MDP Preliminaries

Nan Jiang

April 10, 2019

【强化学习理论 57】StatisticalRL 1

张楚珩
清华大学 交叉信息院博士在读

这是UIUC姜楠老师开设的CS598统计强化学习（理论）课程的第一讲（第一部分），由于最近在研究state abstraction相关的东西，因此准备借此机会刷一下这个课程，强化一下自己在RL理论方面的水平。

原文传送门

CS598 Note1
nanjiang.cs.illinois.edu

备注

此系列Notes写的很好，如果大家有兴趣可以直接去看Note。我这里主要贴截图，在重难点的地方讲解一下（如果我能搞懂的话）。大家不懂的地方可以上来看看我这里有没有讲清楚。

一、马可夫决策过程

首先是定义

- State space $\mathcal{S}$. In this course we only consider finite state spaces.
- Action space $\mathcal{A}$. In this course we only consider finite action spaces.
- Transition function $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$, where $\Delta(\mathcal{S})$ is the space of probability distributions over $\mathcal{S}$ (i.e., the probability simplex). $P(s'|s, a)$ is the probability of transitioning into state s' upon taking action a in state s.
- Reward function $R : \mathcal{S} \times \mathcal{A} \to [0, R_{\max}]$, where $R_{\max} > 0$ is a constant. $R(s, a)$ is the immediate reward associated with taking action a in state s.
- Discount factor $\gamma \in [0, 1)$, which defines a horizon for the problem.

比较特别的是

- 这里的概率转移函数 P 是state-action到状态空间上概率分布 $\Delta(\mathcal{S})$ 的函数，写的比较形式化；
- 假设了奖励都是非负数，并且有界，应该是为了后面的分析方便；这其实不影响，有界不是一个很强的假设，并且非负只需要把reward都平移一下即可。

1.1. 与环境的交互

In a given MDP $M = (\mathcal{S}, \mathcal{A}, P, R, \gamma)$, the agent interacts with the environment according to the following protocol: the agent starts at some state s_1; at each time step $t = 1, 2, \ldots$, the agent takes an action $a_t \in \mathcal{A}$, obtains the immediate reward $r_t = R(s_t, a_t)$, and observes the next state s_{t+1} sampled from $P(s_t, a_t)$, or $s_{t+1} \sim P(s_t, a_t)$. The interaction record

$$\tau = (s_1, a_1, r_1, s_2, \ldots, s_{H+1})$$

is called a *trajectory* of length H.

In some situations, it is necessary to specify how the initial state s_1 is generated. We consider s_1 sampled from an initial distribution $\mu \in \Delta(\mathcal{S})$. When μ is of importance to the discussion, we include it as part of the MDP definition, and write $M = (\mathcal{S}, \mathcal{A}, P, R, \gamma, \mu)$.

定义轨迹什么都很普通，需要注意的是，有的时候分析起来初始状态分布 μ 比较关键，这时候就把它显式写到MDP里面。

1.2. 策略和价值函数

A (deterministic and stationary) policy $\pi : \mathcal{S} \to \mathcal{A}$ specifies a decision-making strategy in which the agent chooses actions adaptively based on the current state, i.e., $a_t = \pi(s_t)$. More generally, the agent

may also choose actions according to a stochastic policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$, and with a slight abuse of notation we write $a_t \sim \pi(s_t)$. A deterministic policy is its special case when $\pi(s)$ is a point mass for all $s \in \mathcal{S}$.

策略主要分为确定性策略（deterministic policy）和随机策略（stochastic policy），注意到这里主要讲的是稳定（stationary）的策略，个人理解是每次遇到同样的状态都会按照同样的概率分布去选择行动，这样的策略叫做稳定的。而如果策略取决于现在是走的第几步（timestep）那么就是non-stationay的。本专栏讲的Distributional RL涉及到这个。

这里对于确定性策略和随机策略都用了同样的notation π 。

The goal of the agent is to choose a policy π to maximize the expected discounted sum of rewards, or *value*:

$$\mathbb{E}\Big[\sum_{t=1}^{\infty}\gamma^{t-1}r_t \mid \pi, s_1\Big]. \tag{1}$$

目标是最大化expected discounted sum of rewards。

The expectation is with respect to the randomness of the trajectory, that is, the randomness in state transitions and the stochasticity of π. Notice that, since r_t is nonnegative and upper bounded by $R_{\max}$, we have

$$0 \le \sum_{t=1}^{\infty}\gamma^{t-1}r_t \le \sum_{t=1}^{\infty}\gamma^{t-1}R_{\max} = \frac{R_{\max}}{1-\gamma}. \tag{2}$$

Hence, the discounted sum of rewards (or the discounted **return**) along any actual trajectory is always bounded in range $[0, \frac{R_{\max}}{1-\gamma}]$, and so is its expectation of any form. This fact will be important when we later analyze the error propagation of planning and learning algorithms.

