

# 计算机科学188：人工智能导论

## 2024年春季

## 笔记15

作者（其他所有笔记）：尼基尔·夏尔马

作者（贝叶斯网络笔记）：乔希·胡格和杰基·梁，由王蕾吉娜编辑

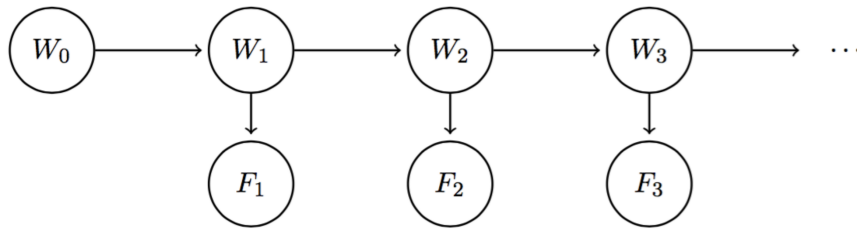
作者（逻辑笔记）：亨利·朱，由考佩林编辑

致谢（机器学习与逻辑笔记）：部分内容改编自教材《人工智能：一种现代方法》。

最后更新时间：2023年8月26日

### 隐马尔可夫模型

通过马尔可夫模型，我们了解了如何通过一系列随机变量来纳入随时间的变化。例如，如果我们想使用上述标准马尔可夫模型来预测第10天的天气，我们可以从初始分布  $P(W_0)$  开始，并使用转移模型通过小前向算法来计算  $P(W_{10})$ 。然而，在时间  $t = 0$  到时间  $t = 10$  之间，我们可能会收集到新的气象证据，这可能会影响我们对任何给定时间步的天气概率分布的信念。简单来说，如果天气预报第10天有 80% 的降雨概率，但第9天晚上天空晴朗，那么这个80%的概率可能会大幅下降。这正是隐马尔可夫模型所帮助我们解决的问题——它允许我们在每个时间步观察一些证据，这些证据可能会影响每个状态下的信念分布。我们天气模型的隐马尔可夫模型可以用如下的贝叶斯网络结构来描述：



与普通马尔可夫模型不同，我们现在有两种不同类型的节点。为了区分这两种节点，我们将每个  $W_i$  称为状态变量，将每个天气预报  $F_i$  称为证据变量。由于  $W_i$  编码了我们对第  $i$  天天气概率分布的信念，那么第  $i$  天的天气预报依赖于这种信念应该是一个自然的结果。该模型暗示了与标准马尔可夫模型类似的条件独立关系，对于证据变量还有一组额外的关系：

$$\begin{aligned} F_1 &\perp\!\!\!\perp W_0 | W_1 \\ \forall i &= 2, \dots, n; \quad W_i \perp\!\!\!\perp \{W_0, \dots, W_{i-2}, F_1, \dots, F_{i-1}\} | W_{i-1} \\ \forall i &= 2, \dots, n; \quad F_i \perp\!\!\!\perp \{W_0, \dots, W_{i-1}, F_1, \dots, F_{i-1}\} | W_i \end{aligned}$$

与马尔可夫模型一样，隐马尔可夫模型假设转移模型  $P(W_{i+1} | W_i)$  是平稳的。隐马尔可夫模型还做了额外的简化假设，即传感器模型

$P(F_i | W_i)$  也是静止的。因此，任何隐马尔可夫模型都可以仅用三个概率表来紧凑地表示：初始分布、转移模型和传感器模型。

作为关于符号的最后一点说明，我们将定义在时间  $i$  且已观察到所有证据  $F_1, \dots, F_i$  时的信念分布：

$$B(W_i) = P(W_i | f_1, \dots, f_i)$$

类似地，我们将  $B'(W_i)$  定义为在时间  $i$  且已观察到证据  $f_1, \dots, f_{i-1}$  时的信念分布：

$$B'(W_i) = P(W_i | f_1, \dots, f_{i-1})$$

将  $e_i$  定义为在时间步  $i$  观察到的证据，你有时可能会看到来自时间步  $1 \leq i \leq t$  的聚合证据以以下形式重新表示：

$$e_{1:t} = e_1, \dots, e_t$$

在此表示法下， $P(W_i | f_1, \dots, f_{i-1})$  可写为  $P(W_i | f_{1:(i-1)})$ 。此表示法将在后续章节中变得相关，我们将在那里讨论时间推移更新，这些更新会将新证据迭代地纳入我们的天气模型。

