隐马尔可夫模型-入门篇

摘要：隐马尔可夫模型（Hidden Markov Model, HMM）是序列数据处理和统计学习的一种重要概率模型，已被成功应用于许多工程任务中。本文首先介绍隐马尔可夫的基本概念，然后分别叙述隐马尔科夫模型的三个基本问题及解决方法，最后讲述 HMM 在词性标注中的实际应用。

一 背景介绍

隐马尔可夫模型(Hidden Markov Model, HMM)作为一种统计分析模型， 创立于 20 世纪 70 年代， 80 年代得到了传播和发展并成功应用于声学信号的建模中， 到目前为止， 它仍然被认为是实现快速精确语音识别系统最成功的方法。作为信号处理的一个重要方向，HMM 广泛应用于图像处理，模式识别，语音人工合成和生物信号处理等领域的研究中，并取得了诸多重要的成果。近年来，很多研究者把 HMM 应用于计算机视觉、 金融市场的波动性分析和经济预算等新兴领域中。

HMM 是一种用参数表示的用于描述随机过程统计特性的概率模型，是一个双重随机过程, 由两个部分组成:马尔可夫链和一般随机过程。 其中马尔可夫链用来描述状态的转移，用转移概率描述。一般随机过程用来描述状态与观察序列间的关系，用观察值概率描述。下面讲对 HMM 展开详细介绍。

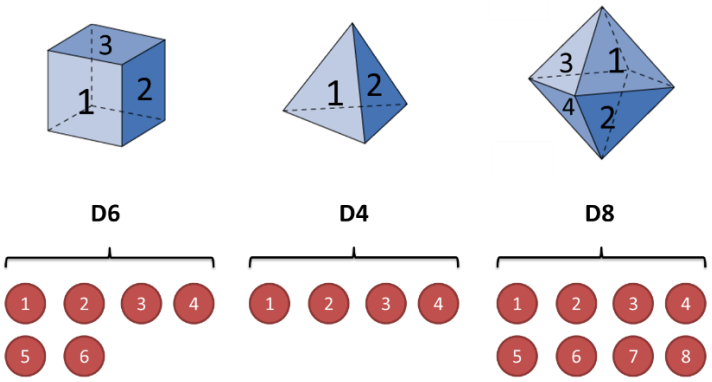
二 HMM 基本原理

2.1 一个故事

首先讲一个生动但不一定有趣的故事来说明一下隐马尔可夫过程。如果你是初次接触 HMM，请耐心把这个故事看完。

假设我手里有三个不同的骰子。第一个骰子是我们平常见的骰子（称这个骰子为 D6），6 个面，每个面（1，2，3，4，5，6）出现的概率是 1/6。第二个骰子是个四面体（称这个骰子为 D4），每个面（1，2，3，4）出现的概率是 1/4。第三个骰子有八个面（称这个骰子为 D8），每个面（1，2，3，4，5，6，7，8）出现的概率是 1/8。

三种骰子和掷骰子可能产生的结果



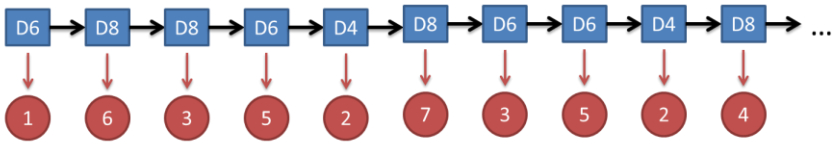
假设我们开始掷骰子，我们先从三个骰子里挑一个，挑到每一个骰子的概率都是 1/3。这就是初始状态概率（the initial state probabilities）。然后我们掷骰子，得到一个数字，是 1，2，3，4，5，6，7，8 中的一个。不停的重复上述过程，我们会得到一串数字，每个数字都是 1，2，3，4，5，6，7，8 中的一个。例如我们可能得到这么一串数字（掷骰子 10 次）：1 6 3 5 2 7 3 5 2 4。这串数字叫做可见状态链。

但是在隐马尔可夫模型中，我们不仅仅有这么一串可见状态链，还有一串隐含状态链。在这个例子里，这串隐含状态链就是你用的骰子的序列。比如，隐含状态链有可能是：D6 D8 D8 D6 D4 D8 D6 D6 D4 D8。