刚刚对于reward的假设这里就能得到一个小结论了，即优化的目标的上界为 $\frac{R_{max}}{1-\gamma}$ 。

Note that for a fixed policy, its value may differ for different choice of s_1, and we define the value function $V_M^{\pi} : \mathcal{S} \to \mathbb{R}$ as

$$V_M^{\pi}(s) = \mathbb{E}\Big[\sum_{t=1}^{\infty}\gamma^{t-1}r_t \mid \pi, s_1 = s\Big],$$

which is the value obtained by following policy π starting at state s. Similarly we define the action-value (or Q-value) function $Q_M^{\pi} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ as

$$Q_M^{\pi}(s,a) = \mathbb{E}\Big[\sum_{t=1}^{\infty}\gamma^{t-1}r_t \mid \pi, s_1 = s, a_1 = a\Big].$$

V函数和Q函数的定义。

1.3. Policy evaluation 下的 Bellman 方程

Policy evaluation讲的是给定一个策略，然后估计其价值函数；与之相对的是control，即求一个最优策略。

Policy evaluation下V函数和Q函数的关系如下。

Based on the principles of dynamic programming, V^π and Q^π can be computed using the following *Bellman equations for policy evaluation*: $\forall s \in \mathcal{S}, a \in \mathcal{A}$,

$$V^\pi(s) = Q^\pi(s, \pi(s)).$$
$$Q^\pi(s,a) = R(s,a) + \gamma \mathbb{E}_{s' \sim P(s,a)}\left[V^\pi(s')\right]. \quad (3)$$

为了书写方便，这里就直接当策略是确定性的，如果是随机策略，就在外面套一个期望。

下面是对于理论推导比较关键的一个准备工作，即把相关的量都表示成矩阵-向量形式。

考虑到 V^π 是关于状态的函数，因此对于每个状态 $s \in \mathcal{S}$ 都有一个状态函数值，这样 V^π 可以被写成一个向量 $\mathbb{R}^{|\mathcal{S}|}$ ；同理， R 和 Q^π 可以被写成向量 $\mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$ 。

Define P^π as the transition matrix for policy π with dimension $|\mathcal{S}| \times |\mathcal{S}|$, whose (s, s')-th entry is

$$[P^\pi]_{s,s'} = \mathbb{E}_{a\sim\pi(s)}[P(s'|s,a)].$$

In fact, this matrix describes a Markov chain induced by MDP M and policy π. Its s-th row is the distribution over next-states upon taking actions according to π at state s, which we also write as $[P(s,\pi)]^\top$.

关于策略 π 的转移概率可以表示为矩阵，矩阵中的每个元素表示了从一个状态 s ，在策略 π 下一步转移到另外一个状态 s' 的概率。它同时包含了策略的随机性和环境的随机性。如果规定从状态 s 出发，那么 $P(s,\pi)$ 就是一个向量，表示从状态 s 出发转移到状态空间中其他状态的概率。

Similarly define R^π as the reward vector for policy π with dimension $|\mathcal{S}| \times 1$, whose s-th entry is

$$[R^\pi]_s = \mathbb{E}_{a\sim\pi(s)}[R(s,a)].$$

关于策略 π 的奖励函数可以表示为向量形式，它表示从每个状态出发按照策略 π 行动能够得到的期望单步奖励。

下面开始推导Bellman Equation

Then from Equation 3 we have

$$\begin{aligned}[V^\pi]_s = Q^\pi(s,\pi(s)) &= [R^\pi]_s + \gamma \mathbb{E}_{a\sim\pi(s)}\mathbb{E}_{s'\sim P(s,a)}\left[V^\pi(s')\right] \\ &= [R^\pi]_s + \gamma \mathbb{E}_{s'\sim P(s,\pi)}\left[V^\pi(s')\right] \\ &= [R^\pi]_s + \gamma \langle P(s,\pi), V^\pi \rangle,\end{aligned}$$

where $\langle \cdot, \cdot \rangle$ is dot product.

第一行就是利用的前面的关于V函数和Q函数之间关系的定义，第二行就是把策略的随机性和环境的随机性都吸收到单步转移概率矩阵里面，第三行就利用前面的定义做了改写。

由于上式对于每个状态 s 都成立，这样就能写成矩阵形式

$$V^\pi = R^\pi + \gamma P^\pi V^\pi \quad \Rightarrow \quad (\mathbf{I}_{|\mathcal{S}|} - \gamma P^\pi) V^\pi = R^\pi,$$

where $\mathbf{I}_{|\mathcal{S}|}$ is the identity matrix.

