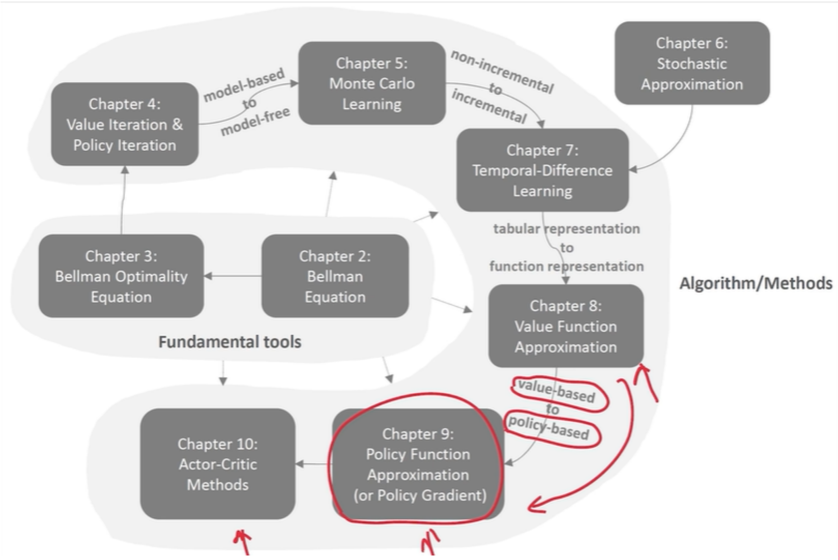
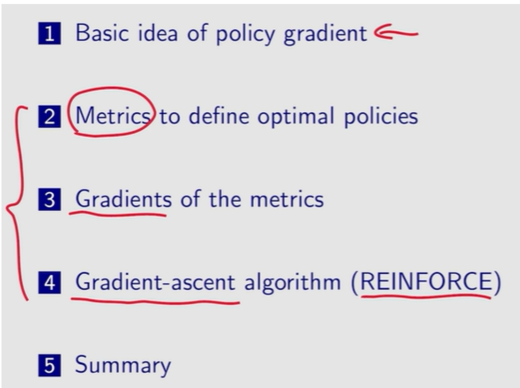
RL-9-策略梯度方法

[第9课-策略梯度方法（该方法的基本思路）_哔哩哔哩_bilibili]

目前最流行的方法；policy-base 直接建立有关于policy的目标函数，优化该目标得到最优策略；



主要内容：



基本概念

- 函数代替表格来表示 π；
- 优点：可以表示连续的s，泛化性也更好；

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 $\pi_\theta(a,s)$.

三个基本的不同点

如何定义最优策略?

- 表格型：对所有state s，$v_{\pi^*}(s) \geq v_\pi(s)$；
- 函数型：使用scaler metrics；

如何获取一个action的概率?

- 需要通过网络进行一次计算；

如何更新策略?

- 通过改变函数的参数 θ 来更新策略；

求解问题的基本思路

- 定义目标函数/metrics； （怎么取？）
- 优化，求解最优参数（最优policy） （如何计算gradients？）

Metrics

average value

定义为所有state-value的加权平均，权重d为S出现的概率分布；

The first metric is the average state value or simply called average value. In particular, the metric is defined as

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

- $\bar{v}_\pi$ 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

$$\bar{v}_\pi = \mathbb{E}[v_\pi(S)] = \sum_s p(s) V_\pi(s)$$

where $S \sim d$.

如何确定分布d？有两种情况：

- d 独立于 π
  - 求梯度的时候比较简单；
  - 写成 d_0；
  - 所有state出现的概率相同，则均匀分布；

- 非常关心某一个状态 s_0 ，则极端情况下 $d_0(s_0)=1$；
- d 与 π 有关
  - 平稳概率：直接求解 d_π，不动点 $d_\pi^T P_\pi = d_\pi^T$；
  - 在策略pi下，每个状态会被访问的概率；
- 也可以写出另一种形式

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

Answer: First, clarify and understand this metric.

- It starts from $S_0 \sim d$ and then $A_0, R_1, S_1, A_1, R_2, S_2, \ldots$
- $A_t \sim \pi(S_t)$ and $R_{t+1}, S_{t+1} \sim p(R_{t+1}|S_t, A_t), p(S_{t+1}|S_t, A_t)$

Then, we know this metric is the same as the average value because

$$J(\theta)=\mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t R_{t+1}\right]=\sum_{s\in\mathcal{S}}d(s)\mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t R_{t+1}|S_0=s\right]$$

average reward

定义为单步reward的平均；

The second metric is average one-step reward or simply average reward. In particular, the metric is

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

where $S \sim d_\pi$. Here,

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

is the average of the one-step immediate reward that can be obtained starting from state s, and

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

- The weight d_π is the stationary distribution.
- As its name suggests, $\bar{r}_\pi$ is simply a weighted average of the one-step immediate rewards.

