C1-16

Machine Learning
by Andrew Ng, Stanford Engineering

Xiaojie Zhou

szxjzhou@163.com

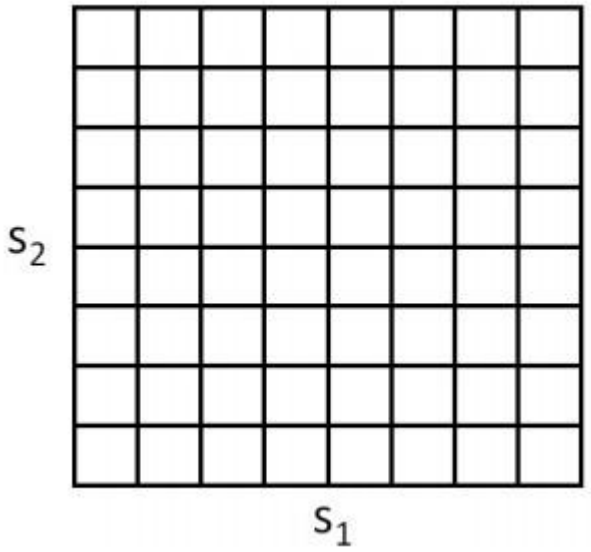
2016.10.1

第十七集：离散与维数灾难

- 离散化(Discretization)
- 模拟器(Simulators)
- 拟合值迭代与Q函数(Fitted Value Iteration and Q Function)
- 近似策略迭代(Approximate Policy Iteration)

第十七集：离散与维数灾难

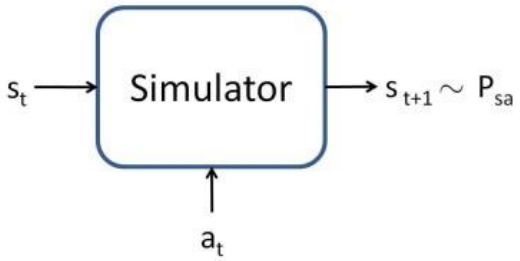
- 离散化
  - 在之前所述的马尔可夫决策模型中要求状态集S是离散有限的集合
    - 如果集合取值无限多，那么状态转移的情况也会有无限多，没办法进行分析
      - 但事实上这是存在的，比如说要给一辆车导航，那么它的所在位置就会在一个连续的区间中取值；比如说要控制一辆车的速度，那么它的速度也会在一个连续的区间内取值
    - 由于连续的值我们没法应对，因此我们很容易地会想到将其变成离散的值进行处理



左图显示了将一个二维的连续状态区间离散化的做法（高维空间的也可以采用类似做法）。这样做确实可以解决连续值无法直接处理的问题，但效果并不好。一方面用离散值去表达连续值肯定会丢失精度，另一方面离散的区间过多会导致状态的数目极大（而且数目会随着维数指数级上升），出现维数灾难(curse of dimension)的问题，非常不便于后续的处理

第十七集：离散与维数灾难

- 模拟器
  - 由于直接进行状态的离散化效果可能并不太好
    - 我们更加期待的效果是输入连续范围内取值的状态s，有限离散范围内取值的动作a，最后可以根据状态s、动作a以某一概率（状态转移概率p_{sa}）转移到下一状态s′
      - 在这里我们认为动作是离散的，原因在于一般情况下动作集的维数比较小，可以直接进行比较准确的离散化，但是状态集合的维数一般比动作集合高得多，离散化问题明显



这一过程的背后可以看成是经过了一个黑箱的模拟器，它以连续范围内取值的状态s，有限离散范围内取值的动作a为输入，根据状态转移概率p_{sa}转换到下一状态s'，并输出

