隐马尔科夫模型

隐马尔科夫模型（Hidden Markov Model(HMM)）是可用于标注问题的统计学习模型，描述由隐藏的马尔科夫链随机生成观察序列的过程，属于生成模型，可以应用于语音识别，词性标注，音字转换，概率问题发等自然语言处理的各个应用。

本文的总结主要根据李航《统计机器学习方法》，隐马尔科夫模型内容，但对概率计算问题，做了更清楚的补充，从而使证明更易懂。而且对**维特比算法做了更清晰的数学说明**（独立推导和说明，个人认为比原书内容更清晰，欢迎批评指正）。

基本概念

隐马尔科夫模型的定义

隐马尔可夫模型 ：隐马尔科夫模型是关于时序的概率模型，描述一个由隐藏的马尔可夫链随机生成不可观察的状态的随机序列，再由各个状态生成一个观察而产生观测随机序列的过程。

状态序列（state sequence）:隐马尔科夫链随机生成的状态的序列

观测序列(observation sequence) :每个状态生成一个观察，而由此产生的观察的随机序列

序列每个位置看作一个时刻。

隐马尔科夫模型由初始概率分布，状态转移概率分布以及观测概率分布确定，形式如下：

状态集合 :$Q=(q_1,q_2...q_N)$, **观察集合** ：$V=(v_1,v_2...v_M)$

N ：可能的状态数 M 可能的观察数

状态序列 ：$I=(i_1,i_2....i_T)$ **观察序列** ：$O=(o_1,o_2...o_T)$

隐马尔科夫模型的两个基本假设 ：

1、**齐次马尔科夫模型假设** （有限历时假设）：隐藏的马尔科夫链在任意时刻t的状态只依赖于前一时刻的状态，与其他时刻的状态以及观测无关，也与t无关。

$p(i_t|i_{t-1},o_{t-1},...i_1,o_1)=p(i_t|i_{t-1})$

2、**观察独立假设** ：任意时刻的观测只依赖于该时刻的马儿可分链的状态，与其他观察及状态无关。

$p(o_t|i_T,o_T,...i_1,o_1)=p(o_t|i_t)$

3、**时间不变性假设** ：条件依赖不随时间改变而改变

根据上述假设，可以定义：

状态转移概率矩阵 ： $A=[a_{ij}]_{N*N}$，其中$a_{ij}=p(i_{t+1}=q_j|i_t=q_i)$

t时刻处于状态q_i的条件下在t+1转移到状态q_j的概率

在马尔科夫模型中，状态转移概率a_{ij}必须满足以下条件

$0 \leq a_{ij}$

$\sum_{j=1}^{N} a_{ij} = 1$

B 是观测概率矩阵 ：$B = [b_j(k)]_{[N*M]}$,其中$b_j(k)$是在t时刻处于状态q_j的条件下生成观察v_k的概率。

π：是初始状态概率向量：π_i t=1时刻处于状态q_i的概率

隐马尔科夫模型由初始状态向量π，状态转移概率矩阵A和观察概率矩阵B决定，π和A决定状态序列，B决定观察序列，因此隐马尔科夫模型可以用三元符号表示，即$\lambda = (A, B, \pi)$

隐马尔科夫可以用于标注，这时状态对应着标记，标注问题是给定观察序列预测器对应的标记序列。

隐马尔科夫模型的3个基本问题：

1、**概率计算问题** ：给定模型$\lambda = (A, B, \pi)$和观察序列$O = (o_1, o_2 \dots o_T)$，计算在模型λ下观察序列O出现的概率：$P(o|\lambda)$.

2、**学习问题** ：已知观察序列$O = (o_1, o_2 \dots o_T)$,根据最大似然估计，估计模型参数，使得模型在λ下观察序列概率$p(O|\lambda)$最大。

3、**预测问题** ：也称解码问题，已知模型$\lambda = (A, B, \pi)$和观察序列$O = (o_1, o_2 \dots o_T)$,求给定观察序列条件概率$P(I|O, \lambda)$最大的状态序列$I = (i_1, i_2 \dots . i_T)$。

