C1-15

Machine Learning
by Andrew Ng, Stanford Engineering

Xiaojie Zhou

szxjzhou@163.com

2016.9.25

第十六集：马尔可夫决策过程

- 马尔可夫决策过程(Markov Decision Processes (MDP))
- 值函数(Value Function)
- 值迭代(Value Iteration)
- 策略迭代(Policy Iteration)
- 模型学习(Model Learning)

第十六集：马尔可夫决策过程

- 马尔可夫决策过程
  - 在讲解马尔可夫决策过程前，先来介绍马尔可夫模型
    - 马尔可夫模型解决的是对于数据序列的建模问题
      - 比如语音信号可以看成是由一个个的词组合而成的有序序列，一个月的天气也可看成是由一天天的天气组成的序列
    - 马尔可夫模型里面分为两种
      - 1、马尔可夫模型(Marcov Model)：得到的是监督的数据，对于已观测到的状态序列进行建模，目的是得到对应的状态跳转关系，从而能对未来的状态进行预测
      - 2、隐马尔可夫模型(Hidden Marcov Model)：得到的是非监督的数据，对于已观测到的特征进行建模，目的是从特征中得到对应的隐状态，并在得到隐状态的跳转关系后能对于未来的特征进行预测

第十六集：马尔可夫决策过程

- 马尔可夫决策过程
  - 马尔可夫模型
    - 得到的是监督的数据，对于已观测到的状态序列进行建模，目的是得到对应的状态跳转关系，从而能对未来的状态进行预测
    - 在马尔可夫模型中给定了状态集合$S=\{s_1,s_2,...,s_{|S|}\}$，通过持续时间t的观测即可得到某组对应的状态序列$Z=\{z_1,z_2,...,z_t\}$
      - 观测可以看成一个随机过程，而这些观测得到的状态是一个随机变量。对于时刻t的观测结果s_t来说，可能由前面的多个状态所决定，甚至还可能由后面未知的状态所决定
      - 而这样分析问题会过于复杂，因此在马尔可夫模型中提出了两点前提，这也被称为马尔可夫假设(Marcov assumptions)
        - 1、Limited Horizon Assumption：某一时刻的状态只与前一时刻的状态有关（在马尔可夫模型中认为前一时刻的状态已经囊括了足够多我们需要的信息）

$$P(z_t|z_{t-1},z_{t-2},...,z_1)=P(z_t|z_{t-1})$$

        - 2、Stationary Process Assumption：某一状态A到另一状态B的跳转不随时间的推移而改变（只要A、B是确定的）

第十六集：马尔可夫决策过程

- 马尔可夫决策过程
  - 马尔可夫模型
    - 基于前面两点假设我们可以通过构造一个状态转移矩阵来表示从某个状态到另一个状态的转移概率
      - 但这个时候有个很麻烦的问题：初始状态应该是什么？
        - 一种最直观的想法：从状态集合中随便指定一个作为初始状态
          - 但这样不准确，假如对于天气有晴天、多云、下雨三个状态，如果我们强令初始状态为晴天，那就暗示了第一天一定会是晴天，这显然不准确
        - 因此更科学的办法，在前面再新加一个状态s_0，表示初始状态
          - 初始状态是一个确定的状态（0时刻状态z_0一定是s_0），不属于状态集合中任意一个状态，只能从初始状态到达状态集合中的状态，但不能从状态集合中状态转换到初始状态
          - 由此可得最终的状态转换表（以天气有晴天、多云、下雨三个状态为例）：

$$A = \begin{array}{c|cccc} & s_0 & s_{sun} & s_{cloud} & s_{rain} \\ s_0 & 0 & .33 & .33 & .33 \\ s_{sun} & 0 & .8 & .1 & .1 \\ s_{cloud} & 0 & .2 & .6 & .2 \\ s_{rain} & 0 & .1 & .2 & .7 \end{array}$$

其中s_0为初始状态，s_{sun}, s_{cloud}, s_{rain}为状态集合中状态，矩阵中元素显示了状态跳转的概率。从中可以看出只能从初始状态到达状态集合中的状态，但不能从状态集合中状态转换到初始状态。而且不难看出，不可能跳转至非状态集合中的状态

第十六集：马尔可夫决策过程

- 马尔可夫决策过程
  - 马尔可夫模型
    - 对于马尔可夫模型来说我们关心两个问题
      - 1、在给定状态转换矩阵的前提下，怎样计算某个状态序列出现的概率

假设需要计算概率的序列为$\vec{z}=(z_1,z_2,\ldots,z_t)$，状态转移矩阵A，由此在给定状态转换矩阵的前提下，序列$\vec{z}$出现的概率为：

$$p\left(\vec{z}\right)=p(z_1,z_2,\ldots,z_t;A)$$

接下来，需要将初始状态$z_0=s_0$加进去，由于这一初始状态为确定性状态（即 0 时刻状态z_0一定是s_0），因此有：

$$p\left(\vec{z}\right)=p(z_0,z_1,z_2,\ldots,z_t;A)$$

由链式法则可知：

$$p\left(\vec{z}\right)=p(z_t|z_{t-1},z_{t-2},\ldots,z_0;A)p(z_{t-1}|z_{t-2},\ldots,z_0;A)\ldots p(z_1|z_0;A)$$

基于 Limited Horizon Assumption 假设，某一时刻的状态只与前一时刻的状态有关，有：

$$p\left(\vec{z}\right)=p(z_t|z_{t-1};A)p(z_{t-1}|z_{t-2};A)\ldots p(z_1|z_0;A)=\prod_{i=1}^{t}p(z_i|z_{i-1};A)=\prod_{i=1}^{t}A_{z_{i-1}z_i}$$

$$A=\begin{array}{c|cccc} & s_0 & s_{sun} & s_{cloud} & s_{rain} \\ \hline s_0 & 0 & .33 & .33 & .33 \\ s_{sun} & 0 & .8 & .1 & .1 \\ s_{cloud} & 0 & .2 & .6 & .2 \\ s_{rain} & 0 & .1 & .2 & .7 \end{array}$$

$P(z_1=s_{sun},z_2=s_{cloud},z_3=s_{rain},z_4=s_{rain},z_5=s_{cloud})$ which can be factored as $P(s_{sun}|s_0)P(s_{cloud}|s_{sun})P(s_{rain}|s_{cloud})P(s_{rain}|s_{rain})P(s_{cloud}|s_{rain})=.33\times.1\times.2\times.7\times.2$.

第十六集：马尔可夫决策过程

- 马尔可夫决策过程
  - 马尔可夫模型
    - 对于马尔可夫模型来说我们关心两个问题
      - 2、给定多组状态序列如何预测出对应的状态转移矩阵

由于给定了多组状态序列$\vec{z}^{(1)},\vec{z}^{(2)},\dots,\vec{z}^{(m)}$，对应时间长度为$t^{(1)},t^{(2)},\dots,t^{(m)}$，因此和之前一样，我们有理由相信这些序列是在给定状态转换矩阵A的情况下最有可能得到的。由此可用极大似然法解决问题，有：

$$\ell(\mathrm{A})=\log\prod_{l=1}^{m}p\left(\vec{z}^{(l)};A\right)=\log\prod_{l=1}^{m}\prod_{i=1}^{t^{(l)}}A_{z_{i-1}^{(l)}z_i^{(l)}}=\sum_{l=1}^{m}\sum_{i=1}^{t^{(l)}}\log A_{z_{i-1}^{(l)}z_i^{(l)}}$$

$$=\sum_{l=1}^{m}\sum_{i=0}^{|S|}\sum_{j=0}^{|S|}\sum_{k=1}^{t^{(l)}}\left(1\left\{z_{k-1}^{(l)}=s_i\wedge z_k^{(l)}=s_j\right\}\log A_{ij}\right)$$

下面要求对$\ell(\mathrm{A})$求极大值以得到状态转换矩阵A的最优值（因为$\ell(\mathrm{A})$是线性非负加权，因此一定是凸函数，极大值等于最大值），但由于状态转换矩阵A不能跳转到非状态集合中的状态，因此要满足任意一行的概率之和为 1，即$\sum_{j=0}^{|S|}A_{ij}=1,i=0,1,2,\dots,|S|$.由此可得广义 Lagrange 算子为：

$$\mathcal{L}(\mathrm{A},\alpha)=\sum_{l=1}^{m}\sum_{i=0}^{|S|}\sum_{j=0}^{|S|}\sum_{k=1}^{t^{(l)}}\left(1\left\{z_{k-1}^{(l)}=s_i\wedge z_k^{(l)}=s_j\right\}\log A_{ij}\right)+\sum_{i=0}^{|S|}\left(\alpha_i\left(1-\sum_{j=0}^{|S|}A_{ij}\right)\right)$$

由于目标函数$\ell(\mathrm{A})$为凸函数，而等式约束$\sum_{j=0}^{|S|}A_{ij}=1$为仿射函数，因此本问题为凸优化问题。由于存在状态转换矩阵A使得$\sum_{j=0}^{|S|}A_{ij}=1$，因此满足 Slater 条件，可直接通过 KKT 条件进行求解，因此有：

$$\frac{\partial\mathcal{L}(\mathrm{A},\alpha)}{\partial A_{ij}}=0\Leftrightarrow\frac{\partial}{\partial A_{ij}}\left[\sum_{l=1}^{m}\sum_{i=0}^{|S|}\sum_{j=0}^{|S|}\sum_{k=1}^{t^{(l)}}\left(1\left\{z_{k-1}^{(l)}=s_i\wedge z_k^{(l)}=s_j\right\}\log A_{ij}\right)+\sum_{i=0}^{|S|}\left(\alpha_i\left(1-\sum_{j=0}^{|S|}A_{ij}\right)\right)\right]=0$$

$$\Leftrightarrow\frac{\partial}{\partial A_{ij}}\sum_{l=1}^{m}\sum_{k=1}^{t^{(l)}}\left(1\left\{z_{k-1}^{(l)}=s_i\wedge z_k^{(l)}=s_j\right\}\log A_{ij}\right)-\frac{\partial}{\partial A_{ij}}\alpha_iA_{ij}=0$$

$$\Leftrightarrow\frac{1}{A_{ij}}\sum_{l=1}^{m}\sum_{k=1}^{t^{(l)}}1\left\{z_{k-1}^{(l)}=s_i\wedge z_k^{(l)}=s_j\right\}-\alpha_i=0\Leftrightarrow A_{ij}=\frac{1}{\alpha_i}\sum_{l=1}^{m}\sum_{k=1}^{t^{(l)}}1\left\{z_{k-1}^{(l)}=s_i\wedge z_k^{(l)}=s_j\right\}$$

