统计概率模型

1、高斯判别分析

2、朴素贝叶斯

3、隐马尔可夫模型

4、最大熵马尔科夫模型

5，条件随机场

6，马尔科夫决策过程

一、高斯判别分析

一、生成模型

机器学习模型有一种分类方式是：判别模型和生成模型。它们之前的区别在于判别模型是直接从数据特征到标签，而生成模型是通过标签到特征。形式化的表示就是是否使用了贝叶斯公式：

$$\max P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}$$
$$\rightarrow \max P(X|Y)P(Y)$$

机器学习模型从概率的角度来看就是最大$P(Y|X)$的条件概率，判别模型的思想是直接最大化这个概率（Fisher线性判别，线性感知机），生成模型则是通过贝叶斯模型最大后验概率$P(X|Y)P(Y)$，其中$P(X|Y)$可以看作是从标签d生成数据，$P(Y)$则是标签的先验概率。

基本上从标签到数据的模型都是基于对样本的统计，以下的模型都是基于数据的统计（但不全是生成模型），所以我将这部分模型都归类到统计概率模型。

二、高斯判别分析

高斯判别分析是一个典型的生成模型，其假设$P(X|Y)$服从一个高斯分布，$P(Y)$服从一个伯努利分布通过统计样本来确定高斯分布和伯努利分布的参数，进而通过最大后验概率来进行分类。

假设数据在标签为Y下，特征为X的条件概率为$P(X|Y)$服从多元高斯分布 $X - N(\mu, \Sigma)$，其中μ为均值，Σ为协方差矩阵。则有：

$$P(X|Y) = \frac{1}{2\pi^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}} exp(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu))$$

而先验分布$P(Y)$服从伯努利分布$y \sim Bernoulli(\phi)$，当$y \in (-1, 1)$时，那么是一元伯努利分布，当$y \in (1，2, \dots k)$时，同样可以像Logistic推广到SoftMax一样处理多元伯努利分布。下面以一元伯努利分布为例计算完整的高斯判别模型的概率：

$$y \sim Bernoulli(\phi)$$
$$(x|y=0) \sim N(\mu_0, \Sigma_0)$$
$$(x|y=1) \sim N(\mu_1, \Sigma_1)$$

$$p(y) = \phi^y(1-\phi)^{1-y}$$
$$p(x|y=0) = \frac{1}{2\pi^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}}exp(-\frac{1}{2}(x-\mu_0)^T\Sigma_0^{-1}(x-\mu_0))$$
$$p(x|y=0) = \frac{1}{2\pi^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}}exp(-\frac{1}{2}(x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1))$$

最大化后验概率即为：

$$arg\max_y P(y|x) = \frac{p(x|y)p(y)}{p(x)}$$
$$\rightarrow arg\max_y p(x|y)p(y)$$

极大似然函数有：

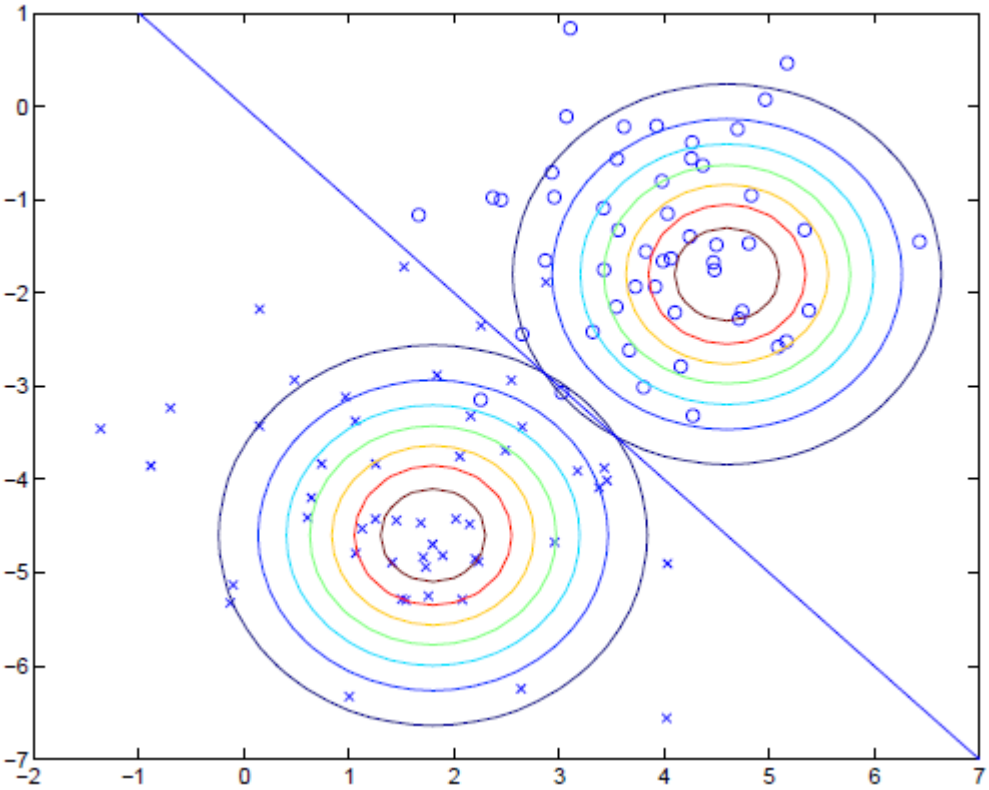
$$\max L(\phi,\mu_0,\mu_1,\Sigma_0,\Sigma_1) = \prod_{i=1}^m p(x^{(i)},y^{(i)};\phi,\mu_0,\mu_1,\Sigma_0,\Sigma_1)$$
$$\Leftrightarrow log\prod_{i=1}^m p(x^{(i)},y^{(i)};\phi,\mu_0,\mu_1,\Sigma_0,\Sigma_1)$$

最大似然估计得到参数如下：

$$\phi = \frac{1}{m}\sum_{i=1}^m 1(y^{(i)}=1)$$
$$\mu_0 = \frac{\sum_{i=1}^m 1(y^{(i)}=0)x^{(i)}}{\sum_{i=1}^m 1(y^{(i)}=0)}$$
$$\mu_1 = \frac{\sum_{i=1}^m 1(y^{(i)}=1)x^{(i)}}{\sum_{i=1}^m 1(y^{(i)}=1)}$$
$$\Sigma = \sum_{i=1}^m (x^{(i)}-\mu_{y^{(i)}})(x^{(i)}-\mu_{y^{(i)}})^T$$