概率计算问题

1、直接计算法（暴力解法）

直接根据公式进行计算：

将公式展开，可以得到：

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

根据状态转移概率矩阵，状态序列$I = (i_1, i_2 \dots i_T)$的概率是

$P(I|\lambda) = \pi_{i_1} a_{i_1 i_2} a_{i_2 i_3} \dots a_{i_{T-1} i_T}$

固定状态序列$I = (i_1, i_2 \dots i_T)$，观察序列$O = (o_1, o_2 \dots o_T)$的概率为：

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

将上边两式放到一起，最终得到：

$P(O|\lambda) = \sum_I P(I|\lambda)P(O|I, \lambda) = \sum_{i_1, i_2 \dots i_T} \pi_{i_1} b_{i_1}(o_1) a_{i_1 i_2} b_{i_2}(o_2) a_{i_2 i_3} \dots a_{i_{T-1} i_T} b_{i_T}(o_T)$

利用上述公式计算量很大($O(N^T)$),实践上不可行。

下面介绍计算观察序列的有效算法：前向-后向算法(forward-backward algorithm)

前向算法

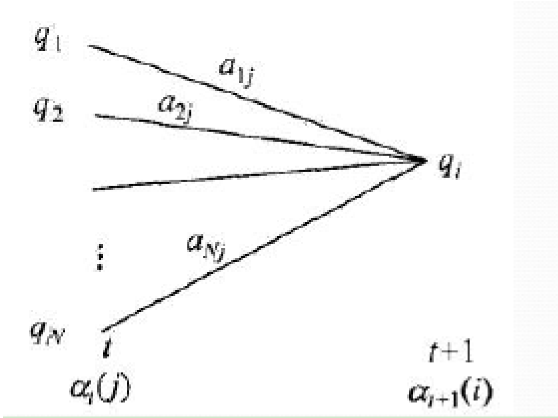
前向概率 :给定马尔科夫模型λ,定义到时刻t部分观察序列为$o_1, o_2 \dots o_t$且状态为q_i的概率。

$\alpha_t(i) = P(o_1, o_2, \dots o_t, i_t = q_i|\lambda)$

前向算法采用递推方式求解：

初始值：$\alpha_1(i) = \pi_i b_i(o_1)$ 第一时刻观察到o_1且第一时刻状态为q_i的概率

递推公式 ：对t=1,2...,T-1, $\alpha_{t+1}(i)=[\sum_{j=1}^{N}\alpha_t(j)a_{ji}]b_i(o_{t+1})$



递推式比较复杂，下面具体说明：

根据定义$\alpha_t(j)$表示到t时刻，观察到$o_1,o_2...o_t$且t时刻处于q_j的概率

$\alpha_t(j)a_{ji}$ 表示t时刻，观察到$o_1,o_2...o_t$且t时刻处于q_j，t+1时刻状态处于q_i的概率

对t时刻所有可能的N个状态q_j求和：$[\sum_{j=1}^{N}\alpha_t(j)a_{ji}]$ 得到t时刻观察到$o_1,o_2...o_t$，且t+1时刻状态处于q_i的概率（如上图所示）

用上式乘上$b_i(o_{t+1})$表示t时刻观察到$o_1,o_2...o_t$，且t+1时刻状态处于q_i的概率，且t+1时刻观察到o_{t+1}时刻概率。

终止：$P(O|\lambda)=\sum_{i=1}^{N}\alpha_T(i)$

将所有t+1时刻所有可能的状态相加得到：T时刻观察到$o_1,o_2...o_T$的概率，即$P(O|\lambda)=\sum_{i=1}^{N}\alpha_T(i)$

计算复杂度分析 ：

前向算法高效的关键是局部计算前向概率，然后利用路径结构将前向概率递推到全局。

递推公式中，根据t时刻N个$\alpha_t(j)$，计算t+1时刻$\alpha_{t+1}(i)$主要计算量为$[\sum_{j=1}^{N}\alpha_t(j)a_{ji}]$,计算复杂度为N，为了计算$t+2$时刻前向概率，需要t+1时刻所有可能的N个状态，因此每个时间点总的时间复杂度为$O(N^2)$，所以该算法总的时间复杂度为$O(TN^2)$。

后向算法

后向概率 :给定马尔科夫模型λ，定义t时刻状态为q_i的条件下，从t+1时刻到T部分观察序列为$o_{t+1},o_{t+1},...o_T$的概率。

初始值 ：$\beta_T(i)=1$,T时刻之后没有观察序列，定义为1

递推公式 ：对t =T-1，T-2,...,1 $\beta_t(i)=\sum_{j=1}^{N}a_{ij}b_j(o_{t+1})\beta_{t+1}(j)$