接下来需要求α_i的值，这时候需要用上$\sum_{j=0}^{|S|}A_{ij}=1$的条件，将A_{ij}的结果代入有：

$$\sum_{j=0}^{|S|}\left(\frac{1}{\alpha_i}\sum_{l=1}^{m}\sum_{k=1}^{t^{(l)}}1\left\{z_{k-1}^{(l)}=s_i\wedge z_k^{(l)}=s_j\right\}\right)=1\Leftrightarrow\alpha_i=\sum_{j=0}^{|S|}\sum_{l=1}^{m}\sum_{k=1}^{t^{(l)}}1\left\{z_{k-1}^{(l)}=s_i\wedge z_k^{(l)}=s_j\right\}=\sum_{l=1}^{m}\sum_{k=1}^{t^{(l)}}1\left\{z_{k-1}^{(l)}=s_i\right\}$$

将α_i代回即可得到最终A_{ij}的表达式：

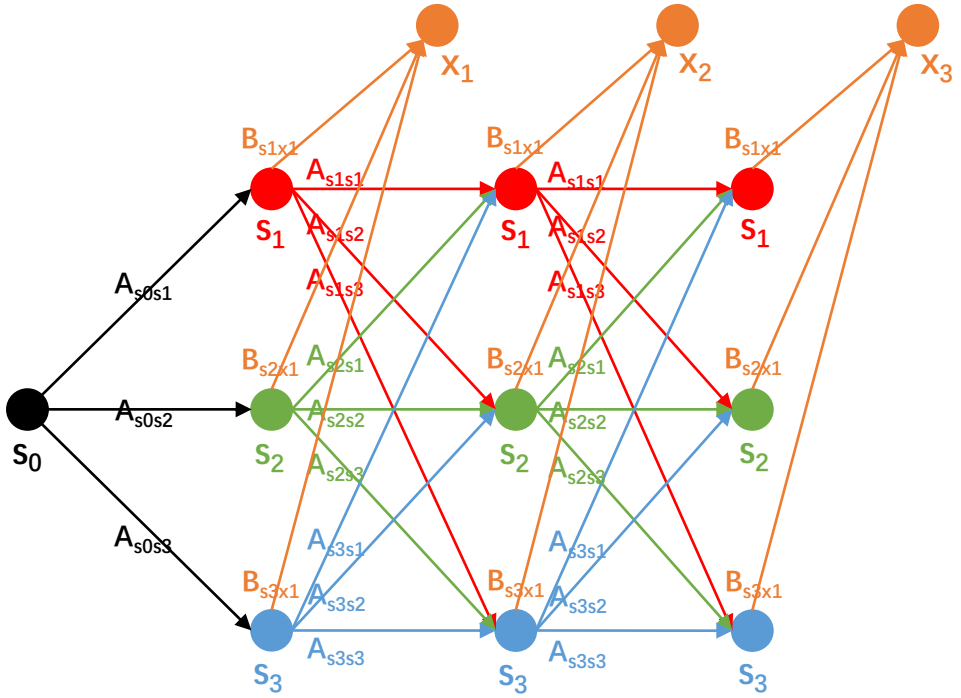
$$A_{ij}=\frac{\sum_{l=1}^{m}\sum_{k=1}^{t^{(l)}}1\left\{z_{k-1}^{(l)}=s_i\wedge z_k^{(l)}=s_j\right\}}{\sum_{l=1}^{m}\sum_{k=1}^{t^{(l)}}1\left\{z_{k-1}^{(l)}=s_i\right\}}$$

第十六集：马尔可夫决策过程

- 马尔可夫决策过程
  - 隐马尔可夫模型
    - 得到的是非监督的数据，对于已观测到的特征进行建模，目的是从特征中得到对应的隐状态，并在得到隐状态的跳转关系后能对于未来的特征进行预测
    - 因此在马尔可夫模型中给定了特征集合$V=\{v_1,v_2,...,v_{|V|}\}$，通过持续时间t的观测即可得到某组观测到的特征序列$x=\{x_1,x_2,...,x_T\}$，我们可以假设这些序列对应的状态集合$z=\{z_1,z_2,...,z_T\}$
      - 由于在隐马尔可夫模型中观测到的是特征，而不是状态，因此在这里假设的状态z是一个未确定的隐状态，而这个隐状态与状态集$S=\{s_1,s_2,...,s_{|S|}\}$中的某些确定的状态相关
      - 这里面蕴含了一个重要的OUTPUT INDEPENDENT ASSUMPTION：某一时刻的观测特征只与这一时刻的隐状态相关，与前后的观测特征和隐状态均无关
        - 举个例子：假设我们要用马尔可夫模型预测未来的天气状态，但我们并不知道过去的天气状态序列，只知道过去每一天小卖部的冰激凌销量序列。因此我们要用冰激凌销量序列来预测未来的天气
          - 在这个例子里面所有的天气状态组成一个状态集，但我们并不能直接观测到状态集中的状态，而只能观测到冰激凌销量这一特征序列。那么冰激凌销量和状态集中状态的关系用隐状态所表示，可以以一定概率属于状态集合中的某个状态
            - 比如说：状态集合里面有“晴天”、“雨天”，然后观测到冰激凌销量序列为{100个，50个}。对于这100个冰激凌销量来说，并不能直接下结论说这一天一定是晴天（虽然晴天确实卖的多），而只能说这一天80%是晴天，20%是雨天。这也就是为什么说隐状态可以以一定概率属于状态集合中的某个状态

第十六集：马尔可夫决策过程

- 马尔可夫决策过程
  - 隐马尔可夫模型
    - 基于前面三点假设（包括马尔可夫模型的两个假设和隐马尔可夫模型新加的一点假设）我们可以构造一个状态转移矩阵A来表示状态间的转移概率，一个观测矩阵B来表示状态到观测数据的转移概率。整个状态转移过程如下图所示



左图展示了一个典型的隐马尔可夫序列，其中观测特征序列为$\{x_1,x_2,x_3\}$，状态集合为$S=\{s_1,s_2,s_3\}$.和之前一样，这里也会遇到初始状态的表示问题。在隐马尔可夫模型的解决办法同样也是在前面再新加一个确定的状态s_0，表示初始状态。矩阵A表示状态间的转移概率，任何一个状态都有到状态集合中任意状态的转换概率；而矩阵B则表示状态到观测数据的转移概率，其含义是观测特征有多大可能由该状态生成

第十六集：马尔可夫决策过程

- 马尔可夫决策过程
  - 隐马尔可夫模型
    - 对于隐马尔可夫模型来说我们关心三个问题
      - 1、在给定状态转换矩阵A和观测矩阵B的前提下，怎样计算某个状态序列出现的概率

假设需要计算概率的特征序列$\vec{x} = (x_1, x_2, \dots, x_t)$，在给定状态转移矩阵 A 和观测矩阵 B 下，该序列出现的概率为：

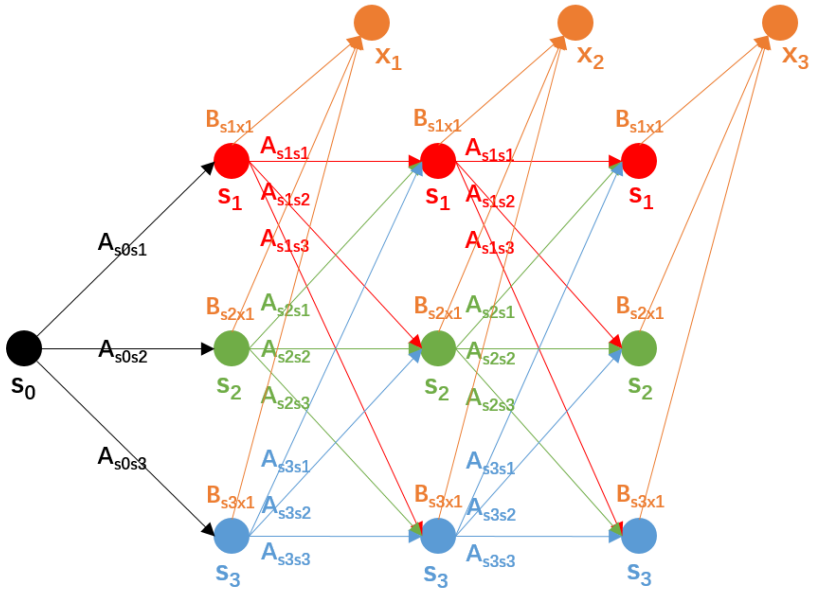
$$\mathrm{p}\left(\vec{x}; \mathrm{A}, \mathrm{B}\right) = \sum_{\vec{z}} p\left(\vec{x}, \vec{z}; A, B\right) = \sum_{\vec{z}} p\left(\vec{x} \middle| \vec{z}; A, B\right) p\left(\vec{z}; A, B\right)$$

基于隐马尔可夫模型的三点假设有：

$$\mathrm{p}\left(\vec{x}; \mathrm{A}, \mathrm{B}\right) = \sum_{\vec{z}} \left(\prod_{t=1}^{T} p\left(x_t | z_t; B\right) \right) \left(\prod_{t=1}^{T} p\left(z_t | z_{t-1}; A\right) \right) = \sum_{\vec{z}} \left(\prod_{t=1}^{T} B_{z_t x_t} \right) \left(\prod_{t=1}^{T} A_{z_{t-1} z_t} \right)$$

第十六集：马尔可夫决策过程

- 马尔可夫决策过程
  - 隐马尔可夫模型
    - 状态序列出现的概率计算



但与此同时我们发现这样做运算复杂度极高，因为$\vec{z}$需要取遍状态集合$S=\{s_1,s_2,\ldots,s_{|S|}\}$，算法复杂度高至$O\left(|S|^T\right)$，比如左图所示的情况来说$p(x_1;A,B)$的计算为：

$$p(x_1;A,B)=A_{s_0s_1}B_{s_1x_1}+A_{s_0s_2}B_{s_2x_1}+A_{s_0s_3}B_{s_3x_1}$$

这只有3项，看似还比较简单，但是等到计算$p(x_1x_2;A,B)$就很麻烦了：

$$p(x_1x_2;A,B)=p(x_1x_2,z_2=s_1;A,B)+p(x_1x_2,z_2=s_2;A,B)+p(x_1x_2,z_2=s_3;A,B)$$