    - 现在的问题变成了如何学习出这样的一个控制器，即在给定状态s和动作a的前提下计算出会以多大的状态转移概率p_{sa}转移到哪些状态s′

第十七集：离散与维数灾难

- 模拟器
  - 对于模拟器的学习来说无非需要知道输入为状态s，进行a操作下会转换到的状态s'及对应的状态转移概率p_{sa}
    - 一种方法是直接通过物理的计算获得

假设一个只能直走或者后退的小车，其质量为m，可以直接控制该车的加速度a，而我们关心这个车的速度状态v。

由此可知这个车在某一时刻t的状态$state_t = v_t$是在一维连续空间中取值的，车在某一时刻t的动作$action_t = a$是在一维空间中取值的，而我们要做的是在输入$state_t$和$action_t$的情况下将转换到的下一时刻$t+1$的状态$state_{t+1}$表示出来。由于：

$$state_{t+1} = state_t + (\Delta t)\frac{d}{dt}state_t$$

而在这个例子中$state_{t+1} = v_{t+1}, state_t = v_t$，代入有：

$$v_{t+1} = v_t + (\Delta t)\frac{d}{dt}v_t = v_t + (\Delta t)a$$

从而在给定某一时刻t的状态$state_t = v_t$和动作$action_t = a$的情况下即可得到下一时刻$t+1$的状态$state_{t+1}$.而这样得到的$state_{t+1}$是确定的，因此这模型也被称为确定性模型。在确定性模型中，只要给定了某一时刻t的状态$state_t$和动作$action_t$，最终一定会到达某个状态。而对于非确定性模型来说给定了某一时刻t的状态$state_t$和动作$action_t$不绝对到达某个状态，而是以某一概率到达某一状态

第十七集：离散与维数灾难

- 模拟器
  - 但是不是所有情况我们都能很好地对应到物理模型上，这使得输入为状态s，进行a操作下会转换到的状态s'不能直接通过公式计算出
    - 但如果我们给定了一系列测试到的马尔可夫决策过程，然后问s,a与s'的关系是什么，就可以等价转换为一个监督学习问题

如下所示，假设现在有m次持续时间为l的马尔可夫决策过程

$$s_0^{(1)} \xrightarrow{a_0^{(1)}} s_1^{(1)} \xrightarrow{a_1^{(1)}} \dots \xrightarrow{a_{l^{(1)}-1}^{(1)}} s_{l^{(1)}-1}^{(1)}$$
$$s_0^{(2)} \xrightarrow{a_0^{(2)}} s_1^{(2)} \xrightarrow{a_1^{(2)}} \dots \xrightarrow{a_{l^{(2)}-1}^{(2)}} s_{l^{(2)}-1}^{(2)}$$
$$\dots$$
$$s_0^{(m)} \xrightarrow{a_0^{(m)}} s_1^{(m)} \xrightarrow{a_1^{(m)}} \dots \xrightarrow{a_{l^{(m)}-1}^{(m)}} s_{l^{(m)}-1}^{(m)}$$

这时我们并不知道给定某一时刻t的状态$state_t$和动作$action_t$的情况下如何得到下一时刻t+1的状态$state_{t+1}$，那么我们就可以假设一个模型表示出他们之间的关系。比如我们可以假设它们之间服从线性关系，即$state_{t+1} = Astate_t + \mathrm{B}action_t$（其中$A,B$为系数），再通过已知的马尔可夫决策过程通过如下的目标函数将系数A,B学习出来即可

$$\min_{A,B} \sum_{i=1}^{m} \sum_{t=0}^{l^{(i)}-1} \left\| s_{t+1}^{(i)} - \left(As_t^{(i)} + Ba_t^{(i)} \right) \right\|^2$$

而这就相当于将模型学习问题转换为监督学习里面的求解线性回归参数的问题（实际上我们还可以换用其他模型进行训练来表达出$state_t, action_t$和$state_{t+1}$之间的关系）。与此同时不难发现这样训练出来的是个确定性模型，如果要表示出非确定性，一种方法是采用类似生成学习算法的模型，认为$state_{t+1}$服从一个与$state_t, action_t$有关的分布；还有一种方法是直接在训练的结果后面加上一个随机变量ε，变成$state_{t+1} = Astate_t + \mathrm{B}action_t + \varepsilon$

