

ASR 补充材料：马尔可夫链、HMM 与 GMM

Supplementary Material: 经典声学模型与算法推导

王帅

南京大学智能科学与技术学院

2026 年秋季



南京大學
NANJING UNIVERSITY



智能科学与技术学院
School of Intelligence Science and Technology

本材料配合《ASR Foundation》使用，供课后自主阅读。

阅读前的基础

条件概率与贝叶斯定理、向量与矩阵运算，以及声学特征的基本概念。

读完后应能解释

- 马尔可夫链与 HMM 的区别，隐藏状态与观测各表示什么
- 前向算法、Viterbi 和 Baum–Welch 分别解决什么问题
- GMM 如何描述声学特征，DNN 如何替代 GMM

- 1 **建立概念**: 阅读马尔可夫链、HMM 定义与语音识别示例。
- 2 **理解算法**: 先看三个问题及算法总结, 再逐步核对推导。
- 3 **连接 ASR**: 阅读 GMM 发射密度、三音素和状态绑定。
- 4 **拓展阅读**: 了解 DNN-HMM 的后验与似然转换。

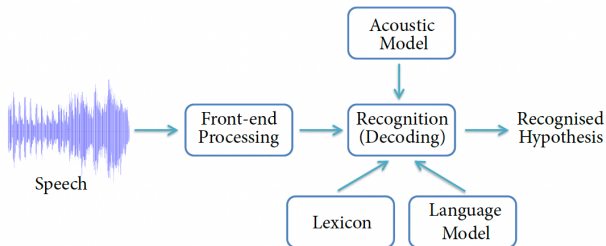
自学方法

用末尾的两状态例题手算前向与 Viterbi, 比较“路径求和”与“最佳路径”。推导时注意每个变量的条件、时间范围和归一化分母。

- 1 阅读准备与 ASR 概率框架
- 2 马尔可夫链与隐藏状态
- 3 隐马尔可夫模型与三个基本问题
- 4 GMM-HMM
- 5 从单音素到上下文相关模型
- 6 DNN-HMM 系统
- 7 总结、自测与参考资料

$$\hat{\mathbf{W}} = \arg \max_{\mathbf{W}} p(\mathbf{O} | \mathbf{W}) P(\mathbf{W}),$$
$$p(\mathbf{O} | \mathbf{W}) = \sum_{\mathbf{L}} p(\mathbf{O} | \mathbf{L}) P(\mathbf{L} | \mathbf{W}).$$

- $\mathbf{O}$: 声学特征序列。 $\mathbf{W}$: 词序列。 $\mathbf{L}$: 发音序列。
- 声学模型 $p(\mathbf{O} | \mathbf{L})$: 给定声学单元, 计算观测的似然。
- 发音词典 $P(\mathbf{L} | \mathbf{W})$: 词序列可能对应哪些发音。
- 语言模型 $P(\mathbf{W})$: 候选词序列的先验概率。



- 前端把语音波形转换为声学特征序列。
- HMM 描述声学状态的先后顺序与持续时间。
- GMM 或 DNN 给出声学状态分数，解码器结合词典与语言模型搜索文字。

符号	含义
$\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_T)$	观测序列，与 ASR 中的声学特征序列 $\mathbf{O}$ 对应
q_t, s_i, N	时刻 t 的隐藏状态、状态取值及状态总数
$A = [a_{ij}], \pi$	状态转移矩阵与初始状态分布
$B, b_j(\mathbf{x}_t)$	发射分布及在状态 s_j 下的观测概率或密度
$\lambda = (A, B, \pi)$	HMM 的参数
$Q = (q_1, \dots, q_T)$	完整隐藏状态序列
$w_{jk}, \mu_{jk}, \Sigma_{jk}$	状态 j 的第 k 个高斯分量的权重、均值和协方差

如何对“序列”建模？

回顾我们的问题



- 前面的概率框架说明，声学模型 $p(\mathbf{O}|\mathbf{L})$ 的目标是计算给定一个声学建模单元序列 $\mathbf{L}$ 时，观测到声学特征序列 $\mathbf{O}$ 的概率。
- 这里的核心挑战是：语音信号是一个**时间序列**。它的特性是随时间动态变化的。
- 我们需要一个强大的数学工具来描述这种随时间演进的、具有“记忆”的序列过程。
- 让我们从一个最简单的序列模型开始：**马尔可夫链**。

定义 (一阶马尔可夫性质)

给定当前状态后，下一时刻的状态与更早的状态条件独立：

$$P(q_{t+1} \mid q_t, \dots, q_1) = P(q_{t+1} \mid q_t).$$

用初始分布 π 和转移矩阵 A 描述链。每行转移概率之和为 1。

天气例子

按“晴、阴、雨”排列状态，转移矩阵为

$$A = \begin{pmatrix} 0.7 & 0.3 & 0 \\ 0.4 & 0.2 & 0.4 \\ 0.2 & 0.3 & 0.5 \end{pmatrix}.$$



例子：一个简单的天气模型

- 如果今天是晴天，明天有 70% 的概率还是晴天，30% 的概率会阴天。
- 如果今天是阴天，明天有 40% 的概率会放晴，40% 的概率继续下雨，20% 的概率保持阴天
- 如果今天是雨天，明天有 50% 的概率继续下雨，30% 概率变为阴天，20% 概率放晴。