## 前向算法

利用上述条件概率假设和条件概率表的边缘化属性，我们可以推导出  $B(W_i)$  和  $B'(W_{i+1})$  之间的一种关系，其形式与迷你前向算法的更新规则相同。我们首先使用边缘化：

$$B'(W_{i+1}) = P(W_{i+1} | f_1, \dots, f_i) = \sum_{w_i} P(W_{i+1}, w_i | f_1, \dots, f_i)$$

然后用链式法则将其重新表示如下：

$$B'(W_{i+1}) = P(W_{i+1} | f_1, \dots, f_i) = \sum_{w_i} P(W_{i+1} | w_i, f_1, \dots, f_i) P(w_i | f_1, \dots, f_i)$$

注意到  $P(w_i | f_1, \dots, f_i)$  就是  $B(w_i)$  且  $W_{i+1} \perp \{f_1, \dots, f_i\} | W_i$ ，这简化为我们最终得到的  $B(W_i)$  与  $B'(W_{i+1})$  之间的关系：

$$B'(W_{i+1}) = \sum_{w_i} P(W_{i+1} | w_i) B(w_i)$$

现在让我们考虑如何推导出  $B'(W_{i+1})$  和  $B(W_{i+1})$  之间的关系。通过应用条件概率的定义（带有额外的条件），我们可以看到

$$B(W_{i+1}) = P(W_{i+1} | f_1, \dots, f_{i+1}) = \frac{P(W_{i+1}, f_{i+1} | f_1, \dots, f_i)}{P(f_{i+1} | f_1, \dots, f_i)}$$

在处理条件概率时，一个常用的技巧是将归一化延迟到我们需要归一化概率的时候，我们现在就采用这个技巧。更具体地说，由于上述  $B(W_{i+1})$  展开式中的分母是  $B(W_{i+1})$  所表示的概率表中每个项的公共部分，我们可以省略实际除以  $P(f_{i+1} | f_1, \dots, f_i)$  的操作。相反，我们可以简单地注意到  $B(W_{i+1})$  与  $P(W_{i+1}, f_{i+1} | f_1, \dots, f_i)$  成比例：

$$B(W_{i+1}) \propto P(W_{i+1}, f_{i+1} | f_1, \dots, f_i)$$

比例常数等于  $P(f_{i+1} | f_1, \dots, f_i)$ 。每当我们决定要恢复信念分布  $B(W_{i+1})$  时，我们可以将每个计算值除以这个比例常数。现在，使用链式法则我们可以观察到以下情况：

$$B(W_{i+1}) \propto P(W_{i+1}, f_{i+1} | f_1, \dots, f_i) = P(f_{i+1} | W_{i+1}, f_1, \dots, f_i) P(W_{i+1} | f_1, \dots, f_i)$$

