HMM简介

隐式马尔科夫模型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生成观测序列的过程，属于生成模型。

HMM的基本概念

HMM的定义

定义(HMM)：隐式马尔科夫模型是关于时序的概率模型，描述由一个隐藏的马尔科夫链随机生成的不可观测的状态随机序列，再由各个状态生成一个观测而产生观测随机序列的过程。HMM随机生成的状态序列为状态序列，每个状态生成一个观测，产生观测序列。序列的每一个位置可以看作一个时刻。

HMM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。形式定义如下：

设Q是所有可能的状态的集合，V是所有可能的观测的集合：

$$Q=\{q_1,\cdots,q_N\},V=\{v_1,\cdots,v_M\}$$

I是长度为T的状态序列，O是对应的观测序列：

$$I=\{i_1,\cdots,i_T\},O=\{o_1,\cdots,o_T\}$$

A是状态转移概率矩阵：

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

其中$a_{ij}=P(i_{t+1}=q_j|i_t=q_i)$是在时刻$t$处于状态$q_i$的条件在下一时刻$t+1$转移到状态$q_j$的概率。

B是观测概率矩阵：

$$B=[b_j(k)]_{N\times M}$$

其中$b_j(k)=P(o_t=v_k|i_t=q_j)$是在时刻$t$处于状态$q_j$的条件下生成观测$v_k$的概率。

π是初始状态概率向量：

$$\pi=(\pi_i)$$

其中$\pi_i=P(i_1=q_i)$是时刻$t=1$处于状态q_i的概率。

HMM模型由初始状态概率向量π、状态转移概率矩阵A和观测概率矩阵B决定，因此，HMM可由三元符号表示：

$$\lambda=(A,B,\pi)$$

从定义可知，HMM作了两个基本假设：

1. 齐次马尔科夫性假设：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，与其他时刻无关

$$P(i_t|i_{t-1},o_{t-1},\cdots,i_1,o_1)=P(i_t|i_{t-1}),t=1,\cdots,T$$

2. 观测独立性假设：假设在任意时刻的观测只依赖于该时刻的状态

$$P(o_t|i_T,o_T,\cdots,i_1,o_1)=P(o_t|i_t)$$

HMM序列生成过程

算法(观测序列的生成)：

1. 按照初始状态分布π产生状态i_1
2. 令$t=1$
3. 按照状态i_t的观测概率分布$b_{i_t}(k)$生成o_t
4. 按照状态i_t的状态转移概率分布产生状态i_{t+1}
5. 令$t=t+1$，继续2,3步直到$t=T$

HMM的三个基本问题

1. 概率计算问题。给定模型λ和观测序列O，计算在模型λ下观测序列O出现的概率$P(O|\lambda)$。
2. 学习问题。已知观测序列O，估计模型λ参数，使得$P(O|\lambda)$最大。
3. 预测问题。已知模型λ和观测序列O，求对给定观测序列条件概率$P(I|O)$最大的状态序列I。

概率计算算法

现在来介绍观测序列概率$P(O|\lambda)$的前向与后向算法。首先来介绍概念上可行但计算上不可行的直接计算法。

直接计算法

给定模型λ和观测序列O，计算$P(O|\lambda)$。最直接的方法就是按照概率公式直接计算，列举所有可能的长度为T的状态序列I，求各个状态序列I与观测序列O的联合概率$P(O,I|\lambda)$，然后求和，得到$P(O|\lambda)$。

状态序列$I=(i_1,\cdots,i_T)$的概率是

$$P(I|\lambda)=\pi_{i_1}a_{i_1i_2}\cdots a_{i_{t-1}i_T}$$

对固定的I，观测序列$O=(o_1,\cdots,o_T)$的概率为

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

所以O,I的联合概率为

$$P(O,I|\lambda)=P(O|I,\lambda)P(I|\lambda)=\pi_{i_1}b_{i_1}(o_1)a_{i_1i_2}\cdots a_{i_{T-1}i_T}b_{i_T}(o_T)$$

然后，对所有的I求和，就得到了$P(O|\lambda)$：

$$P(O|\lambda)=\sum_I P(O|I,\lambda)P(I|\lambda)=\sum_{i_1,\cdots,i_T}\pi_{i_1}b_{i_1}(o_1)a_{i_1i_2}\cdots a_{i_{T-1}i_T}b_{i_T}(o_T)$$