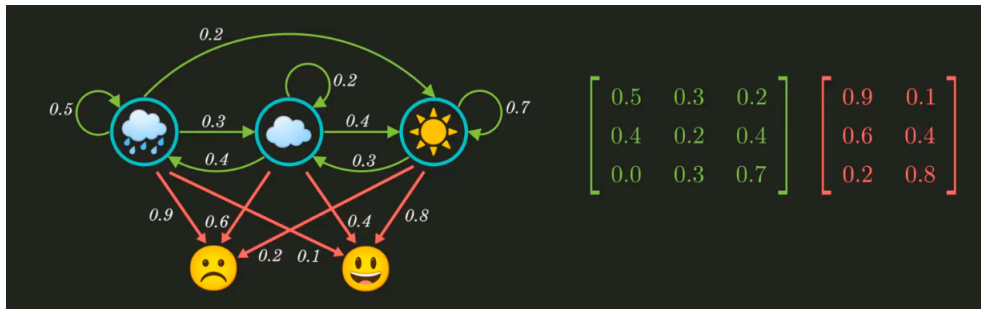
当我们无法直接观测状态时

- 想象一下，你在地，无法直接观测，则原来的天气状态是“隐藏”的。
- 但你有当地的一个朋友，他每天都会出门，并打电话告诉你他今天的心情。
- 他可能的心情有：{开心, 难过}。
- 朋友的心情表现，很大概率是和天气相关的：
 - 晴天时，他更可能 开心。
 - 雨天或者阴天时，他更可能 难过。

任务：根据朋友的心情序列，推断出天气序列

马尔可夫链的局限性

当我们无法直接观测状态时...



任务：根据朋友的心情序列，推断出天气序列

这个问题引出了一个更强大的模型：隐马尔可夫模型。

- 它包含一条**隐藏的、我们无法观测的**马尔可夫状态链（例如：天气序列）。
- 它还包含一个**可观测的**符号序列（例如：朋友的心情序列）。
- 每一个隐藏状态都会以一定的概率“发射”出某个可观测的符号。

HMM 的两大核心要素：

- **状态转移概率：**

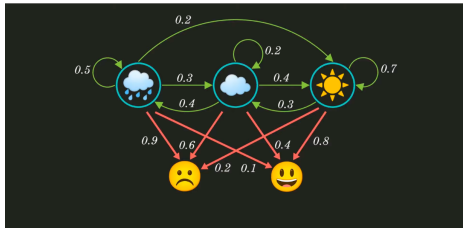
隐藏状态之间如何切换？

(e.g., 从晴天到雨天的概率)

- **发射概率 (观测概率)：**

在某个隐藏状态下，观测到某个现象的概率是多少？

(e.g., 在晴天时，心情难过的概率)



HMM 如何应用于语音识别？

建立“天气”和“语音”之间的联系



外地猜天气

- **隐藏状态:**
天气的真实序列
(晴 → 晴 → 雨)
- **观测序列:**
朋友的心情序列
(开心, 开心, 难过)
- **我们的目标:**
根据心情序列, 推断出最可能的天气序列。

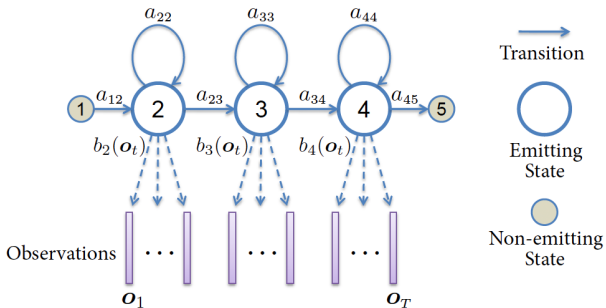
语音识别系统

- **隐藏状态:**
构成单词的 **音素** 序列
(/h/ → /e/ → /l/ → /o/)
- **观测序列:**
从语音信号中提取的 **声学特征** (MFCC)
序列 $(\vec{o}_1, \vec{o}_2, \vec{o}_3, \dots, \vec{o}_T)$
- **我们的目标:**
根据声学特征序列, **解码** 出最可能的音素序列, 进而得到词序列。

HMM 成为了连接 **不可见的语言单元 (音素)** 和 **可测量的声学信号** 之间的完美桥梁。

为什么使用 HMM?

- 语音信号具有时序性和序列性，其特性随时间演进。
- HMM 能够优雅地对这种时变序列进行建模。
- 语音的观测值 (声学特征) 是连续的，但其背后的本质状态是离散的 (如音素)，这与 HMM 的假设完美契合。



- 初始分布: $\pi_i = P(q_1 = s_i)$ 。
- 转移分布: $a_{ij} = P(q_{t+1} = s_j \mid q_t = s_i)$ 。
- 发射分布: $b_j(\mathbf{x}_t) = p(\mathbf{x}_t \mid q_t = s_j)$ 。

两个条件独立假设

状态满足一阶马尔可夫性质。给定当前状态，当前观测与其他状态及观测条件独立。

因此，状态与观测的联合分布为

$$p(Q, \mathbf{X} \mid \lambda) = \pi_{q_1} b_{q_1}(\mathbf{x}_1) \prod_{t=2}^T a_{q_{t-1}, q_t} b_{q_t}(\mathbf{x}_t).$$

