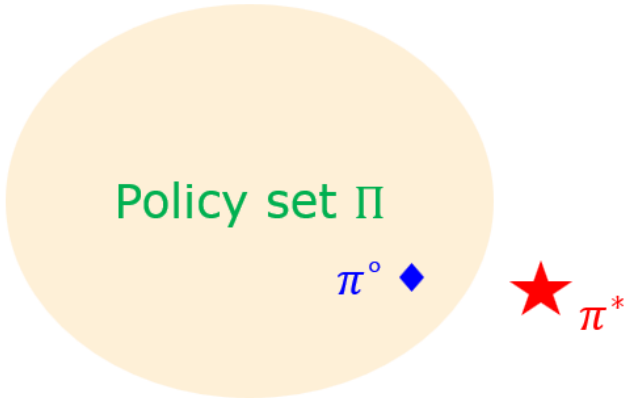
$$\pi^*(s) = \arg\max_{a \in \mathcal{A}} \sum_{s' \in \mathcal{S}} \mathcal{P}^a_{ss'} \left(r + \gamma v^* \left(s' \right) \right), \forall s \in \mathcal{S}$$

$$\pi^\circ = \arg\max_{\pi \in \Pi} J(\pi(s))$$

- Π is the allowable policy set from designers
- π^* is the optimal policy coming from each state element (Bellman)
- π° is optimal policy from overall RL criterion maximization



Case (1): $\pi^* \in \Pi$



Case (2): $\pi^* \notin \Pi$

Overall RL objective function

Case (1): $\pi^*(s)$ is inside allowable policy set Π

1^{st} step:

$$J(\pi^*) \leq J(\pi^\circ) = \max_\pi J(\pi)$$

2^{nd} step:

$$\begin{aligned} J(\pi^\circ) &= \max_\pi \mathbb{E}_{s\sim d(s)} \{v^\pi(s)\} \\ &\leq \max_\pi \mathbb{E}_{s\sim d(s)} \left\{\max_\pi v^\pi(s)\right\} \\ &= \mathbb{E}_{s\sim d(s)} \left\{\max_\pi v^\pi(s)\right\} \\ &= \mathbb{E}_{s\sim d(s)} \{v^*(s)\} \\ &= J(\pi^*) \end{aligned}$$

Conclusion

$$J(\pi^*) = J(\pi^\circ)$$

$$\mathbb{E}_{s\sim d(s)} \left\{\max_\pi v^\pi(s)\right\} = \max_\pi \mathbb{E}_{s\sim d(s)} \{v^\pi(s)\}$$

- The initial state distribution does not affect the optimal policy

Overall RL objective function

Case (2): $\pi^*(s)$ is NOT inside allowable policy set Π

Only the inequality holds

$$\max_{\pi} \mathbb{E}_{s\sim d(s)} \left\{ v^{\pi}(s) \right\} \leq \mathbb{E}_{s\sim d(s)} \left\{ \max_{\pi} v^{\pi}(s) \right\}$$

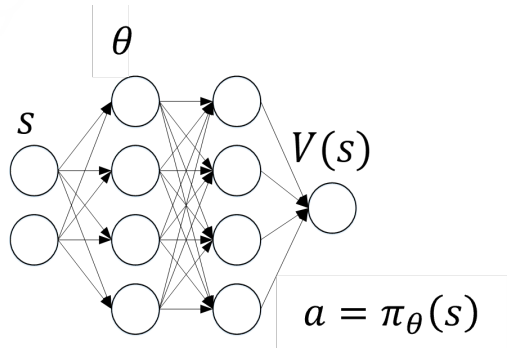
$$J(\pi^{\circ}) \leq J(\pi^*)$$

Conclusion

- Policy $\pi^{\circ}(s) \in \Pi$ gives a less optimal policy than π^*
- π° becomes dependent of initial state distribution $d(s)$

Hint: Policy set Π should be as large as possible

- Neural network is a good choice!



