强化学习基本原理及编程实现07：基于策略梯度的方法

郭宪

2019.11.10

人工智能学院
College of Artificial Intelligence

Nankai University

强化学习的问题形式化

序贯决策问题可以形式化为：

$$\max_{\pi} J(\pi)=\sum_{s,a}\mu^{\pi}(s)\pi(s,a)R(s,a)$$

$$s.t.\ \mu^{\pi}(s')=\sum_{s,a}\mu^{\pi}(s)\pi(s,a)T(s,a,s')$$

$$1=\sum_{s,a}\mu^{\pi}(s)\pi(s,a)$$

$$\pi\left(s,a\right)\geq 0,\ \forall s\in S, a\in A$$

1. 该问题的原问题是直接得到最优策略。基于直接策略搜索的强化学习。
2. 该问题的对偶问题是得到最优的值函数，然后由值函数构建最优策略。基于值函数的强化学习。

动态规划的本质

动态规划的本质是：将多阶段决策问题通过贝尔曼方程转化为多个单阶段的决策问题

离散贝尔曼方程：

$$J^*[x(j),j] = \min_{u(j)\in U} \min_{\{u(j+1),\cdots,u(N-1)\}\in U} \left\{ L[x(j),u(j),j] + \sum_{k=j+1}^{N-1} L[x[k],u[k],k] \right\}$$

$$= \min_{\substack{u(j)\in U \\ x(j+1)\in X}} \left\{ L(x(j),u(j),j) + J^*[x(j+1),j+1] \right\}$$

$$= \min_{\substack{u(j)\in U \\ x(j+1)\in X}} \left\{ L(x(j),u(j),j) + J^*[f[x(j),u(j),j],j+1] \right\}$$

求出值函数后，通过贪婪策略重构出最优策略

最优控制中的动态规划

动态规划的本质是：将多阶段决策问题通过贝尔曼方程转化为多个单阶段的决策问题

连续贝尔曼方程：

$$J^*[x(t),t]=\min_{u[t,t+\Delta t]}\{\min_{u[t+\Delta t,t_f]}\left[\int_t^{t+\Delta t}L(x(\tau),u(\tau),\tau)\right]d\tau+\int_{t+\Delta t}^{t_f}L(x(\tau),u(\tau),\tau)d\tau+\varphi[x(t_f),t_f]\}$$

$$=\min_{\substack{u(\tau)\in U\\ t\le\tau\le t+dt}}\left\{\int_t^{t+dt}L[x(\tau),u(\tau),\tau]d\tau+J^*[x(t)+dx(t),t+dt]\right\} \quad （1）$$

将 $J^*[x(t)+dx(t),t+dt]$ 进行泰勒展开有：

$$J^*[x(t)+dx(t),t+dt]=J^*[x(t),t]+\frac{\partial J^*[x(t),t]}{\partial x^T(t)}dx(t)+\frac{\partial J^*[x(t),t]}{\partial t}dt+\varepsilon[dx(t),dt] \quad （2）$$

将（2）带入（1），并令 $dt\to 0$ Hamilton-Jacobi-Bellman方程

胡寿松等， 最优控制理论与系统，科学出版社，2005

$$-\frac{\partial J^*[x(t),t]}{\partial t}=\min_{u(t)\in U}\{L[x(t),u(t),t]+\frac{\partial J^*[x(t),t]}{\partial x^T(t)}f[x(t),u(t),t]\}\xRightarrow{\text{重构}}\min_{u(t)\in U}\left\{L[x(t),u(t),t]+\frac{\partial J^*[x(t),t]}{\partial x^T(t)}f[x(t),u(t),t]\right\}$$