此处 q_t 在下标中表示状态索引。离散观测使用概率，连续观测使用概率密度。

- ① 评估 (Evaluation): 给定模型 λ , 计算观测序列 $\mathbf{X}$ 出现的概率 $p(\mathbf{X}|\lambda)$ 。
 - 算法: 前向算法 (Forward Algorithm)。
- ② 解码 (Decoding): 给定模型 λ 和观测序列 $\mathbf{X}$, 找到最可能的状态序列 Q 。
 - 算法: 维特比算法 (Viterbi Algorithm)。
- ③ 学习 (Learning): 给定观测序列 $\mathbf{X}$, 调整模型参数 λ 使得 $p(\mathbf{X}|\lambda)$ 最大。
 - 算法: Baum-Welch 算法 (即 EM 算法在 HMM 中的特例)。

问题一：评估 (Evaluation)

如何计算 $p(\mathbf{X}|\lambda)$?



目标

给定一个观测序列 $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T)$ 和一个 HMM 模型 $\lambda = (A, B, \pi)$ ，计算从该模型中观测到序列 $\mathbf{X}$ 的概率 $p(\mathbf{X}|\lambda)$ 。

● 应用场景:

- 比如我们训练了两个 HMM，一个代表单词“hello”，一个代表“world”。
- 给定一段新的语音特征 $\mathbf{X}$ ，我们可以计算 $p(\mathbf{X}|\lambda_{\text{hello}})$ 和 $p(\mathbf{X}|\lambda_{\text{world}})$ 。
- 哪个概率更高，我们就认为这段语音更可能是哪个单词。这是最简单的识别思路。

核心挑战

我们不知道隐藏的状态序列是什么！

- 我们可以穷举所有可能的、长度为 T 的隐藏状态序列 $Q = (q_1, q_2, \dots, q_T)$ 。
- 对于其中**任意一条**状态路径 Q ，计算它和观测序列 $\mathbf{X}$ 同时发生的联合概率 $p(\mathbf{X}, Q|\lambda)$ ：

$$p(\mathbf{X}, Q|\lambda) = \pi_{q_1} b_{q_1}(\mathbf{x}_1) \cdot a_{q_1 q_2} b_{q_2}(\mathbf{x}_2) \cdots a_{q_{T-1} q_T} b_{q_T}(\mathbf{x}_T)$$

- 然后，把**所有可能路径**的概率全部加起来，根据全概率公式得到最终结果：

$$p(\mathbf{X}|\lambda) = \sum_{\text{所有可能的 } Q} p(\mathbf{X}, Q|\lambda)$$

为什么不可行？

- 假设有 N 个隐藏状态，序列长度为 T 。
- 那么总共有 N^T 条不同的状态路径！
- 这是一个指数级的计算量。对于一段几秒钟的语音 (T 可能上千)，这是天文数字，计算上完全无法实现。

定义 (前向变量)

定义前向变量 $\alpha_t(i)$ 为：在给定模型 λ 的情况下，到时刻 t 为止，观测到的部分序列是 $\mathbf{x}_1, \dots, \mathbf{x}_t$ ，并且在时刻 t 恰好处于状态 s_i 的**联合概率**。

$$\alpha_t(i) = p(\mathbf{x}_1, \dots, \mathbf{x}_t, q_t = s_i | \lambda)$$

- 这个变量是算法的基石，它存储了到达某个中间点的“累积概率”。
- 我们的目标是利用 $\alpha_t(\cdot)$ 来递推计算 $\alpha_{t+1}(\cdot)$ 。

我们来计算第一个前向变量 $\alpha_1(i)$ 。

- 根据定义, $\alpha_1(i) = p(\mathbf{x}_1, q_1 = s_i | \lambda)$ 。
- 这件事要发生, 需要两步:
 - ① 首先, 初始状态必须是 s_i 。这个概率是 π_i 。
 - ② 接着, 在状态 s_i 下, 必须发射出观测值 $\mathbf{x}_1$ 。这个概率是 $b_i(\mathbf{x}_1)$ 。
- 我们将它们的概率相乘:

$$\alpha_1(i) = \pi_i b_i(\mathbf{x}_1)$$

这为我们的递推提供了起点。我们为所有 N 个状态都计算出这个初始值。

2. 递推 (从 t 到 $t+1$)

假设我们已经计算出了时刻 t 的所有前向变量 $\alpha_t(1), \alpha_t(2), \dots, \alpha_t(N)$ 。如何计算 $\alpha_{t+1}(j)$?