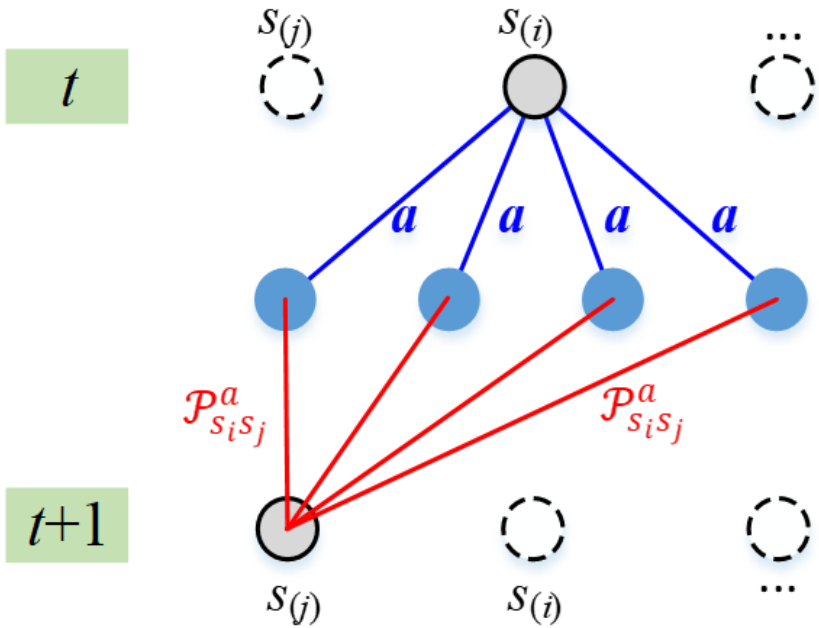
Stationary State Distribution

One-step transition probability

Given policy $\pi(a \mid s)$ and environment model $p(s' \mid s, a)$

$$\mathcal{S} = \{s_1, s_2, \cdots, s_n\}$$

$$\zeta_{i,j} = \sum_{a \in \mathcal{A}} \pi(a \mid s = s_i)\, p\left(s' = s_j \mid s = s_i, a\right)$$



Stationary State Distribution

State distribution at time t

Occurrence frequency of a specific state s_i at time t

$$d_t(s_i) = \Pr\{S_t = s_i\}$$

"Stationary" refers to "stationary in time"

$$\boldsymbol{d}_{t+1} = H_{n\times n}\boldsymbol{d}_t$$

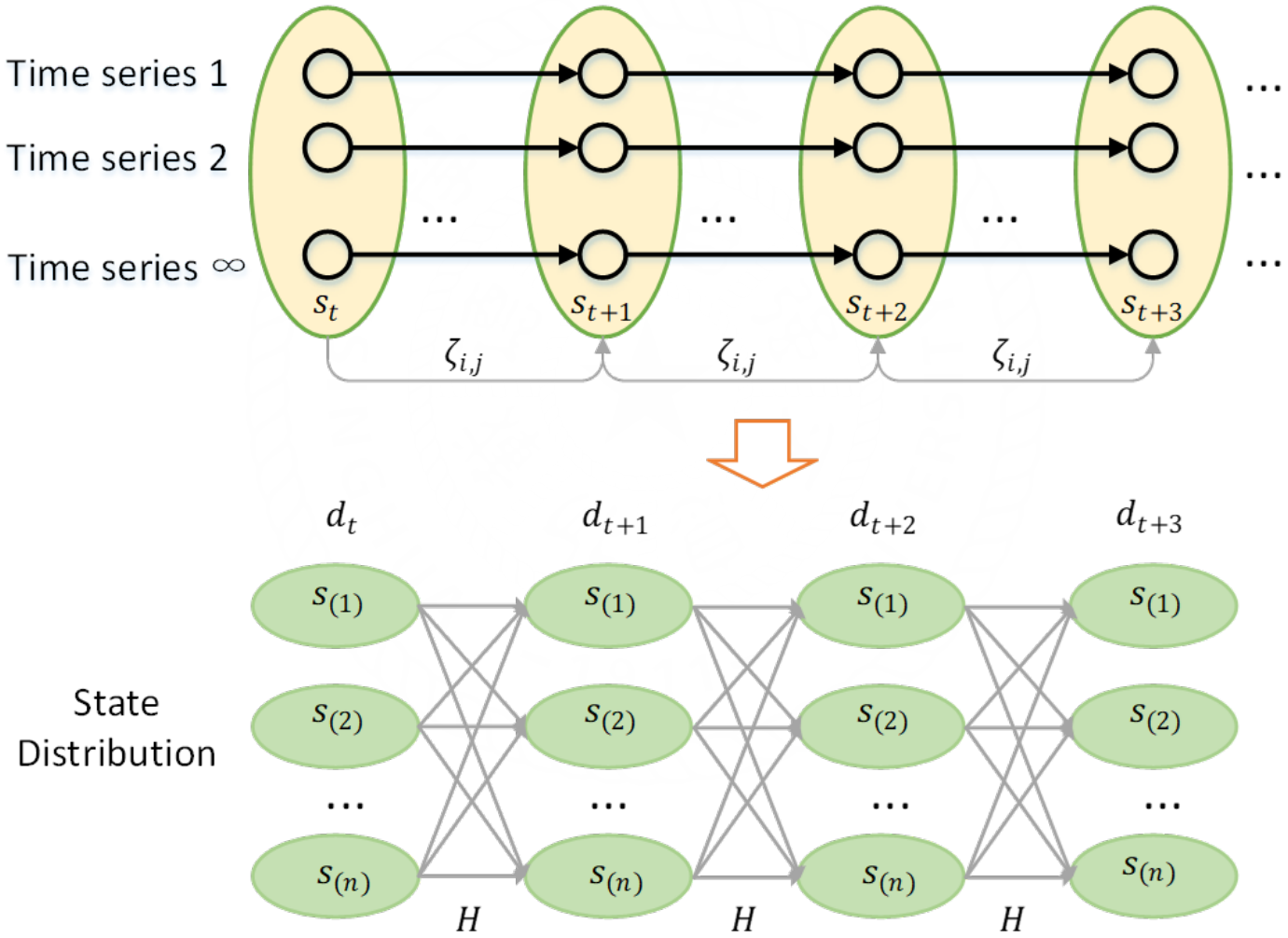
$$\boldsymbol{H}_{n\times n} = \begin{bmatrix} \zeta_{1,1} & \cdots & \zeta_{n,1} \\ \vdots & \ddots & \vdots \\ \zeta_{1,n} & \cdots & \zeta_{n,n} \end{bmatrix} \quad \boldsymbol{d}_t = \begin{bmatrix} d_t(s_1) & d_t(s_2) & \cdots & d_t(s_n) \end{bmatrix}^{\mathrm{T}}$$