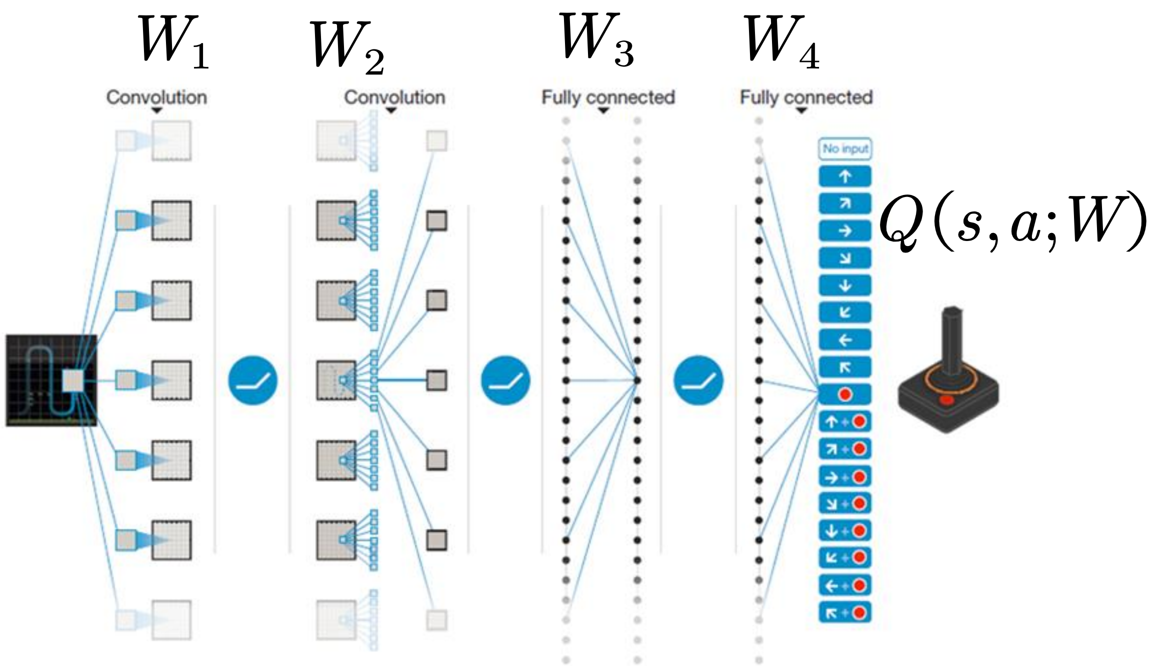
为什么要策略搜索？

表格型强化学习：

	a_1	a_2	a_3	a_4	a_5
s_1	$Q(s_1,a_1)$	$Q(s_1,a_2)$	$Q(s_1,a_3)$	$Q(s_1,a_4)$	$Q(s_1,a_5)$
s_2	$Q(s_2,a_1)$	…	…	…	…
s_3	…	…	…	…	…
s_4	…	…	…	…	…
s_5	…	…	…	…	…
s_6	…	…	…	…	…
s_7	…	…	…	…	…