其中：

$$\begin{aligned}p(x_1x_2,z_2=s_1;A,B)&=p(x_1x_2,z_1=s_1,z_2=s_1;A,B)+p(x_1x_2,z_1=s_2,z_2=s_1;A,B)+p(x_1x_2,z_1=s_3,z_2=s_1;A,B)\\&=A_{s_0s_1}B_{s_1x_1}A_{s_1s_1}B_{s_1x_2}+A_{s_0s_2}B_{s_2x_1}A_{s_2s_1}B_{s_1x_2}+A_{s_0s_3}B_{s_3x_1}A_{s_3s_1}B_{s_1x_2}\end{aligned}$$

$$\begin{aligned}p(x_1x_2,z_2=s_2;A,B)&=p(x_1x_2,z_1=s_1,z_2=s_2;A,B)+p(x_1x_2,z_1=s_2,z_2=s_2;A,B)+p(x_1x_2,z_1=s_3,z_2=s_2;A,B)\\&=A_{s_0s_1}B_{s_1x_1}A_{s_1s_2}B_{s_2x_2}+A_{s_0s_2}B_{s_2x_1}A_{s_2s_2}B_{s_2x_2}+A_{s_0s_3}B_{s_3x_1}A_{s_3s_2}B_{s_2x_2}\end{aligned}$$

$$\begin{aligned}p(x_1x_2,z_2=s_3;A,B)&=p(x_1x_2,z_1=s_1,z_2=s_3;A,B)+p(x_1x_2,z_1=s_2,z_2=s_3;A,B)+p(x_1x_2,z_1=s_3,z_2=s_3;A,B)\\&=A_{s_0s_1}B_{s_1x_1}A_{s_1s_3}B_{s_3x_2}+A_{s_0s_2}B_{s_2x_1}A_{s_2s_3}B_{s_3x_2}+A_{s_0s_3}B_{s_3x_1}A_{s_3s_3}B_{s_3x_2}\end{aligned}$$

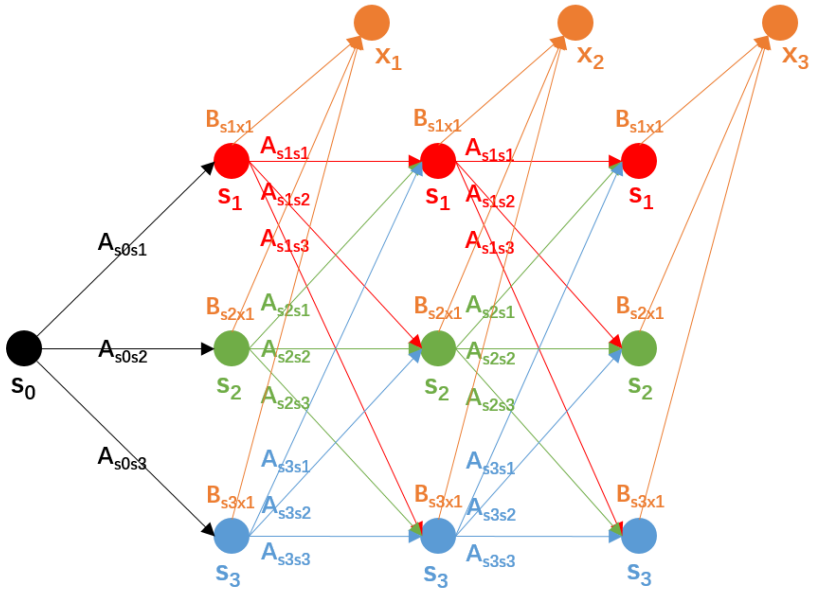
因此：

$$\begin{aligned}p(x_1x_2;A,B)&=A_{s_0s_1}B_{s_1x_1}A_{s_1s_1}B_{s_1x_2}+A_{s_0s_2}B_{s_2x_1}A_{s_2s_1}B_{s_1x_2}+A_{s_0s_3}B_{s_3x_1}A_{s_3s_1}B_{s_1x_2}+A_{s_0s_1}B_{s_1x_1}A_{s_1s_2}B_{s_2x_2}\\&+A_{s_0s_2}B_{s_2x_1}A_{s_2s_2}B_{s_2x_2}+A_{s_0s_3}B_{s_3x_1}A_{s_3s_2}B_{s_2x_2}+A_{s_0s_1}B_{s_1x_1}A_{s_1s_3}B_{s_3x_2}+A_{s_0s_2}B_{s_2x_1}A_{s_2s_3}B_{s_3x_2}\\&+A_{s_0s_3}B_{s_3x_1}A_{s_3s_3}B_{s_3x_2}\end{aligned}$$

变成了3^2项

第十六集：马尔可夫决策过程

- 马尔可夫决策过程
  - 隐马尔可夫模型
    - 状态序列出现的概率计算



之前使用定义进行计算的方法过于复杂，但通过对比计算的过程发现$\mathrm{p}(x_1;\mathrm{A},\mathrm{B})$的结果对于$\mathrm{p}(x_1x_2;\mathrm{A},\mathrm{B})$的计算有很大作用，这是因为在隐马尔可夫模型中下一状态与上一状态相关，因此

$$\mathrm{p}(z_1,z_2,\dots,z_T)=\prod_{t=1}^{T}p(z_t|z_{t-1})$$

也就是说一种加快运算的方法是将生成截止到上一时刻状态序列的概率$\prod_{t=1}^{T-1}p(z_t|z_{t-1})$，由此引出了一种前向算法(Forward Procedure)用动态规划的办法来解决这个问题，对此该算法的流程如下：

首先记$\alpha_i(t)=p(x_1,x_2,\dots,x_t,z_t=s_i;A,B)$为观测序列满足$x_1,x_2,\dots,x_t$的情况下，在$t$时刻状态为$s_i$的概率（其中$i=1,2,\dots,|S|$）

1、初始化

$$\alpha_i(1)=A_{s_0s_i}B_{s_ix_1}$$

2、归纳计算

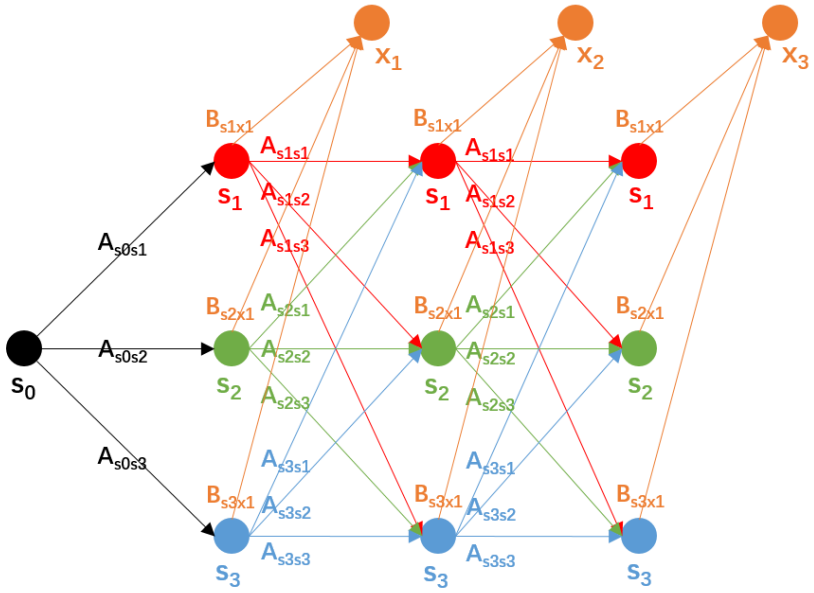
$$\alpha_i(t)=\left(\sum_{j=1}^{|S|}\alpha_j(t-1)A_{s_js_i}\right)B_{s_ix_t},2\le t\le T$$

3、终结

$$\mathrm{p}(\vec{x};\mathrm{A},\mathrm{B})=\sum_{i=1}^{|S|}\alpha_i(t)$$

第十六集：马尔可夫决策过程

- 马尔可夫决策过程
  - 隐马尔可夫模型
    - 状态序列出现的概率计算



实际上除了前向算法从前往后递推外，也可以反过来从后往前递推，这种方法被称为后向算法(Backward Procedure)。首先记$\beta_i(t)=p(x_T,x_{T-1},\dots,x_t,z_t=s_i;A,B)$为观测序列满足$x_t,x_{t+2},\dots,x_T$的情况下，在$t$时刻状态为$s_i$的概率（其中$i=1,2,\dots,|S|$）

1、 初始化：当$t=T$时，$\beta_i(T)=p(x_T,z_T=s_i;A,B)=B_{s_ix_t}$

2、 归纳计算

$$\beta_i(t)=p(x_T,\dots,x_t,z_t=s_i;A,B)=\sum_{j=1}^{|S|}p\left(x_T,\dots,x_t,z_t=s_i,z_{t+1}=s_j;A,B\right)$$

$$=\sum_{j=1}^{|S|}B_{s_ix_t}A_{s_is_j}\beta_j(t+1),1\le t\le T-1$$

3、 终结

$$p\left(\vec{x};A,B\right)=\sum_{i=1}^{|S|}A_{s_0s_i}\beta_i(1)$$

第十六集：马尔可夫决策过程

- 马尔可夫决策过程
  - 隐马尔可夫模型
    - 对于隐马尔可夫模型来说我们关心三个问题
      - 2、假如每个隐状态只能对应到某一个状态集中状态，那么观测到的特征序列x最可能对应到的状态序列s是什么？

对于这个问题来说，无非要求观测到的特征序列$\vec{\mathrm{x}}=(x_1,x_2,\ldots,x_T)$对应的隐状态序列$\vec{\mathrm{z}}=(z_1,z_2,\ldots,z_T)$，其中某一时刻的隐状态$z_t$只能确定对应到状态集合$\mathrm{S}=\{\mathrm{s}_1,\mathrm{s}_2,\ldots,\mathrm{s}_{|S|}\}$中某一个确定的状态。由此可知$\vec{\mathrm{z}}$一共有$\mathrm{S}^T$种可能，而目的是求出某一个$\vec{\mathrm{z}}$使：

$$arg\max_{\vec{\mathrm{z}}} p\left(\vec{\mathrm{z}}\middle|\vec{\mathrm{x}};A,B\right)$$

由贝叶斯公式有：

$$arg\max_{\vec{\mathrm{z}}}\frac{p\left(\vec{\mathrm{x}},\vec{\mathrm{z}};A,B\right)}{\sum_{\vec{\mathrm{z}}}p\left(\vec{\mathrm{x}},\vec{\mathrm{z}};A,B\right)}$$

由于$\sum_{\vec{\mathrm{z}}}p\left(\vec{\mathrm{x}},\vec{\mathrm{z}};A,B\right)$为常数（其中$\vec{\mathrm{x}}$是给定的），因此这相当于

$$arg\max_{\vec{\mathrm{z}}} p\left(\vec{\mathrm{x}},\vec{\mathrm{z}};A,B\right)$$

对此最简单的办法是将所有序列都枚举一遍，计算每个序列的概率

$$arg\max_{\vec{\mathrm{z}}}\prod_{t=1}^{T}A_{z_{t-1}z_t}B_{z_tx_t}$$

但要枚举S^T种可能的序列计算量过大，对此可以使用剪枝的办法，即对于每个t时刻，我们将一些概率较低的子序列直接去除或者只取t时刻概率最高的子序列（Viterbi 算法）

第十六集：马尔可夫决策过程

- 马尔可夫决策过程
  - 隐马尔可夫模型
    - 对于隐马尔可夫模型来说我们关心三个问题
      - 3、给定多组状态序列如何预测出对应的状态转移矩阵A和观测矩阵B？