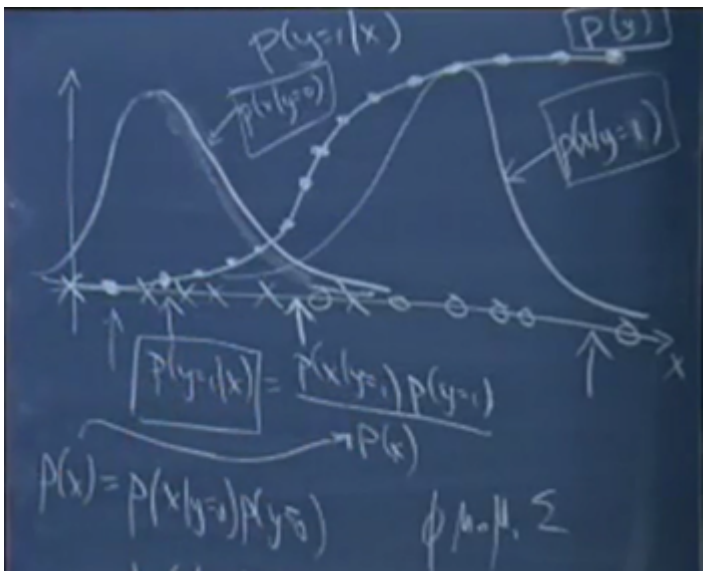
其中$1(y^{(i)}=1)$为指示函数，同时假设$\Sigma_0 = \Sigma_1 = \Sigma$，$\Sigma$反映一类数据分布的方差，可以看出最大似然估计的参数值就是基于对样本的一个统计。

下图为一个简单的高斯判别模型示意图：



从上图可以看出，高斯判别模型通过建立两类样本的特征模型，对于二分类问题，然后通过比较后验概率的大小来得到一个分类边界。

回过头来再看最小错误贝叶斯决策（Logistic回归）与一维高斯判别模型，有趣的是最后得到的决策函数也类似于sigmoid函数。



高斯判别模型与Logistic回归比较

高斯判别模型的假设是$P(X|Y)$服从一个高斯分布，$P(Y)$服从一个伯努利分布

Logistic回归的概率解释中可以看出它的假设是$P(Y|X,\theta)$服从伯努利分布

由高斯判别分析模型可以得到，加上一些推导可以得到：

$$p(y=1|x;\phi,\mu_0,\mu_1,\Sigma)=\frac{1}{1+e^{-\theta^T x}}$$

其中，θ是参数ϕ，μ_0, μ_1, Σ的某种函数。也就是说高斯判别模型是Logistic回归模型中的一种特例。

这里我们就可以发现高斯判别模型的假设强于Logistic模型，也就是说Logistic回归模型的鲁棒性更强。这就表示在数据量足够大时，跟倾向于选择Logistic回归模型。而在数据量较小，且$P(X|Y)$服从一个高斯分布非常合理时，选择高斯判别分析模型更适合。

二、朴素贝叶斯

一、朴素贝叶斯

朴素贝叶斯模型也是一个典型的生成模型，一般用来处理非数值数据。其核心假设是特征之间的条件概率是相互独立的。同样由贝叶斯公式有：

$$\begin{aligned} P(Y|X) &= \frac{P(X|Y)P(Y)}{P(X)} \\ &= \frac{P(Y)}{P(X)}\prod_{i=1}^{n} p(x_i|Y) \end{aligned}$$

下面以垃圾邮件分类介绍两类问题的朴素贝叶斯模型：

垃圾邮件分类任务是一个基本文本分类任务，涉及到NLP的初步知识-文本的one-hot表示。由于机器学习模型通常是一个数学模型，而非数值型属性是不能直接处理，所以一般对邮件的特征进行编码。首先将所有的邮件中出现的词统计出来作为一个词典，并对每一个词进行编码向量化（即词序）。一封邮件对应的One-hot表示如下：

$$x^{(j)} = \begin{bmatrix} 1 \\ 0 \\ . \\ 0 \\ . \\ 1 \end{bmatrix}$$

其中j表示第j封邮件，$x_i^{(j)} \in (0,1)$，i表示词典中的第i个词，如果第i个词在第j封邮件中出现则，$x_i^{(j)} = 1$，反之为0。可以看出这种表示忽略了文本的大量信息，上下文，词出现的次数等。

由上面的公式有，一封邮件是垃圾邮件的概率可以表示为下式：

$$P(Y=1|X) = \frac{P(X|Y=1)P(Y=1)}{P(X)}$$

其中$P(X|Y=1)$似然函数为在垃圾邮件下产生X的条件概率，$P(Y=1)$为垃圾邮件的先验概率，$P(X) = \sum_{\Omega} P(X)$对于所有样本都是一致，近似忽略。

由朴素贝叶斯的条件概率独立性假设有条件概率如下：

$$\begin{aligned} P(x_1, \dots x_i \dots x_{5000}|y=1) &= P(x_1|y=1)\cdot\dots\cdot P(x_i|y=1, x_{i-1}\dots x_1)\cdot\dots\cdot P(x_{5000}|y=1, x_{4999}\dots x_1) \\ &= P(x_1|y=1)\cdot\dots\cdot P(x_i|y=1)\cdot\dots\cdot P(x_{5000}|y=1) \\ &= \prod_{i=1}^{n} P(x_i|y=1) \end{aligned}$$

其中i表示第i个特征。所以，对于每一封邮件属于哪一类的概率为都有：

$$P(Y=k|X) == \prod_{i=1}^{n} P(y=k|x_i)$$

邮件之间独立，所以目标函数最大化所有邮件属于各自类的概率为：

$$\begin{aligned}\max P(Y=k|X) &= \prod_{j=1}^{m} \frac{P(x^{(j)}|y^{(j)}=k)P(y^{(j)}=k)}{P(x^{(j)})} \\ &= \prod_{j=1}^{m}\prod_{i=1}^{n} \frac{P(x_i^{(j)}|y^{(j)}=k)P(y^{(j)}=k)}{P(x^{(j)})} \\ &\approx \max \prod_{j=1}^{m}\prod_{i=1}^{n} P(x_i^{(j)}|y^{(j)}=k)P(y^{(j)}=k)\end{aligned}$$

从上式可以看出朴素贝叶斯的参数是$P(x_i|y=k)$，$P(y=k)$，即所有邮件类别的先验，以及在某一类下出现某个词的概率。由极大似然估计参数值即为其期望。

$$P(y=k) = \sum_{j=1}^{m} \frac{1(y^{(j)}=k)}{m}, k=0,1$$

$$P(x_i=a_{il}|y^{(j)}=k) = \frac{\sum_{j=1}^{m} 1(x_i^{(j)}=a_{il}, y^{(j)}=k)}{\sum_{j=1}^{m} 1(y^{(j)}=k)}, i=1..n$$

其中k表示类别，对应垃圾邮件分类取值为（$0,1$），i表示第i个特征，l表示特征的取值。由于垃圾邮件中采用one-hot编码，所以x_i的取值为（$0,1$），1表示出现。当以上参数确定之后，对于一封新的邮件，根据估计的参数和贝叶斯公式求得样本属于哪一类的概率。最后一封邮件属于哪一类的概率参数表示如下：

$$P(Y=k|X) = \prod_{i=1}^{n} \frac{P(x_i=a_{il}|y)P(y=k)}{P(x)}$$

由于one-hot编码比较特殊，$P(x_i=0|Y=k)+P(x_i=1|Y=k)=1, a_{il}\in\{0,1\}$。由于所有类的概率加和为1，垃圾邮件为二分类，所以邮件属于概率大于$\frac{1}{2}$的那一类。