$$\pi(s) = arg \max_a Q(s,a)$$

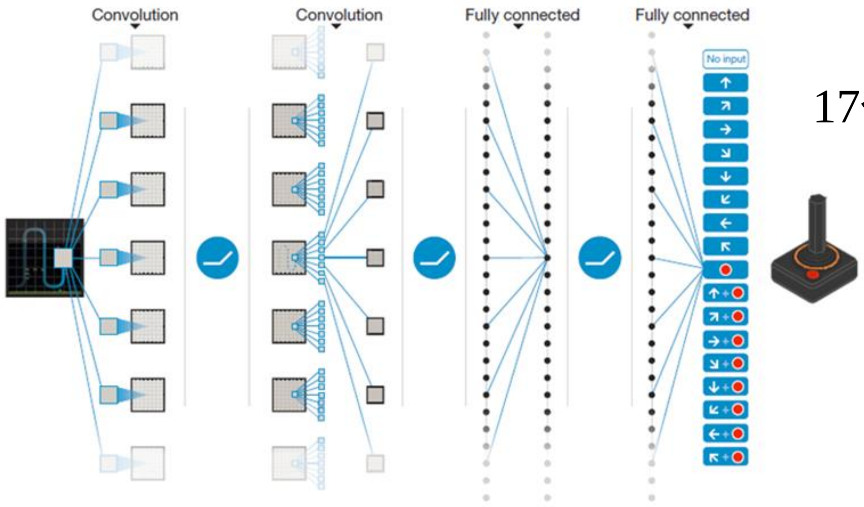
函数逼近强化学习：



$$\pi(s) = arg \max_a Q_W(s,a)$$

1. MC方法； 2. TD方法

动作的选择一定要值函数？



17个动 作

利用值函数学习时，策略改进需要求解：

$$arg \ \max_{a} \ Q_w(s,a)$$

当要解决的问题的动作空间很大或连续时，该式无法求解。

动作的选择一定要行为值函数吗？

未必！

可以直接得到最优的策略。

策略的定义：

一个策略 π是给定状态s时，动作集上的一个分布：

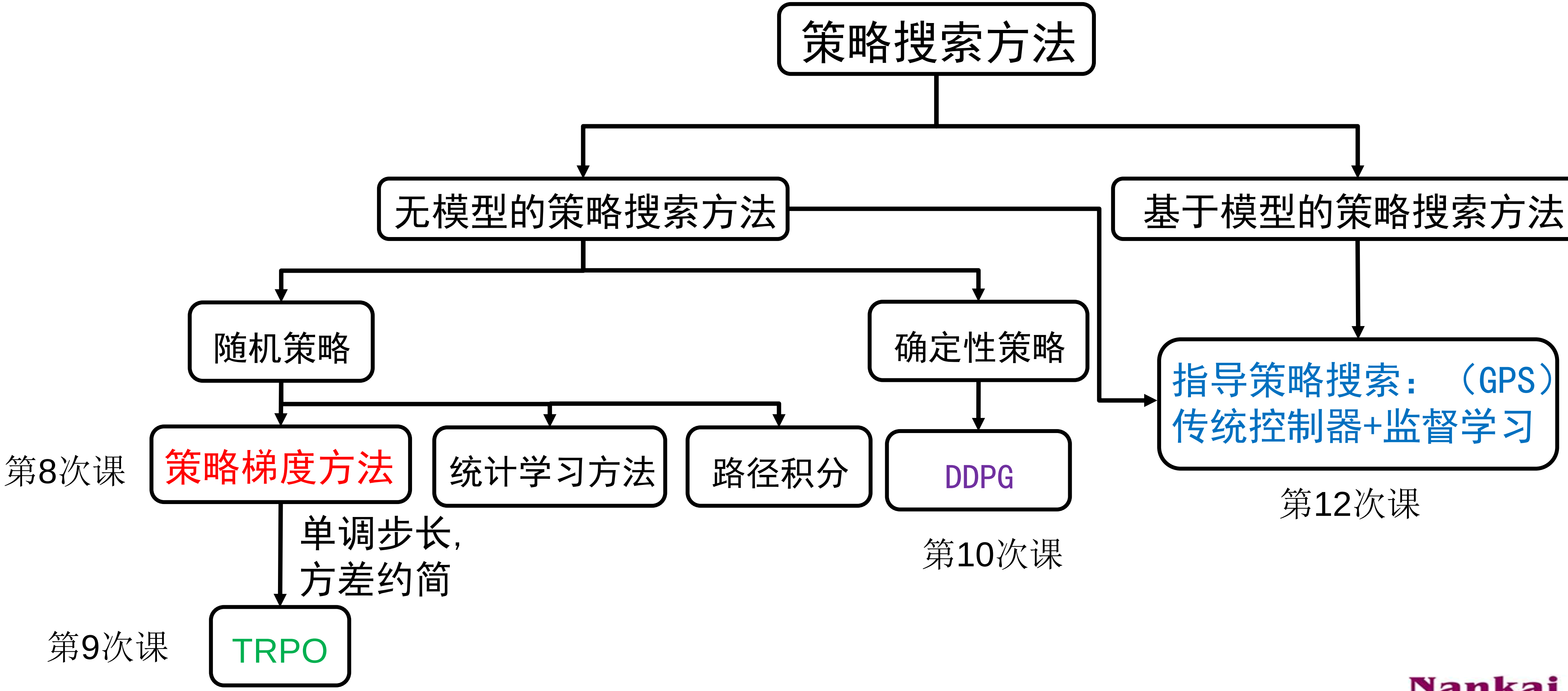
$$\pi(a|s) = p[A_t = a|S_t = s]$$

常见的直接策略搜索

常用的直接策略搜索的方法：

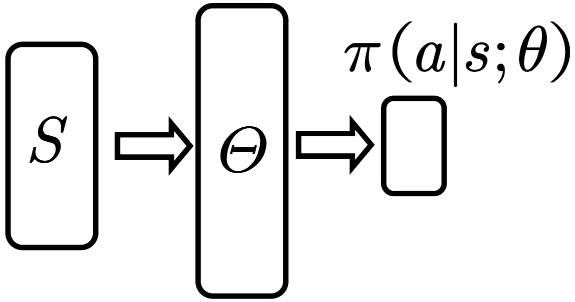
1. 利用梯度的方法
2. 基于EM的方法
3. 基于路径积分的方法
4. 基于模型的方法
5. 基于粒子滤波的方法