根据之前所述的与隐马尔可夫模型相关的条件独立性假设， $P(f_{i+1} | W_{i+1}, f_1, \dots, f_i)$  等同于简单的  $P(f_{i+1} | W_{i+1})$ ，并且根据定义为  $P(W_{i+1} | f_1, \dots, f_i) = B'(W_{i+1})$ 。这使我们能够以最终形式表达  $B'(W_{i+1})$  和  $B(W_{i+1})$  之间的关系：

$$B(W_{i+1}) \propto P(f_{i+1} | W_{i+1}) B'(W_{i+1})$$

将我们刚刚推导得到的两个关系结合起来，就产生了一种迭代算法，称为前向算法，它是早期迷你前向算法在隐马尔可夫模型中的类似算法：

$$B(W_{i+1}) \propto P(f_{i+1} | W_{i+1}) \sum_{w_i} P(W_{i+1} | w_i) B(w_i)$$

前向算法可以被认为由两个不同的步骤组成：时间推移更新，它对应于从  $B(W_i)$  确定  $B'(W_{i+1})$ ；以及观测更新，它对应于从  $B'(W_{i+1})$  确定  $B(W_{i+1})$ 。因此，为了将我们的信念分布向前推进一个时间步长（即从  $B(W_i)$  计算  $B(W_{i+1})$ ），我们必须首先通过时间推移更新将模型的状态向前推进一个时间步长，然后通过观测更新纳入该时间步长的新证据。考虑以下初始分布、转移模型和传感器模型：

$W_0$	$B(W_0)$	$W_{i+1}$	$W_i$	$P(W_{i+1}   W_i)$	$F_i$	$W_i$	$P(F_i   W_i)$
<i>sun</i>	0.8	<i>sun</i>	<i>sun</i>	0.6	<i>good</i>	<i>sun</i>	0.8
<i>rain</i>	0.2	<i>rain</i>	<i>sun</i>	0.4	<i>bad</i>	<i>sun</i>	0.2
		<i>sun</i>	<i>rain</i>	0.1	<i>good</i>	<i>rain</i>	0.3
		<i>rain</i>	<i>rain</i>	0.9	<i>bad</i>	<i>rain</i>	0.7

为了计算  $B(W_1)$ ，我们首先进行时间更新以得到  $B'(W_1)$ ：

$$\begin{aligned}
 B'(W_1 = \text{sun}) &= \sum_{w_0} P(W_1 = \text{sun} | w_0) B(w_0) \\
 &= P(W_1 = \text{sun} | W_0 = \text{sun}) B(W_0 = \text{sun}) + P(W_1 = \text{sun} | W_0 = \text{rain}) B(W_0 = \text{rain}) \\
 &= 0.6 \cdot 0.8 + 0.1 \cdot 0.2 = \boxed{0.5} \\
 B'(W_1 = \text{rain}) &= \sum_{w_0} P(W_1 = \text{rain} | w_0) B(w_0) \\
 &= P(W_1 = \text{rain} | W_0 = \text{sun}) B(W_0 = \text{sun}) + P(W_1 = \text{rain} | W_0 = \text{rain}) B(W_0 = \text{rain}) \\
 &= 0.4 \cdot 0.8 + 0.9 \cdot 0.2 = \boxed{0.5}
 \end{aligned}$$

因此：

$W_1$	$B'(W_1)$
<i>sun</i>	0.5
<i>rain</i>	0.5

接下来，我们假设第1天的天气预报是好天气（即  $F_1 = \text{好}$ ），并进行观测更新以得到  $B(W_1)$ ：

$$\begin{aligned} B(W_1 = \text{sun}) &\propto P(F_1 = \text{good} | W_1 = \text{sun}) B'(W_1 = \text{sun}) = 0.8 \cdot 0.5 = \boxed{0.4} \\ B(W_1 = \text{rain}) &\propto P(F_1 = \text{good} | W_1 = \text{rain}) B'(W_1 = \text{rain}) = 0.3 \cdot 0.5 = \boxed{0.15} \end{aligned}$$

最后一步是对  $B(W_1)$  进行归一化，注意  $B(W_1)$  表格中的各项之和为  $0.4 + 0.15 = 0.55$ ：

$$\begin{aligned} B(W_1 = \text{sun}) &= 0.4 / 0.55 = \frac{8}{11} \\ B(W_1 = \text{rain}) &= 0.15 / 0.55 = \frac{3}{11} \end{aligned}$$