- 要在 $t+1$ 时刻到达状态 s_j ，我们必须在 t 时刻处于某个状态 s_i (可以是任意一个)。
- 从 t 时刻的状态 s_i **转移到** $t+1$ 时刻的状态 s_j 的概率是 a_{ij} 。
- 到达 t 时刻状态 s_i 的总概率 (我们已经算过了!) 是 $\alpha_t(i)$ 。
- 所以，从 t 时刻的某个状态 s_i 出发，在 $t+1$ 时刻到达 s_j 的概率是 $\alpha_t(i)a_{ij}$ 。
- 把所有可能的上一个状态 s_i ($i = 1, \dots, N$) 的情况**全部加起来**，就得到了到达 $t+1$ 时刻状态 s_j 的总路径概率： $\sum_{i=1}^N \alpha_t(i)a_{ij}$ 。
- 最后，状态 s_j 还要**发射**出观测值 $\mathbf{x}_{t+1}$ ，其概率为 $b_j(\mathbf{x}_{t+1})$ 。

固定模型参数 λ ，对时刻 t 的状态进行求和：

$$\begin{aligned}\alpha_{t+1}(j) &= \sum_{i=1}^N p(\mathbf{x}_{1:t+1}, q_t = s_i, q_{t+1} = s_j \mid \lambda) \\ &= \sum_{i=1}^N p(\mathbf{x}_{1:t}, q_t = s_i \mid \lambda) P(q_{t+1} = s_j \mid q_t = s_i) \\ &\quad \cdot p(\mathbf{x}_{t+1} \mid q_{t+1} = s_j) \\ &= \left[\sum_{i=1}^N \alpha_t(i) a_{ij} \right] b_j(\mathbf{x}_{t+1}).\end{aligned}$$

每一步用到了什么

第二步分别使用状态的马尔可夫性质与观测的条件独立假设。 $\mathbf{x}_{1:t}$ 表示前 t 个观测。

3. 终止

- 当我们递推到最后一个时刻 T 时，我们得到了一组前向变量 $\alpha_T(1), \alpha_T(2), \dots, \alpha_T(N)$ 。
- 回忆 $\alpha_T(i)$ 的定义： $p(\mathbf{x}_1, \dots, \mathbf{x}_T, q_T = s_i | \lambda)$ 。
- 这是观测到完整序列 $\mathbf{X}$ ，并且在 T 时刻以状态 s_i **结尾** 的概率。
- 我们最终想求的 $p(\mathbf{X} | \lambda)$ 并不关心结尾是哪个状态。所以，我们只需要把所有可能的结尾状态的概率**全部加起来**即可。

$$p(\mathbf{X} | \lambda) = \sum_{i=1}^N \alpha_T(i)$$

效率

每一步递推的计算量大约是 $O(N^2)$ 。总共需要递推 $T - 1$ 次。所以总的计算复杂度是 $O(N^2T)$ ，这是一个线性时间复杂度，远优于 $O(N^T)$ ！

目标：计算 $p(\mathbf{X}|\lambda)$

定义前向变量 $\alpha_t(i) = p(\mathbf{x}_1, \dots, \mathbf{x}_t, q_t = s_i | \lambda)$ 。

1. 初始化:

$$\alpha_1(i) = \pi_i b_i(\mathbf{x}_1), \quad i = 1, \dots, N$$

2. 递推: For $t = 1, \dots, T - 1$:

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^N \alpha_t(i) a_{ij} \right] b_j(\mathbf{x}_{t+1}), \quad j = 1, \dots, N$$

3. 终止:

$$p(\mathbf{X}|\lambda) = \sum_{i=1}^N \alpha_T(i)$$

问题二：解码 (Decoding)

找到最优的隐藏状态序列



目标

给定观测序列 $\mathbf{X}$ 和模型 λ ，找到一条**最有可能**的隐藏状态序列 $Q^* = (q_1^*, q_2^*, \dots, q_T^*)$ 。

$$Q^* = \arg \max_Q P(Q|\mathbf{X}, \lambda)$$

- 这正是语音识别的核心任务：我们看到了声学特征序列（观测），想知道它背后最有可能的音素序列（隐藏状态）是什么。
- 这个问题和“评估”问题非常相似，但有一个关键区别：
 - **前向算法**是把所有路径的概率 **加起来** ($\sum$)。
 - **维特比算法**是要找到那条概率 **最大**的路径 ($\max$)。
- 幸运的是，我们同样可以使用动态规划来解决。

维特比算法：动态规划思想

定义新变量 $\delta_t(i)$ 和 $\psi_t(i)$



定义 (维特比变量 $\delta_t(i)$)

定义 $\delta_t(i)$ 为：到时刻 t 为止，观测到部分序列 $\mathbf{x}_1, \dots, \mathbf{x}_t$ ，并且在时刻 t 到达状态 s_i 的**所有**路径中，**概率最大的那一条路径的概率**。

$$\delta_t(i) = \max_{q_1, \dots, q_{t-1}} p(q_1, \dots, q_{t-1}, q_t = s_i, \mathbf{x}_1, \dots, \mathbf{x}_t | \lambda)$$

定义 (路径记录变量 $\psi_t(i)$)

为了能够最终找出这条路径，我们还需要一个“记事本” $\psi_t(i)$ ，它记录了在时刻 t 到达状态 s_i 的最优路径，其**上一步 (在 $t-1$ 时刻)** 是哪个状态。

$$\psi_t(i) = \arg \max_{1 \leq j \leq N} [\delta_{t-1}(j) a_{ji}], \quad t \geq 2$$