第十七集：离散与维数灾难

- 拟合值迭代与Q函数
  - 现在通过离散化或者使用模拟器的办法可以解决连续区间内取值的状态的表示和转换的问题
    - 也就是解决了状态转移概率p_{sa}的表示问题，给定上一时刻的状态和动作，我们能够知道以什么样的概率转到下一个状态
      - 如果是确定性模型，那么将100%转到某个状态，而转到其他状态均为0%
      - 如果是非确定性模型，那么可能转到多个状态（可转到的状态数可能无限多）
    - 接下来要考虑如何找出这样的决策过程的最优策略

还是和之前一样假设用$V^*(s)$表示在给定初始状态s下的最优值函数，其依然满足：

$$V^*(s)=\max_{\pi} V^{\pi}(s)$$

同样地由于 Bellman 方程$V^{\pi}(s)=R(s)+\gamma\sum_{s'\in S}p_{s,\pi(s)}\left(s'\right)V^{\pi}\left(s'\right)$有：

$$V^*(s)=R(s)+\max_{a\in A}\gamma\sum_{s'\in S}p_{s,a}\left(s'\right)V^*\left(s'\right)$$

由此在上一节中得到了值迭代的方法求解最优策略：

1、 初始化：

$$V(s)=0$$

2、 重复迭代直至收敛：

$$V(s)=R(s)+\max_{a\in A}\gamma\sum_{s'\in S}p_{s,a}\left(s'\right)\mathrm{V}\left(s'\right)$$

这里面由于现在状态s从离散值变成了连续值，因此重复迭代的部分变成了

$$V(s)=R(s)+\gamma\max_{a\in A}\int_{s'}p_{s,a}\left(s'\right)\mathrm{V}\left(s'\right)ds'$$

由于这里面$p_{s,a}\left(s'\right)$为给定初始状态s和动作a的情况下关于s'的概率，而$\mathrm{V}\left(s'\right)$为关于$s'$的函数，因此整体相当于在给定初始状态$s$和动作$a$的情况下对于$\mathrm{V}\left(s'\right)$求期望，由此可得：

$$V(s)=R(s)+\gamma\max_{a\in A}E_{s'\sim p_{s,a}}\left[\mathrm{V}\left(s'\right)\right]$$

因此对于状态在连续空间中取值的情况值迭代方法转换为：

1、 初始化：

$$V(s)=0$$

2、 重复迭代直至收敛：

$$V(s)=R(s)+\gamma\max_{a\in A}E_{s'\sim p_{s,a}}\left[\mathrm{V}\left(s'\right)\right]$$

第十七集：离散与维数灾难

- 拟合值迭代与Q函数

$$V(s)=R(s)+\gamma\max_{a\in A}E_{s'\sim p_{s,a}}\left[\mathrm{V}\left(s'\right)\right]$$

从状态集S中某个状态s来看，值函数$V(s)$的计算会很难求，主要原因在于$\mathrm{V}\left(s'\right)$（$p_{s,a}\left(s'\right)$通过模拟器方法在大部分情况下是可以近似求得的）。设想一下，在原本状态在离散空间取值的状态下这是通过枚举所有状态$s'\in S$得出的，但是在连续空间中因为状态数是无限的，我们不可能枚举与之相关的所有的$\mathrm{V}\left(s'\right)$.而且我们也不可能计算出全部状态$s$下的值函数$\mathrm{V}(s)$。