Stationary state distribution (SSD) | $(t \to \infty)$

$$\boldsymbol{d}(s) = \boldsymbol{H}\,\boldsymbol{d}(s)$$

Stationary State Distribution

Random variable vs State distribution



Stationary State Distribution

Some properties of SSD

(1) Any finite, irreducible, and ergodic Markov chain has a unique SSD

(2) For any $i, j \in \mathcal{S}$, the following limit exists, independent of initial state S_0

$$\lim_{t \to \infty} \Pr\{S_t = s_j \mid S_0 = s_i\}$$

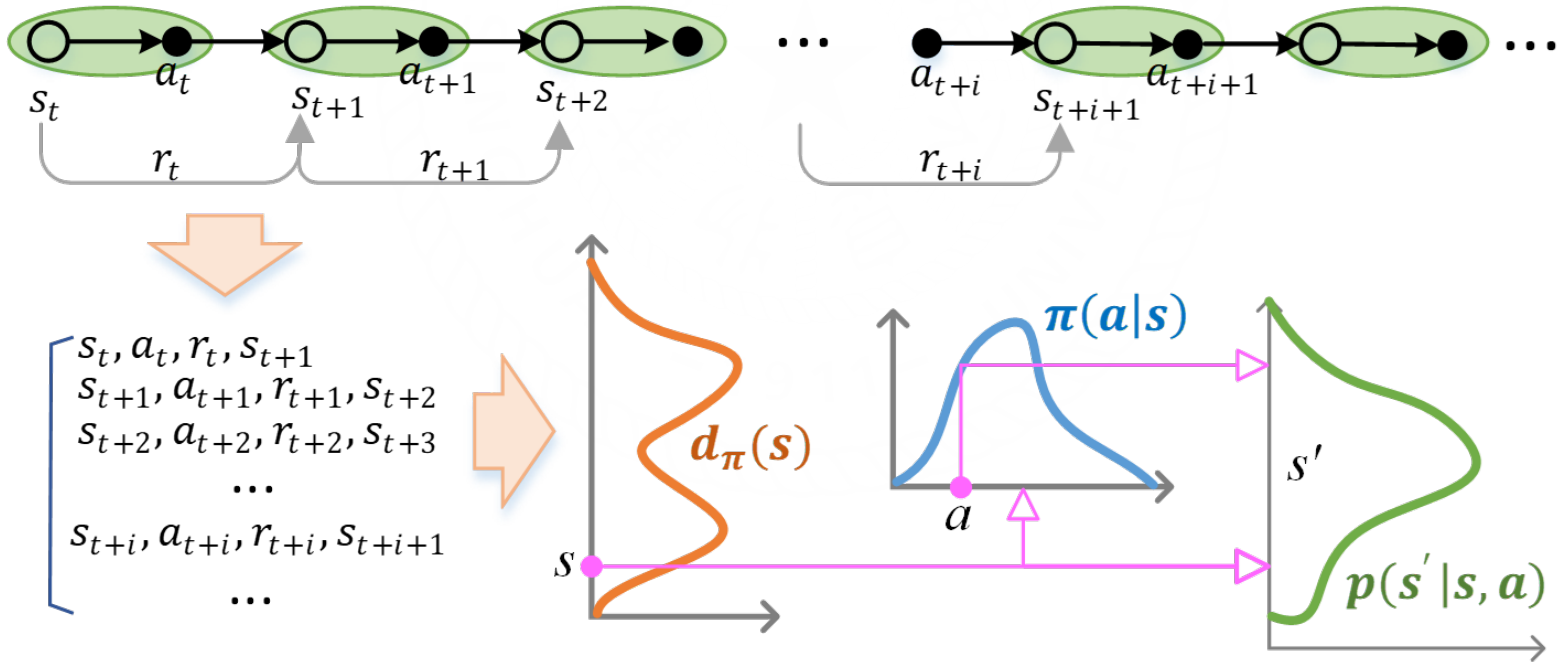
(3) In an MDP, the SSD under policy π is