维特比算法：动态规划思想

定义新变量 $\delta_t(i)$ 和 $\psi_t(i)$



$$\alpha_t(i) = p(\mathbf{x}_1, \dots, \mathbf{x}_t, q_t = s_i)$$

$$\delta_t(i) = \max_{q_1, \dots, q_{t-1}} p(q_1, \dots, q_{t-1}, q_t = s_i, \mathbf{x}_1, \dots, \mathbf{x}_t | \lambda)$$

对比 $\alpha_t(i)$ 和 $\delta_t(i)$

- $\alpha_t(i)$ 是所有到达 (t, s_i) 的路径概率的**和**。
- $\delta_t(i)$ 是所有到达 (t, s_i) 的路径概率的**最大值**。

1. 初始化 ($t=1$)

- 对于 $\delta_1(i)$ ，在 $t=1$ 时刻，路径只有一步。所以不存在“多条路径取最大”的问题。
- 它的计算方式和 $\alpha_1(i)$ 完全一样：

$$\delta_1(i) = \pi_i b_i(\mathbf{x}_1)$$

- 对于路径记录 $\psi_1(i)$ ，因为 $t=1$ 是起点，没有“上一步”，所以我们将其记为 0。

$$\psi_1(i) = 0$$

2. 递推 (从 $t-1$ 到 t)

假设我们已经计算出了 $t-1$ 时刻的所有 $\delta_{t-1}(i)$ 。如何计算 $\delta_t(j)$?

- 要找到在 t 时刻到达 s_j 的最优路径，这条路径必然经过 $t-1$ 时刻的某个状态 s_i 。
- 对于**每一个可能的前驱状态** s_i ，从它延伸到 s_j 的路径概率是：

$$\text{路径概率} = \underbrace{\delta_{t-1}(i)}_{\text{到 } s_i \text{ 的最优路径概率}} \times \underbrace{a_{ij}}_{\text{转移概率}}$$

- 我们要从这 N 个可能的前驱状态中，**选出使得这个路径概率最大的那一个**。

$$\max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}]$$

- 最后，状态 s_j 还要发射出观测值 $\mathbf{x}_t$ ，所以再乘上 $b_j(\mathbf{x}_t)$ 。

- 计算最大概率 $\delta_t(j)$:

$$\delta_t(j) = \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}] b_j(\mathbf{x}_t)$$

- 记录最优路径的前驱状态 $\psi_t(j)$: 我们不仅要知道最大概率是多少, 还要记下是**哪个** i 让我们得到了这个最大值。

$$\psi_t(j) = \arg \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}]$$

核心操作

前向算法的核心是 $\sum$, 而维特比算法的核心是 $\max$ 和 $\arg \max$ 。

- 当递推到最后时刻 T 时，我们计算出了 $\delta_T(1), \dots, \delta_T(N)$ 。
- 整个序列的最优路径的概率，就是这些值中的最大值。

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

- 最优路径的**最后一个状态**，就是使 $\delta_T(i)$ 取得最大值的那个状态 i 。

$$q_T^* = \arg \max_{1 \leq i \leq N} \delta_T(i)$$

- 我们已经知道了最优路径的终点 q_T^* 。
- 那么路径的上一个点 q_{T-1}^* 是什么呢？答案就记录在我们的“记事本” ψ 中！

$$q_{T-1}^* = \psi_T(q_T^*)$$

- 再往前一步： $q_{T-2}^* = \psi_{T-1}(q_{T-1}^*)$
- ... 以此类推，直到我们回溯到 q_1^* 。

$$q_t^* = \psi_{t+1}(q_{t+1}^*) \quad \text{for } t = T-1, T-2, \dots, 1$$

最终，我们就得到了完整的最优状态序列 $Q^* = (q_1^*, \dots, q_T^*)$ 。

目标：找到最优状态序列 $Q^* = \arg \max_Q p(Q, \mathbf{X}|\lambda)$

定义 $\delta_t(i)$ 为到时间 t 且处于状态 s_i 的所有路径中的最大概率， $\psi_t(i)$ 为其前驱状态索引。

1. 初始化：

$$\delta_1(i) = \pi_i b_i(\mathbf{x}_1), \quad \psi_1(i) = 0$$

2. 递推 ($t = 2, \dots, T$):

$$\delta_t(j) = \left(\max_{1 \leq i \leq N} \delta_{t-1}(i) a_{ij} \right) b_j(\mathbf{x}_t), \quad \psi_t(j) = \arg \max_{1 \leq i \leq N} \delta_{t-1}(i) a_{ij}$$

3. 终止与回溯：

$$P^* = \max_{1 \leq i \leq N} \delta_T(i), \quad q_T^* = \arg \max_{1 \leq i \leq N} \delta_T(i), \quad q_t^* = \psi_{t+1}(q_{t+1}^*) \quad (t = T-1, \dots, 1)$$

最终得到最优状态序列 $Q^* = (q_1^*, q_2^*, \dots, q_T^*)$ 。

问题三：学习 (Learning)

如何找到最优的模型参数 λ ?