在马尔可夫模型中我们采取极大似然的办法在给定多组状态序列预测对应的状态转移矩阵A，但是在隐马尔可夫模型中每个观测样本对应的状态不定，可能以一定概率属于状态集中的每个状态。因此又遇到了给定观测样本属于状态集中状态的概率$\mathrm{Q}\left(\vec{z}\right)=p\left(\vec{z}\middle|\vec{x};A,B\right)$，即可通过极大似然得到状态转移矩阵A和观测矩阵B；给定状态转移矩阵A和观测矩阵B即可计算每个观测样本属于状态集中状态的概率$\mathrm{Q}\left(\vec{z}\right)=p\left(\vec{z}\middle|\vec{x};A,B\right)$的问题。对此还是需要通过 EM 算法解决问题：

在这里不妨假设有m个样本特征序列，记为$\vec{x}^{(1)},\vec{x}^{(2)},\ldots,\vec{x}^{(m)}$，对应的隐状态序列为$\vec{z}^{(1)},\vec{z}^{(2)},\ldots,\vec{z}^{(m)}$，对应的序列长度为$T^{(1)},T^{(2)},\ldots,T^{(m)}$

在 E-Step 中计算每个观测样本属于状态集中状态的概率（其中$\vec{z}^{(l)}\in S^{T^{(l)}}$）：

$$\mathrm{Q}\left(\vec{z}^{(l)}\right)=p\left(\vec{z}^{(l)}\middle|\vec{x}^{(l)};A,B\right)$$

在 M-Step 中计算状态转移矩阵A和观测矩阵B的参数

$$A,B=arg\max_{A,B}\sum_{l=1}^{m}\sum_{\vec{z}^{(l)}}\mathrm{Q}\left(\vec{z}^{(l)}\right)\log\frac{p(\vec{x}^{(l)},\vec{z}^{(l)};A,B)}{\mathrm{Q}\left(\vec{z}^{(l)}\right)}$$

其中由于不允许状态集合中状态转移到非状态集合的状态，因此要求状态转移矩阵A每一行的概率值和为 1；与此同时，由于不允许观测样本属于非状态集合的状态，因此要求观测矩阵B每一行的概率值和为 1

$$\sum_{j=0}^{|S|}A_{ij}=1,i=0,1,2,\ldots,|S|$$

$$\sum_{k=1}^{|V|}B_{ik}=1,i=1,2,\ldots,|S|$$

第十六集：马尔可夫决策过程

- 马尔可夫决策过程
  - 隐马尔可夫模型
    - 状态转移矩阵A和观测矩阵B的求解

$$A,B=arg\max_{A,B}\sum_{i=1}^{m}\sum_{\vec{z}^{(i)}}\mathrm{Q}\left(\vec{z}^{(i)}\right)\log\frac{p(\vec{x}^{(i)},\vec{z}^{(i)};A,B)}{\mathrm{Q}\left(\vec{z}^{(i)}\right)}$$

首先对优化的目标函数进行化简，由于$\mathrm{Q}\left(\vec{z}^{(i)}\right)$是在 E-Step 中给定的常数，因此可化简为：

$$A,B=arg\max_{A,B}\sum_{i=1}^{m}\sum_{\vec{z}^{(i)}}\mathrm{Q}\left(\vec{z}^{(i)}\right)\log p(\vec{x}^{(i)},\vec{z}^{(i)};A,B)$$

代入$p(\vec{x}^{(i)},\vec{z}^{(i)};A,B)$有：

$$\begin{aligned}A,B&=arg\max_{A,B}\sum_{i=1}^{m}\sum_{\vec{z}^{(i)}}\mathrm{Q}\left(\vec{z}^{(i)}\right)\log\left(\prod_{t=1}^{T^{(i)}}p\left(x_t^{(i)}|z_t^{(i)};B\right)\right)\left(\prod_{t=1}^{T^{(i)}}p\left(z_t^{(i)}|z_{t-1}^{(i)};A\right)\right)\\&=arg\max_{A,B}\sum_{i=1}^{m}\sum_{\vec{z}^{(i)}}\mathrm{Q}\left(\vec{z}^{(i)}\right)\sum_{t=1}^{T^{(i)}}\left(\log B_{z_t^{(i)}x_t^{(i)}}+\log A_{z_{t-1}^{(i)}z_t^{(i)}}\right)\\&=arg\max_{A,B}\sum_{i=1}^{m}\sum_{\vec{z}^{(i)}}\mathrm{Q}\left(\vec{z}^{(i)}\right)\sum_{t=0}^{|S|}\sum_{j=1}^{|S|}\sum_{k=1}^{|V|}\sum_{t=1}^{T^{(i)}}\left(p\left(z_t^{(i)}=s_j,x_t^{(i)}=v_k\right)\log B_{jk}+p\left(z_{t-1}^{(i)}=s_i,z_t^{(i)}=s_j\right)\log A_{ij}\right)\end{aligned}$$

由于存在两个仿射的等式约束$\sum_{j=0}^{|S|}A_{ij}=1,\sum_{k=1}^{|V|}B_{ik}=1$，而且目标函数$\sum_{i=1}^{m}\sum_{\vec{z}^{(i)}}\mathrm{Q}\left(\vec{z}^{(i)}\right)\log\frac{p(\vec{x}^{(i)},\vec{z}^{(i)};A,B)}{\mathrm{Q}\left(\vec{z}^{(i)}\right)}$是关于矩阵$A,B$的非负线性加权，是凸函数。因此整个问题又是一个凸优化问题，而且不难发现这一问题也同样满足 Slater 条件。对此可以通过 KKT 条件来完成对问题的求解。在此之前需要写出广义 Lagrange 算子：

$$\begin{aligned}\mathcal{L}(\mathrm{A},\mathrm{B},\delta,\epsilon)=&\sum_{i=1}^{m}\sum_{\vec{z}^{(i)}}\mathrm{Q}\left(\vec{z}^{(i)}\right)\sum_{i=0}^{|S|}\sum_{j=1}^{|S|}\sum_{k=1}^{|V|}\sum_{t=1}^{T^{(i)}}\left(p\left(z_t^{(i)}=s_j,x_t^{(i)}=v_k\right)\log B_{jk}+p\left(z_{t-1}^{(i)}=s_i,z_t^{(i)}=s_j\right)\log A_{ij}\right)\\&+\sum_{j=1}^{|S|}\epsilon_j\left(1-\sum_{k=1}^{|V|}B_{jk}\right)+\sum_{i=0}^{|S|}\delta_i\left(1-\sum_{j=0}^{|S|}A_{ij}\right)\end{aligned}$$

由 KKT 条件有：

$$\begin{aligned}&\frac{\partial\mathcal{L}(\mathrm{A},\mathrm{B},\delta,\epsilon)}{\partial A_{ij}}=0\Leftrightarrow\sum_{i=1}^{m}\sum_{\vec{z}^{(i)}}\mathrm{Q}\left(\vec{z}^{(i)}\right)\frac{1}{A_{ij}}\sum_{t=1}^{T^{(i)}}p\left(z_{t-1}^{(i)}=s_i,z_t^{(i)}=s_j\right)-\delta_i=0\\&\Leftrightarrow A_{ij}=\frac{1}{\delta_i}\sum_{i=1}^{m}\sum_{\vec{z}^{(i)}}\mathrm{Q}\left(\vec{z}^{(i)}\right)\sum_{t=1}^{T^{(i)}}p\left(z_{t-1}^{(i)}=s_i,z_t^{(i)}=s_j\right)\\&\frac{\partial\mathcal{L}(\mathrm{A},\mathrm{B},\delta,\epsilon)}{\partial B_{jk}}=0\Leftrightarrow\sum_{i=1}^{m}\sum_{\vec{z}^{(i)}}\mathrm{Q}\left(\vec{z}^{(i)}\right)\frac{1}{B_{jk}}\sum_{t=1}^{T^{(i)}}p\left(z_t^{(i)}=s_j,x_t^{(i)}=v_k\right)-\epsilon_j=0\\&\Leftrightarrow B_{jk}=\frac{1}{\epsilon_j}\sum_{i=1}^{m}\sum_{\vec{z}^{(i)}}\mathrm{Q}\left(\vec{z}^{(i)}\right)\sum_{t=1}^{T^{(i)}}p\left(z_t^{(i)}=s_j,x_t^{(i)}=v_k\right)\end{aligned}$$

接下来需要解出这里面的参数δ_i和ϵ_j，方法很简单，直接将A_{ij}和B_{jk}代回原等式约束即可：

$$\begin{aligned}&\sum_{j=0}^{|S|}A_{ij}=1\Leftrightarrow\sum_{j=0}^{|S|}\frac{1}{\delta_i}\sum_{i=1}^{m}\sum_{\vec{z}^{(i)}}\mathrm{Q}\left(\vec{z}^{(i)}\right)\sum_{t=1}^{T^{(i)}}p\left(z_{t-1}^{(i)}=s_i,z_t^{(i)}=s_j\right)-1=0\Leftrightarrow\delta_i\\&=\sum_{j=0}^{|S|}\sum_{i=1}^{m}\sum_{\vec{z}^{(i)}}\mathrm{Q}\left(\vec{z}^{(i)}\right)\sum_{t=1}^{T^{(i)}}p\left(z_{t-1}^{(i)}=s_i,z_t^{(i)}=s_j\right)=\sum_{i=1}^{m}\sum_{\vec{z}^{(i)}}\mathrm{Q}\left(\vec{z}^{(i)}\right)\sum_{t=1}^{T^{(i)}}p\left(z_{t-1}^{(i)}=s_i\right)\\&\sum_{k=1}^{|V|}B_{jk}=1\Leftrightarrow\sum_{k=1}^{|V|}\frac{1}{\epsilon_j}\sum_{i=1}^{m}\sum_{\vec{z}^{(i)}}\mathrm{Q}\left(\vec{z}^{(i)}\right)\sum_{t=1}^{T^{(i)}}p\left(z_t^{(i)}=s_j,x_t^{(i)}=v_k\right)-1=0\Leftrightarrow\epsilon_j\\&=\sum_{k=1}^{|V|}\sum_{i=1}^{m}\sum_{\vec{z}^{(i)}}\mathrm{Q}\left(\vec{z}^{(i)}\right)\sum_{t=1}^{T^{(i)}}p\left(z_t^{(i)}=s_j,x_t^{(i)}=v_k\right)=\sum_{i=1}^{m}\sum_{\vec{z}^{(i)}}\mathrm{Q}\left(\vec{z}^{(i)}\right)\sum_{t=1}^{T^{(i)}}p\left(z_t^{(i)}=s_j\right)\end{aligned}$$