为了使模型更具普适性，考虑到当某一特征没有在训练集中出现过，即某一个单词在某一类下没有出现过，或者某一单词在某一类下都出现过（意味着不出现的条件概率为0）。但不能说该单词在这一类下的条件概率为0。又或者在所有类中都未出现（即原始训练集中没有的词，而词典中有的词，即词典不依赖于训练集)。当来一个新样本时，如果不做处理，那么只要有一个分量的概率为0，由于特征之间的条件概率独立，连乘形式只要有一个为0，即整个概率为0，无意义。

拉普拉斯平滑：

$$P(x_i=a_{il}|y^{(j)}=k) = \frac{\sum_{j=1}^{m} 1(x_i^{(j)}=a_{il}, y^{(j)}=k)+1}{\sum_{j=1}^{m} 1(y^{(j)}=k)+\Omega(a_{il})}, i=1..n$$

其中$\Omega(a_{il})$为第i个特征分量x_i的可能取值数。

二、N元多项分布模型

同样，上述贝叶斯模型中只考虑单词是否出现，即单词特征x_i服从伯努利分布，样本$X^{(j)}$服从n次独立的伯努利分布。而忽略了一个单词可能出现次数对邮件分类的影响。假设要统计某一单词出现的次数，那么有$x_i \in \{0,1,\dots k\}$多项分布。只考虑单词是否出现的贝叶斯模型叫multi-variate Bernoulli event model，后者叫multinational event model。 同样以邮件分类问题介绍multinational event model，在之前的模型中，我们首先建立词典，并且特征向量长度为词典长度，并且从词典出发，对于邮件出现过的单词，在对应词典的位置标记为1，反之标记为0产生一个特征向量x_i。而multinational event model则从邮件出发， 表示邮件中第i个单词，其值表示第i个单词在字典中出现的位置，那么x_i的取值则有$|V|$，其中V表示字典长度。这样一封邮件可以表示为$(x_1, x_2, \dots, x_{n_j})$，$n_j$表示第j封邮件的长度。这相当于掷一枚有V面的骰子n次，将观测值记录下来形成一封邮件。假设出现某一点的情况与第几次掷无关，也就是单词在邮件中出现的位置无关，而且每一次投掷都是独立的，即单词之间出现的事件是独立的。

文档的表示：

1)one-hot表示

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ . \\ x_{3234} \\ . \\ x_{50000} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ . \\ 1 \\ . \\ 0 \end{bmatrix}$$

2)编号表示

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ . \\ x_{45} \\ . \\ x_{203} \end{bmatrix} = \begin{bmatrix} 4790 \\ 34689 \\ 24567 \\ . \\ 23 \\ . \\ 415 \end{bmatrix}$$

可以看出两者方式的样本表示不同之处在于一个以词典维度对邮件中的词是否出现进行0-1编码，一个是以邮件维度对邮件中的词在词典中的编号进行编码，这就导致了两者表示的维度不同，特征服从的分布也不同。

一封邮件属于垃圾邮件的概率由贝叶斯公式有：

$$P(Y=1|X) = \frac{P(X|Y=1)P(Y=1)}{P(X)}$$

其中$P(X|Y=1)$似然函数，在垃圾邮件下产生X的条件概率，$P(Y=1)$为垃圾邮件的先验概率，$P(X) = \sum_{\Omega} P(X)$对于所有样本都是一致，近似忽略。

由朴素贝叶斯的条件概率独立性假设有条件概率如下：

$$\begin{aligned} P(x_1, \dots x_i \dots x_{n_j}|y=1) &= P(x_1|y=1)\cdot\dots\cdot P(x_i|y=1, x_{i-1}\dots x_1)\cdot\dots\cdot P(x_{n_j}|y=1, x_{4999}\dots x_1) \\ &= P(x_1|y=1)\cdot\dots\cdot P(x_i|y=1)\cdot\dots\cdot P(x_{n_j}|y=1) \\ &= \prod_{i=1}^{n_j} P(x_i|y=1) \end{aligned}$$

其中$x_i \in \{0,1,\dots|V|\}$。同样最大化似然函数：

$$\max P(Y=k|X)=\prod_{j=1}^{m}\frac{P(x^{(j)}|y^{(j)}=k)P(y^{(j)}=k)}{P(x^{(j)})}$$

$$=\prod_{j=1}^{m}\prod_{i=1}^{n_j}\frac{P(x_i^{(j)}|y^{(j)}=k)P(y^{(j)}=k)}{P(x^{(j)})}$$

$$\approx \max\prod_{j=1}^{m}\prod_{i=1}^{n_j}P(x_i^{(j)}|y^{(j)}=k)P(y^{(j)}=k)$$

其中n_j表示第j封邮件的长度。所以上式中的参数有$P(y=k)$，$P(x_i=a_v|y=k)$。由最大似然估计有：

$$P(y=k)=\sum_{j=1}^{m}\frac{1(y^{(j)}=k)}{m},k=0,1$$

$$P(x_i=a_v|y^{(j)}=k)=\frac{\sum_{j=1}^{m}1(x_i^{(j)}=a_v,y^{(j)}=k)}{\sum_{j=1}^{m}1(y^{(j)}=k)},a_v\in\{0,1,\dots|V|\}$$

其中$x_i=a_v$与i无关，我们需要求的是x_i所有可能的取值。

最后一封邮件属于哪一类的概率参数表示如下：

$$P(Y=k|X)=\prod_{i=1}^{n_j}\frac{P(x_i=a_v|y)P(y=k)}{P(x)}$$

其中n_j表示第j封邮件的长度,$x_i=a_v$表示邮件第i个词在词典中的编号。

one-hot表示和编号表示：

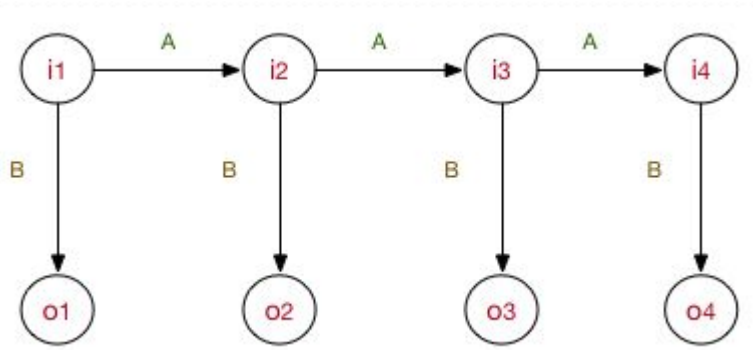
两种表示最大的差别在于包含的语义信息，one-hot表示信息不够丰富，只有0-1，所以需要高的维度，而编号表示信息相对丰富，维度低。然而on-hot表示是可以直接度量两个样本之间的相似性的（0-1表示是否存在，有语义意义的），而编号表示则不能直接度量两个样本之间的相似性（在词典中的编号是无语义的），但是可以把编号表示放回到集合中去度量两个样本的重合度。所以编号表示可以看作是one-hot的一种降维表示。