$$d_\pi\left(s_{(j)}\right) = \lim_{t \to \infty} \Pr\left\{s_t = s_{(j)} \mid s_0 = s_{(i)}\right\}$$

Stationary State Distribution

Graphic understanding

- Limiting distributions that can start from any initial state distribution
- Temporal order of samples becomes meaningless since each sample could occur randomly an infinite times.



Metric 1: Average value

The first metric is the average state value or called average value:

$$\bar{v}_\pi = \sum_{s \in \mathcal{S}} d(s) v_\pi(s)$$

- $\bar{v}_\pi$ is a weighted average of the state values.
- $d(s) \geq 0$ is the weight for state s.

Since $\sum_{s \in \mathcal{S}} d(s) = 1$, we can interpret $d(s)$ as a probability distribution. Then, the metric can be written as

$$\bar{v}_\pi = \mathbb{E}_{S \sim d}[v_\pi(S)]$$

Metric 1: Average value

How to select the distribution d? There are two cases.

Case 1: d is independent of the policy π.

- This case is relatively simple because the gradient of the metric is easier to calculate: $\nabla_\theta \bar{v}_\pi = d^T \nabla_\theta v_\pi$
- In this case, we specifically denote d as d_0 and $\bar{v}_\pi$ as $\bar{v}_\pi^0$.

How to select d_0 ?

- One trivial way is to treat all the states equally important and hence select $d_0(s) = 1/|\mathcal{S}|$.
- Another important case is that we are only interested in a specific state s_0.

  For example, the episodes in some tasks always start from the same state s_0.

  Then, we only care about the long-term return starting from s_0. In this case,

$$d_0(s_0) = 1, \quad d_0(s \neq s_0) = 0$$

  In this case, $\bar{v}_\pi = v_\pi(s_0)$

Metric 1: Average value

How to select the distribution d ? There are two cases.

Case 2: d depends on the policy π.

- A common way is to select d as $d_\pi(s)$, which is the stationary distribution under π.

The interpretation of selecting d_π is as follows.

- d_π reflects the long-run behavior of the Markov decision process (MDP) under a given policy π.
- If one state is frequently visited in the long run, it is more important and deserves more weight.
- If a state is hardly visited, then we give it less weight.

Metric 1: Average value

$$\begin{aligned}\max_{\theta} J(\theta) &= \mathbb{E}_{s_t \sim d(s_t)} \left\{ v^{\pi} \left(s_t \right) \right\} \\ &= \int d\left(s_t \right) v^{\pi_\theta} \left(s_t \right) ds_t \\ &= \mathbb{E}_{s_t, a_t, s_{t+1}, \ldots \sim \rho_{\pi_\theta}} \left\{ \sum_{\tau=t}^{\infty} \gamma^{\tau - t} r_\tau \right\}\end{aligned}$$

ρ_{π_θ} is the joint probability of states and actions in the trajectory - Revisit the value function in terms of the trajectory concept

$$v^{\pi}(s) = \mathbb{E}_{a_t, s_{t+1}, a_{t+1}, s_{t+2}, a_{t+2}, \ldots \sim \rho_{\pi_\theta}} \left\{ \sum_{\tau=t}^{\infty} \gamma^{\tau - t} r_\tau \mid s_t = s \right\}$$

$$q^{\pi}(s, a) = \mathbb{E}_{s_{t+1}, a_{t+1}, s_{t+2}, a_{t+2}, \ldots \sim \rho_{\pi_\theta}} \left\{ \sum_{\tau=t}^{\infty} \gamma^{\tau - t} r_\tau \mid s_t = s, a_t = a \right\}$$

Metric 1: Average value

An important equivalent expression:

You will see the following metric often in the literature:

$$J(\theta) = \lim_{n\to\infty} \mathbb{E}\left[\sum_{t=0}^{n} \gamma^t R_{t+1}\right] = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R_{t+1}\right]$$

Question: What is its relationship to the metric we introduced just now?