同时注意到以下矩阵是可逆的，证明的方法在下面写得很清楚了，即证明对于任意的向量，矩阵向量乘都不等于零，那么该矩阵就是非奇异矩阵，非奇异矩阵和可逆矩阵是等价的。

where $\mathbf{I}_{|\mathcal{S}|}$ is the identity matrix. Now we notice that matrix $(\mathbf{I}_{|\mathcal{S}|} - \gamma P^\pi)$ is always invertible. In fact, for any non-zero vector $x \in \mathbb{R}^{|\mathcal{S}|}$,

$$\begin{aligned}
&\left\|(\mathbf{I}_{|\mathcal{S}|} - \gamma P^\pi)x\right\|_\infty = \|x - \gamma P^\pi x\|_\infty \\
&\geq \|x\|_\infty - \gamma \|P^\pi x\|_\infty && \text{(triangular inequality for norms)} \\
&\geq \|x\|_\infty - \gamma \|x\|_\infty && \text{(each element of } P^\pi x \text{ is a convex average of } x\text{)} \\
&= (1-\gamma)\|x\|_\infty > 0 && (\gamma < 1, x \neq \mathbf{0})
\end{aligned}$$

So we can conclude that

$$V^\pi = (\mathbf{I}_{|\mathcal{S}|} - \gamma P^\pi)^{-1} R^\pi. \tag{4}$$

另外注意到如果奖励只依赖于当前的状态，那么 V^π 关于策略 π 的部分就只有前面一项，并且 V^π 可以看做是对于各个状态上奖励的线性组合，组合系数就是前面这一项对应的矩阵，这里叫做 discounted state occupancy，我看到有些论文里面叫state visitation frequency，是类似的东西。

State occupancy

When the reward function only depends on the current state, i.e., $R(s,a) = R(s)$, R^π is independent of π, and Equation 4 exhibits an interesting structure: implies that the value of a policy is *linear* in rewards, and the rows of the matrix $(\mathbf{I}_{|\mathcal{S}|} - \gamma P^\pi)^{-1}$ give the linear coefficients that depend on the initial state. Such coefficients, often represented as a vector, are called *discounted state occupancy* (or state occupancy for short). It can be interpreted as the expected number of times that each state is visited along a trajectory, where later visits are discounted more heavily.[1]

1.4. Optimal control 下的 Bellman 方程

There always exists a stationary and deterministic policy that simultaneously maximizes $V^{\pi}(s)$ for all $s \in \mathcal{S}$ and maximizes $Q^{\pi}(s,a)$ for all $s \in \mathcal{S}, a \in \mathcal{A}$ [1], and we denote this *optimal policy* as $\pi_M^\star$ (or $\pi^\star$). We use $V^\star$ as a shorthand for $V^{\pi^\star}$, and $Q^\star$ similarly.

$V^\star$ and $Q^\star$ satisfy the following set of *Bellman optimality equations* [2]: $\forall s \in \mathcal{S}, a \in \mathcal{A}$,

$$\begin{aligned} V^\star(s) &= \max_{a \in \mathcal{A}} Q^\star(s,a). \\ Q^\star(s,a) &= R(s,a) + \gamma \mathbb{E}_{s' \sim P(s,a)}\left[V^\star(s')\right]. \end{aligned} \tag{5}$$

Once we have $Q^\star$, we can obtain $\pi^\star$ by choosing actions greedily (with arbitrary tie-breaking mechanisms):

$$\pi^\star(s) = \arg\max_{a \in \mathcal{A}} Q^\star(s,a), \ \forall s \in \mathcal{S}.$$

We use shorthand π_Q to denote the procedure of turning a Q-value function into its greedy policy, and the above equation can be written as

$$\pi^\star = \pi_{Q^\star}.$$

都是比较平常的定义。值得一提的是，***总存在deterministic and stationary的策略能够同时对于所有状态最大化其V函数，对于所有的状态和行动最大化其Q函数***，该定理note里面没给证明。下面给出Bellman optimality operator的定义。对于policy evaluation情况其实也有相应的Bellman operator，区别在于optimal的情况下含有一个取max的操作，而policy evaluation的情况下是对于策略求期望。

To facilitate future discussions, define the *Bellman optimality operator* $\mathcal{T}_M : \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|} \to \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$ (or simply $\mathcal{T}$) as follows: when applied to some vector $f \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$,

$$(\mathcal{T}f)(s,a) := R(s,a) + \gamma \langle P(s,a), V_f \rangle, \tag{6}$$

where $V_f(\cdot) := \max_{a \in \mathcal{A}} f(\cdot, a)$. This allows us to rewrite Equation 5 in the following concise form, which implies that $Q^\star$ is the fixed point of the operator $\mathcal{T}$:

$$Q^\star = \mathcal{T}Q^\star.$$

1.5. MDP Setup

这里提出了其他一些可能的（前面没有涉及到的）MDP设定，并且给出了它们和本课程所讨论MDP设定直接的联系。

Finite horizon and episodic setting

我们这里讨论的是infinite horizon discounted setting。

一种比较常见的是infinite-horizon average reward setting，这种设定下由于是对于奖励取平均，累

积奖励是不会发散啦，但是所面临的问题是需要额外的条件才能有一个良好的价值函数定义（比如遍历性条件）。如何理解这一点呢？考虑一个MDP有两个状态 s_1 和 s_2 ，对于任意行动 $s_1 \to s_2$ 产生奖励 $r > 0$ ， $s_2 \to s_2$ 产生奖励 0 。在这样的定义下，无论 r 取值如何，每个状态上的价值函数都为 0 。这显然很不好。