三、隐马尔可夫模型

一、隐马尔科夫模型定义

隐马尔科夫模型是一种时序的概率模型，描述由一个隐的马尔科夫链随机生成的不可观察的隐状态序列，在每一个隐状态下随机产生观察值构成一个可观测的随机序列。其中关键是状态序列是满足马尔科夫性质的，且可观测序列是由隐藏的状态序列以一定的概率随机生成。

在自然语言中文分词中，由于自然语言是有明显的上下文关系的，即当前字与其前后出现的字都是有关系的。为了表示前一个字对当前字的影响，我们用一个隐状态来表示前的语义状态，用在前一个状态下转移到发射出当前字的隐状态的概率表示前一个字对当前字的影响。整个来说就是把上下文字对字的影响转化成状态对状态的影响。而用发射概率来表示状态到字的关系。值得注意的是隐马尔可夫模型中$p(o_t,i_t|i_{t-1})=p(o_t|i_t)p(i_t|i_{t-1})$，即$i_{t-1}$与$o_t$之间独立作用$i_t$。



隐马尔科夫模型由状态集，观测集，初始状态转移概率，状态转移概率，以及发射概率确定。

形式化定义为：

所有可能的隐藏状态集Q，所有可能的观察值集V，其中n是可能的状态数，m是可能的观察数。

$$Q = \{q_1, q_2, \ldots, q_n\}, V = \{v_1, v_2, \ldots, v_m\}$$

假设I是长度为T的隐状态序列，O是其对应的观测值序列。

$$I = \{i_1, i_2, . i_t . . , i_T\}, O = \{o_1, o_2, . o_t . . , o_T\}$$

A是状态转移概率矩阵：

$$A = [a_{ij}]_{n \times n}$$

其中$a_{ij} = p(i_{t+1} = q_j | i_t = q_i)$,表示第$t$时刻在$q_i$状态下转移到第$t+1$时刻状态$q_j$的概率。

B是发射概率矩阵，在隐状态确定之后发射出观测状态的概率：

$$B = [b_j(k)]_{n \times m}$$

其中$b_j(k) = p(o_t = v_k | i_t = q_j)$，表示在状态$q_j$下发射出$v_k$的概率。

π是初始状态的概率分布：

$$\pi = (\pi_i)$$

其中$\pi_i = p(i_1 = q_i)$，表示在$t = 1$时刻状态为q_i的概率。

由此，马尔科夫模型定义完成。至于为何这样定义，隐状态的意义是什么，就是模型的价值所在，如何理解隐状态也是一种个人体会。

有了隐马尔科夫模型，接下来看隐马尔科夫模型能做什么？

1、给定一个确定的隐马尔科夫模型（参数$\lambda = \{A, B, \pi\}$确定）和观察序列O，计算在该参数下观察序列的输出概率 。

概率计算，由于观测序列的产生于隐状态是相关的，所以需要从隐状态的转移概率入手，通过发射概率间接的转化到观察序列。一般情况下该观测序列对应的隐状态序列有多个，把所有隐状态可能的序列结合观察序列求概率，再求和。

2、学习问题，已知观察序列O，估计模型参数$\lambda = \{A, B, \pi\}$，使得在该模型下观测序列的概率最大。

学习问题，假设在不知道模型参数的情况下，而我们有大量的观察序列，那么这些大量的观察序列一定不是偶然是这样,而不是那样的。从概率的角度来讲，是这样，而不是那样的原因就是，是这样的概率大于是那样的概率。如果有大量的观察序列，那么其中必然隐藏了模型的信息。

3、预测问题，已知模型的参数$\lambda = \{A, B, \pi\}$和观察序列O，求解一条使得该观测序列概率最大的隐状态序列 。这样概率计算类似，只需要求最大的即可。

好了，对应上面的三个问题，分别有三个算法求解对应的问题。

1概率计算-前向后向算法

2 参数学习-最大似然估计（有监督），Baum-Walch（无监督）

3 预测-Viterbi算法

一、概率计算(观察序列的概率)

给定一个确定的隐马尔科夫模型（参数$\lambda = \{A, B, \pi\}$确定）和观察序列$O = \{o_1, o_2, .. o_t .., o_T\}$，计算在该参数下观察序列的输出概率 。最直接的方法是计算所有可能的概率，即：

$$P(O|\lambda) = p(O, I|\lambda) = p(O|I, \lambda)p(I|\lambda)$$

其中$I = i_1, i_2, .., i_T$，这T个状态我们是看不见的，且没个时刻i_t的取值都有N中，由于隐状态与观察状态无关，其概率为：

$$p(I|\lambda) = \pi_{i1} a_{i_1 i_2} a_{i_2 i_3} \dots a_{i_{T-1} i_T}, i = 1, 2...N$$

由于$a_{i_t i_{t+1}}$的取值有N^2种，但序列前后有一个相同的状态，所以整个$p(I|\lambda)$的复杂度是TN^T。

而在参数和隐状态都确定的条件下，产生观察序列$O = \{o_1, o_2, .. o_t .., o_T\}$的概率为：

$$p(O|I, \lambda) = b_{i_1(o_1)} b_{i_2(o_2)}, \dots, b_{i_T(o_T)}$$

即整个T时刻的发射概率的乘积。

因此在给定参数的条件下，产生观察序列$O = \{o_1, o_2, .. o_t .., o_T\}$的概率为

$$\begin{aligned} P(O|\lambda) &= p(O, I|\lambda) = p(O|I, \lambda)p(I|\lambda) \\ &= \pi_{i1} b_{i_1(o_1)} a_{i_1 i_2} b_{i_2(o_2)} a_{i_2 i_3} \dots a_{i_{T-1} i_T} b_{i_T(o_T)} \end{aligned}$$

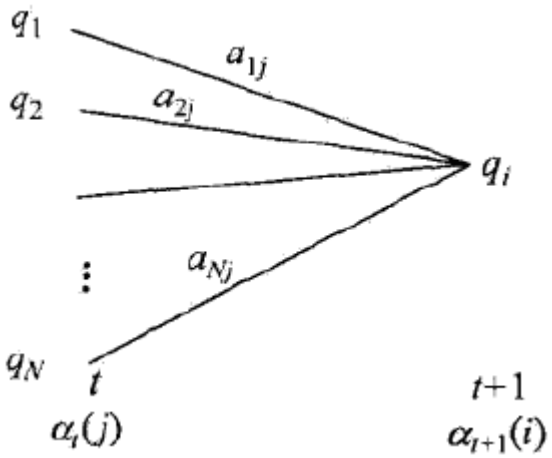
算法的复杂度为TN^T。之所以算法的复杂度高是直接计算$a_{i_{t-1} i_t}$和$a_{i_t i_{t+1}}$，而忽略了序列之间的递推关系。