Answer: They are the same. That is because

$$\begin{aligned} J(\theta) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R_{t+1}\right] &= \sum_{s\in\mathcal{S}} d(s)\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R_{t+1} \mid S_0 = s\right] \\ &= \sum_{s\in\mathcal{S}} d(s) v_\pi(s) \\ &= \bar{v}_\pi \end{aligned}$$

Metric 2: Average Reward

The second metric is average one-step reward or simply average reward:

$$\bar{r}_\pi \doteq \sum_{s \in \mathcal{S}} d_\pi(s) r_\pi(s) = \mathbb{E}\left[r_\pi(S)\right]$$

where $S \sim d_\pi$,

$$r_\pi(s) = \sum_{a \in \mathcal{A}} \pi(a \mid s) r(s, a)$$

$$r(s, a) = \mathbb{E}[R \mid s, a] = \sum_r r p(r \mid s, a)$$

Remarks:

- $\bar{r}_\pi$ is simply a weighted average of immediate rewards.
- $r_\pi(s)$ is the average immediate reward that can be obtained from s.
- d_π is the stationary distribution.

Metric 2: average reward

An important equivalent expression:

- Suppose an agent follows a given policy and generate a trajectory with the rewards as $(R_1, R_2, \ldots)$.
- The average single-step reward along this trajectory is

$$\lim_{n\to\infty} \frac{1}{n}\mathbb{E}\left[R_1 + R_2 + \cdots + R_n \mid S_0 = s_0\right]$$
$$= \lim_{n\to\infty} \frac{1}{n}\mathbb{E}\left[\sum_{t=0}^{n-1} R_{t+1} \mid S_0 = s_0\right]$$

where s_0 is the starting state of the trajectory.

Metrics to define optimal policies - Remarks

An important fact is that

$$\begin{aligned}\lim_{n\to\infty}\frac{1}{n}\mathbb{E}\left[\sum_{t=0}^{n-1}R_{t+1}\mid S_0=s_0\right] &= \lim_{n\to\infty}\frac{1}{n}\mathbb{E}\left[\sum_{t=0}^{n-1}R_{t+1}\right]\\ &= \sum_s d_\pi(s)r_\pi(s)\\ &= \bar{r}_\pi\end{aligned}$$

Remarks:

- Highlight: the starting state s_0 does not matter.

Summary of the two metrics

Metric	Expression 1	Expression 2	Expression 3
$\bar{v}_\pi$	$\sum_{s\in\mathcal{S}} d(s)v_\pi(s)$	$\mathbb{E}_{S\sim d}\left[v_\pi(S)\right]$	$\lim_{n\to\infty}\mathbb{E}\left[\sum_{t=0}^{n}\gamma^t R_{t+1}\right]$
$\bar{r}_\pi$	$\sum_{s\in\mathcal{S}} d_\pi(s)r_\pi(s)$	$\mathbb{E}_{S\sim d_\pi}\left[r_\pi(S)\right]$	$\lim_{n\to\infty}\frac{1}{n}\mathbb{E}\left[\sum_{t=0}^{n-1} R_{t+1}\right]$

Table: Summary of the different but equivalent expressions of $\bar{v}_\pi$ and $\bar{r}_\pi$.

Summary of the two metrics

Remark 1 about the metrics:

- All these metrics are functions of π.
- Since π is parameterized by θ, these metrics are functions of θ.
- In other words, different values of θ can generate different metric values.

Therefore, we can search for the optimal values of θ to maximize these metrics.

This is the basic idea of policy gradient methods.

Summary of the two metrics

Remark 2 about the metrics:

- One complication is that the metrics can be defined in either the discounted case where $\gamma \in (0, 1)$ or the undiscounted case where $\gamma = 1$.
- The undiscounted case is nontrivial.
- We only consider the discounted case so far.

Summary of the two metrics

Remark 3 about the metrics:

- What is the relationship between $\bar{r}_\pi$ and $\bar{v}_\pi$?
- The two metrics are equivalent (not equal) to each other. Specifically, in the discounted case where $\gamma < 1$, it holds that

$$\bar{r}_\pi = (1-\gamma)\bar{v}_\pi$$

Therefore, they can be maximized simultaneously.