策略搜索方法分类

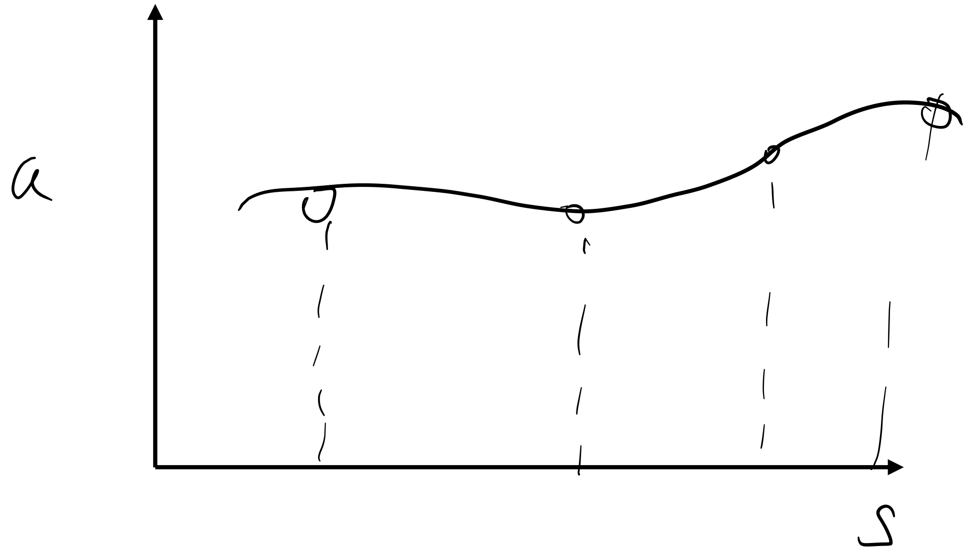

策略搜索方法
无模型的策略搜索方法
基于模型的策略搜索方法
随机策略
确定性策略
指导策略搜索：（GPS）
传统控制器+监督学习
第8次课
策略梯度方法
统计学习方法
路径积分
DDPG
第12次课
单调步长，
方差约简
第10次课
第9次课
TRPO

策略进行参数化

策略的表示：



学习：Θ



如何获得样本点？

标签？ **值函数的帮助！强化学习**

策略搜索的好处

（1） 对于离散的动作空间，参数化策略可表示为：

$$\pi(a|s;\theta)=\frac{e^{h(s,a;\theta)}}{\Sigma_b e^{h(s,b;\theta)}}$$

该策略表示可以逼近一个确定性策略，而 $\varepsilon-greedy$ 不能逼近确定性策略。

与玻尔兹曼策略的区别：

$$\pi(a|s;\theta)=\frac{e^{\frac{Q(s,a)}{\tau}}}{\Sigma_b e^{\frac{Q(s,b)}{\tau}}}$$

但是玻尔兹曼策略不允许逼近确定性策略。动作值估计将收敛到相应的真实值。收敛到行为值函数对应的概率真实分布。

（2） 参数化的策略可以逼近任意概率分布，不受行为值函数的限制

（3） 策略是更简单的函数逼近，如PID控制

（4） 策略参数化更容易加入先验知识

似然率策略梯度

用 τ 表示一组状态-行为序列$s_0, u_0, \cdots, s_H, u_H$

重 载符号：$R(\tau)=\sum_{t=0}^{H} R(s_t, u_t)$

目标函数为：

$$U(\theta)=E\left(\sum_{t=0}^{H} R(s_t, u_t); \pi_\theta\right)=\sum_{\tau} P(\tau;\theta) R(\tau)$$