给定一个（或多个）观测序列 $\mathbf{X}$ ，找到一组模型参数 $\lambda = (A, B, \pi)$ ，使得从该模型生成观测序列 $\mathbf{X}$ 的概率 $p(\mathbf{X}|\lambda)$ 最大。

$$\lambda^* = \arg \max_{\lambda} p(\mathbf{X}|\lambda)$$

- 这是三个问题中最重要的，也是最难的。因为它回答了“如何从数据中训练出一个 HMM 模型”。
- **核心困难：这是一个“鸡生蛋，蛋生鸡”的问题。**
 - 如果我们**知道**隐藏状态序列 Q ，那么估计参数 λ 将会非常简单（只需要统计频率即可）。
 - 如果我们**知道**模型参数 λ ，我们就可以推断隐藏状态序列 Q （例如用维特比算法）。
- 但现实是，我们两者都不知道！我们只有观测序列 $\mathbf{X}$ 。

解决方案

使用一种迭代算法——Baum-Welch 算法，它是著名的期望最大化 (Expectation-Maximization, EM) 算法在 HMM 上的一个特例。

若训练数据中真实状态序列已知，参数可以通过频率统计获得。

$$\begin{aligned}\hat{\pi}_i &= \frac{\text{以 } s_i \text{ 开始的序列数}}{\text{总序列数}}, \\ \hat{a}_{ij} &= \frac{\text{从 } s_i \text{ 转移到 } s_j \text{ 的次数}}{\text{从 } s_i \text{ 出发的转移次数}}, \\ \hat{b}_j(k) &= \frac{\text{在 } s_j \text{ 下观测到 } v_k \text{ 的次数}}{\text{处于 } s_j \text{ 的次数}}.\end{aligned}$$

实际训练的困难

最后一式针对离散观测。通常只有观测序列，隐藏状态未知，无法直接统计上述次数。EM用期望计数替代这些真实计数。

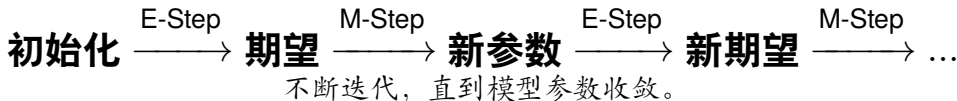
既然不能直接“计数”，那我们可以“软计数”——即计算**期望次数**。

E-Step (期望)

- 固定当前的模型参数 λ 。
- 基于这个模型和观测 X ，我们不再猜测唯一的隐藏状态序列，而是计算出所有我们关心的事件的“期望”：
 - 在时刻 t 处于状态 s_i 的**概率**。
 - 在时刻 t 从 s_i 转移到 s_j 的**概率**。
- 这些概率可以看作是“软计数”或“期望计数”。

M-Step (最大化)

- 假设 E-Step 计算出的“期望计数”就是真实的、准确的计数。
- 使用上一页的频率统计公式，只不过把“硬计数”替换为 E-Step 得到的“期望计数”，来重新估计一组新的、更好的参数 λ^{new} 。



定义 (后向变量)

给定时刻 t 的状态, 后续观测序列出现的概率:

$$\beta_t(i) = p(\mathbf{x}_{t+1}, \dots, \mathbf{x}_T \mid q_t = s_i, \lambda).$$

- 初始化: $\beta_T(i) = 1$, 因为最后时刻之后没有观测。
- 从 $t = T - 1$ 向前递推至 $t = 1$:

$$\beta_t(i) = \sum_{j=1}^N a_{ij} b_j(\mathbf{x}_{t+1}) \beta_{t+1}(j).$$

每一项依次对应: 转移到 s_j 、生成下一个观测、生成剩余观测。对所有 s_j 求和。

$$\gamma_t(i) = P(q_t = s_i \mid \mathbf{X}, \lambda).$$

- $\alpha_t(i)$ 包含从开始到时刻 t 的观测和状态信息。
- $\beta_t(i)$ 包含从 $t+1$ 到结尾的观测信息。
- 两者相乘得到所有经过 (t, s_i) 的路径概率之和。

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{p(\mathbf{X} \mid \lambda)} = \frac{\alpha_t(i)\beta_t(i)}{\sum_j \alpha_t(j)\beta_t(j)}.$$

检查

对任意 t , $\sum_i \gamma_t(i) = 1$ 。分母是完整观测序列的概率或密度。

$$\xi_t(i, j) = P(q_t = s_i, q_{t+1} = s_j \mid \mathbf{X}, \lambda).$$

- 前半段: $\alpha_t(i)$ 。
- 一步转移与发射: $a_{ij}b_j(\mathbf{x}_{t+1})$ 。
- 后半段: $\beta_{t+1}(j)$ 。

$$\xi_t(i, j) = \frac{\alpha_t(i)a_{ij}b_j(\mathbf{x}_{t+1})\beta_{t+1}(j)}{p(\mathbf{X} \mid \lambda)}.$$

时间范围是 $t = 1, \dots, T-1$ 。对所有 (i, j) 求和为 1, 且 $\sum_j \xi_t(i, j) = \gamma_t(i)$ 。

下面给出单条序列、离散观测的重估公式：

$$\begin{aligned}\hat{\pi}_i &= \gamma_1(i), \\ \hat{a}_{ij} &= \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)}, \\ \hat{b}_j(k) &= \frac{\sum_{t=1}^T \gamma_t(j) \mathbb{I}[\mathbf{x}_t = v_k]}{\sum_{t=1}^T \gamma_t(j)}.\end{aligned}$$