但是上述公式的复杂度是$O(TN^T)$的，不可行。

前向算法

定义(前向概率)： 给定模型λ，定义到时刻t部分观测序列为$o_1,o_2,\cdots,o_t$且状态为q_i的概率为前向概率，记作

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

可以通过递推地求前向概率$\alpha_t(i)$和观测序列概率$P(O|\lambda)$。

算法(前向算法)：

1. 初值

$$\alpha_1(i)=\pi_ib_i(o_1),i=1,2,\cdots,N$$

2. 递推，对$t=1,2,\cdots,T-1$

$$\alpha_{t+1}(i)=\left[\sum_{j=1}^{N}\alpha_t(j)a_{ji}\right]b_i(o_{t+1}),i=1,2,\cdots,N$$

3. 终止

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

利用前向算法的计算量是$O(N^2T)$。

后向算法

定义(后向概率)：给定λ，定义在时刻t状态为q_i的条件下，从$t+1$到T的部分观测序列为$o_{t+1},\cdots,o_T$的概率为后向概率，记作：

$$\beta_t(i)=P(o_{t+1},\cdots,o_T|i_t=q_i,\lambda)$$

算法(后向算法)：

1. 初值

$$\beta_T(i)=1,i=1,\cdots,N$$

2. 对$t=T-1,\cdots,1$

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

3. 终止

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

利用前向概率和后向概率可以将观测序列概率$P(O|\lambda)$统一写成

$$P(O|\lambda)=\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_t(i)a_{ij}b_j(o_{t+1})\beta_{t+1}(j),t=1,\cdots,T-1$$

一些概率与期望值的计算

1. 给定模型λ和观测O，在时刻t处于状态q_i的概率，记作

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

可以通过前向后向概率计算。事实上，

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

且知：

$$\alpha_t(i)\beta_t(i)=P(i_t=q_i,O|\lambda)$$

于是得到

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

2. 给定模型λ和观测O，在时刻t处于状态q_i且在时刻$t+1$处于状态q_j的概率，记作

$$\xi_t(i,j) = P(i_t = q_i, i_{t+1} = q_j | O, \lambda)$$

   可以通过向后向概率计算：

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

   而

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

   所以

$$\xi_t(i,j) = \frac{\alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{\sum\limits_{i=1}^{N}\sum\limits_{j=1}^{N} \alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}$$

3. 将$\gamma_t(i)$和$\xi_t(i,j)$对各个时刻t求和，可以得到一些有用的期望值：
   - 在观测O下状态i出现的期望值

$$\sum_{t=1}^{T} \gamma_t(i)$$

   - 在观测O下由状态i转移的期望值

$$\sum_{t=1}^{T-1} \gamma_t(i)$$

   - 在观测O下由状态i转移到状态j的期望值

$$\sum_{t=1}^{T-1} \xi_t(i,j)$$

学习算法

监督学习算法

假设已给训练数据包含S个长度相同的观测序列和对应的状态序列，那么可以利用极大似然估计。

1. 转移概率a_{ij}的概率

   设样本中时刻t处于状态i到时刻$t+1$位于状态j的频数为A_{ij}，那么

$$\hat{a}_{ij} = \frac{A_{ij}}{\sum_{j=1}^{N} A_{ij}}, i = 1, \cdots, N; j = 1, \cdots, N$$

2. 观测概率$b_j(k)$的估计

   设样本中状态为j且观测为k的频数为B_{jk}，则状态为j观测为k的频率$b_j(k)$为

$$\hat{b}_{jk} = \frac{B_{jk}}{\sum_{k=1}^{M} B_{jk}}$$

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

Baum-Welch算法

假设只给定了观测序列$\{O_1,\cdots,O_S\}$，而没有给定状态序列，目标是学习λ。我们将观测序列看作观测数据O，而状态序列为不可观测的隐数据I，那么HMM事实上是一个含有隐变量的概率模型

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

1. 确定完全数据的对数似然

$$\log P(O,I|\lambda)$$

2. E步：求$Q(\lambda,\bar{\lambda})$

$$Q(\lambda,\bar{\lambda})=\sum_I \log P(O,I|\lambda)P(O,I|\bar{\lambda})$$

   其中

$$P(O,I|\lambda)=\pi_{i_1}b_{i_1}(o_1)a_{i_1i_2}b_{i_2}(o_2)\cdots a_{i_{T-1}i_T}b_{i_T}(o_T)$$

   于是$Q(\lambda,\bar{\lambda})$可以写作

$$Q(\lambda,\bar{\lambda})=\sum_I \log \pi_{i_1}P(O,I|\bar{\lambda})$$
$$+\sum_I(\sum_{t=1}^{T-1}\log a_{i_ti_{t+1}})P(O,I|\bar{\lambda})+\sum_I(\sum_{t=1}^{T}\log b_{i_t}(o_t))P(O,I|\bar{\lambda})$$