强化学习的目标是找到最优参数 θ 使得：

$$\max_\theta U(\theta)=\max_\theta \sum_{\tau} P(\tau;\theta) R(\tau)$$

目标函数对参数求导：

$$\begin{aligned}
\nabla_\theta U(\theta) &= \nabla_\theta \sum_{\tau} P(\tau;\theta) R(\tau) \\
&= \sum_{\tau} \nabla_\theta P(\tau;\theta) R(\tau) \\
&= \sum_{\tau} \frac{P(\tau;\theta)}{P(\tau;\theta)} \nabla_\theta P(\tau;\theta) R(\tau) \\
&= \sum_{\tau} P(\tau;\theta) \frac{\nabla_\theta P(\tau;\theta) R(\tau)}{P(\tau;\theta)} \\
&= \sum_{\tau} P(\tau;\theta) \nabla_\theta \log P(\tau;\theta) R(\tau)
\end{aligned}$$

利用经验平均估计策略的梯度：

$$\nabla_\theta U(\theta) \approx \hat{g}=\frac{1}{m} \sum_{i=1}^{m} \nabla_\theta \log P(\tau;\theta) R(\tau)$$

从重要性采样的视角进行推导

目标函数为：

$$U(\theta)=E\left(\sum_{t=0}^{H}R(s_t,u_t);\pi_\theta\right)=\sum_{\tau}P(\tau;\theta)R(\tau)$$

利用重要性采样，可以利用已知参数为 θ_{old} 的策略产生的数据，对任意参数为θ 的策略进行评估：

$$U(\theta)=E_{\tau\sim\theta_{old}}\left[\frac{P(\tau|\theta)}{P(\tau|\theta_{old})}R(\tau)\right]$$

导数为：

$$\nabla_\theta U(\theta)=E_{\tau\sim\theta_{old}}\left[\frac{\nabla_\theta P(\tau|\theta)}{P(\tau|\theta_{old})}R(\tau)\right]$$

同分布评价：

$$\nabla_\theta U(\theta)|_{\theta=\theta_{old}}=E_{\tau\sim\theta_{old}}\left[\frac{\nabla_\theta P(\tau|\theta)|_{\theta_{old}}}{P(\tau|\theta_{old})}R(\tau)\right]$$

$$\nabla_\theta U(\theta)|_{\theta=\theta_{old}}=E_{\tau\sim\theta_{old}}[\nabla_\theta \log P(\tau|\theta)|_{\theta_{old}}R(\tau)]$$

不仅仅能推导出策略梯度公式，我们还能得到新的损失函数：

$$U(\theta)=E_{\tau\sim\theta_{old}}\left[\frac{P(\tau|\theta)}{P(\tau|\theta_{old})}R(\tau)\right]$$

似然率策略梯度的直观理解

利用经验平均估计策略的梯度：

$$\nabla_\theta U(\theta) \approx \hat{g} = \frac{1}{m}\sum_{i=1}^{m} \nabla_\theta \log P(\tau;\theta) R(\tau)$$

$$\nabla_\theta \log P(\tau^{(i)};\theta) R(\tau^{(i)})$$

增加高回报路径的概率，减小低回报路径的概率

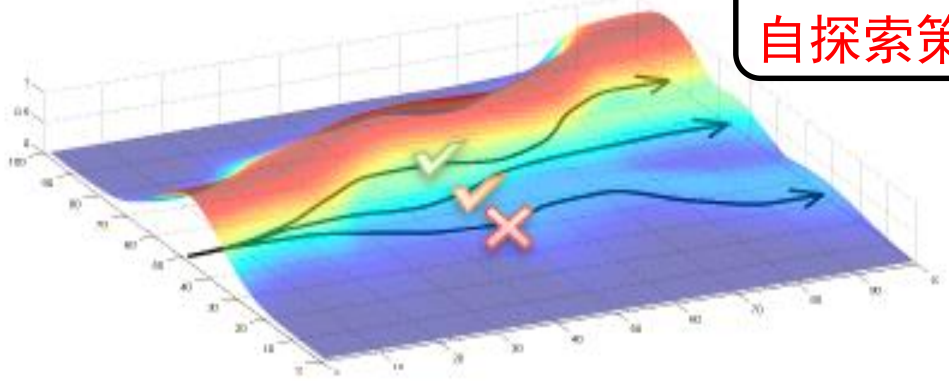
$$\nabla_\theta \log P(\tau^{(j)};\theta) R(\tau^{(j)})$$