将δ_i,ϵ_j代回即可得A_{ij}和B_{jk}的表达式：

$$\begin{aligned}A_{ij}&=\frac{\sum_{i=1}^{m}\sum_{\vec{z}^{(i)}}\mathrm{Q}\left(\vec{z}^{(i)}\right)\sum_{t=1}^{T^{(i)}}p\left(z_{t-1}^{(i)}=s_i,z_t^{(i)}=s_j\right)}{\sum_{i=1}^{m}\sum_{\vec{z}^{(i)}}\mathrm{Q}\left(\vec{z}^{(i)}\right)\sum_{t=1}^{T^{(i)}}p\left(z_{t-1}^{(i)}=s_i\right)}\\B_{jk}&=\frac{\sum_{i=1}^{m}\sum_{\vec{z}^{(i)}}\mathrm{Q}\left(\vec{z}^{(i)}\right)\sum_{t=1}^{T^{(i)}}p\left(z_t^{(i)}=s_j,x_t^{(i)}=v_k\right)}{\sum_{i=1}^{m}\sum_{\vec{z}^{(i)}}\mathrm{Q}\left(\vec{z}^{(i)}\right)\sum_{t=1}^{T^{(i)}}p\left(z_t^{(i)}=s_j\right)}\end{aligned}$$

第十六集：马尔可夫决策过程

- 马尔可夫决策过程
  - 隐马尔可夫模型
    - 状态转移矩阵A和观测矩阵B的求解

进一步观察 EM 算法求解状态转移矩阵A和观测矩阵B的过程发现，过程看似简单，但实际上非常不好算。这是因为每个观测样本不一定完全属于某个状态，而是以一定概率属于状态集中的状态。这导致了对于一个长度为T的观测序列来说，每个t时刻可属于的状态数为S，整个序列可属于的状态数为S^T，情况太多使得$\mathrm{Q}\left(\vec{z}^{(l)}\right)=p\left(\vec{z}^{(l)}\middle|\vec{x}^{(l)};A,B\right)$在 E-step 中很难得到。由此我们将$A_{ij}$和$B_{jk}$继续推导，看看会出现什么情况：

首先看看A_{ij}的分子部分：

$$\begin{aligned}\sum_{l=1}^{m}\sum_{\vec{z}^{(l)}}\mathrm{Q}\left(\vec{z}^{(l)}\right)\sum_{t=1}^{T^{(l)}}p\left(z_{t-1}^{(l)}=s_i,z_t^{(l)}=s_j\right)&=\sum_{l=1}^{m}\sum_{t=1}^{T^{(l)}}\sum_{\vec{z}^{(l)}}p\left(z_{t-1}^{(l)}=s_i,z_t^{(l)}=s_j\right)\mathrm{Q}\left(\vec{z}^{(l)}\right)\\&=\sum_{l=1}^{m}\sum_{t=1}^{T^{(l)}}\sum_{\vec{z}^{(l)}}p\left(z_{t-1}^{(l)}=s_i,z_t^{(l)}=s_j\right)p\left(\vec{z}^{(l)}\middle|\vec{x}^{(l)};A,B\right)\\&=\sum_{l=1}^{m}\sum_{t=1}^{T^{(l)}}\sum_{\vec{z}^{(l)}}p\left(z_{t-1}^{(l)}=s_i,z_t^{(l)}=s_j\right)\frac{p\left(\vec{z}^{(l)},\vec{x}^{(l)};A,B\right)}{p\left(\vec{x}^{(l)};A,B\right)}\\&=\sum_{l=1}^{m}\frac{1}{p\left(\vec{x}^{(l)};A,B\right)}\sum_{t=1}^{T^{(l)}}\sum_{\vec{z}^{(l)}}p\left(z_{t-1}^{(l)}=s_i,z_t^{(l)}=s_j\right)p\left(\vec{z}^{(l)},\vec{x}^{(l)};A,B\right)\end{aligned}$$

由于$\sum_{\vec{z}^{(l)}}p\left(\vec{z}^{(l)},\vec{x}^{(l)};A,B\right)$的意义是在给定状态转移矩阵A和观测矩阵B的情况下，观测特征序列$\vec{x}^{(l)}$和对应隐状态序列$\vec{z}^{(l)}$同时发生的概率，$p\left(z_{t-1}^{(l)}=s_i,z_t^{(l)}=s_j\right)$的意义是第$l$个序列在$t-1$时刻的状态为$s_i$而$t$时刻的状态为$s_j$的概率。因此$\sum_{\vec{z}^{(l)}}p\left(z_{t-1}^{(l)}=s_i,z_t^{(l)}=s_j\right)p\left(\vec{z}^{(l)},\vec{x}^{(l)};A,B\right)$等价于求在给定状态转移矩阵A和观测矩阵B的情况下，观测特征序列$\vec{x}^{(l)}$的$t-1$时刻的状态为s_i而t时刻的状态为s_j的概率。而这个概率是有快速算法进行计算的，因为之前已经提过$\alpha_i(t)=p(x_1,\dots,x_t,z_t=s_i)$，表示的是序列满足$x_1,\dots,x_t$的情况下$t$时刻的状态为$s_i$的概率；$\beta_i(t)=p(x_T,\dots,x_t,z_t=s_i)$，表示的是序列满足$x_t,\dots,x_T$的情况下$t+1$时刻的状态为$s_i$的概率，因此有：

$$\begin{aligned}&\sum_{\vec{z}^{(l)}}p\left(z_{t-1}^{(l)}=s_i,z_t^{(l)}=s_j\right)p\left(\vec{z}^{(l)},\vec{x}^{(l)};A,B\right)\\&=p\left(x_1^{(l)},\dots,x_{t-1}^{(l)},z_{t-1}^{(l)}=s_i\right)p\left(z_{t-1}^{(l)}=s_i,z_t^{(l)}=s_j\right)p\left(x_T^{(l)},\dots,x_t^{(l)},z_t^{(l)}=s_j\right)\\&=\alpha_i^{(l)}(t-1)A_{ij}\beta_j^{(l)}(t)\end{aligned}$$

在这个方法综合了前向算法和后向算法，因此这一也被称为前向后向算法(Forward-Backward Algorithm)，因此有：

$$\begin{aligned}\sum_{l=1}^{m}\sum_{\vec{z}^{(l)}}\mathrm{Q}\left(\vec{z}^{(l)}\right)\sum_{t=1}^{T^{(l)}}p\left(z_{t-1}^{(l)}=s_i,z_t^{(l)}=s_j\right)&=\sum_{l=1}^{m}\frac{1}{p\left(\vec{x}^{(l)};A,B\right)}\sum_{t=1}^{T^{(l)}}\sum_{\vec{z}^{(l)}}p\left(z_{t-1}^{(l)}=s_i,z_t^{(l)}=s_j\right)p\left(\vec{z}^{(l)},\vec{x}^{(l)};A,B\right)\\&=\sum_{l=1}^{m}\frac{1}{p\left(\vec{x}^{(l)};A,B\right)}\sum_{t=1}^{T^{(l)}}\alpha_i^{(l)}(t-1)A_{ij}\beta_j^{(l)}(t)\end{aligned}$$

同理可得A_{ij}的分母部分：

$$\begin{aligned}\sum_{l=1}^{m}\sum_{\vec{z}^{(l)}}\mathrm{Q}\left(\vec{z}^{(l)}\right)\sum_{t=1}^{T^{(l)}}p\left(z_{t-1}^{(l)}=s_i\right)&=\sum_{l=1}^{m}\sum_{t=1}^{T^{(l)}}\sum_{\vec{z}^{(l)}}p\left(z_{t-1}^{(l)}=s_i\right)\mathrm{Q}\left(\vec{z}^{(l)}\right)\\&=\sum_{l=1}^{m}\sum_{j=1}^{|S|}\sum_{t=1}^{T^{(l)}}\sum_{\vec{z}^{(l)}}p\left(z_{t-1}^{(l)}=s_i,z_t^{(l)}=s_j\right)p\left(\vec{z}^{(l)}\middle|\vec{x}^{(l)};A,B\right)\\&=\sum_{l=1}^{m}\frac{1}{p\left(\vec{x}^{(l)};A,B\right)}\sum_{j=1}^{|S|}\sum_{t=1}^{T^{(l)}}\sum_{\vec{z}^{(l)}}p\left(z_{t-1}^{(l)}=s_i,z_t^{(l)}=s_j\right)p\left(\vec{z}^{(l)},\vec{x}^{(l)};A,B\right)\\&=\sum_{l=1}^{m}\frac{1}{p\left(\vec{x}^{(l)};A,B\right)}\sum_{j=1}^{|S|}\sum_{t=1}^{T^{(l)}}\alpha_i^{(l)}(t-1)A_{ij}\beta_j^{(l)}(t)\end{aligned}$$

第十六集：马尔可夫决策过程

- 马尔可夫决策过程
  - 隐马尔可夫模型
    - 状态转移矩阵A和观测矩阵B的求解

同时对于观测矩阵B也可采取类似的办法，对于B_{jk}的分子部分有：

$$\sum_{l=1}^{m}\sum_{\vec{z}^{(l)}}\mathrm{Q}\left(\vec{z}^{(l)}\right)\sum_{t=1}^{T^{(l)}}p\left(z_t^{(l)}=s_j,x_t^{(l)}=v_k\right)=\sum_{l=1}^{m}\sum_{t=1}^{T^{(l)}}\sum_{\vec{z}^{(l)}}p\left(z_t^{(l)}=s_j,x_t^{(l)}=v_k\right)\mathrm{Q}\left(\vec{z}^{(l)}\right)$$

$$=\sum_{l=1}^{m}\sum_{t=1}^{T^{(l)}}\sum_{\vec{z}^{(l)}}p\left(z_t^{(l)}=s_j,x_t^{(l)}=v_k\right)p\left(\vec{z}^{(l)}\middle|\vec{x}^{(l)};A,B\right)$$

$$=\sum_{l=1}^{m}\frac{1}{p\left(\vec{x}^{(l)};A,B\right)}\sum_{i=0}^{|S|}\sum_{t=1}^{T^{(l)}}\sum_{\vec{z}^{(l)}}p\left(z_{t-1}^{(l)}=s_i,z_t^{(l)}=s_j,x_t^{(l)}=v_k\right)p\left(\vec{z}^{(l)},\vec{x}^{(l)};A,B\right)$$

$$=\sum_{l=1}^{m}\frac{1}{p\left(\vec{x}^{(l)};A,B\right)}\sum_{i=0}^{|S|}\sum_{t=1}^{T^{(l)}}\sum_{\vec{z}^{(l)}}p\left(z_{t-1}^{(l)}=s_i,z_t^{(l)}=s_j\right)p\left(x_t^{(l)}=v_k\right)p\left(\vec{z}^{(l)},\vec{x}^{(l)};A,B\right)$$

$$=\sum_{l=1}^{m}\frac{1}{p\left(\vec{x}^{(l)};A,B\right)}\sum_{i=0}^{|S|}\sum_{t=1}^{T^{(l)}}1\left\{x_t^{(l)}=v_k\right\}\alpha_i^{(l)}(t-1)A_{ij}\beta_j^{(l)}(t)$$