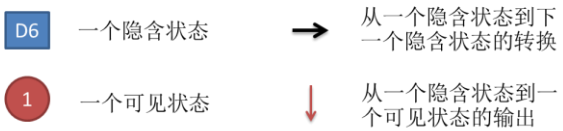
一般来说，HMM 中说到的马尔可夫链其实是指隐含状态链，因为隐含状态（骰子）之间存在转换概率（transition probability）。在我们这个例子里，D6 的下一个状态是 D4，D6，D8 的概率都是 1/3。D4，D8 的下一个状态是 D4，D6，D8 的转换概率也都一样是 1/3。这样设定是为了最开始容易说清楚，但是我们其实是可以随意设定转换概率的。比如，我们可以这样定义，D6 后面不能接 D4，D6 后面是 D6 的概率是 0.9，是 D8 的概率是 0.1。这样就是一个新的 HMM。

同样的，尽管可见状态之间没有转换概率，但是隐含状态和可见状态之间有一个概率叫做输出概率（emission probability）。就我们的例子来说，六面骰（D6）产生 1 的输出概率是 1/6。产生 2，3，4，5，6 的概率也都是 1/6。我们同样可以对输出概率进行其他定义。比如，我有一个被赌场动过手脚的六面骰子，掷出来是 1 的概率更大，是 1/2，掷出来是 2，3，4，5，6 的概率是 1/10。

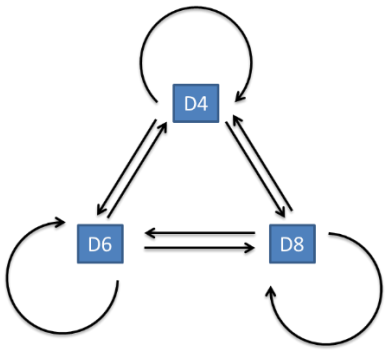
隐马尔可夫模型示意图



图例说明：



隐含状态转换关系示意图



2.2 HMM 定义

从上面的例子中，我们已经对隐马尔可夫过程有了一个大致的了解，现在我们给出 HMM 的形式化定义，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。

$$Q=\{q_1,q_2,...,q_N\},\quad V=\{v_1,v_2,...,v_M\}$$

其中，N 是所有可能的状态数，M 是可能的观测数。

I 是长度为 T 的状态序列，O 是对应的观测序列。

$$I=(i_1,i_2,...,i_T),\quad O=(o_1,o_2,...,o_T)$$

（1）A 是状态转移概率矩阵：

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

其中，

$$a_{ij}=P(i_{t+1}=q_j\,|\,i_t=q_i),\quad i=1,2,...,N;\ j=1,2,...,N$$

是在时刻 t 处于状态 q_i 的条件下在时刻 t+1 转移到状态 q_j 的概率。

（2）B 是观测概率矩阵：

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

其中，

$$b_j(k)=P(o_t=v_k\,|\,i_t=q_j),\quad k=1,2,...,M;\ j=1,2,...,N$$

是在时刻 t 处于状态 q_j 的条件下生成观测 v_k 的概率。

（3）π 是初始状态概率向量：

$$\pi=(\pi_i)$$

其中，

$$\pi_i=P(i_1=q_i),\quad i=1,2,...,N$$

是时刻 t=1 处于状态 q_i 的概率。

隐马尔可夫模型是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。π 和 A 决定状态序列，B 决定观测序列。因此，隐马尔可夫模型 λ 可以用三元符号表示，即

$$\lambda=（\pi, A, B）$$

A，B，π 称为隐马尔可夫模型的三要素。

上面所述 HMM 的三个关键元素实际可以分成两部分，其一为 Markov 链，由 π、A 描述，另一部分是一个随机过程，由 B 描述。

三 HMM 的三个基本问题

其实对于 HMM 来说，如果提前知道所有隐含状态之间的转换概率和所有隐含状态到所有可见状态之间的输出概率，做模拟是相当容易的。但是应用 HMM 模型时候呢，往往是缺失了一部分信息的。比如上面那个故事中，有时候你知道骰子有几种，每种骰子是什么，但是不知道掷出来的骰子序列；有时候你只是看到了很多次掷骰子的结果，剩下的什么都不知道。如果应用算法去估计这些缺失的信息，就成了一个很重要的问题。