下面介绍隐马尔可夫概率计算问题中的前向-后向算法

前向概率：在给定模型的参数和观察序列$O = \{o_1, o_2, .. o_t\}$下，$a_t(i)$表示t时刻$a_t = i$的前向概率（从$t = 1$时刻到t时刻观察序列$O = \{o_1, o_2, .. o_t\}$,$a_t = i$）：

$$a_t(i) = p(o_1, o_2, .., o_t, i_t = q_i|\lambda)$$

由前向递推关系$a_t(i)$等于在所有可能的前一状态转移到当前状态（同时t时刻发射出观测值o_t）的概率之和



因此前向算法计算如下：

1)初值：

$$a_1(i) = \pi_{i1} b_i(o_1), i = 1..n$$

2)前向递推:

$$a_{t+1}(i) = [\sum_{j=1}^{n} a_t(j) a_{ji}] b_i(o_{t+1})$$

3)求和：

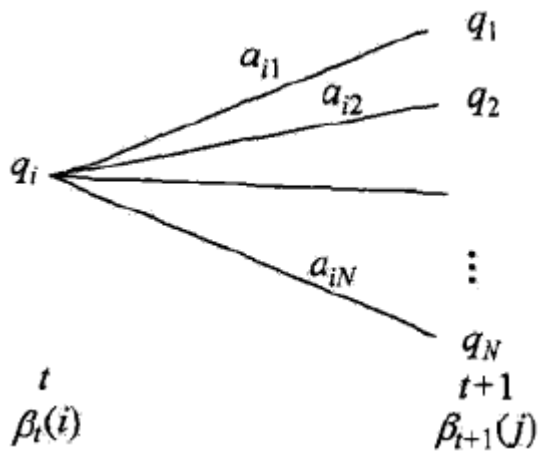
$$p(O|\lambda) = \sum_{i=1}^{n} a_T(i)$$

后向概率：在给定模型的参数和观察序列$O = \{o_{t+1}, o_{t+2}.., o_T\}$下，$\beta_t(i)$表示$t$时刻$a_t = i$的后向概率（从$t$时刻到$T$时刻观察序列$O = \{o_{t+1}, o_{t+2}.., o_T\}$，$a_t = i$）：

$$\beta_t(i) = p(o_{t+1}, o_{t+2}, .., o_T, i_t = q_i|\lambda)$$

值得注意的是，后向概率表示序列从t时刻到T时刻的概率，所以$\beta_t(i) \leq \beta_{t+1}(j)$

由后向递推关系$\beta_t(i)$等于所有可能的后一状态逆转移到当前状态（同时$t+1$时刻发射出观测值o_{t+1}）的概率之和



因此后向算法计算如下：

1)初值：

$$\beta_T(i) = 1, i = 1, 2...n$$

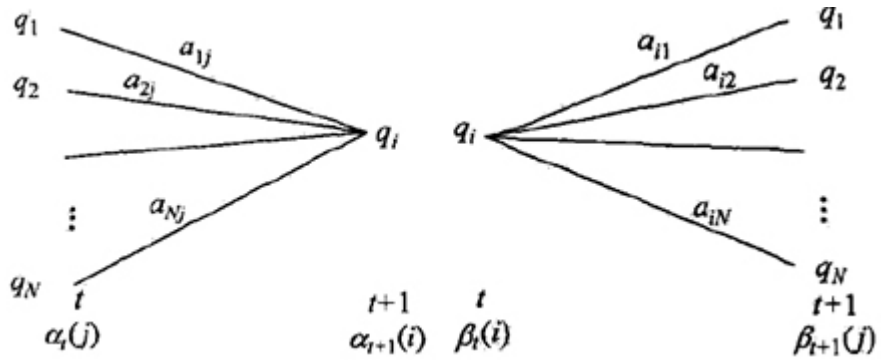
2)反向递推：

$$\beta_t(i) = \sum_{j=1}^{n} a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)$$

3)求和：

$$p(O|\lambda) = \sum_{i=1}^{n} \pi_{i1} b_i(o_1) \beta_1(i)$$

前向后向算法：



由上面的前向后向算法，固定t时刻的状态$i_t = q_i$，由前向后向算法有：

$$p(O|\lambda) = \sum_{i=1}^{n} \sum_{j=1}^{n} a_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j), t = 1, \ldots, T$$

二、参数估计

一般来讲，隐马尔可夫的参数估计问题分为两种，一种是有监督，一种是无监督的。有监督意味着给定的训练集中观测序列$O = \{o_1, o_2, .. o_t .., o_T\}$和隐状态序列$I = i_1, i_2, .., i_T$，此时对应的参数估计问题就可以直接采用最大似然估计；无监督意味着给定的训练集中只有观测序列$O = \{o_1, o_2, .. o_t .., o_T\}$，此时需要采用EM算法思想，先假设参数，通过期望最大化来获得隐状态序列$I = i_1, i_2, .., i_T$（硬划分隐状态序列对应到值，软化分隐状态序列对应到概率），然后根据隐状态序列来更新参数，不断迭代至收敛。

有监督(最大似然估计)：

转移概率a_{ij}表示从状态i转移到状态j的概率

$$a_{ij} = \frac{A_{ij}}{\sum_{j=1}^{n} A_{ij}}, i = 1..n, j = 1..n$$

其中分子表示从i状态转移到j状态的次数，分母表示从i状态转移到任意状态的次数。

发射概率$b_i(o_k)$表示在状态i下发射出观测值o_k的概率：

$$b_i(o_k) = \frac{B_{ik}}{\sum_{k=1}^{m} B_{ik}}, i = 1..n, k = 1..m$$

其中分子表示在状态i下发射出观测值o_k的次数，分母表示在状态i下发射出任意状态的次数。

初始状态转移概率π_{i1}为样本中初始状态的概率：

$$\pi_{i1} = \frac{a_i}{\sum_{i=1}^{n} a_i}$$

其中分子表示初始状态是i的次数，分母表示所有初始状态出现的次数。

无监督（Baum-Welch）：

隐马尔可夫模型中隐状态其实是一个隐变量，EM算法这类含有隐变量模型的通用求解算法，思路是初始化一个隐变量的概率分布，E步：期望最大化来更新样本的隐变量(值，概率)，M步：在隐变量确定的条件下更新隐变量的概率。

三、状态预测

已知模型的参数$\lambda = \{A, B, \pi\}$和观察序列O，求解一条使得该观测序列概率最大的隐状态序列 。这样概率计算类似，只需要求最大的即可。

维特比算法：维特比算法是一种动态规划算法来求解概率最大路径，也是一种求解最优路径问题。而最优路径中总存在这样一个特性：如果最优路径t时刻通过结点i_t，那么最优路径中从结点i_t到最终结点i_T的部分路径是所有可能从i_t到i_T路径中最优的（同时从i_1到i_t的路径也是最优的）。依据这一特性，我们可以从$t=1$开始递推计算时刻t下状态为i的各种路径的最大概率，直至时刻$t=T$状态为i的最大概率。同时在递推的过程中，我们用一个变量来计住到达最优路径的上一个结点的状态。这样我们就首先确定了$t=T$时刻的状态值i。然后，根据到达该状态的上一个结点状态来递推到$i_{T-1}, ..i_t., i_1$。