对于B_{jk}的分母部分有：

$$\sum_{l=1}^{m}\sum_{\vec{z}^{(l)}}\mathrm{Q}\left(\vec{z}^{(l)}\right)\sum_{t=1}^{T^{(l)}}p\left(z_t^{(l)}=s_j\right)=\sum_{l=1}^{m}\sum_{t=1}^{T^{(l)}}\sum_{\vec{z}^{(l)}}p\left(z_t^{(l)}=s_j\right)\mathrm{Q}\left(\vec{z}^{(l)}\right)=\sum_{l=1}^{m}\sum_{t=1}^{T^{(l)}}\sum_{\vec{z}^{(l)}}p\left(z_t^{(l)}=s_j\right)p\left(\vec{z}^{(l)}\middle|\vec{x}^{(l)};A,B\right)$$

$$=\sum_{l=1}^{m}\frac{1}{p\left(\vec{x}^{(l)};A,B\right)}\sum_{i=0}^{|S|}\sum_{t=1}^{T^{(l)}}\sum_{\vec{z}^{(l)}}p\left(z_{t-1}^{(l)}=s_i,z_t^{(l)}=s_j\right)p\left(\vec{z}^{(l)},\vec{x}^{(l)};A,B\right)$$

$$=\sum_{l=1}^{m}\frac{1}{p\left(\vec{x}^{(l)};A,B\right)}\sum_{i=0}^{|S|}\sum_{t=1}^{T^{(l)}}\alpha_i^{(l)}(t-1)A_{ij}\beta_j^{(l)}(t)$$

由此可得A_{ij}和B_{jk}的表达式：

$$A_{ij}=\frac{\sum_{l=1}^{m}\frac{1}{p\left(\vec{x}^{(l)};A,B\right)}\sum_{t=1}^{T^{(l)}}\alpha_i^{(l)}(t-1)A_{ij}\beta_j^{(l)}(t)}{\sum_{l=1}^{m}\frac{1}{p\left(\vec{x}^{(l)};A,B\right)}\sum_{j=1}^{|S|}\sum_{t=1}^{T^{(l)}}\alpha_i^{(l)}(t-1)A_{ij}\beta_j^{(l)}(t)}=\frac{\sum_{l=1}^{m}\sum_{t=1}^{T^{(l)}}\alpha_i^{(l)}(t-1)A_{ij}\beta_j^{(l)}(t)}{\sum_{l=1}^{m}\sum_{j=1}^{|S|}\sum_{t=1}^{T^{(l)}}\alpha_i^{(l)}(t-1)A_{ij}\beta_j^{(l)}(t)}$$

$$B_{jk}=\frac{\sum_{l=1}^{m}\frac{1}{p\left(\vec{x}^{(l)};A,B\right)}\sum_{i=0}^{|S|}\sum_{t=1}^{T^{(l)}}1\left\{x_t^{(l)}=v_k\right\}\alpha_i^{(l)}(t-1)A_{ij}\beta_j^{(l)}(t)}{\sum_{l=1}^{m}\frac{1}{p\left(\vec{x}^{(l)};A,B\right)}\sum_{i=0}^{|S|}\sum_{t=1}^{T^{(l)}}\alpha_i^{(l)}(t-1)A_{ij}\beta_j^{(l)}(t)}$$

$$=\frac{\sum_{l=1}^{m}\sum_{i=0}^{|S|}\sum_{t=1}^{T^{(l)}}1\left\{x_t^{(l)}=v_k\right\}\alpha_i^{(l)}(t-1)A_{ij}\beta_j^{(l)}(t)}{\sum_{l=1}^{m}\sum_{i=0}^{|S|}\sum_{t=1}^{T^{(l)}}\alpha_i^{(l)}(t-1)A_{ij}\beta_j^{(l)}(t)}$$

即：

$$A_{ij}=\frac{\sum_{l=1}^{m}\sum_{t=1}^{T^{(l)}}\alpha_i^{(l)}(t-1)A_{ij}\beta_j^{(l)}(t)}{\sum_{l=1}^{m}\sum_{j=1}^{|S|}\sum_{t=1}^{T^{(l)}}\alpha_i^{(l)}(t-1)A_{ij}\beta_j^{(l)}(t)}$$

$$B_{jk}=\frac{\sum_{l=1}^{m}\sum_{i=0}^{|S|}\sum_{t=1}^{T^{(l)}}1\left\{x_t^{(l)}=v_k\right\}\alpha_i^{(l)}(t-1)A_{ij}\beta_j^{(l)}(t)}{\sum_{l=1}^{m}\sum_{i=0}^{|S|}\sum_{t=1}^{T^{(l)}}\alpha_i^{(l)}(t-1)A_{ij}\beta_j^{(l)}(t)}$$

第十六集：马尔可夫决策过程

- 马尔可夫决策过程
  - 隐马尔可夫模型
    - 状态转移矩阵A和观测矩阵B的求解

由此可得A_{ij}和B_{jk}的表达式：

$$A_{ij}=\frac{\sum_{l=1}^{m}\frac{1}{p\left(\vec{x}^{(l)};A,B\right)}\sum_{t=1}^{T^{(l)}}\alpha_i^{(l)}(t-1)A_{ij}\beta_j^{(l)}(t)}{\sum_{l=1}^{m}\frac{1}{p\left(\vec{x}^{(l)};A,B\right)}\sum_{j=1}^{|S|}\sum_{t=1}^{T^{(l)}}\alpha_i^{(l)}(t-1)A_{ij}\beta_j^{(l)}(t)}=\frac{\sum_{l=1}^{m}\sum_{t=1}^{T^{(l)}}\alpha_i^{(l)}(t-1)A_{ij}\beta_j^{(l)}(t)}{\sum_{l=1}^{m}\sum_{j=1}^{|S|}\sum_{t=1}^{T^{(l)}}\alpha_i^{(l)}(t-1)A_{ij}\beta_j^{(l)}(t)}$$

$$B_{jk}=\frac{\sum_{l=1}^{m}\frac{1}{p\left(\vec{x}^{(l)};A,B\right)}\sum_{i=0}^{|S|}\sum_{t=1}^{T^{(l)}}1\left\{x_t^{(l)}=v_k\right\}\alpha_i^{(l)}(t-1)A_{ij}\beta_j^{(l)}(t)}{\sum_{l=1}^{m}\frac{1}{p\left(\vec{x}^{(l)};A,B\right)}\sum_{i=0}^{|S|}\sum_{t=1}^{T^{(l)}}\alpha_i^{(l)}(t-1)A_{ij}\beta_j^{(l)}(t)}$$

$$=\frac{\sum_{l=1}^{m}\sum_{i=0}^{|S|}\sum_{t=1}^{T^{(l)}}1\left\{x_t^{(l)}=v_k\right\}\alpha_i^{(l)}(t-1)A_{ij}\beta_j^{(l)}(t)}{\sum_{l=1}^{m}\sum_{i=0}^{|S|}\sum_{t=1}^{T^{(l)}}\alpha_i^{(l)}(t-1)A_{ij}\beta_j^{(l)}(t)}$$

即：

$$A_{ij}=\frac{\sum_{l=1}^{m}\sum_{t=1}^{T^{(l)}}\alpha_i^{(l)}(t-1)A_{ij}\beta_j^{(l)}(t)}{\sum_{l=1}^{m}\sum_{j=1}^{|S|}\sum_{t=1}^{T^{(l)}}\alpha_i^{(l)}(t-1)A_{ij}\beta_j^{(l)}(t)}$$

$$B_{jk}=\frac{\sum_{l=1}^{m}\sum_{i=0}^{|S|}\sum_{t=1}^{T^{(l)}}1\left\{x_t^{(l)}=v_k\right\}\alpha_i^{(l)}(t-1)A_{ij}\beta_j^{(l)}(t)}{\sum_{l=1}^{m}\sum_{i=0}^{|S|}\sum_{t=1}^{T^{(l)}}\alpha_i^{(l)}(t-1)A_{ij}\beta_j^{(l)}(t)}$$

由此可得最终用 EM 算法求解状态转移矩阵A和观测矩阵B的过程

初始化

随机初始化状态转移矩阵A和观测矩阵B（但要满足对于矩阵A,B的基本约束）

E-Step

使用前向后向算法计算每个序列的$\alpha_i^{(l)},\beta_j^{(l)}$和$\gamma_t^{(l)}(i,j)$（其中$l=1\dots m,i=0\dots|S|,j=1\dots|S|$）

$$\gamma_t^{(l)}(i,j)=\alpha_i^{(l)}(t-1)A_{ij}\beta_j^{(l)}(t)$$

M-Step

重新计算状态转移矩阵A和观测矩阵B

$$A_{ij}=\frac{\sum_{l=1}^{m}\sum_{t=1}^{T^{(l)}}\gamma_t^{(l)}(i,j)}{\sum_{l=1}^{m}\sum_{j=1}^{|S|}\sum_{t=1}^{T^{(l)}}\gamma_t^{(l)}(i,j)}$$

$$B_{jk}=\frac{\sum_{l=1}^{m}\sum_{i=0}^{|S|}\sum_{t=1}^{T^{(l)}}1\left\{x_t^{(l)}=v_k\right\}\gamma_t^{(l)}(i,j)}{\sum_{l=1}^{m}\sum_{i=0}^{|S|}\sum_{t=1}^{T^{(l)}}\gamma_t^{(l)}(i,j)}$$

第十六集：马尔可夫决策过程

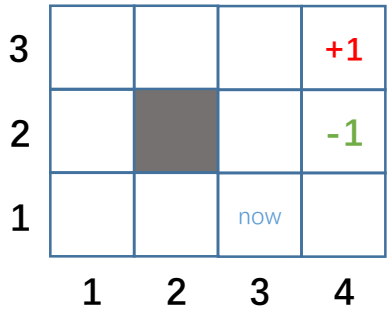
- 马尔可夫决策过程
  - 马尔可夫决策过程是强化学习中的一个重要概念
    - 强化学习的核心在于回报函数，在回报函数中衡量了学习器学习效果，当学习器的学习效果优秀时给予回报，否则给予惩罚
      - 就好比训练小狗，小狗做得好给点奖励，不好给点惩罚
      - 这种机器学习方法被广泛用于自动控制、市场动态分析等数据瞬息万变的领域中
        - 比如说飞机的自动驾驶，每隔一段时间飞机上的传感器就会传回一系列的数据，这数据的变化是不定的；金融市场每隔一段时间都会更新交易价格，这个也是瞬息万变的；在动态路由选择中每个路由器的拥塞情况都是实时更新的...
          - 而我们要做的就是根据现实动态调整自己的策略
        - 从这里我们可以看出这个过程和监督学习的一个很大不同：在强化学习中对于正确根本没有定义，没有一个静态的策略是绝对正确的
          - 监督学习中分类结果对不对用训练数据一训练就知道，然而在强化学习中我们只能根据现实情况进行策略的调整（因此需要一个渐进的决策过程），而这个现实情况不只是当前时刻的情况。如果我们要进行更为准确的趋势分析就必须考虑更前的情况