总结起来，和 HMM 模型相关的算法主要分为三类，

1）评估问题

给定观察序列$O=(o_1,o_2,...,o_T)$和模型 λ={π, A, B}，计算 P(O|λ)。即给定模型和输出观察序列，如何计算从模型生成观察序列的概率，可以把它看作是评估一个模型和给定观察输出序列的匹配程度。

2）解码问题

给定观察序列$O=(o_1,o_2,...,o_T)$和模型 λ={π, A, B}，求某种有意义的情况下最优的相关状态序列I。即给定观测序列，求最有可能的对应的状态序列。

3）学习问题

如何调整模型参数 λ={π, A, B}，对于一个给定的观察序列$O=(o_1,o_2,...,o_T)$，使得 P(O|λ)最大。它试图优化模型的参数来最佳的描述一个给定的观察序列是如何得来的。

下面我们以上面的故事为例子，具体描述一下 HMM 三个基本问题的求解过程。

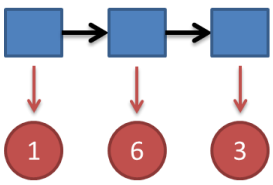
3.1 评估问题

3.1.1 问题描述

比如说你怀疑自己的六面骰被赌场动过手脚了，有可能被换成另一种六面骰，这种六面骰掷出来是 1 的概率更大，是 1/2，掷出来是 2，3，4，5，6 的概率是 1/10。你怎么办？答案很简单，算一算正常的三个骰子掷出一段序列的概率，再算一算不正常的六面骰和另外两个正常骰子掷出这段序列的概率。

3.1.2 求解过程

比如说掷骰子的结果是：



要算用正常的三个骰子掷出这个结果的概率，其实就是将所有可能情况的概率进行加和计算。那么，简单而暴力的方法就是穷举所有的骰子序列，还是计算每个骰子序列对应的概率，再把所有算出来的概率相加，得到的总概率就是我们要求的结果。但是这个方法不能应

用于太长的骰子序列（马尔可夫链）。

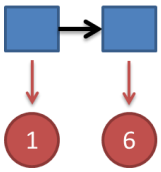
这里我们解决这个问题的算法叫做前向算法（forward algorithm）。我们通过下面这个例子来理解这个算法。

首先，如果我们只掷一次骰子：

看到结果为 1.产生这个结果的总概率可以按照如下计算，总概率为 0.18：

	P1	P2	P3
D6	$\frac{1}{3}*\frac{1}{6}$		
D4	$\frac{1}{3}*\frac{1}{4}$		
D8	$\frac{1}{3}*\frac{1}{8}$		
Total	0.18		

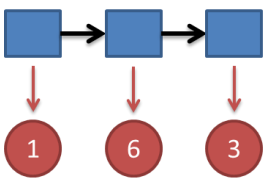
把这个情况拓展，我们掷两次骰子：



看到结果为 1，6.产生这个结果的总概率可以按照如下计算，总概率为 0.05：

	P1	P2	P3
D6	$\frac{1}{3}*\frac{1}{6}$	$P1(D6)*\frac{1}{3}*\frac{1}{6}+P1(D4)*\frac{1}{3}*\frac{1}{6}+P1(D8)*\frac{1}{3}*\frac{1}{6}$	
D4	$\frac{1}{3}*\frac{1}{4}$	$P1(D6)*\frac{1}{3}*0+P1(D4)*\frac{1}{3}*0+P1(D8)*\frac{1}{3}*0$	
D8	$\frac{1}{3}*\frac{1}{8}$	$P1(D6)*\frac{1}{3}*\frac{1}{8}+P1(D4)*\frac{1}{3}*\frac{1}{8}+P1(D8)*\frac{1}{3}*\frac{1}{8}$	
Total	0.18	0.05	