因此，我们需要引入两个变量，从$t=1$时刻到t时刻状态为i的最优路径的概率值，并以此来递推下一时刻状态为i的最优路径，即

$$\sigma_t(i) = \max_{i_1, i_2, .., i_{t-1}} p(i_t = i, i_{t-1}..i_1, o_t..o_1|\lambda), i = 1, 2..n$$

$$\begin{aligned}\sigma_{t+1}(i) &= \max_{i_1, i_2, .., i_t} p(i_{t+1} = i, i_t..i_1, o_{t+1}..o_1|\lambda), i = 1, 2..n, t = 1..T-1 \\ &= \max_{j \in 1..n} \sigma_t(j) a_{ji} b_i(o_{t+1})\end{aligned}$$

同时为了记住到达该路径的上一节点的状态，定义如下变量：

$$\phi_t(i) = arg \max_{j \in 1..n} \sigma_{t-1}(j) a_{ji}, i = 1..n$$

有了上面的两个变量，我们就可以获得隐状态的最优路径

1）初始化

$$\begin{aligned}\sigma_1(i) &= \pi_{i1} b_i(o_1), i = 1..n \\ \phi_1(i) &= 0\end{aligned}$$

2)递推，对$t = 2, 3..T$

$$\sigma_t(i) = \max_{j \in 1..n} \sigma_{t-1}(j) a_{ji} b_i(o_t), i = 1..n$$

$$\phi_t(i) = arg \max_{j \in 1..n} \sigma_{t-1}(j) a_{ji}, i = 1..n$$

3) 终止

$$P^* = \max_{i \in 1..n} \sigma_T(i)$$

$$i_T^* = arg \max_{i \in 1..n} \sigma_T(i)$$

4)最优路径回溯，$t = T - 1..1$

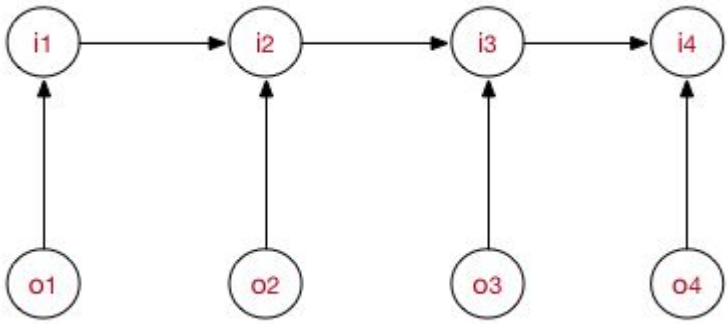
$$i_t^* = \phi_{t+1}(i_t^*)$$

求得最优路径$I = \{i_1^*, i_2^*, .. i_T^*\}$。

其中值得注意的是，$\phi_1(i) = 0$是无用的，在前向递推到T时刻获得最大概率的同时也获得了最优的最终状态i_T^*，回溯的过程只需要从$T-1$开始，不需要任何计算，因为ϕ中保存了到达当前最优路径状态的上一状态。

四、最大熵马尔科夫模型

有最大熵模型和隐马尔可夫模型的基础，再看最大熵马尔科夫模型就直观多了。在隐马尔可夫模型中，$p(o_t, i_t|i_{t-1}) = p(o_t|i_t)p(i_t|i_{t-1})$，即$i_{t-1}$与$o_t$之间独立作用$i_t$。在最大熵马尔科夫模型中则没有这一假设，而直接采用条件概率的形式$p(i_t|o_t, i_{t-1})$输出模型。



结合最大熵模型，不考虑整个序列时，第t时刻的状态可以看作是一个分类问题，采用最大熵模型，由i_{t-1}和o_t，i_t构成分类模型$p(i_t = i|o_t\ ,\ i_{t-1})$，有最大熵模型的结论，我们知道分类模型是一个关于λ的函数，表达式如下：

$$p(i_t = i|o_t\ ,\ i_{t-1}) = \frac{exp(\sum_a \lambda_a f_a(o_t, i_t = i))}{z(o_t, i_{t-1})}$$

$$z(o_t, i_{t-1}) = \sum_i exp(\sum_a \lambda_a f_a(o_t, i_t = i))$$

其中$f_a(o_t, i_t = i))$是联合标签$i_t = i$特征模板，λ_a是特征模板的权重，$z(o_t, i_{t-1})$是联合所有可能的标签$i_t = i, i \in \{1..n\}$特征模板求和，表示归一化因子。对于参数λ的求解，可以采用最大熵模型的使用的优化算法，但是值得注意的是，在优化求解过程中，每个时刻单独归一化，不考虑序列性。

这里，由于笔者之前的误解，对于最大熵模型的特征模板的概率求解采用最大似然估计的方式直接对特征模板进行统计，以其频率作为概率，结果发现还是有效。其中原因可能是我的这种统计方式是基于条件概率服从伯努利分布假设，运用最大似然估计得到模型参数正好是统计频率，而服从伯努利分布假设也有一定的合理性。

$$p(i_t = i|o_t\ ,\ i_{t-1}) = \frac{\sum_a f_a(o_t, i_t = i)}{\sum_i \sum_a f_a(o_t, i_t = i)}$$

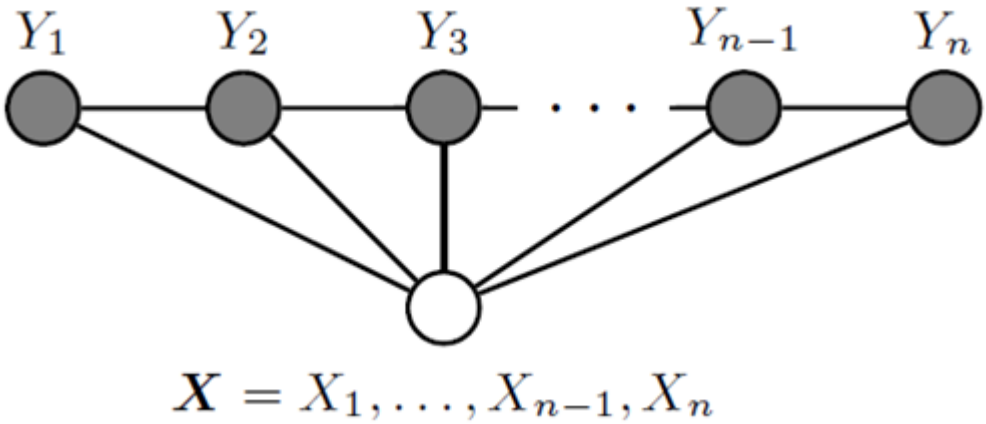
在状态预测中，考虑最大化整个序列的概率，意味着目标函数如下：

$$\max \prod_{t=1}^{T} p(i_t = i|o_t，i_{t-1}), i = 1..n$$

目标函数也就是求解一条最优的状态转移路径，同样可以采用Viterbi算法。

五、条件随机场

条件随机场是一个概率图模型，深入图模型的话实在有太多东西。这里，我们接着隐马尔科夫模型和最大熵马尔科夫模型基础理解条件随机场。在序列标注问题上，条件随机场与两者之间的差异