3. M步：极大化$Q(\lambda,\bar{\lambda})$求参数A,B,π。可以用拉格朗日乘子得到：

$$\pi_i=\frac{P(O,i_1=i|\bar{\lambda})}{P(O|\bar{\lambda})}$$
$$a_{ij}=\frac{\sum_{t=1}^T P(O,i_t=i,i_{t+1}=j|\bar{\lambda})}{\sum_{t=1}^T P(O,i_t=i|\bar{\lambda})}$$
$$b_j(k)=\frac{\sum_{t=1}^T P(O,i_t=j|\bar{\lambda})I(o_t=v_k)}{\sum_{t=1}^T P(O,i_t=j|\bar{\lambda})}$$

4. 最后将上述公式用$\gamma_t(i),\xi_t(i,j)$表示，得到

$$a_{ij}=\frac{\sum\limits_{t=1}^{T-1}\xi_t(i,j)}{\sum\limits_{t=1}^{T-1}\gamma_t(i)}$$
$$b_j(k)=\frac{\sum\limits_{t=1,o_t=v_k}^{T}\gamma_t(j)}{\sum\limits_{t=1}^{T}\gamma_t(j)}$$
$$\pi_i=\gamma_1(i)$$

预测算法

近似算法

近似算法的想法是：在每个时刻t选择在该时刻最有可能出现的状态i_t^*，从而得到一个状态序列$I^*=(i_1^*,i_2^*,\cdots,i_T^*)$作为预测的结果。

给定HMM的λ和O，在时刻t处于状态q_i的概率$\gamma_t(i)$是

$$\gamma_t(i)=\frac{\alpha_t(i)\beta_t(i)}{P(O|\lambda)}=\frac{\alpha_t(i)\beta_t(i)}{\sum\limits_{j=1}^{N}\alpha_t(j)\beta_t(j)}$$

在每一时刻t最有可能的状态i_t^*是

$$i_t^*=\arg\max_{1\le i\le N}[\gamma_t(i)],t=1,\cdots,T$$

近似算法的优点是计算简单，缺点是不能保证预测整体的最优性。

维特比算法

维特比算法使用DP求概率最大路径。最优路径具有这样的特性：如果最优路径在时刻t通过结点i_t^*，那么这一路径从结点i_t^*到i_T^*的部分路径，对于从i_t^*到i_T^*的所有可能的部分路径来说，必须是最优的。若否，则存在另一条更好的部分路径，如果把它和从i_1^*到i_t^*的最优路径连起来，则形成了更好的路径。故我们只需要从$t=1$开始，递推地计算在时刻t状态为i的各条部分路径的最大概率即可。然后倒推回来得到最优路径结点。

首先引入两个变量δ,ψ，定义在时刻t状态为i的所有单个路径$(i_1,\cdots,i_t)$中概率最大值为

$$\delta_t(i)=\max_{i_1,i_2,\cdots,i_{t-1}}P(i_t=i,i_{t-1},\cdots,i_1,o_t,\cdots,o_1|\lambda)\,,\,i=1,2,\cdots,N$$

由此得到δ的递推公式

$$\begin{aligned}\delta_{t+1}(i)&=\max_{i_1,\cdots,i_t}P(i_{t+1}=i,i_t,\cdots,i_1,o_t,\cdots,o_1|\lambda)\\&=\max_{1\le j\le N}[\delta_t(j)a_{ji}]b_i(o_{t+1}),i=1,\cdots,N;t=1,\cdots,T-1\end{aligned}$$

定义在时刻t状态为i的所有单个路径$(i_1,\cdots,i_{t-1},i)$中概率最大的路径的第$t-1$个结点为

$$\psi_t(i)=\arg\max_{1\le j\le N}[\delta_{t-1}(j)a_{ji}],i=1,\cdots,N$$

维特比算法：

1. 初始化

$$\delta_1(i)=\pi_ib_i(o_1),i=1,\cdots,N$$
$$\psi_1(i)=0,i=1,\cdots,N$$

2. 递推，对$t=2,\cdots,T$

$$\delta_t(i)=\max_{1\le j\le N}[\delta_{t-1}(j)a_{ji}]b_i(o_t),i=1,\cdots,N$$
$$\psi_t(i)=\arg\max_{1\le j\le N}[\delta_{t-1}(j)a_{ji}],i=1,\cdots,N$$

3. 终止

$$P^*=\max_{1\le i\le N}\delta_T(i)$$
$$i_T^*=\arg\max_{1\le i\le N}[\delta_T(i)]$$

4. 最优路径回溯

$$i_t^*=\psi_{t+1(i_{t+1}^*)},t=T-1,\cdots,1$$