（统计学习方法中给的说明比较简略，这里采用公式证明说明）

根据定义：

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

$=\sum_{j=1}^{N}p(o_{t+1},o_{t+2},...o_T,i_{t+1}=q_j|i_t=q_i,\lambda)$

$=\sum_{j=1}^{N}p(o_{t+1},o_{t+2},...o_T|i_{t+1}=q_j,i_t=q_i,\lambda).p(i_{t+1}=q_j|i_t=q_i,\lambda)$

根据其次假设,可将上式简化为：

$= \sum_{j=1}^{N} p(o_{t+1}, o_{t+2}, \ldots o_T | i_{t+1} = q_j, \lambda) a_{ij}$

根据概率乘法公式：

$= \sum_{j=1}^{N} p(o_{t+1} | o_{t+2} \ldots . o_T, i_{t+1} = q_j, \lambda) p(o_{t+2}, o_{t+3}, \ldots o_T | i_{t+1} = q_j, \lambda) a_{ij}$

根据其次假设o_{t+1}与 $o_{t+2} \ldots o_T$无关，可以得到：

$= \sum_{j=1}^{N} p(o_{t+1} | i_{t+1} = q_j, \lambda) p(o_{t+2}, o_{t+3}, \ldots o_T | i_{t+1} = q_j, \lambda) a_{ij}$

$= \sum_{j=1}^{N} b_j(o_{t+1}) p(o_{t+2}, o_{t+3}, \ldots o_T | i_{t+1} = q_j, \lambda) a_{ij}$

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

从而得证。

终止式 ：$P(O|\lambda)$ 可以理解为t=1时刻的后向概率之和，从而有：

$P(O|\lambda) = \sum_{i=1}^{N} \pi_i b_i(o_1) \beta_1(i)$

利用前向概率和后向概率的定义可以一起观察序列概率$p(O|\lambda)$：

$P(O|\lambda) = P(o_1, o_2 \ldots o_T | \lambda) = \sum_{i=1}^{N} \sum_{j=1}^{N} P(o_1, o_2 \ldots o_T, i_t = q_i, i_{t+1} = q_j | \lambda)$

$P(o_1, o_2 \ldots o_T, i_t = q_i, i_{t+1} = q_j | \lambda)$

$= P(o_{t+1}, o_{t+2}, .. o_T, i_{t+1} = q_j | o_1, o_2, \ldots o_t, i_t = q_i, \lambda) P(o_1, o_2 \ldots o_t, i_t = q_i | \lambda)$

$= P(o_{t+1}, o_{t+2}, .. o_T, i_{t+1} = q_j | i_t = q_i, \lambda) \alpha_t(i)$
$= P(o_{t+2}, .. o_T | o_{t+1}, i_t = q_i, i_{t+1} = q_j, \lambda) P(i_{t+1} = q_j, o_{t+1} | i_t = q_i) \alpha_t(i)$

$= P(o_{t+2}, .. o_T | i_{t+1} = q_j, \lambda) P(o_{t+1} | i_{t+1} = q_j, i_t = q_i) P(i_{t+1} = q_j | i_t = q_i) \alpha_t(i)$

$= P(o_{t+2}, .. o_T | i_{t+1} = q_j, \lambda) P(o_{t+1} | i_{t+1} = q_j) P(i_{t+1} = q_j | i_t = q_i) \alpha_t(i)$

$= \beta_{t+1}(j) b_j(o_{t+1}) a_{ij} \alpha_t(i)$

从而可以得到：$P(O|\lambda) = \sum_{i=1}^{N} \sum_{j=1}^{N} \beta_{t+1}(j) b_j(o_{t+1}) a_{ij} \alpha_t(i)$

一些概率与期望值的计算

1、给定模型$\boldsymbol{\lambda}$与观察O，在t时刻处于状态q_i的概率，记为：

$\gamma_t(i) = P(i_t = q_i | O, \lambda) = \frac{P(i_t = q_i, O|\lambda)}{P(O|\lambda)}$

$P(i_t = q_i, O|\lambda) = P(o_1, o_2 \ldots o_T, i_t = q_i | \lambda)$

$= P(o_{t+1}, o_{t+2}, \ldots o_T | o_1, o_2, \ldots o_t, i_t = q_i, \lambda) P(o_1, o_2 \ldots o_t, i_t = q_i | \lambda)$