计数的含义

转移次数只统计到 $T-1$ ，状态占用次数统计到 T 。多条序列训练时，累加各序列的期望计数，并对初始状态的后验取平均。

- 1 初始化参数 $\lambda = (A, B, \pi)$ 。
- 2 **E-Step**: 计算前向与后向变量, 再计算状态占用后验 $\gamma_t(i)$ 与转移后验 $\xi_t(i, j)$ 。
- 3 **M-Step**: 根据期望计数更新初始分布、转移矩阵和发射分布。
- 4 用新参数替换旧参数, 继续迭代, 直到似然变化足够小。

迭代的性质

在标准 EM 的精确更新下, 观测似然不下降:

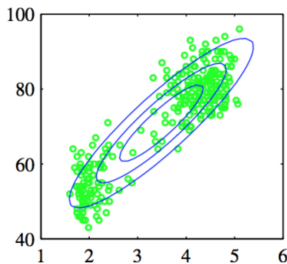
$$p(\mathbf{X} \mid \lambda^{\text{new}}) \geq p(\mathbf{X} \mid \lambda).$$

在常规条件下收敛到驻点, 结果依赖初始化, 不保证全局最优。

- 天气例子使用离散观测，语音特征则是连续向量 $\mathbf{x}_t \in \mathbb{R}^D$ 。
- 因而 $b_j(\mathbf{x}_t) = p(\mathbf{x}_t | q_t = s_j)$ 是概率密度。
- 多维高斯分布是一个简单的选择，但单个分量难以表示多峰分布。

为什么会有多个峰

不同说话人、发音方式与环境会使同一声学状态的特征出现多种分布模式。



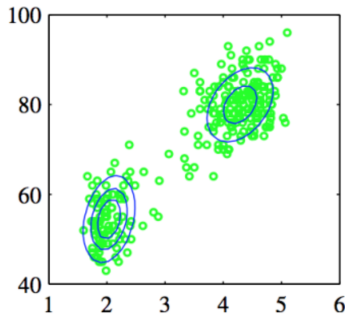
- **核心思想**: 使用多个高斯分布的加权和来共同拟合一个复杂的分布。
- **万能逼近器**: 高斯混合模型 (GMM) 是一个“万能逼近器” (Universal Approximator), 理论上只要分量足够多, 它可以拟合任意形状连续概率分布。

GMM-HMM 的发射概率

状态 j 的发射概率 $b_j(\mathbf{x})$ 由 M 个高斯分量混合而成:

$$b_j(\mathbf{x}) = \sum_{k=1}^M w_{jk} \mathcal{N}(\mathbf{x} | \mu_{jk}, \Sigma_{jk})$$

混合权重满足 $w_{jk} \geq 0$ 、 $\sum_k w_{jk} = 1$ 。每个 HMM 状态 s_j 都拥有自己的一套 GMM 参数: $\{w_{jk}, \mu_{jk}, \Sigma_{jk}\}_{k=1}^M$ 。



为什么用 GMM 作为发射概率模型？



- HMM 中的发射概率 $b_j(\mathbf{x}_t) = p(\mathbf{x}_t | q_t = s_j)$ 需要对声学特征 $\mathbf{x}_t$ 的分布进行建模。
- 声学特征向量 $\mathbf{x}_t$ 是连续的、高维的。
- 单一高斯分布的建模能力太有限，无法捕捉复杂的数据分布。
- 高斯混合模型 (GMM) 是一个万能逼近器 (universal approximator)，理论上可以拟合任意形状连续概率分布。

GMM 公式

状态 j 的发射概率由 M 个高斯分量混合而成：

$$b_j(\mathbf{x}) = \sum_{k=1}^M w_{jk} \mathcal{N}(\mathbf{x}|\mu_{jk}, \Sigma_{jk}), \quad \text{且} \quad \sum_{k=1}^M w_{jk} = 1$$

其中， $\mathcal{N}(\mathbf{x}|\mu, \Sigma)$ 是多维高斯分布：

$$\mathcal{N}(\mathbf{x}|\mu, \Sigma) = \frac{1}{(2\pi)^{D/2} |\Sigma|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x} - \mu)^\top \Sigma^{-1}(\mathbf{x} - \mu)\right]$$

- 仍然使用 Baum–Welch / EM 进行训练。
- 初始分布与转移矩阵的重估公式保持不变：

$$\hat{\pi}_i = \gamma_1(i), \quad \hat{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)}.$$

新增的隐藏变量

除了状态 q_t ，还需要判断观测由状态内的哪个高斯分量 k_t 生成。E-Step 计算状态与分量的联合后验，M-Step 更新 $w_{jk}, \mu_{jk}, \Sigma_{jk}$ 。

核心问题：观测 $\mathbf{x}_t$ 对状态 j 的第 k 个分量贡献了多少期望计数？

给定状态 $q_t = s_j$, 第 k 个高斯分量生成观测的后验为

$$r_t(j, k) = P(k_t = k \mid q_t = s_j, \mathbf{X}, \lambda).$$

利用贝叶斯定理和发射分布的条件独立性:

$$\begin{aligned} r_t(j, k) &= \frac{p(\mathbf{x}_t \mid q_t = s_j, k_t = k) P(k_t = k \mid q_t = s_j)}{p(\mathbf{x}_t \mid q_t = s_j)} \\ &= \frac{w_{jk} \mathcal{N}(\mathbf{x}_t \mid \mu_{jk}, \Sigma_{jk})}{b_j(\mathbf{x}_t)}. \end{aligned}$$