Our definition of value (Equation 1) corresponds to the infinite-horizon discounted setting of MDPs. Popular alternative choices include the finite-horizon undiscounted setting (actual return of a trajectory is $\sum_{t=1}^{H} r_t$ with some finite horizon $H < \infty$) and the infinite-horizon average reward setting (return is $\lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} r_t$). The latter case often requires additional conditions on the transition dynamics (such as ergodicity) so that values can be well-defined [3], and will not be discussed in this course.

另一种比较常见的是finite-horizon undiscounted setting。

这种情况可以构造出了另外的一个MDP使得它和我们前面讨论infinite horizon discounted setting匹配。

The finite-horizon undiscounted (or simply finite-horizon) setting can be emulated using the discounted setting by augmenting the state space. Suppose we have an MDP M with finite horizon H. Define a new MDP $\tilde{M} = (\tilde{\mathcal{S}}, \mathcal{A}, \tilde{P}, \tilde{R}, \gamma)$ such that $\tilde{\mathcal{S}} = \mathcal{S} \times [H] \bigcup \{s_{\text{absorbing}}\}$ ($[H] = \{1, \ldots, H\}$). Essentially we make H copies of the state space and organize them in levels, with an additional absorbing state $s_{\text{absorbing}}$ where all actions transition to itself and yield 0 reward. There is only non-zero transition probability from states at level h to states at level $h+1$ with $\tilde{P}((s', h+1) \mid (s, h), a) = P(s'|s, a)$, and states at the last level (s, H) transition to $s_{\text{absorbing}}$ deterministically. Finally we let $\tilde{R}((s, h), a) = R(s, a)$ and $\gamma = 1$. (In general $\gamma = 1$ may lead to infinite value, but here the agent always loops in the absorbing state after H steps and gets finite total rewards.) The optimal policy for finite-horizon MDPs is generally non-stationary, that is, it depends on both s and the time step h.

The MDP described in the construction above can be viewed as an example of **episodic** tasks: the environment deterministically transitions into an absorbing state after a fixed number of time steps. The absorbing state often corresponds to the notion of termination, and many problems are naturally modeled using an episodic formulation, including board games (a game terminates once the winner is determined) and dialog systems (a session terminates when the conversation is concluded).

Stochastic or negative rewards

对于随机性的奖励或者奖励不仅仅依赖 s_t, a_t （比如可以依赖 s_{t+1} ），我们可以把相应的随机性取期望之后当它是确定性的（marginalize out），这样做不影响价值函数们，只会对于采样的效率有一定的影响。

Our setup assumes that reward r_t only depends on s_t and a_t deterministically. In general, r_t may also depend on s_{t+1} and contain additional noise that is independent from state transitions as well as reward noise in other time steps. As special cases, in inverse RL literature [4, 5], reward only depends on state, and in contextual bandit literature [6], reward depends on the state (or *context* in bandit terminologies) and action but has additional independent noise.

All these setups are equivalent to having a state-action reward with regard to the policy values: define $R(s,a) = \mathbb{E}[r_t | s_t = s, a_t = a]$ where s_{t+1} and the independent noise are marginalized out. The value functions V^π and Q^π for any π remains the same when we substitute in this equivalent reward function. That said, reward randomness may introduce additional noise in the sample trajectories and affect learning efficiency.

如果奖励可能是负数，我们只需要做一个常数的平移即可。

Our setup assumes that $r_t \in [0, R_{\max}]$. This is without loss of generality in the infinite-horizon discounted setting: for any constant $c > 0$, a reward function $R \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$ is equivalent to $R + c\mathbf{1}_{|\mathcal{S}\times\mathcal{A}|}$, as adding c units of reward to each state-action pair simply adds a constant "background" value of $c/(1-\gamma)$ to the value of all policies for all initial states. Therefore, when the rewards may be negative but still have bounded range, e.g., $R(s,a) \in [-a, b]$ with $a, b > 0$, we can add a constant offset $c = a$ to the reward function and define $R_{\max} = a + b$, so that after adding the offset the reward lies in $[0, R_{\max}]$.

编辑于 2019-05-14

强化学习 (Reinforcement Learning)