1. Indirect and Direct RL

2. Basic idea of policy gradient

3. Metrics to define optimal policies
   - 3.0 Influence of Initial State Distribution
   - 3.1 Metric 1: Average value
   - 3.2 Metric 2: Average reward
   - 3.3 Summary of the two metrics

4. Gradients of the metrics

5. Gradient-Ascent algorithm

6. Summary

Gradients of the metrics

Given a metric, we next

- derives its gradient
- and then, apply gradient-based methods to optimize the metric.

The gradient calculation is one of the most complicated parts of policy gradient methods! That is because

- first, we need to distinguish different metrics $\bar{v}_\pi, \bar{r}_\pi, \bar{v}_\pi^0$
- second, we need to distinguish between discounted and undiscounted cases.

Gradients of the metrics

Likelihood Ratio Gradient

The expression of the gradient:

$$\nabla_\theta J(\theta) = \sum_{s\in\mathcal{S}} \eta(s) \sum_{a\in\mathcal{A}} \nabla_\theta \pi(a \mid s,\theta) q_\pi(s,a)$$

The above is a unified expression of many cases:

- $J(\theta)$ can be $\bar{v}_\pi$, $\bar{r}_\pi$, or $\bar{v}^0_\pi$.
- "$=$" may denote strict equality, approximation, or proportional to.
- $\eta(s)$ is a distribution or weight of the states.

The derivation of this expression is very complex.

For most readers, it is sufficient to know this expression.

Gradients of the metrics

A compact and important expression of the gradient:

$$\begin{aligned}\nabla_\theta J(\theta) &= \sum_{s\in\mathcal{S}} \eta(s) \sum_{a\in\mathcal{A}} \nabla_\theta \pi(a \mid s,\theta) q_\pi(s,a) \\ &= \mathbb{E}_{S\sim\eta, A\sim\pi}\Big[\nabla_\theta \ln \pi(A \mid S,\theta) q_\pi(S,A)\Big]\end{aligned}$$

First, why is this helpful expression?

Because we can use samples to approximate the gradient:

$$\nabla_\theta J \approx \nabla_\theta \ln \pi(a \mid s,\theta) q_\pi(s,a)$$

where s, a are samples. This is the idea of Stochastic Gradient Descent (SGD).

Gradients of the metrics

A compact and important expression of the gradient:

$$\begin{aligned}\nabla_\theta J(\theta) &= \sum_{s\in\mathcal{S}} \eta(s) \sum_{a\in\mathcal{A}} \nabla_\theta \pi(a \mid s,\theta) q_\pi(s,a) \\ &= \mathbb{E}_{S\sim\eta, A\sim\pi}\left[\nabla_\theta \ln \pi(A \mid S,\theta) q_\pi(S,A)\right]\end{aligned}$$

Proof:

Consider the function $\ln \pi$ where $\ln$ is the natural logarithm. It is easy to see that

$$\nabla_\theta \ln \pi(a \mid s,\theta) = \frac{\nabla_\theta \pi(a \mid s,\theta)}{\pi(a \mid s,\theta)}$$

and hence

$$\nabla_\theta \pi(a \mid s,\theta) = \pi(a \mid s,\theta) \nabla_\theta \ln \pi(a \mid s,\theta)$$

Gradients of the metrics

A compact and important expression of the gradient:

$$\begin{aligned}\nabla_\theta J(\theta) &= \sum_{s\in\mathcal{S}} \eta(s) \sum_{a\in\mathcal{A}} \nabla_\theta \pi(a \mid s,\theta) q_\pi(s,a) \\ &= \mathbb{E}_{S\sim\eta, A\sim\pi}\left[\nabla_\theta \ln \pi(A \mid S,\theta) q_\pi(S,A)\right]\end{aligned}$$

Proof (continued):

Then, we have

$$\begin{aligned}\nabla_\theta J &= \sum_s \eta(s) \sum_a \nabla_\theta \pi(a \mid s,\theta) q_\pi(s,a) \\ &= \sum_s \eta(s) \sum_a \pi(a \mid s,\theta) \nabla_\theta \ln \pi(a \mid s,\theta) q_\pi(s,a) \\ &= \mathbb{E}_{S\sim\eta}\left[\sum_a \pi(a \mid S,\theta) \nabla_\theta \ln \pi(a \mid S,\theta) q_\pi(S,a)\right] \\ &= \mathbb{E}_{S\sim\eta, A\sim\pi}\left[\nabla_\theta \ln \pi(A \mid S,\theta) q_\pi(S,A)\right]\end{aligned}$$

Gradients of the metrics

Remarks: It is required by $\ln \pi(a \mid s, \theta)$ that for any s, a, θ

$$\pi(a \mid s, \theta) > 0$$

- This can be achieved by using softmax functions that can normalize the entries in a vector from $(-\infty, +\infty)$ to $(0, 1)$.
- For example, for any vector $x = [x_1, \ldots, x_n]^T$,

$$z_i = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}$$

where $z_i \in (0, 1)$ and $\sum_{i=1}^{n} z_i = 1$.