$$\nabla_\theta \log P(\tau^{(i)};\theta) R(\tau^{(i)})$$

只改变经验路径的概率，并不改变路径。

高回报区域

三条路径来自探索策略



策略梯度的直观理解图

路径似然率推导

利用经验平均估计策略的梯度：

$$\nabla_\theta U(\theta) \approx \hat{g} = \frac{1}{m}\sum_{i=1}^{m} \nabla_\theta \log P(\tau^{(i)};\theta) R(\tau^{(i)})$$

$$R(\tau) = \sum_{t=0}^{H} R(s_t, u_t)$$

$$\nabla_\theta \log P(\tau;\theta) ?$$

$$\tau = s_0, u_0, \cdots, s_H, u_H$$

$$P(\tau^{(i)};\theta) = \prod_{t=0}^{H} \underbrace{\boxed{P(s_{t+1}^{(i)}|s_t^{(i)}, u_t^{(i)})}}_{\text{动力学}} \cdot \overbrace{\boxed{\pi_\theta(u_t^{(i)}|s_t^{(i)})}}^{\text{策略}}$$

路径似然率：

$$\nabla_\theta \log P(\tau^{(i)};\theta) = \nabla_\theta \log\left[\prod_{t=0}^{H} P(s_{t+1}^{(i)}|s_t^{(i)}, u_t^{(i)}) \cdot \pi_\theta(u_t^{(i)}|s_t^{(i)})\right]$$

$$= \nabla_\theta \left[\sum_{t=0}^{H} \log P(s_{t+1}^{(i)}|s_t^{(i)}, u_t^{(i)}) + \sum_{t=0}^{H} \log \pi_\theta(u_t^{(i)}|s_t^{(i)})\right]$$

$$= \nabla_\theta \left[\sum_{t=0}^{H} \log \pi_\theta(u_t^{(i)}|s_t^{(i)})\right]$$

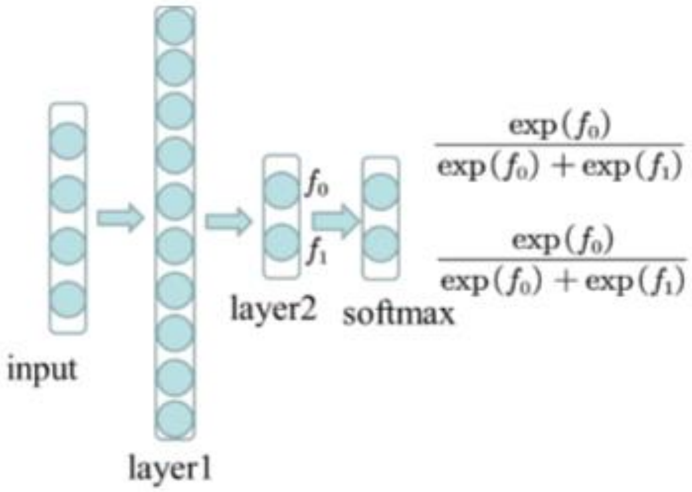
$$= \sum_{t=0}^{H} \nabla_\theta \log \pi_\theta(u_t^{(i)}|s_t^{(i)})$$

只与策略有关，不要求动力学已知

$$\log \pi_\theta(u_t^{(i)}|s_t^{(i)}) \quad ?$$

常见的策略表示：离散动作空间

求解： $\nabla_\theta \log \pi_\theta(u_t^{(i)}|s_t^{(i)})$



常见的策略表示：连续动作空间

求解： $\nabla_\theta \log \pi_\theta(u_t^{(i)} | s_t^{(i)})$

随机策略可以写为确定性策略加随机部分，即：

$$\pi_\theta = \mu_\theta + \varepsilon$$

其中 $\varepsilon \sim N(0, \sigma^2)$ 是均值为零，标准差为 σ 的正态分布

其中确定性部分常见的表示为：

线性策略： $\mu(s) = \phi(s)^T \theta$

径向基策略： $\pi_\theta(s) = \omega^T \phi(s)$,

其中： $\phi_i(s) = \exp\left(-\frac{1}{2}(s-\mu_i)^T D_i (s-\mu_i)\right)$

参数为： $\theta = \{\omega, \mu_i, d_i\}$

以线性策略为例：

$$\pi(u|s) \sim \frac{1}{\sqrt{2\pi}\,\sigma} \exp\left(-\frac{(u-\phi(s)^T\theta)^2}{2\sigma^2}\right)$$

$$\nabla_\theta \log \pi_\theta(u_t^{(i)} | s_t^{(i)}) = \frac{(u_t^{(i)} - \phi(s_t^{(i)})^T \theta)\phi(s_t^{(i)})}{\sigma^2}$$