继续拓展，我们掷三次骰子：



看到结果为 1，6，3.产生这个结果的总概率可以按照如下计算，总概率为 0.03：

	P1	P2	P3
D6	$\frac{1}{3}*\frac{1}{6}$	$P1(D6)*\frac{1}{3}*\frac{1}{6}+P1(D4)*\frac{1}{3}*\frac{1}{6}+P1(D8)*\frac{1}{3}*\frac{1}{6}$	$P2(D6)*\frac{1}{3}*\frac{1}{6}+P2(D4)*\frac{1}{3}*\frac{1}{6}+P2(D8)*\frac{1}{3}*\frac{1}{6}$
D4	$\frac{1}{3}*\frac{1}{4}$	$P1(D6)*\frac{1}{3}*0+P1(D4)*\frac{1}{3}*0+P1(D8)*\frac{1}{3}*0$	$P2(D6)*\frac{1}{3}*\frac{1}{4}+P2(D4)*\frac{1}{3}*\frac{1}{4}+P2(D8)*\frac{1}{3}*\frac{1}{4}$
D8	$\frac{1}{3}*\frac{1}{8}$	$P1(D6)*\frac{1}{3}*\frac{1}{8}+P1(D4)*\frac{1}{3}*\frac{1}{8}+P1(D8)*\frac{1}{3}*\frac{1}{8}$	$P2(D6)*\frac{1}{3}*\frac{1}{8}+P2(D4)*\frac{1}{3}*\frac{1}{8}+P2(D8)*\frac{1}{3}*\frac{1}{8}$
Total	0.18	0.05	0.03

同样的，我们一步一步的算，有多长算多长，再长的马尔可夫链总能算出来的。用同样的方法，也可以算出不正常的六面骰和另外两个正常骰子掷出这段序列的概率，然后我们比较一下这两个概率大小，就能知道你的骰子是不是被人换了。

下面给出该算法的理论定义。

3.1.3 Forward-backward 算法

计算给定模型参数情况下输出序列$O=(o_1,o_2,...,o_T)$的出现概率 P(O| λ)，通常采用前后向（forward-backward）算法，其复杂度为$O(K^2L)$.

2.1.3.1 前向算法

前向概率：给定隐马尔可夫模型 λ，定义到时刻 t 部分观察序列为$O=(o_1,o_2,...,o_T)$且状态为 q_i 的概率为前向概率，记作

$$\alpha_t(i)=P(o_1,o_2,...,o_t,i_t=q_i\mid\lambda)$$

可以递归地求得前向概率$\alpha_t(i)$及观察序列概率 P(O| λ)。具体过程如下：

（1） 初始化

$$\alpha_t(i)=\pi_i b_i(o_1),\ i=1,2,...,N$$

（2） 递推

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

（3） 终止

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

2.1.3.2 后向算法

后向算法和前向算法性质上是一样的，只是递推方向不同。

后向概率：给定隐马尔可夫模型 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 $O_{t+1}\ O_{t+2}\ \ldots O_T$ 的概率为后向概率，记作

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

可以递归地求得前向概率 $\beta_t(i)$ 及观察序列概率 $P(O|\ \lambda\ )$。具体过程如下：

（1） 初始化

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

（2） 递推

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

（3） 终止

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

3.2 解码问题

3.2.1 问题描述

我知道我有三个骰子，分别是六面骰、四面骰、八面骰。我也知道我掷了十次的结果（1 6 3 5 2 7 3 5 2 4），我不知道每次用了哪种骰子，我想知道最有可能的骰子序列。

3.2.2 求解过程

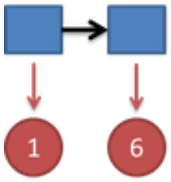
其实最简单而暴力的方法就是穷举所有可能的骰子序列，然后把每个序列对应的概率算出来。然后我们从里面把对应最大概率的序列挑出来就行了。如果马尔可夫链不长，当然可行。如果长的话，穷举的数量太大，就很难完成了。

这里我们解决这个问题的算法叫做 Viterbi 算法。我们通过下面这个例子来理解这个算法。

首先，如果我们只掷一次骰子：

看到结果为 1.对应的最大概率骰子序列就是 D4，因为 D4 产生 1 的概率是 1/4，高于 1/6 和 1/8.