- Specifically, the policy function has the form of

$$\pi(a \mid s, \theta) = \frac{e^{h(s,a,\theta)}}{\sum_{a' \in \mathcal{A}} e^{h(s,a',\theta)}}$$

where $h(s, a, \theta)$ is another function to be learned.

Gradients of the metrics

Remarks:

- Such a form based on the softmax function can be realized by a neural network whose input is s and parameter is θ. The network has $|\mathcal{A}|$ outputs, each of which corresponds to $\pi(a \mid s, \theta)$ for an action a. The activation function of the output layer should be softmax.
- Since $\pi(a \mid s, \theta) > 0$ for all a, the parameterized policy is stochastic and hence exploratory.
  - There also exist deterministic policy gradient (DPG) methods. We will study in the next lecture.

1. Indirect and Direct RL

2. Basic idea of policy gradient

3. Metrics to define optimal policies
   - 3.0 Influence of Initial State Distribution
   - 3.1 Metric 1: Average value
   - 3.2 Metric 2: Average reward
   - 3.3 Summary of the two metrics

4. Gradients of the metrics

5. Gradient-Ascent algorithm

6. Summary

Gradient-Ascent algorithm

Now, we present the first policy gradient algorithm to find optimal policies!

1. The gradient-ascent algorithm maximizing $J(\theta)$ is

$$\begin{aligned}\theta_{t+1} &= \theta_t + \alpha \nabla_\theta J(\theta_t) \\ &= \theta_t + \alpha \mathbb{E}\Big[\nabla_\theta \ln \pi(A \mid S, \theta_t)\, q_\pi(S, A)\Big]\end{aligned}$$

2. Since the true gradient is unknown, we can replace it by a stochastic one:

$$\theta_{t+1} = \theta_t + \alpha \nabla_\theta \ln \pi(a_t \mid s_t, \theta_t)\, q_\pi(s_t, a_t)$$

3. Furthermore, since q_π is unknown, it can be replaced by an estimate:

$$\theta_{t+1} = \theta_t + \alpha \nabla_\theta \ln \pi(a_t \mid s_t, \theta_t)\, q_t(s_t, a_t)$$

Gradient-Ascent algorithm

If $q_\pi(s_t, a_t)$ is estimated by Monte Carlo estimation, the algorithm has a specifics name, REINFORCE.

REINFORCE is one of earliest and simplest policy gradient algorithms.

Many other policy gradient algorithms such as the actor-critic methods can be obtained by extending REINFORCE (next lecture).

Pseudocode: Policy Gradient by Monte Carlo (REINFORCE)

Initialization: Initial parameter $\theta; \gamma \in (0,1); \alpha > 0$.

Goal: Learn an optimal policy to maximize $J(\theta)$.

For each episode, do

Generate an episode $\{s_0, a_0, r_1, \ldots, s_{T-1}, a_{T-1}, r_T\}$ following $\pi(\theta)$.

For $t = 0, 1, \ldots, T-1$:

Value update: $q_t(s_t, a_t) = \sum_{k=t+1}^{T} \gamma^{k-t-1} r_k$

Policy update: $\theta \leftarrow \theta + \alpha \nabla_\theta \ln \pi(a_t \mid s_t, \theta)\, q_t(s_t, a_t)$

Gradient-Ascent algorithm

Remark 1: How to do sampling?

$$\mathbb{E}_{S\sim\eta, A\sim\pi}\left[\nabla_\theta \ln \pi\left(A \mid S, \theta_t\right) q_\pi(S, A)\right] \longrightarrow \nabla_\theta \ln \pi\left(a \mid s, \theta_t\right) q_\pi(s, a)$$

- How to sample S ?
  - $S \sim \eta$, where the distribution η is a long-run behavior under π.
  - In practice, people usually do not care about it.
- How to sample A ?
  - $A \sim \pi(A \mid S, \theta)$. Hence, a_t should be sampled following $\pi\left(\theta_t\right)$ at s_t.
  - Therefore, policy gradient methods are on-policy.