方差参数用来控制策略的探索性

REINFORCE: Monte Carlo Policy Gradient

REINFORCE 更新

$$\theta_{t+1} \doteq \theta_t + \alpha G_t \nabla \log \pi(a_t|s_t;\theta_t)$$

REINFORCE: Monte-Carlo Policy-Gradient Control (episodic) for π_*

Input: a differentiable policy parameterization $\pi(a|s,\boldsymbol{\theta})$
Algorithm parameter: step size $\alpha > 0$
Initialize policy parameter $\boldsymbol{\theta} \in \mathbb{R}^{d'}$ (e.g., to $\mathbf{0}$)

Loop forever (for each episode):
  Generate an episode $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$, following $\pi(\cdot|\cdot,\boldsymbol{\theta})$
  Loop for each step of the episode $t = 0, 1, \ldots, T-1$:
    $G \leftarrow \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k$ (G_t)
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha \gamma^t G \nabla \ln \pi(A_t|S_t,\boldsymbol{\theta})$

减小方差方法：基线

利用经验平均估计策略的梯度：

$$\nabla_\theta U(\theta) \approx \hat{g} = \frac{1}{m}\sum_{i=1}^{m} \nabla_\theta \log P(\tau^{(i)};\theta) R(\tau^{(i)})$$

$$\Downarrow$$

$$\nabla_\theta U(\theta) \approx \hat{g} = \frac{1}{m}\sum_{i=1}^{m} \left(\sum_{t=0}^{H} \nabla_\theta \log \pi_\theta (u_t^{(i)} | s_t^{(i)}) R(\tau^{(i)}) \right)$$

该式给出的策略梯度是无偏的，但是方差很大。

$$\nabla_\theta U(\theta) \approx \hat{g} = \frac{1}{m}\sum_{i=1}^{m} \nabla_\theta \log P(\tau^{(i)};\theta) R(\tau^{(i)})$$

$$= \frac{1}{m}\sum_{i=1}^{m} \nabla_\theta \log P(\tau^{(i)};\theta) \left(R(\tau^{(i)}) - b \right)$$

$$E[\nabla_\theta \log P(\tau;\theta) b]$$

$$= \sum_\tau P(\tau;\theta) \nabla_\theta \log P(\tau;\theta) b$$

$$= \sum_\tau P(\tau;\theta) \frac{\nabla_\theta P(\tau;\theta) b}{P(\tau;\theta)}$$

$$= \sum_\tau \nabla_\theta P(\tau;\theta) b$$

$$= \nabla_\theta \left(\sum_\tau P(\tau;\theta) b \right)$$

$$= \nabla_\theta b$$

$$= 0$$

减小方差方法：基线

利用经验平均估计策略的梯度：

$$\nabla_\theta U(\theta) \approx \hat{g} = \frac{1}{m}\sum_{i=1}^{m} \nabla_\theta \log P(\tau^{(i)};\theta) R(\tau^{(i)})$$

$$\Downarrow$$

$$\nabla_\theta U(\theta) \approx \hat{g} = \frac{1}{m}\sum_{i=1}^{m} \left(\sum_{t=0}^{H} \nabla_\theta \log \pi_\theta(u_t^{(i)}|s_t^{(i)}) R(\tau^{(i)})\right)$$

该式给出的策略梯度是无偏的，但是方差很大。

$$\nabla_\theta U(\theta) \approx \hat{g} = \frac{1}{m}\sum_{i=1}^{m} \nabla_\theta \log P(\tau^{(i)};\theta) R(\tau^{(i)})$$

$$= \frac{1}{m}\sum_{i=1}^{m} \nabla_\theta \log P(\tau^{(i)};\theta) \left(R(\tau^{(i)}) - b\right)$$