把这个情况拓展，我们掷两次骰子：

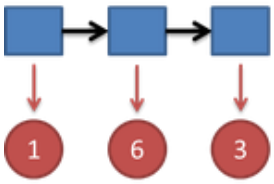


结果为 1，6.这时问题变得复杂起来，我们要计算三个值，分别是第二个骰子是 D6，D4，D8 的最大概率。显然，要取到最大概率，第一个骰子必须为 D4。这时，第二个骰子取到 D6 的最大概率是

$$P2(D6) = P(D4) * P(D4 \to 1) * P(D4 \to D6) * P(D6 \to 6)$$
$$= \frac{1}{3} * \frac{1}{4} * \frac{1}{3} * \frac{1}{6}$$

同样的，我们可以计算第二个骰子是 D4 或 D8 时的最大概率。我们发现，第二个骰子取到 D6 的概率最大。而使这个概率最大时，第一个骰子为 D4。所以最大概率骰子序列就是 D4 D6。

继续拓展，我们掷三次骰子：



同样，我们计算第三个骰子分别是 D6，D4，D8 的最大概率。我们再次发现，要取到最大概率，第二个骰子必须为 D6。这时，第三个骰子取到 D4 的最大概率是

$$P3(D4) = P2(D6) * P(D6 \to D4) * P(D4 \to 3)$$
$$= \frac{1}{216} * \frac{1}{3} * \frac{1}{4}$$

同上，我们可以计算第三个骰子是 D6 或 D8 时的最大概率。我们发现，第三个骰子取

到 D4 的概率最大。而使这个概率最大时，第二个骰子为 D6，第一个骰子为 D4。所以最大概率骰子序列就是 D4 D6 D4。

写到这里，大家应该看出点规律了。既然掷骰子一二三次可以算，掷多少次都可以以此类推。我们发现，我们要求最大概率骰子序列时要做这么几件事情。首先，不管序列多长，要从序列长度为 1 算起，算序列长度为 1 时取到每个骰子的最大概率。然后，逐渐增加长度，每增加一次长度，重新算一遍在这个长度下最后一个位置取到每个骰子的最大概率。因为上一个长度下的取到每个骰子的最大概率都算过了，重新计算的话其实不难。当我们算到最后一位时，就知道最后一位是哪个骰子的概率最大了。然后，我们要把对应这个最大概率的序列从后往前推出来。

下面给出该算法的理论定义。

3.2.3 Viterbi 算法

Viterbi 算法采用动态规划算法，复杂度为$O(K^2L)$，其中 K 和 L 分别为状态个数和序列长度。

首先导入两个变量 δ 和 ψ。定义在时刻 t 状态为 *i* 的所有单个状态路径$(i_1,i_2,...,i_t)$ 中概率最大值为

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

由定义可得变量 δ 的递推公式：

$$\begin{aligned}\delta_{t+1}(i)&=\max_{i_1,i_2,...,i_t} P(i_{t+1}=i,i_t,...,i_1,o_{t+1},...,o_1\mid\lambda)\\&=\max_{1\le j\le N}\left[\delta_t(j)a_{ji}\right]b_i(o_{t+1}),\ i=1,2,...,N;t=1,2,...,T-1\end{aligned}$$

定义在时刻 t 状态为 *i* 的所有单个路径$(i_1,i_2,...,i_{t-1},i)$中概率最大的路径的第 t-1 个结点为

$$\psi_t(i)=\arg\max_{1\le j\le N}\left[\delta_{t-1}(j)a_{ji}\right],\ i=1,2,...,N$$

Viterbi 算法具体过程如下：

（1） 初始化

$$\delta_1(i)=\pi_i b_i(o_1),\ i=1,2,...,N$$

$$\psi_1(i)=0,\ i=1,2,...,N$$

（2） 递推：对 t=2,3，…，T

$$\delta_t(i) = \max_{1\le j\le N}\left[\delta_{t-1}(j)a_{ji}\right]b_i(o_t),\ i=1,2,...,N$$

$$\psi_t(i) = \arg\max_{1\le j\le N}\left[\delta_{t-1}(j)a_{ji}\right],\ i=1,2,...,N$$

（3） 终止

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

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

（4） 最优路径回溯，对 $t=T-1,T-2,...,1$

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

求得最优路径 $I^* = (i_1^*, i_2^*, ..., i_T^*)$.