等价形式：一个经过无穷多步的轨迹中获得的reward平均到每一步上的值；

- Suppose an agent follows a given policy and generate a trajectory with the rewards as $(R_{t+1}, R_{t+2}, \ldots)$.
- The average single-step reward along this trajectory is

$$\begin{aligned}&\lim_{n\to\infty}\frac{1}{n}\mathbb{E}\left[R_{t+1}+R_{t+2}+\cdots+R_{t+n}|S_t=s_0\right]\\ &=\lim_{n\to\infty}\frac{1}{n}\mathbb{E}\left[\sum_{k=1}^{n}R_{t+k}|S_t=s_0\right]\end{aligned}$$

where s_0 is the starting state of the trajectory.

忽略其实状态（既然是无穷多步，根据无穷级数的性质，那么其实位置不重要）

$$\lim_{n\to\infty}\frac{1}{n}\mathbb{E}\left[\sum_{k=1}^{n}R_{t+k}|S_t=s_0\right]=\lim_{n\to\infty}\frac{1}{n}\mathbb{E}\left[\sum_{k=1}^{n}R_{t+k}\right]$$

- 以上2个metrics都是策略的函数；函数使用参数theta描述；

- 考虑了discounted case 和 undiscounted case；
- 这两个metrics是等价的，可以相互转化

Remark 3 about the metrics:

- Intuitively, $\bar{r}_\pi$ is more short-sighted because it merely considers the immediate rewards, whereas $\bar{v}_\pi$ considers the total reward overall steps.
- However, the two metrics are equivalent to each other. In the discounted case where $\gamma < 1$, it holds that

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

See the proof in the book.

Gradients of the metrics

- 这些metrics的梯度计算是整个方法中最复杂的部分！
  统一表示

Summary of the results about the gradients:

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

where

- $J(\theta)$ 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.

$=$
$\approx$
$\propto$

等价表达

A compact and useful form of the gradient:

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

where $S \sim \eta$ and $A \sim \pi(A|S,\theta)$.

Why is this expression useful?

- Because we can use samples to approximate the gradient!

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

推导

Then, we have

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

可以用随机梯度下降代替expectation；

重要性质

- $\pi(a|s,\theta) > 0$，使用softmax函数来保证；同时满足了归一化条件；

Some remarks: Because we need to calculate $\ln \pi(a|s,\theta)$, we must ensure that for all s, a, θ

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

- This can be archived 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 $\sum_{i=1}^{n} z_i = 1$.
- Then, 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.

h是feature function，用一个神经网络代替；

- 上面的softmax，action如果是无穷多个怎么办？这种方法失效，要用DPG（deterministic policy gradient）

梯度上升算法-REINFORCE

- Furthermore, since q_π is unknown, it can be approximated:

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

  There are different methods to approximate $q_\pi(s_t,a_t)$
  - In this lecture, Monte-Carlo based method, *REINFORCE*
  - In the next lecture, TD method and more

由于 q_π 是未知的，因此需要近似表达；

采样

- S 服从 d 分布，d理论上是平稳分布，实际上不太考虑；
- A服从pi分布，那么应该根据 $\pi(\theta_t)$ 得到 s_t；on-policy的策略；

理解算法

变形

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:

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

优化 $\pi(a_t|s_t)$ 值；若beta>0，那么更新得到的theta得到的新的pi确实比之前的更大，也就是梯度上升；

- 从q出发，更新了更好的pi，也就是exploitation；
- 对于分母上，若之前选择的pi很小，那么下一次更新时就会增大pi，也就是exploration；

REINFORCE

- q_t：用MC策略求得；

Pseudocode: Policy Gradient by Monte Carlo (REINFORCE)

Initialization: A parameterized function $\pi(a|s,\theta)$, $\gamma \in (0,1)$, and $\alpha > 0$.
Aim: Search for an optimal policy maximizing $J(\theta)$.

For the kth iteration, do

Select s_0 and generate an episode following $\pi(\theta_k)$. Suppose the episode is $\{s_0, a_0, r_1, \ldots, s_{T-1}, a_{T-1}, r_T\}$.

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

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

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

$\theta_k = \theta_T$

MC是off-line的方法，要采集完所有数据才能更新；