1）隐马尔科夫模型是一个生成模型，其假设当前时刻状态只与上一状态有关，而当前的观测值只与当前的状态有关，所以独立性假设非常强。

2）最大熵马尔科夫模型则通过特征模板的定义克服了独立性假设问题。基于熵原理，在满足所有条件经验期望的条件下，熵最大的为最好的模型，也就导出了对数线性模型，是一个标准的判别模型。

3）条件随机场同最大熵马尔科夫模型非常一致，也是一个基于特征模板的判别模型。然而在序列标注问题上，最大熵马尔科夫模型将每个时刻看作是一个分类问题，每时刻独立归一化，这就导致标注偏置问题，条件随机场则归一化作用于整个序列。

结合最大熵模型，模型输出条件概率$P(Y|X)$，假设所有的特征模板为$f_i(x,y)$，（其中一些书中分为转移特征和发射特征）。最大化条件概率为：

$$P_w(y|x) = \frac{1}{z_w(x)}\left(exp\sum_i w_i f_i(x,y)\right)$$
$$z_w(x) = \sum_y exp(\sum_i w_i f_i(x,y))$$

考虑整个序列的条件概率$P(Y|X)$，条件随机场的目标函数是最大化$P(Y|X)$：

$$P_w(Y|X) = \frac{1}{z_w(x)}\left(exp\sum_i \sum_{t=1}^{T} w_i f_i(x,y)\right)$$
$$z_w(X) = \sum_y exp(\sum_i \sum_{t=1}^{T} w_i f_i(x,y))$$

对于参数的学习，同样可以采用最大熵模型使用的优化算法，比如梯度下降的方法。

状态预测问题就是一个最大化序列概率获得状态序列：

$$
\begin{aligned}
y^* &= arg\max_y P_w(y|x) \\
&= arg\max_y \frac{exp(\sum_i w_i f_i(x,y))}{z_w(x)} \\
&= arg\max_y (\sum_i w_i f_i(x,y))
\end{aligned}
$$

其中归一化因子忽略，分子的指数形式单调递增也忽略。所以整个序列的状态等价于在最优的状态序列下特征模板与特征模板权重乘积和最大。就此，条件随机场就变成了一个特征模板定义的问题了，特征模板直接决定条件随机场的性能。

六、马尔科夫决策过程

在机器学习算法（有监督，无监督，弱监督）中，马尔科夫决策过程是弱监督中的一类叫增强学习。增加学习与传统的有监督和无监督，弱监督方法不同的地方是，这些方法都是一次性决定最终结果的，而无法刻画一个决策过程，无法直接定义每一次决策的优劣，也就是说每一次的决策信息都是弱信息，所以某种程度上讲，强化学习也属于弱监督学习。从模型角度来看，也属于马尔科夫模型，其与隐马尔科夫模型有非常强的可比性。

下面是一个常用的马尔科夫模型的划分关系

	不考虑动作	**考虑动作**
状态完全可见	马尔科夫链(MC)	马尔科夫决策过程(MDP)
状态不完全可见	隐马尔科夫模型(HMM)	不完全可观察马尔科夫决策过程(POMDP)

马尔科夫决策过程

马尔科夫决策过程由五元组组成$\{S, A, P_{sa}, \gamma, R\}$

S：表示转态集合

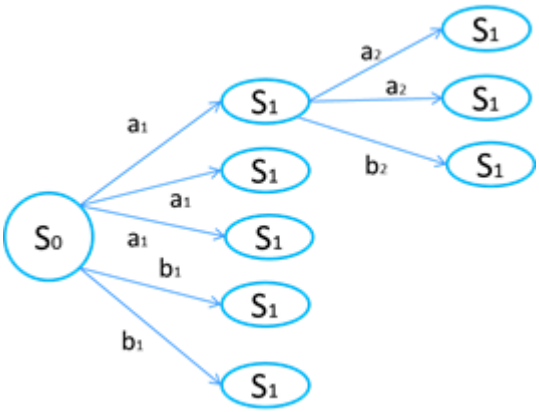
A：表示一组动作

P_{sa}：表示在某一状态S_i下，采取动作A_i，转移到S_{i+1}转态的概率，也就是说在确定的状态下采取相应的动作之后不能完全确定下一状态，而是以一定的概率确定下一状态。

γ：表示决策过程的一个阻尼系数，用户定义回报在决策过程中随时间打折扣，加快决策国产的收敛

R：表示在该状态下的一个回报$R(s)$，有时由动作和状态共同决定回报该时刻的回报$R(a,s)$。

有了上面的定义之后，一个完整的马尔科夫决策过程状态转移图如下：



该过程表示从S_0出发，有决策函数来选择相应的动作a_0，然后以概率$P_{a,s}$到达下一状态$S_i \in S\{P_{sa}\}$，这里的S_i只是表示第i时刻的状态，而S_i的值属于状态集。

回报函数定义之后，整个决策过程的累积回报如下：

$$R(s_0,a_0)+\gamma^1 R(s_1,a_1)+....+\gamma^i R(s_i,a_i)+...$$

当回报函数与状态无关累积回报如下：

$$R(s_0)+\gamma^1 R(s_1)+....+\gamma^i R(s_i)+...$$

其中γ为折扣因子，随着决策不断进行，回报不断打折扣。

当定义不同决策函数时，我们会得到不同的回报，因此就定义了一个决策到回报的函数。在整个决策过程中，给定决策函数$a=\pi(s)$—在s状态下采取$\pi(s)$动作。因此，从状态$s=s_0$出发，采用决策函数$a=\pi(s)$，有累积回报函数如下：

$$V^{\pi}(s)=E[R(s_0)+\gamma^1 R(s_1)+....+\gamma^i R(s_i)+...|s=s_0,\pi]$$

直接最大化累积回报函数不易，从递推角度来看，由贝尔曼方程有：

$$\begin{aligned}V^{\pi}(s)&=E[R(s_0)]+\gamma(E[R(s_1)+\gamma R(s_2)+...\gamma^{i-1}R(s_i)...|s=s_1,\pi])\\&=R(s_0)+\gamma\sum_{s_1\in S}P_{s_0\pi(s_0)}V^{\pi}(s)\end{aligned}$$

其中$R(s_0)=V(s_0->s_0)$为立即回报，$s_1\in S$表示由s_0采取π_{s_0}动作之后转移到下一个状态集中，具体到哪个状态的概率为$P_{s_0\pi(s_0)}$。其解释性可以理解为下象棋最终的累积回报为输赢，在第s状态下的累积回报则是当前状态下的立即回报以及未来的回报。第一项为立即回报，第二项就是未来的回报。

有了上面的贝尔曼方程，我们的目标就是最大化任意状态下出发的累积回报函数$V^{\pi}(s)$，其中$a=\pi(s)$也是一个决策函数，但是在累积回报函数中它是我们需要优化的变量。目标函数如下：

$$\begin{aligned}V^*(s)&=\max_{\pi}V^{\pi}(s)\\&=R(s)+\max_{\pi}\gamma\sum_{s_i\in S}P_{sa}(s_1)V^{\pi}(s_1)\end{aligned}$$