Gradient-Ascent algorithm

Remark 2: How to interpret this algorithm?

Since

$$\nabla_\theta \ln \pi\left(a_t \mid s_t, \theta_t\right) = \frac{\nabla_\theta \pi\left(a_t \mid s_t, \theta_t\right)}{\pi\left(a_t \mid s_t, \theta_t\right)}$$

the algorithm can be rewritten as

$$\begin{aligned}\theta_{t+1} &= \theta_t + \alpha \nabla_\theta \ln \pi\left(a_t \mid s_t, \theta_t\right) q_t\left(s_t, a_t\right) \\ &= \theta_t + \alpha \underbrace{\left(\frac{q_t\left(s_t, a_t\right)}{\pi\left(a_t \mid s_t, \theta_t\right)}\right)}_{\beta_t} \nabla_\theta \pi\left(a_t \mid s_t, \theta_t\right).\end{aligned}$$

Therefore, we have the important expression of the algorithm:

$$\theta_{t+1} = \theta_t + \alpha \beta_t \nabla_\theta \pi\left(a_t \mid s_t, \theta_t\right)$$

Gradient-Ascent algorithm

The interpretation of

$$\theta_{t+1} = \theta_t + \alpha\beta_t \nabla_\theta \pi\left(a_t \mid s_t, \theta_t\right)$$

is as follows. Suppose that α is sufficiently small.

- If $\beta_t > 0$, the probability of choosing (s_t, a_t) is increased:

$$\pi\left(a_t \mid s_t, \theta_{t+1}\right) > \pi\left(a_t \mid s_t, \theta_t\right)$$

- If $\beta_t < 0$, the probability of choosing (s_t, a_t) is lower:

$$\pi\left(a_t \mid s_t, \theta_{t+1}\right) < \pi\left(a_t \mid s_t, \theta_t\right)$$

Math: When $\theta_{t+1} - \theta_t$ is sufficiently small, the definition of differential implies

$$\begin{aligned}
\pi\left(a_t \mid s_t, \theta_{t+1}\right) &\approx \pi\left(a_t \mid s_t, \theta_t\right) + \left(\nabla_\theta \pi\left(a_t \mid s_t, \theta_t\right)\right)^T \left(\theta_{t+1} - \theta_t\right) \\
&= \pi\left(a_t \mid s_t, \theta_t\right) + \alpha\beta_t \left(\nabla_\theta \pi\left(a_t \mid s_t, \theta_t\right)\right)^T \left(\nabla_\theta \pi\left(a_t \mid s_t, \theta_t\right)\right) \\
&= \pi\left(a_t \mid s_t, \theta_t\right) + \alpha\beta_t \left\|\nabla_\theta \pi\left(a_t \mid s_t, \theta_t\right)\right\|^2
\end{aligned}$$

Gradient-Ascent algorithm

$$\theta_{t+1} = \theta_t + \alpha \underbrace{\left(\frac{q_t(s_t, a_t)}{\pi(a_t \mid s_t, \theta_t)} \right)}_{\beta_t} \nabla_\theta \pi(a_t \mid s_t, \theta_t)$$

Interpretation (continued): β_t can balance exploration and exploitation.

The reason is as follows.

- First, β_t is proportional to $q_t(s_t, a_t)$.

$$\text{greater } q_t(s_t, a_t) \Longrightarrow \text{ greater } \beta_t \Longrightarrow \text{ greater } \pi(a_t \mid s_t, \theta_{t+1})$$

Therefore, the algorithm intends to exploit actions with greater values.

- Second, β_t is inversely proportional to $\pi(a_t \mid s_t, \theta_t)$ (when $q_t(s_t, a_t) > 0$).

$$\text{smaller } \pi(a_t \mid s_t, \theta_t) \Longrightarrow \text{ greater } \beta_t \Longrightarrow \text{ greater } \pi(a_t \mid s_t, \theta_{t+1})$$

Therefore, the algorithm intends to explore actions that have low probabilities.

1. Indirect and Direct RL

2. Basic idea of policy gradient

3. Metrics to define optimal policies
   - 3.0 Influence of Initial State Distribution
   - 3.1 Metric 1: Average value
   - 3.2 Metric 2: Average reward
   - 3.3 Summary of the two metrics

4. Gradients of the metrics

5. Gradient-Ascent algorithm

6. Summary

Summary

Contents of this lecture:

- Metrics for optimality
- Gradients of the metrics
- Gradient-ascent algorithm
- A special case: REINFORCE

Next lecture: Actor-critic