$= P(o_{t+1}, o_{t+2}, \ldots o_T | i_t = q_i, \lambda) P(o_1, o_2 \ldots o_t, i_t = q_i | \lambda)$

=$\alpha_t(i) \beta_t(i)$

$\gamma_t(i) = \frac{\alpha_t(i) \beta_t(i)}{\sum_{j=1}^{N} \alpha_t(i) \beta_t(i)}$

2、给定模型$\boldsymbol{\lambda}$与观察O，在t时刻处于状态q_i,且t+1时刻处于q_j的概率：

$\xi(i,j) = P(i_t = q_i, i_{t+1} = q_j|O,\lambda) = \frac{P(i_t=q_i,i_{t+1}=q_jO|\lambda)}{\sum_{i=1}^N \sum_{j=1}^N P(i_t=q_i,i_{t+1}=q_jO|\lambda)}$

$P(i_t = q_i, i_{t+1} = q_jO|\lambda) = \beta_{t+1}(j)b_j(o_{t+1})a_{ij}\alpha_t(i)$

从而有$\xi(i,j) = \frac{\beta_{t+1}(j)b_j(o_{t+1})a_{ij}\alpha_t(i)}{\sum_{i=1}^N \sum_{j=1}^N \beta_{t+1}(j)b_j(o_{t+1})a_{ij}\alpha_t(i)}$

学习算法

隐马尔科夫模型的学习，根据训练数据是包括观察序列和对应的状态序列还是只有观察序列，可分为监督学习和非监督学习，本节首先介绍监督学习，然后介绍非监督学习算法（Baum-Welch算法）

监督学习算法

假设已给训练数据包含S个长度相同的观察序列和对应的状态序列（比如词性标注中对应的词和词性）{$(O_1,I_1),(O_2,I_2)...(O_S,I_S)$},那么可以利用极大似然估计算法来估计隐马尔科夫模型的参数。具体估计如下：

1、转移概率a_{ij}的估计，样本中t时刻处于q_i条件下，t+1时刻处于q_j所占百分比

样本中t时刻处于q_i，t+1时刻处于q_j的频数为A_{ij},

那么$\hat{a}_{ij} = \frac{A_{ij}}{\sum_{j=1}^N A_{ij}}$

2、观察概率$b_j(k)$的估计,状态为q_j时，观测状态为v_k所占百分比

样本中状态为q_j且观察状态为v_k的频数为B_{jk},

那么$\hat{b}_j(k) = \frac{B_{jk}}{\sum_{k=1}^M B_{jk}}$

3、初始状态概率π_i的估计$\hat{\pi}$为S个样本中初始状态为q_i的概率。

非监督学习算法：Baum-Welch算法

假设给定训练数据只包含S个长度为T的观察序列{$O_1,O_2...O_S$},而没有对应的状态。将观察序列数据看作观察数据O，状态序列看作不客管处的因数据I，那么隐马尔科夫模型事实上是一个含有隐变量的概率模型。

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

参数学习可以通过EM算法实现。EM算法的推导比较复杂，这里就不再做说明，具体可以参考《统计机器学习方法》隐马尔科夫模型Baum-Welch算法部分，这里只给出参数估计形式。

Baum-welch模型参数估计形式

$a_{ij} = \frac{\sum_{t=1}^{T-1} \xi(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$

$b_j(k) = \frac{\sum_{t=1,o_t=v_k} \gamma_t(i)}{\sum_{t=1} \gamma_t(i)}$

$\pi_i = \gamma_1(i)$

预测算法

预测问题是在给定观察序列O和模型λ的条件下，找到概率最大的状态序列I^*的问题

数学表述为$I^* = argmax_I P(I|O,\lambda)$

近似算法

近似算法是为判别模型，采取贪心策略在给定模型和观察序列O条件下，每个时刻最优可能出现的状态i_t^*，

$\gamma_t(i)$为给定模型λ与观察O，在t时刻处于状态q_i的概率

那么：$i_t^* = argmax_{1\le i\le N}[\gamma_t(i)], t=1,2...T$

从而得到一个状态序列$I^* = (i_1^*, i_2^* \dots i_T^*)$

优点：计算简单

缺点： 贪心策略无法保证最优解

维特比算法

给定模型λ和观测序列O，求取最大的状态序列，数学上可以表示为：$I^* = argmax_I P(I|O,\lambda)$,但上式不容易求解。维特比算法采用**概率生成模型**，根据贝叶斯公式，$P(I|O,\lambda) = \frac{P(I,O|\lambda)}{P(O|\lambda)}$，由于状态序列I与观察序列O无关，因此$I^* = argmax_I P(I|O,\lambda) = argmax_I P(I,O|\lambda)$:

即在给定模型参数λ下，求解最大的联合概率对应的序列。

首先我们来分析t+1时刻与t时刻最佳序列之间的关系：

假设从初始到t+1时刻状态为q_j最佳序列中，t时刻状态分别为q_i,之前序列使用$i_1*, \dots i_{t-1}^*$

$p(i_{t+1}=q_j, o_{t+1}, i_t=q_i, o_t, \dots\dots i_1^*, o_1|\lambda)$

$= p(i_{t+1}=q_j, o_{t+1}|i_t=q_i, o_t, \dots\dots i_1^*, o_1, \lambda)p(i_t=q_i, o_t \dots i_1^*, o_1|\lambda)$

$= p(i_{t+1}=q_j, o_{t+1}|i_t=q_i, \lambda)p(i_t=q_i, o_t \dots i_1^*, o_1|\lambda)$

$= p(o_{t+1}|i_{t+1}=q_j, i_t=q_i, \lambda)p(i_{t+1}=q_j|i_t=q_i, \lambda)p(i_t=q_i, o_t \dots i_1^*, o_1|\lambda)$

$= p(o_{t+1}|i_{t+1}=q_j, \lambda)p(i_{t+1}=q_j|i_t=q_i, \lambda)p(i_t=q_i, o_t \dots i_1^*, o_1|\lambda)$

$= b_j(o_{t+1})a_{ij}p(i_t=q_i, o_t \dots i_1^*, o_1|\lambda)$

分析上式，如果i,j确定，那么$b_j(o_{t+1})$与a_{ij}也确定，

因此，如果初始点到t+1时刻状态为q_j的最佳序列中前t个序列为$i_1^*, i_2^*, \dots i_{t-1}^*, i_t = q_i$。那么从初始点到t时刻状态为$q_i$的最大序列为$i_1^*, i_2^*, \dots i_{t-1}^*, i_t = q_i$。即如果一个序列是最佳序列，那么它的子序列也是一个最佳序列而后者只，是前者小一些规模的相同问题。

为了求解t+1时刻最佳序列，我们需要保证$b_j(o_{t+1})a_{ij}p(i_t=q_i, o_t \dots i_1^*, o_1|\lambda)$最大，

因此如果我们t时刻，状态为q_i的最大概率为$\delta_t(i)$,

那么t+1时刻状态为q_j的最大概率$\delta_{t+1}(j) = max_{1\le i\le N}\delta_t(i)a_{ij}b_j(o_{t+1})$.

为了求得t+1时刻状态为q_j的最大概率，我们只需要t时刻每一个可能的状态的最大概率（a_{ij}与$b_j(o_{t+1})$只与模型有关，与时间无关），从而将问题的规模缩小，我们可以采用相同的策略将问题进一步缩小。从而可以采用递推的方式进行求解。

根据$\delta_{t+1}(j) = max_{1\le i\le N}\delta_t(i)a_{ij}b_j(o_{t+1})$ ，我们也可以确定t+1时刻状态为q_j的最佳序列对应的t时刻的状态。

定义$\phi_t(j) = argmax_{1\leq i\leq N}\delta_t(i)a_{ij}b_j(o_{t+1}) = argmax_{1\leq i\leq N}\delta_t(i)a_{ij}$为t+1时刻状态为$q_j$的最佳序列对应的t时刻的状态。

初始值：t=1时刻，状态为q_i,观测序列为o_1的最大概率为$\delta_1(i) = \pi_i b_i(o_1)$，$\phi_0 = 0$

递推公式：

$\delta_{t+1}(j) = max_{1\leq i\leq N}\delta_t(i)a_{ij}b_j(o_{t+1})$

$\phi_t(j) = argmax_{1\leq i\leq N}\delta_t(i)a_{ij}b_j(o_{t+1}) = argmax_{1\leq i\leq N}\delta_t(i)a_{ij}$

终止式：$p^* = max_{1\leq j\leq N}\delta_T(i)$ $i_T^* = argmax_{1\leq j\leq N}\delta_T(i)$