如何取b使得方差最小？

令$X = \nabla_\theta \log P(\tau^{(i)};\theta) \left(R(\tau^{(i)}) - b\right)$

则X的方差为：

与b无关

$$Var(X) = E\left(X - \bar{X}\right)^2 = EX^2 - E\bar{X}^2$$

$$\frac{\partial Var(X)}{\partial b} = E\left(X \frac{\partial X}{\partial b}\right) = 0$$

$$b = \frac{E_p\left[\left(\sum_{t=0}^{H} \nabla_\theta \log \pi_\theta(u_t^{(i)}|s_t^{(i)})\right)^2 R(\tau)\right]}{E_p\left[\left(\sum_{t=0}^{H} \nabla_\theta \log \pi_\theta(u_t^{(i)}|s_t^{(i)})\right)^2\right]}$$

减小方差方法：修改值函数

REINFORCE方法，1992, R.J.Williams

$$\nabla_\theta U(\theta) \approx \frac{1}{m}\sum_{i=1}^{m}\left(\sum_{t=0}^{H}\nabla_\theta \log\pi_\theta(u_t^{(i)}|s_t^{(i)})\left(R(\tau^{(i)})-b\right)\right)$$

Policy Gradient Theorem:1999, R. Sutton

当前的动作与过去的回报无关：

$$E_p[\partial_\theta \log\pi_\theta(u_t|x_t,t)r_j]=0 \quad for \quad j<t$$

$$\nabla_\theta U(\theta) \approx \frac{1}{m}\sum_{i=1}^{m}\sum_{t=0}^{H-1}\nabla_\theta \log\pi_\theta(u_t^{(i)}|s_t^{(i)})\left(\sum_{k=t}^{H-1}\left(R(s_k^{(i)})-b\right)\right)$$

τ： $x_0, u_0, r_0, x_1, u_1, r_1, \cdots$

$$R(\tau)=\sum_{t=0}^{H}R(s_t,u_t)$$

τ： $x_0, u_0, r_0, s_1, u_1, r_1, s_2, u_2, r_2 \ldots$

减小方差方法：修改值函数

当前的动作与过去的回报无关：

$$E_p[\partial_\theta \log \pi_\theta(u_t|x_t,t) r_j] = 0 \quad for \quad j < t$$

$$\nabla_\theta U(\theta) \approx \frac{1}{m}\sum_{i=1}^{m}\sum_{t=0}^{H-1}\nabla_\theta \log \pi_\theta(u_t^{(i)}|s_t^{(i)})\left(\sum_{k=t}^{H-1}(R(s_k^{(i)})-b)\right)$$

τ: $x_0, u_0, r_0, s_1, u_1, r_1, s_2, u_2, r_2 \ldots$

当前的回报只与过去的动作有关 (G(PO)MDP) ：

$$\nabla_\theta U(\theta) \approx \frac{1}{m}\sum_{i=1}^{m}\sum_{j=0}^{H-1}\left(\sum_{t=0}^{j}\nabla_\theta \log \pi_\theta(u_t^{(i)}|s_t^{(i)})\,(r_j - b_j)\right)$$

τ: $x_0, u_0, r_0, s_1, u_1, r_1, s_2, u_2, r_2 \ldots$

Mujoco环境配置

（1）官方网站https://www.roboti.us/license.html注册，得到注册码和许可文件

（2）官方网站https://www.roboti.us/index.html 下载mjpro131 win64，并解压缩到文件夹中

（3）将注册码和许可文件复制到压缩文件夹中

（4）安装mujoco_py。pip install mujoco_py==0.5.7

（5）在mujoco_py的文件夹下的config.py文件中修改_key_path 和 mjpro_path

（6）在mujoco_py的文件夹下的mjlib.py文件中修改bin/mujoco131.lib为bin/mujoco131.dll

（7）在mujoco_py的文件夹下的platname_targdir.py中修改platname==”win”

（8）若要用gym需要修改成0.9.1版本， pip install gym==0.9.1

https://github.com/reinforcement-learning-kr/pg_travel

第六次作业

1. 阅读《Reinforcement Learning： An Introduction》第13章

2. 利用策略梯度的方法控制倒立摆

3. 利用策略梯度的方法解决其他问题。

4. 利用策略梯度的方法解决闲聊机器人对话生成问题