因此，我们最终得到的  $B(W_1)$  表格如下：

$W_1$	$B'(W_1)$
<i>sun</i>	$8/11$
<i>rain</i>	$3/11$

注意观测天气预报的结果。由于气象员预测天气良好，我们对天气晴朗的信念从时间更新后的  $\frac{1}{2}$  增加到观测更新后的  $\frac{8}{11}$ 。

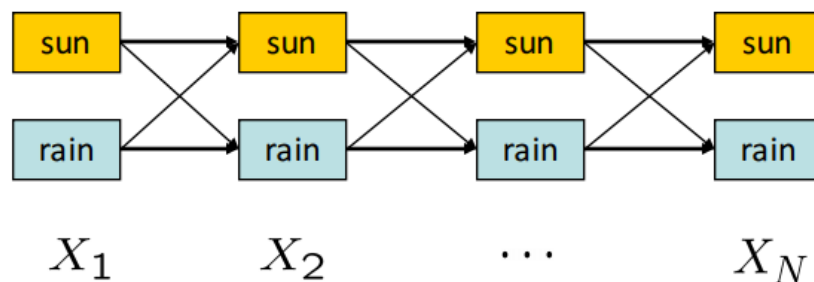
作为临别提示，上述讨论的归一化技巧在处理隐马尔可夫模型时实际上可以显著简化计算。如果我们从某个初始分布开始，并对计算时间  $t$  时的信念分布感兴趣，我们可以使用前向算法迭代地计算  $B(W_1), \dots, B(W_t)$ ，并在最后仅通过将  $B(W_t)$  的表中的每个条目除以其条目的总和来进行一次归一化。

## 维特比算法

在向前算法中，我们使用递归求解了  $P(X_N | e_{1:N})$ ，即给定到目前为止观察到的证据变量时，系统可能处于的状态上的概率分布。与隐马尔可夫模型相关的另一个重要问题是：给定到目前为止观察到的证据变量，系统遵循的最可能的隐藏状态序列是什么？换句话说，我们要求解  $\arg\max_{x_{1:N}} P(x_{1:N} | e_{1:N}) = \arg\max_{x_{1:N}} P(x_{1:N}, e_{1:N})$ 。这个轨迹也可以使用维特比算法通过动态规划来求解。

该算法由两遍计算组成：第一遍按时间正向运行，根据到目前为止观察到的证据计算到每个（状态，时间）元组的最佳路径的概率。第二遍按时间反向运行：首先，它找到概率最高的路径上的终端状态，然后沿着通向该状态的路径（这一定是最佳路径）反向遍历时间。

为了直观展示该算法，考虑以下状态网格，即一个随时间变化的状态和转移的图：



在这个具有两种可能隐藏状态（晴天或雨天）的隐马尔可夫模型中，我们希望计算从  $X_1$  到  $X_N$  的最高概率路径（为每个时间步分配一个状态）。从  $X_{t-1}$  到  $X_t$  的边的权重等于  $P(X_t | X_{t-1})P(E_t | X_t)$ ，路径的概率通过其边权重的乘积来计算。权重公式中的第一项表示特定转移的可能性，第二项表示观察到的证据与结果状态的匹配程度。

回想一下：

$$P(X_{1:N}, e_{1:N}) = P(X_1)P(e_1|X_1) \prod_{t=2}^N P(X_t|X_{t-1})P(e_t|X_t)$$

前向算法计算（直至归一化）

$$P(X_N, e_{1:N}) = \sum_{x_1, \dots, x_{N-1}} P(X_N, x_{1:N-1}, e_{1:N})$$

在维特比算法中，我们想要计算

$$\arg \max_{x_1, \dots, x_N} P(x_{1:N}, e_{1:N})$$

以找到隐藏状态序列的最大似然估计。注意，乘积中的每一项恰好是从层  $t-1$  到层  $t$  的边权重的表达式。因此，网格上沿路径的权重乘积给出了给定证据时该路径的概率。