对此我们会想到一个解决方案是借用状态集S中抽样得到的有限个状态s去估计值函数$V(s)$的通用表达形式，一种经典的做法是：选取我们所需的特征（比如之前例子中小车的速度特征）组成向量$\phi(s)$，表示在s状态下这些特征的数值，然后认为值函数$V(s)$是$\phi(s)$的线性组合，即：

$$V(s)=\theta^{T}\phi(s)$$

并由此得到了对于状态集S中任意状态s的值函数$V(s)$的计算方法：

1、 在状态集S中任选m个状态，记为$s^{(1)},s^{(2)},\ldots,s^{(m)}$

2、 初始化参数θ的值为 0

3、 计算所选状态的最优值函数$V\left(s^{(1)}\right),V\left(s^{(2)}\right),\ldots,V\left(s^{(m)}\right)$

4、 如下式所示训练参数θ

$$\theta=arg\min_{\theta}\frac{1}{2}\sum_{i=1}^{m}\left(\theta^{T}\phi\left(s^{(i)}\right)-V\left(s^{(i)}\right)\right)^{2}$$

然而这里面还有一个问题，第 3 步中任意状态s下的最优值函数$\mathrm{V}(s)$怎么求？因为之前讲过我们不可能枚举与之相关的所有的$\mathrm{V}\left(s'\right)$.对此，我们也只能采取采样估计的办法，先按照之前在模拟器中计算出的状态转移情况抽取出可能跳转到的k个状态，记为$s'_1,s'_2,\ldots,s'_k$，然后用上一步迭代得到的$V(s)$表达式去求得在最优策略下这k个状态的值函数$\mathrm{V}\left(s'_1\right),\mathrm{V}\left(s'_2\right),\ldots,V\left(s'_k\right)$，再求和即可得到$\mathrm{V}(s)$.总体的算法流程如下所示（这一算法也被称为拟合值迭代法）：

1、 在状态集S中任选m个状态，记为$s^{(1)},s^{(2)},\ldots,s^{(m)}$

2、 初始化参数θ的值为 0

3、 重复迭代下面各步直至收敛

a) 计算所选状态的最优值函数$V\left(s^{(i)}\right),i=1,2,\ldots,m$

i. 对于动作集A中任意动作a

1. 按照之前在模拟器中计算出的状态转移情况抽取出可能跳转到的k个状态$s'_1,s'_2,\ldots,s'_k$

2. 用上一步迭代得到的$V(s)$表达式求关于动作a的函数$\mathrm{Q}(a)$

$$\mathrm{Q}(a)=\frac{1}{k}\sum_{j=1}^{k}R\left(s^{(i)}\right)+\gamma V\left(s'_j\right)$$

ii. 求取最优值函数$V\left(s^{(i)}\right)$

$$V\left(s^{(i)}\right)=\max_{a\in A}\mathrm{Q}(a)$$

b) 如下式所示训练参数θ

$$\theta=arg\min_{\theta}\frac{1}{2}\sum_{i=1}^{m}\left(\theta^{T}\phi\left(s^{(i)}\right)-V\left(s^{(i)}\right)\right)^{2}$$

第十七集：离散与维数灾难

- 近似策略迭代
  - 现在通过拟合值迭代可以近似得到最优值函数，但现在的问题是如何通过该最优值函数得到最优策略

$$\pi^*(s) = \arg\max_{a \in A} \sum_{s' \in S} P_{sa}(s') V^*(s')$$

    - 对于确定性的系统，能跳转到的目标状态s′只有一个，因此直接枚举所有动作a的取值，代入计算即可比较出哪个策略是最优策略
    - 而对于非确定性系统，能跳转到的状态无限多，不能列举所有的情况进行比较，只能采取下面这个近似的式子进行求解

$$\pi^*(s) = arg\max_{a \in A} \sum_{s' \in S} p_{s,a}\left(s'\right) V^*\left(s'\right) = arg\max_{a \in A} E_{s' \sim p_{s,a}}\left[V^*\left(s'\right)\right] \approx arg\max_{a \in A} V^*\left(E_{s' \sim p_{s,a}}\left[s'\right]\right)$$