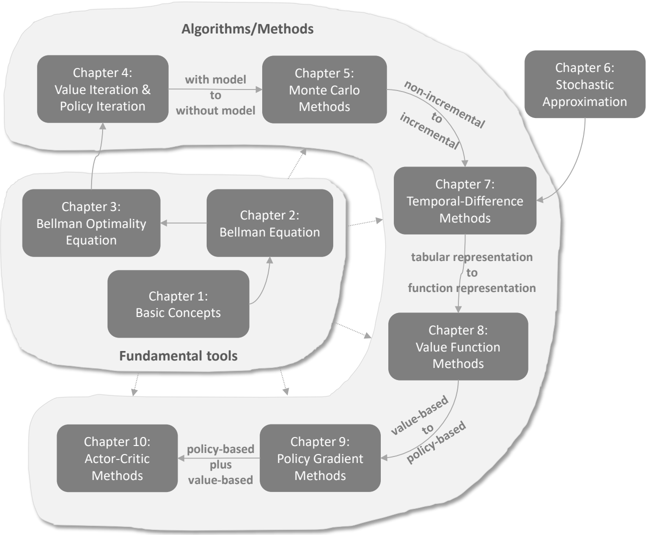
Lecture 9: Policy Gradient Methods

Shiyu Zhao

Department of Artificial Intelligence

Westlake University

Outline


Algorithms/Methods
Chapter 4: Value Iteration & Policy Iteration
with model to without model
Chapter 5: Monte Carlo Methods
non-incremental to incremental
Chapter 6: Stochastic Approximation
Chapter 7: Temporal-Difference Methods
tabular representation to function representation
Chapter 8: Value Function Methods
value-based to policy-based
Chapter 9: Policy Gradient Methods
policy-based plus value-based
Chapter 10: Actor-Critic Methods
Chapter 3: Bellman Optimality Equation
Chapter 2: Bellman Equation
Chapter 1: Basic Concepts
Fundamental tools

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)

Outline

Outline

Basic idea of policy gradient

Previously, policies have been represented by tables:

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

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

Basic idea of policy gradient

Now, policies can be represented by parameterized functions:

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

where $\theta \in \mathbb{R}^m$ is a parameter vector.

- The function can be, for example, a neural network, whose input is s, output is the probability to take each action, and parameter is θ.
- **Advantage:** when the state space is large, the tabular representation will be of low efficiency in terms of storage and generalization.
- The function representation is also sometimes written as $\pi(a, s, \theta)$, $\pi_\theta(a|s)$, or $\pi_\theta(a, s)$.

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*.

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 $\pi(a|s,\theta)$ given the function structure and the parameter.

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.

- What appropriate metrics should be used?
- How to calculate the gradients of the metrics?

These questions will be answered in detail in this lecture.

Outline

Outline

Metric 1: average value

The first metric is the average state value or simply 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

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

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. Details of stationary distribution can be found in the last lecture and the book.
- The interpretation of selecting d_π is as follows.
  - d_π reflects the long-run behavior of the Markov decision process 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.

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

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

Outline

Metric 2: average reward

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

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

where $S \sim d_\pi$,

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

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

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.

An important equivalent expression:

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

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

where s_0 is the starting state of the trajectory.

Metrics to define optimal policies - Remarks

An important fact is that

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

Remarks:

- Highlight: the starting state s_0 does not matter.
- The derivation of the equation is nontrivial and can be found in my book.

Outline

Summary of the two metrics

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

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

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.

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 in this book. For details about the undiscounted case, see the book.

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. See the proof in the book.

Outline

Gradients of the metrics

Given a metric, we next

- derive 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 discounted and undiscounted cases.

Gradients of the metrics

I simply give the expression of the gradient without proof:

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

The above is a unified expression of many cases:

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

The derivation of this expression is **very complex**.
Details are not given here. Interested readers can read my book.
For most readers, it is sufficient to know this expression.

A compact and important expression of the gradient:

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

First, why is this expression useful?

- Because we can use samples to approximate the gradient:

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

  where s, a are samples. This is the idea of stochastic gradient descent.

Gradients of the metrics

A compact and important expression of the gradient:

$$\nabla_\theta J(\theta) = \sum_{s\in\mathcal{S}} \eta(s) \sum_{a\in\mathcal{A}} \nabla_\theta \pi(a|s,\theta) q_\pi(s,a)$$
$$= \mathbb{E}_{S\sim\eta, A\sim\pi}\big[\nabla_\theta \ln \pi(A|S,\theta) q_\pi(S,A)\big]$$

Second, how to prove the above equation?

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

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

and hence

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

A compact and important expression of the gradient:

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

Proof (continued): Then, we have

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

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

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

- 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

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

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

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 $\pi(a|s,\theta)$ for an action a. The activation function of the output layer should be softmax.
- Since $\pi(a|s,\theta) > 0$ 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.

Outline

Gradient-ascent algorithm

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

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

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

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

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

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

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

- 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 θ; $\gamma \in (0,1)$; $\alpha > 0$.
Goal: Learn an optimal policy to maximize $J(\theta)$.

For each episode, do
    Generate an episode $\{s_0, a_0, r_1, \dots, s_{T-1}, a_{T-1}, r_T\}$ following $\pi(\theta)$.
    For $t = 0, 1, \dots, T-1$:
        Value update: $q_t(s_t, a_t) = \sum_{k=t+1}^{T} \gamma^{k-t-1} r_k$
        Policy update: $\theta \leftarrow \theta + \alpha \nabla_\theta \ln \pi(a_t|s_t, \theta) q_t(s_t, a_t)$

Remark 1: How to do sampling?

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

- 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?
  - $A \sim \pi(A|S,\theta)$. Hence, a_t should be sampled following $\pi(\theta_t)$ at s_t.
  - Therefore, policy gradient methods are on-policy.

Remark 2: How to interpret this algorithm?

Since

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

the algorithm can be rewritten as

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

Therefore, we have the important expression of the algorithm:

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

The interpretation of

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

is as follows. Suppose that α is sufficiently small.

Interpretation:

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

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

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

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

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

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

Gradient-ascent algorithm

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

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

The reason is as follows.

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

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

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

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

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

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

Outline

Summary

Contents of this lecture:

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

Next lecture: Actor-critic