第十六集：马尔可夫决策过程

- 马尔可夫决策过程
  - 在强化学习中这种渐进的决策过程就被称为马尔可夫决策过程，马尔可夫决策过程由下列五部分组成
    - 状态(state)集合S：当前所处的状态
      - 比如自动驾驶中有汽车所处的位置、速度等状态信息
    - 动作(action)集合A：可能的动作
      - 比如自动驾驶中有转弯、加速、刹车等动作
    - 状态转换概率分布P_{sa}：说明在s状态下有多大可能执行a动作
    - 折现系数(discount factor)γ：$0 \leq \gamma < 1$，对于状态的权重
    - 回报函数(reward function)：对于当前所处状态s的回报

第十六集：马尔可夫决策过程

- 马尔可夫决策过程
  - 下面用一个例子说明马尔可夫决策过程的五元组



现在借机器人导航的例子来解释马尔可夫决策过程的五个部分，假设图中灰色区块为不可通行的障碍物。我们想让机器人到达右上方的区块，因此给那个区块赋予回报+1；而我们不想到达下面的那个区块，因此给那个区块赋予回报-1，而在其他区域由于有油耗因此赋予回报-0.03，这就是在那这些状态下对应的回报函数

R((4,3))=+1
R((4,2))=-1
R(other state)=-0.03

从图中可以看出机器人有11个位置可走，因此一共有11个状态，而这11个状态造成了状态集。同时假定机器人只能从上、下、左、右四个方向中选一个方向移动，因此一共有4个动作，这些动作共同构成了动作集

由于控制的准确度不高，我们在命令机器人向上移动时，仅有80%可能会向上，而另有10%的可能向左，10%的可能向右。然后机器人在撞到障碍物时会停下。假设当前所处位置为(3,1)，那么其对应状态转换概率分布为：

$P_{s=(3,1)\ a=North}((4,1))=0.1$
$P_{s=(3,1)\ a=North}((2,1))=0.1$
$P_{s=(3,1)\ a=North}((3,2))=0.8$
……

第十六集：马尔可夫决策过程

- 马尔可夫决策过程
  - 接下来正式来看马尔可夫决策过程是如何进行的
    - 首先先看一下和之前提到的马尔可夫模型、隐马尔可夫模型之间的关系

	不考虑动作	考虑动作
已知状态	马尔可夫模型	马尔可夫决策模型
未知状态	隐马尔可夫模型	部分可观测的马尔可夫决策模型

- 从中可以看出马尔可夫决策模型中最重要的区别在于其带有动作，而且观测到的结果是直接的状态序列，其过程如下所示

$$s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} s_2 \xrightarrow{a_2} s_3 \xrightarrow{a_3} \cdots$$

初始状态为s_0，然后在执行了动作集中某一动作a_0后会以$P_{s=s0,a=a0}$的概率跳转到状态集中的某一状态s_1，之后继续执行下一动作a_1，执行完毕后又会以$P_{s=s1,a=a1}$的概率跳转到状态集中的某一状态s_2.

第十六集：马尔可夫决策过程

- 值函数
  - 对马尔可夫决策的评价
    - 在给定一个马尔可夫决策过程的情况下，我们可以通过回报函数来评判这个过程的好坏

$$R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \cdots$$

左边的式子给出了马尔可夫决策过程的一种经典的评判形式，这里的R(s)表示在该状态下的回报。由于每件事情都有一个时效性，因此越往后这件事情将显得越不重要（或者说这件事情所带来的回报越低），因此在这里我们引入了折现函数γ，使得越靠前的状态重要性越大

    - 而马尔可夫决策过程的目标就是选取动作集中的动作使得对应过程的总回报的期望值是最大的

$$\mathrm{E}\left[R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \cdots\right]$$

$$V^{\pi}(s) = \mathrm{E}\left[R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \cdots \middle| s_0 = s, \pi\right]$$

左上的式子即为马尔可夫决策过程的目标函数，要选取动作集中的动作使得该总回报的期望值最大。而我们所做(executing)的策略(policy)π可以看成是从状态集S到动作集A之间的映射，表明某一状态下要对应采取的动作。因此我们可定义如左下所示的值函数(value function)$V^{\pi}(s)$，为初始状态为s，给定策略π的情况下总回报关于策略π的期望值

第十六集：马尔可夫决策过程

- 值函数
  - 接下来对于值函数做更进一步的推导

我们将马尔可夫决策过程的值函数$V^{\pi}(s)$定义为初始状态为s，给定策略π的情况下总回报关于策略π的期望值，即

$$V^{\pi}(s) = E\left[R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \cdots | s_0 = s, \pi\right]$$

其中由于$s_0 = s$，是一个给定的值与所选策略π没有关系，因此可以在$V^{\pi}(s)$表达式中可以直接将$R(s_0)$提出来，令$R(s_0) = R(s)$，从而得到：

$$V^{\pi}(s) = R(s) + E\left[\gamma R(s_1) + \gamma^2 R(s_2) + \cdots | \pi\right]$$

在这里面$R(s)$是一开始就带有的回报，因此我们称其为立即回报(immediate reward)，后面的部分称为后续回报。由于γ是折现系数，是一个常数，同样与所选策略π无关，有：

$$V^{\pi}(s) = R(s) + \gamma E\left[R(s_1) + \gamma R(s_2) + \cdots | \pi\right]$$

在这里并没有明确给出下一阶段确定到达的状态s_1，我们只知道上一状态$s_0 = s$和所用策略π，并在策略π中指定了在某一状态下所对应执行的操作。因此，我们令$\pi(s)$表示在状态为s的情况下所采取的动作。根据之前所述，执行某一动作不能保证 100%能到达我们预先所想的状态（就跟之前的例子一样，指挥机器人向上走，机器人可能会 10%向左走），而是服从状态转换概率跳转到某一状态，因此我们需要考虑状态集合S中所有可能的状态，有：

$$V^{\pi}(s) = R(s) + \gamma \sum_{s' \in S} p_{s=s, a=\pi(s)}(s_1 = s') E\left[R(s_1 = s') + \gamma R(s_2) + \cdots | \pi\right]$$

在这里又发现$E\left[R(s_1 = s') + \gamma R(s_2) + \cdots | \pi\right]$中初始状态$s_1$确定为$s'$，因此这变成了给定初始状态为$s'$给定策略$\pi$的情况下总回报关于策略$\pi$的期望值，这与$V^{\pi}(s')$是等价的，因此：

$$V^{\pi}(s) = R(s) + \gamma \sum_{s' \in S} p_{s=s, a=\pi(s)}(s') V^{\pi}(s')$$

而这也被称为 Bellman 方程(Bellman equations)，类似于隐马尔可夫模型的前向算法，这一等式给出了马尔可夫决策过程中在给定策略π的情况下，值函数$V^{\pi}(s)$的高效算法：对于每一个状态集合S中的状态s，写出其 Bellman 方程$V^{\pi}(s)$，从而最终得到了总共有s个参数的s个 Bellman 方程，通过求解线性方程组的方法即可解出状态集合S中的任一状态s为初始状态，在给定策略π的情况下的值函数$V^{\pi}(s)$

第十六集：马尔可夫决策过程

- 值函数
  - 在之前的内容中讲述了马尔可夫决策过程中，在给定策略π的情况下，值函数$V^{\pi}(s)$的高效算法
    - 那么不仅会想，反过来如果给定了初始状态s，使得值函数取最大值（称其为"最优值函数"(optimal value function)）的最优策略π是什么？

在此用$V^{*}(s)$表示在给定初始状态s下的最优值函数，其满足：

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

由 Bellman 方程$V^{\pi}(s)=R(s)+\gamma\sum_{s'\in S}p_{s=s,a=\pi(s)}(s')V^{\pi}(s')$有：

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

其中不难发现$R(s)$是立即回报，γ是折现系数均与所选策略π无关，因此我们要找的能使给定初始状态s下的最优值函数$V^{*}(s)$的策略$\pi^{*}(s)$只需满足：

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

从中不难发现对于状态集S中任意的初始状态s，策略$\pi^{*}(s)$均满足不等式：

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

由此可知该表达式下的策略$\pi^{*}(s)$对于状态集S中任意的初始状态s都能得到最优值函数$V^{*}(s)$，因此在实际应用中我们无需对于不同的初始状态s考虑不同的最优策略$\pi^{*}(s)$表达形式，而只需通过$arg\max_{a\in A}\sum_{s'\in S}p_{s=s,a=a}(s')V^{*}(s')$去求$\pi^{*}(s)$

第十六集：马尔可夫决策过程

- 值迭代
  - 下面说明如何求解最优策略π

在给定初始状态s下的最优值函数$V^*(s)$满足下面形式：

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

从中可知，要找的最优策略$\pi^*(s)$能使得值函数$V(s)$取得最大值，因此可以通过求最优值函数$V^*(s)$间接求最优策略$\pi^*(s)$.而对于最优值函数可以通过迭代的方法进行求解，此方法也被称为值迭代(value iteration)方法：

1、 对于状态集S中任意的初始状态s初始化：

$$V(s) = 0$$

2、 对于状态集S中任意的初始状态s重复迭代直至收敛：

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

由于状态集S中任意的初始状态s的值函数会受到其他状态下值函数的影响，因此特别强调值迭代方法中需要对于状态集S中任意的初始状态s更新计算值函数$V(s)$，而不能只算某一初始状态下的值函数。而这种影响也使得算法的第 2 步有两种计算策略：

1、 同步更新法(synchronous update) :在计算每一个状态下的值函数，得到新的$V(s)$值后，先存下来，不立即更新。待状态集S中任意的初始状态s的新的$V(s)$值都计算完毕后，再统一更新所有状态的$V(s)$值，用于下一轮的计算

2、 异步更新法(asynchronous update) :在计算某一个状态下的值函数，得到新的$V(s)$值后，立即更新该状态的$V(s)$值，用于其他状态的更新计算

然而不论是哪种方法都能保证对于状态集S中任意的初始状态s的$V(s)$值都是只升不降的，而$V(s)$是有上界的，因此其结果一定会收敛，最终一定能求出最优值函数$V^*(s)$，并可得到对应的最优策略$\pi^*(s)$

第十六集：马尔可夫决策过程

- 策略迭代
  - 另一种求解最优策略π的方法

通过值迭代求最优值函数$V^*(s)$间接求最优策略$\pi^*(s)$的方法外，还可以直接迭代求最优策略$\pi^*(s)$，此方法也被称为策略迭代(policy iteration)方法：

1、 随机初始化策略π

2、 重复迭代直至收敛

2.1、通过 Bellman 方程求解该策略下的值函数$V^{\pi}(s), s \in S$

2.2、用 2.1 求出的值函数更新集合S中的任一状态s下的策略π

$$\pi(s) = arg\max_{a \in A} \sum_{s' \in S} p_{s=s,a=a}(s') V^{\pi}(s)$$

整个重复迭代的过程类似于 EM 算法，最终算法收敛时会直接得到对应的最优策略$\pi^*(s)$.其中 2.1 的具体算法为：对于每一个状态集合S中的状态s，写出其 Bellman 方程$V^{\pi}(s)$，从而最终得到了总共有s个参数的s个 Bellman 方程，通过求解线性方程组的方法即可解出状态集合S中的任一状态s为初始状态，在给定策略π的情况下的值函数$V^{\pi}(s)$；而 2.2 中采用的贪心的思想(greedy with expect to V)得到新的策略π