我们可以求解所有可能隐藏状态上的联合概率表，但这会导致指数级的空间成本。给定这样一个表，我们可以使用动态规划在多项式时间内计算出最佳路径。然而，由于我们可以使用动态规划来计算最佳路径，所以在任何给定时间我们不一定需要整个表。

定义  $m_t[x_t] = \max_{x_{1:t-1}} P(x_{1:t}, e_{1:t})$ ，即从任意  $x_0$  出发且基于目前已观察到的证据，在时刻  $t$  到达给定  $x_t$  的路径的最大概率。这与从步骤1到  $t$  通过网格的最高权重路径相同。另请注意

$$m_t[x_t] = \max_{x_{1:t-1}} P(e_t|x_t)P(x_t|x_{t-1})P(x_{1:t-1}, e_{1:t-1}) \quad (1)$$

$$= P(e_t|x_t) \max_{x_{t-1}} P(x_t|x_{t-1}) \max_{x_{1:t-2}} P(x_{1:t-1}, e_{1:t-1}) \quad (2)$$

$$= P(e_t|x_t) \max_{x_{t-1}} P(x_t|x_{t-1})m_{t-1}[x_{t-1}]. \quad (3)$$

这表明我们可以通过动态规划为所有  $m_t$  递归地计算  $t$ 。这使得确定最可能路径的最后状态  $x_N$  成为可能，但我们仍然需要一种回溯方法来重建整个路径。让我们定义

$a_t[x_t] = P(e_t|x_t) \arg \max_{x_{t-1}} P(x_t|x_{t-1})m_{t-1}[x_{t-1}] = \arg \max_{x_{t-1}} P(x_t|x_{t-1})m_{t-1}[x_{t-1}]$  来跟踪沿着到  $x_t$  的最佳路径的最后一次转移。现在我们可以概述该算法。

**Result:** Most likely sequence of hidden states  $x_{1:N}^*$

```
/* Forward pass
for  $t = 1$  to  $N$  do
    for  $x_t \in \mathcal{X}$  do
        if  $t = 1$  then
             $m_t[x_t] = P(x_t)P(e_0|x_t)$ 
        else
             $a_t[x_t] = \arg \max_{x_{t-1}} P(x_t|x_{t-1})m_{t-1}[x_{t-1}]$ ;
             $m_t[x_t] = P(e_t|x_t)P(x_t|a_t[x_t])m_{t-1}[a_t[x_t]]$ ;
        end
    end
end
/* Find the most likely path's ending point
 $x_N^* = \arg \max_{x_N} m_N[x_N]$ ;
/* Work backwards through our most likely path and find the hidden
states
for  $t = N$  to  $2$  do
     $x_{t-1}^* = a_t[x_t^*]$ ;
end
```

注意，我们的  $a$  数组定义了一组  $N$  序列，其中每个序列都是特定终止状态  $x_N$  下最可能的序列。一旦我们完成前向传播，我们查看  $N$  序列的可能性，选择最佳的那个，并在反向传播中进行重构。因此，我们在多项式空间和时间内计算出了对我们证据的最可能解释。

## 总结

在本笔记中，我们介绍了两种新型模型：

- 马尔可夫模型，它对具有马尔可夫性质的时间相关随机变量进行编码。我们可以使用最小前向算法通过概率推理在我们选择的任何时间步计算马尔可夫模型的信念分布。
- 隐马尔可夫模型，它是具有附加属性的马尔可夫模型，即在每个时间步都可以观察到可能影响我们信念分布的新证据。为了使用隐马尔可夫模型计算任何给定时间步的信念分布，我们使用前向算法。

有时，对这些模型进行精确推理的计算成本可能过高，在这种情况下，我们可以使用粒子滤波作为一种近似推理方法。