由目标函数可以看出，最优回报和最优决策一一对应。最大化累积回报对应的决策函数π就是最有决策，最有决策对应的累积回报也是最大累积回报，所以最有决策如下：

$$\pi^*(s)=arg\max_{a\in A}\sum_{s_i\in S}P_{sa}(s_1)V*(s_1)$$

有了最优决策和最大累积回报，那么必定有下式：

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

也就是说最优决策下对应的累积回报一定不小于一般的决策下的累积回报。

值得注意是，最优决策是出于全局考虑的，是从所有状态下出发到得到的累积回报的加和最大，这就意味着决策函数不保证其中每一个状态出发根据决策函数得到的累积回报都是最大的。

最优决策

也许上面的目标函数还不清晰，如何求解最有决策，如何最大化累积回报

下面结合例子来介绍如何求解上面的目标函数。且说明累积回报函数本身就是一个过程的累积回报，回报函数R才是每一步的回报。

→	→	→	+1
↑	无	↑	−1
←	←	←	←

0.85	0.90	0.93	+1
0.82	无	0.69	−1
0.78	0.75	0.71	0.49

下面再来看求解上述最优问题，其中 就是以s为初始状态沿着决策函数走到结束状态的累积回报。

值迭代

1将每一个初始状态为s的$V(s)$初始化为0,目标状态累积回报为1

2循环直到收敛{

对于每一个初始状态s，对V进行更新$V(s) = R(s) + \gamma \max_{a \in A} \sum_{s_1 \in S} P_{s_0\pi(s_0)} V^{\pi}(s)$

}

可以看出，更新第一次所有的$V_s := R(s)$，也就是说都只看眼下的立即回报，然后由于奖励状态和惩罚状态的分布不同，由靠近奖励状态和惩罚状态的状态决策逐渐导向到初始状态的决策，这也就是累积回报不断更新的原因（动力）。但是值得思考的还是最终会不会收敛到最优累积回报（暂时不作讨论）。

内循环迭代的的处理方法有两种：

1同步迭代：即在一次循环过程中，累积回报不更新，而是计算完所有的累积回报之后，再统一更新。

2异步迭代，即在一次循环过程中，每计算完一个初始状态下累积回报就立即更新，不需要等到所有的累积回报都计算出来之后再更新。

可以看出两种迭代方式造成不同的原因是第二项，因为立即更新之后，再计算下一个初始状态下的累积回报与暂时不更新得到的累积回报肯定不一样，拿第一次更新为例，同步更新第一次 $V(s) = R(s)$，而异步更新则第一次内循环中，除了第一次更新的s会出现 $V(s) = R(s)$，剩下的都有 $V(s) \neq R(s)$，值得肯定的是异步迭代的收敛速度肯定是快于同步迭代。

策略迭代

值迭代是使累积回报值最优为目标进行迭代，而策略迭代是借助累积回报最优即策略最优的等价性，进行策略迭代。

1随机指定一个策略 $\pi : S \to A$ 。

2循环直到收敛{

a:令$V := V^{\pi}$

b:对于每一个状态s，对$\pi(s)$做更新$\pi_s := arg\max_{a \in A} \sum_{s_1 \in S} P_{sa}(s_1)V(s_1)$

}

这里要说明的是a步是通过前面的贝尔曼方程，以解方程的形式求解出每一个状态下的累积回报：

$$V(s) = R(s) + \gamma \sum_{s_1 \in S} P_{s_0\pi(s_0)} V^{\pi}(s)$$

在b步则是根据累积回报值，重新更新决策$\pi(s)$。

同样，收敛性也是值得探讨的，这里简单的思考一下，由于奖励状态和惩罚状态的分布，以及累积回报唯一确定决策函数，那么未达到最优决策，必然累积回报和决策函数处于不稳定的状态，而只有当到达最优决策时，才有

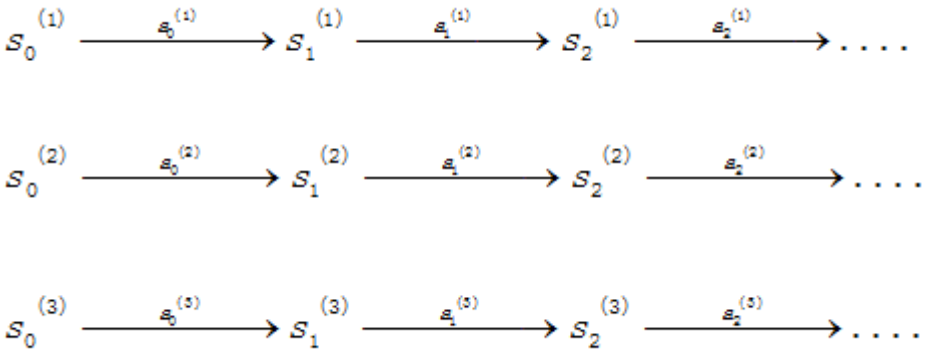
$$V^*(s) = V^{\pi^*}(s) \geq V^{\pi}(s)$$

所以该过程就是在a步由决策函数确定累积回报，然后最大化累积回报来更新决策，如此反复，则有最优决策。值迭代和策略迭代比较：可以看出策略迭代涉及从决策函数到累积回报的解线性方程组的步骤，值迭代则是反复的，所以策略迭代更适合处理少量状态的情况，一般10000以内还是可以接受的。

MDP中的参数估计

回过头来再来看前面的马尔科夫决策过程的定义是一个五元组，一般情况下，五元组应该是我们更加特定的问题建立马尔科夫决策模型时该确定的，并在此基础上来求解最优决策。所以在求解最优决策之前，我们还需更加实际问题建立马尔科夫模型，建模过程就是确定五元组的过程，其中我们仅考虑状态转移概率 ，那么也就是一个参数估计过程。（其他参数一般都好确定，或设定）

假设，在时间过程中，我们有下面的状态转移路径：



其中$s_i^{(j)}$表示i步，第j条转移路径对应的状态，$a_i^{(j)}$是$s_i^{(j)}$状态下执行的动作，每一条转移路径中状态数都是有限的，在实际过程中，每一个状态转移路径都要进入终结状态。如果我们获得了很多上面的转移路径，那么我们就可以来估计参数P_{sa}

$$P_{sa} = \frac{s_0 a s_1}{s_0 a}$$

分子是在s_t状态下采取a动作都转移到s_{t+11}的次数，分母是在s状态下采取a动作的次数。为了避免$\frac{0}{0}$的情况，同样采用拉普拉斯平滑。也就是说当到达的状态是样本中为为到达过的状态，那么在该状态下的执行的动作达到下一状态的概率均分。上面的这种估计方法是从历史数据中进行统计的，同样该方法适合于在线更新。对于立即回报函数的估计，一般根据实际情况学习或者设定。

所以整个马尔科夫决策过程流程如下（以策略迭代为例）：

1 随机初始化策略 π：$S \to A$ 。

2循环直到收敛{

a在样本上统计该策略下每个状态转移的次数，来估计P_{sa}和R

b使用估计到参数来更新对应决策函数下的累积回报V

c根据更新的累积回报V重新进行决策，即更新π

}

整个流程就是在策略迭代的基础上，同时进行了参数估计。