值迭代相对于策略迭代来说比较容易算，但是收敛也比较慢。对于规模比较小的马尔可夫决策过程来说，策略迭代一般能够更快地收敛。但是对于规模很大（状态很多）的马尔可夫决策过程来说，值迭代好算的优势就发挥出来了（不需要求解线性方程组）

第十六集：马尔可夫决策过程

- 模型学习
  - 通过值迭代和策略迭代，我们可以得到在给定了初始状态s下的最优策略
    - 但与此同时发现，不论是值迭代还是策略迭代都需要状态转换概率分布P_{sa}的信息，但在实际情况下这个信息不那么容易能直接得到
    - 而在马尔可夫决策过程的其它四个要素的信息（状态集合S、动作集合A、折现函数γ、回报函数R）都是预定义好的
      - 当回报函数不好具体定义时常认为R(s)是在多次决策中的状态s下所得回报的均值
      - 对此采取的办法是尝试通过数据来估计状态转换概率分布（极大似然），而这需要借助于如下所示的多次决策序列

$$
\begin{aligned}
&s_0^{(1)} \xrightarrow{a_0^{(1)}} s_1^{(1)} \xrightarrow{a_1^{(1)}} s_2^{(1)} \xrightarrow{a_2^{(1)}} s_3^{(1)} \xrightarrow{a_3^{(1)}} \dots \\
&s_0^{(2)} \xrightarrow{a_0^{(2)}} s_1^{(2)} \xrightarrow{a_1^{(2)}} s_2^{(2)} \xrightarrow{a_2^{(2)}} s_3^{(2)} \xrightarrow{a_3^{(2)}} \dots \\
&\dots
\end{aligned}
$$

第十六集：马尔可夫决策过程

- 模型学习

假设我们已知如下所示的m条马尔可夫决策过程的状态转移链$\vec{z}^{(1)},\vec{z}^{(2)},\dots,\vec{z}^{(m)}$，每条链的长度分别为$\mathrm{t}^{(1)},\mathrm{t}^{(2)},\dots,\mathrm{t}^{(m)}$

$$s_0^{(1)} \xrightarrow{a_0^{(1)}} s_1^{(1)} \xrightarrow{a_1^{(1)}} s_2^{(1)} \xrightarrow{a_2^{(1)}} s_3^{(1)} \xrightarrow{a_3^{(1)}} \dots$$
$$s_0^{(2)} \xrightarrow{a_0^{(2)}} s_1^{(2)} \xrightarrow{a_1^{(2)}} s_2^{(2)} \xrightarrow{a_2^{(2)}} s_3^{(2)} \xrightarrow{a_3^{(2)}} \dots$$
$$\dots$$

不失一般性的，可假设状态集合$(s_1,s_2,\dots,s_{|S|})$，动作集合$(a_1,a_2,\dots,a_{|A|})$，第i条马尔可夫决策过程的状态序列$\vec{s}^{(i)}=\left(s_0^{(i)},s_1^{(i)},\dots,s_{\mathrm{t}^{(i)}-1}^{(i)}\right)$，动作序列$\vec{a}^{(i)}=\left(a_0^{(i)},a_1^{(i)},\dots,a_{\mathrm{t}^{(i)}-2}^{(i)}\right)$，那么可知$\vec{z}^{(i)}=\left(\vec{s}^{(i)},\vec{a}^{(i)}\right)$.和之前一样我们认为这些序列是在给定每个序列的初始状态$s_0^{(i)}$，策略为π的情况下最有可能得到的。由此可用极大似然法解决问题，有：

$$\ell(p_{sa})=\log\prod_{l=1}^{m}p\left(\vec{z}^{(l)};p_{sa}\right)=\log\prod_{l=1}^{m}p\left(\vec{s}^{(l)},\vec{a}^{(l)};p_{sa}\right)=\log\prod_{l=1}^{m}\prod_{k=1}^{t^{(l)}-1}p_{s_{k-1}^{(l)},a_{k-1}^{(l)}}\left(s_k^{(l)}\right)=\sum_{l=1}^{m}\sum_{k=1}^{t^{(l)}-1}\log p_{s_{k-1}^{(l)},a_{k-1}^{(l)}}\left(s_k^{(l)}\right)$$
$$=\sum_{l=1}^{m}\sum_{s=1}^{|S|}\sum_{a=1}^{|A|}\sum_{e=1}^{|S|}\sum_{k=1}^{t^{(l)}-1}\left(1\left\{s_{k-1}^{(l)}=s_s\wedge a_{k-1}^{(l)}=a_a\wedge s_k^{(l)}=s_e\right\}\log p_{s_s,a_a}(s_e)\right)$$

下面要求对$\ell(p_{sa})$求极大似然以得到状态转换概率分布p_{sa}的最优值（因为$\ell(p_{sa})$是非负线性加权，因此一定是凸函数，极大值等于最大值），但由于转换到的下一阶段的结果一定在状态集中，因此要求$\sum_{e=1}^{|S|}p_{s,a}(s_e)=1$，由此可得广义 Lagrange 算子为：

$$\ell(p_{sa},\alpha)=\sum_{l=1}^{m}\sum_{s=1}^{|S|}\sum_{a=1}^{|A|}\sum_{e=1}^{|S|}\sum_{k=1}^{t^{(l)}-1}\left(1\left\{s_{k-1}^{(l)}=s_s\wedge a_{k-1}^{(l)}=a_a\wedge s_k^{(l)}=s_e\right\}\log p_{s_s,a_a}(s_e)\right)$$
$$+\sum_{s=1}^{|S|}\sum_{a=1}^{|A|}\left(\alpha_{sa}\left(1-\sum_{e=1}^{|S|}p_{s_s,a_a}(s_e)\right)\right)$$

由于目标函数$\ell(p_{sa})$为凸函数，而等式约束$\sum_{e=1}^{|S|}p_{s,a}(s_e)=1$为仿射函数，因此本问题为凸优化问题，并且易知其满足 Slater 条件，可直接通过 KKT 方法进行求解，因此有：

$$\frac{\partial\ell(p_{sa},\alpha)}{\partial p_{s_s,a_a}(s_e)}=0$$
$$\Leftrightarrow\frac{\partial}{\partial p_{s_s,a_a}(s_e)}\left[\sum_{l=1}^{m}\sum_{k=1}^{t^{(l)}-1}\left(1\left\{s_{k-1}^{(l)}=s_s\wedge a_{k-1}^{(l)}=a_a\wedge s_k^{(l)}=s_e\right\}\log p_{s_s,a_a}(s_e)\right)-\alpha_{s_s,a_a}p_{s_s,a_a}(s_e)\right]=0$$
$$\Leftrightarrow\frac{1}{p_{s_s,a_a}(s_e)}\sum_{l=1}^{m}\sum_{k=1}^{t^{(l)}-1}1\left\{s_{k-1}^{(l)}=s_s\wedge a_{k-1}^{(l)}=a_a\wedge s_k^{(l)}=s_e\right\}-\alpha_{s_s,a_a}=0$$
$$\Leftrightarrow p_{s_s,a_a}(s_e)=\frac{1}{\alpha_{s_s,a_a}}\sum_{l=1}^{m}\sum_{k=1}^{t^{(l)}-1}1\left\{s_{k-1}^{(l)}=s_s\wedge a_{k-1}^{(l)}=a_a\wedge s_k^{(l)}=s_e\right\}$$

接下来需要求解α_{s_s,a_a}的值，将其代入$\sum_{e=1}^{|S|}p_{s,a}(s_e)=1$中有：

$$\sum_{e=1}^{|S|}\left(\frac{1}{\alpha_{s_s,a_a}}\sum_{l=1}^{m}\sum_{k=1}^{t^{(l)}-1}1\left\{s_{k-1}^{(l)}=s_s\wedge a_{k-1}^{(l)}=a_a\wedge s_k^{(l)}=s_e\right\}\right)=1$$
$$\Leftrightarrow\alpha_{s_s,a_a}=\sum_{e=1}^{|S|}\sum_{l=1}^{m}\sum_{k=1}^{t^{(l)}-1}1\left\{s_{k-1}^{(l)}=s_s\wedge a_{k-1}^{(l)}=a_a\wedge s_k^{(l)}=s_e\right\}=\sum_{l=1}^{m}\sum_{k=1}^{t^{(l)}-1}1\left\{s_{k-1}^{(l)}=s_s\wedge a_{k-1}^{(l)}=a_a\right\}$$

代回即可得$p_{s_s,a_a}(s_e)$的表达形式：

$$p_{s_s,a_a}(s_e)=\frac{\sum_{l=1}^{m}\sum_{k=1}^{t^{(l)}-1}1\left\{s_{k-1}^{(l)}=s_s\wedge a_{k-1}^{(l)}=a_a\wedge s_k^{(l)}=s_e\right\}}{\sum_{l=1}^{m}\sum_{k=1}^{t^{(l)}-1}1\left\{s_{k-1}^{(l)}=s_s\wedge a_{k-1}^{(l)}=a_a\right\}}$$

即状态转换概率分布$p_{s,a}(s')$等于在状态s下执行动作a到达状态s'的次数除以状态s下执行动作a的总次数

$$p_{s,a}(s')=\frac{\#times\ \ we\ \ took\ \ action\ \ a\ \ in\ \ state\ \ s\ \ and\ \ got\ \ to\ \ s'}{\#times\ \ we\ \ took\ \ action\ \ a\ \ in\ \ state\ \ s}$$

然而有些情况下会出现在状态s下没有执行过动作a的情况，这会导致分子分母全为 0.为避免这种情况，我们可以采用平滑的办法，或者令其值为$\frac{1}{|S|}$（意为在给定状态s下执行动作a跳转到任意状态集中状态的机会均等）

第十六集：马尔可夫决策过程

- 模型学习
  - 最终可将模型学习的状态转换概率分布和求解最优策略结合在一起，从而得到学习马尔可夫决策模型的方法
    - 1、随机初始化策略π
    - 2、循环迭代直至收敛
      - 2-1、根据策略π进行多次马尔可夫决策过程，从而得到多组马尔可夫决策过程序列，以估计状态转换概率分布P_{sa}和可能的回报函数R（如无定义回报函数时）
      - 2-2、通过值迭代的方法，使用2-1中估计到的参数来更新V(s)
        - 这里有一种加快运行的方法，用上一次迭代得到的V(s)进行本次值迭代的初始化，这比原来值迭代中直接将V(s)初始化为0再运行迭代要来得快
      - 2-3、通过策略迭代的方法，根据2-2中更新的V(s)来更新策略π