3.3 学习问题

对于 HMM 的参数选择和优化问题，目前使用较广的处理方法是 Baum-Welch 算法（也就是 EM 算法）。该算法是一种迭代算法，初始时刻由用户给出各参数的经验估计值，通过不断迭代，使各个参数逐渐趋向更为合理的较优值。因为 EM 算法较为复杂，难度较大，这里不再详细讲解，如果有兴趣的话，可以找些关于 EM 算法的资料自行查看。

四 HMM 的应用

下面描述一个 HMM 应用在词性标注上的例子。

词性标注（Part-of-Speech tagging 或 POS tagging)是指对于句子中的每个词都指派一个合适的词性，也就是要确定每个词是名词、动词、形容词或其他词性的过程，又称词类标注或者简称标注。词性标注是自然语言处理中的一项基础任务，在语音识别、信息检索及自然语言处理的许多领域都发挥着重要的作用。

以 Brown 语料库中的句子为例，词性标注的任务指的是，对于输入句子：

The Fulton County Grand Jury said Friday an investigation of Atlanta’s recent primary election produced “ no evidence ” that any irregularities took place .

需要为句子中的每个单词标上一个合适的词性标记，既输出含有词性标记句子：

The/at-tl Fulton/np-tl County/nn-tl Grand/jj-tl Jury/nn-tl said/vbd Friday/nr an/at investigation/nn of/in Atlanta’s/np$ recent/jj primary/jj election/nn produced/vbn “/“ no/at evidence/nn ”/” that/cs any/dti irregularities/nns took/vbd place/nn ./.

在进行词性标注时，前提条件之一便是选择什么样的标记集？Brown 语料库标记集有 87 个，而英语中其他标记集多数是从 Brown 语料库中的标记集发展而来的，如最常用的 Penn Treebank 标记集，包含 45 个标记，是小标记集。汉语标记集中常用的有北大《人民日报》语料库词性标记集、计算所汉语词性标记集等。

确定使用的标记集，之后便是如何进行词性标注了！如果每个单词仅仅对应一个词性标记，那么词性标注就非常容易了。但是语言本身的复杂性导致了并非每一个单词只有一个词性标记，而存在一部分单词有多个词性标记可以选择，如 book 这个单词，既可以是动词（book that flight），也可以是名词(hand me that book)，因此，词性标注的关键问题就是消解这样的歧义，也就是对于句子中的每一个单词在一定的上下文中选择恰如其分的标记。

实际中，英语中的大多数单词都是没有歧义的，也就是这些单词只有一个单独的标记。但是，英语中的最常用单词很多都是有歧义的，因此，任何一个词性标注算法的关键归根结底还是如何解决词性标注中的歧义消解问题。

如何建立一个与词性标注问题相关联的 HMM 模型？首先必须确定 HMM 模型中的隐藏状态和观察符号，也可以说成观察状态，由于我们是根据输入句子输出词性序列，因此可以将词性标记序列作为隐藏状态，而把句子中的单词作为观察符号，那么对于 Brown 语料库来说，就有 87 个隐藏状态（标记集）和将近 4 万多个观察符号（词型）。确定了隐藏状态和观察符号，我们就可以根据训练语料库的性质来学习 HMM 的各项参数了。

HMM 还可以应用在人的行为分析、网络安全和信息抽取中，还有人讲 HMM 用于金融、管理和心理情绪等建模中。随着时代的发展，HMM 必将有更广泛的应用。

结语

隐马尔可夫模型（Hidden Markov Model, HMM）是可用于标注问题的统计学习的模型，描述由隐藏的马尔可夫链随机生成观测序列的过程，属于生成模型。本文以一个例子为主线，用理论结合实际的方法讲解了 HMM 的基本原理和三个基本问题，以及三个问题的求解方法。最后，详细讲述了一个 HMM 在词性标注中的实际应用。

引用

1. http://www.zhihu.com/question/20962240
2. 朱明,郭春生. 隐马尔可夫模型及其最新应用与发展[J]. 计算机系统应用. 2010(09)
3. 李航. 统计学习方法. 北京：清华大学出版社. 2012
4. http://zipperary.com/2013/10/15/an-introduction-to-hmm/
5. http://blog.csdn.net/daringpig/article/details/8072794