检查

给定状态 j , 所有分量的责任度满足 $\sum_k r_t(j, k) = 1$ 。

$$\gamma_t(j, k) = P(q_t = s_j, k_t = k \mid \mathbf{X}, \lambda).$$

由条件概率的乘法规则：

$$\begin{aligned}\gamma_t(j, k) &= P(q_t = s_j \mid \mathbf{X}, \lambda) P(k_t = k \mid q_t = s_j, \mathbf{X}, \lambda) \\ &= \gamma_t(j) r_t(j, k) \\ &= \gamma_t(j) \frac{w_{jk} \mathcal{N}(\mathbf{x}_t \mid \mu_{jk}, \Sigma_{jk})}{b_j(\mathbf{x}_t)}.\end{aligned}$$

用于参数更新的软计数

$\sum_k \gamma_t(j, k) = \gamma_t(j)$ 。这个联合后验表示观测对状态 j 的第 k 个分量的期望贡献。

GMM-HMM 的 M-Step: 重估 GMM 参数

用“期望贡献度”进行加权平均



有了 $\gamma_t(j, k)$, 我们就可以把它作为权重, 来更新 GMM 的参数。这和标准 GMM 的 EM 算法非常相似。

更新混合权重 w_{jk}

新权重是分量 (j, k) 的总期望贡献度, 除以状态 j 的总期望贡献度。

$$\hat{w}_{jk} = \frac{\sum_{t=1}^T \gamma_t(j, k)}{\sum_{t=1}^T \sum_{m=1}^M \gamma_t(j, m)} = \frac{\sum_{t=1}^T \gamma_t(j, k)}{\sum_{t=1}^T \gamma_t(j)}$$

GMM-HMM 的 M-Step: 重估 GMM 参数

用“期望贡献度”进行加权平均



有了 $\gamma_t(j, k)$, 我们就可以把它作为权重, 来更新 GMM 的参数。这和标准 GMM 的 EM 算法非常相似。

更新均值 μ_{jk}

新均值是所有观测向量 $\mathbf{x}_t$ 的加权平均, 权重为 $\gamma_t(j, k)$ 。

$$\hat{\mu}_{jk} = \frac{\sum_{t=1}^T \gamma_t(j, k) \mathbf{x}_t}{\sum_{t=1}^T \gamma_t(j, k)}$$

GMM-HMM 的 M-Step: 重估 GMM 参数

用“期望贡献度”进行加权平均



有了 $\gamma_t(j, k)$, 我们就可以把它作为权重, 来更新 GMM 的参数。这和标准 GMM 的 EM 算法非常相似。

更新协方差 Σ_{jk}

新协方差是每个观测向量与新均值之差的平方的加权平均。

$$\hat{\Sigma}_{jk} = \frac{\sum_{t=1}^T \gamma_t(j, k) (\mathbf{x}_t - \hat{\mu}_{jk})(\mathbf{x}_t - \hat{\mu}_{jk})^\top}{\sum_{t=1}^T \gamma_t(j, k)}$$

- 到目前为止，我们讨论的 HMM 模型（如为每个音素 /ah/, /b/, /p/ ... 建立一个 HMM）是**上下文无关 (Context-Independent)** 的。
- 这意味着，音素 /ah/ 的发音模型，在任何情况下都是完全一样的。
- **现实情况**：发音会受到协同发音 (Coarticulation) 的严重影响。

例如，音素 /ah/ 的发音

- 在单词“pub” (p-ah-b) 中的发音
- 在单词“tub” (t-ah-b) 中的发音
- 在单词“fuss” (f-ah-s) 中的发音

这三个 /ah/ 的声学实现（即语音特征）是截然不同的，因为它们受到了前后音素的影响。

结论

单音素模型无法捕捉这种上下文相关性，导致模型精度不足。

定义 (三音素)

用左邻音素、中心音素和右邻音素共同定义声学模型，记作 L-C+R。

同一中心音素，不同上下文

- pub 中的 /ah/: p-ah+b。
- fuss 中的 /ah/: f-ah+s。

两种上下文可使用不同 HMM 参数，以描述协同发音造成的差异。

代价

模型数量随上下文组合增加，带来参数量和训练数据覆盖的问题。

- 假设一个语言有 40 个基本音素。
- 理论上，可能存在的三音素数量为：

$$40(\text{左}) \times 40(\text{中}) \times 40(\text{右}) = 40^3 = 64,000 \text{ 个}$$

- 如果每个三音素 HMM 有 3 个状态，每个状态的 GMM 需要均值、方差等参数...
- 这会导致模型参数的总量急剧膨胀，变得异常庞大！

问题一：参数爆炸 (Parameter Explosion)

模型参数过多，不仅需要巨大的存储空间，也使得训练变得极其困难和耗时。

- 在一个非常大的语音训练数据库中，许多理论上可能存在的三音素，实际上可能**从未出现过**，或者只出现过一两次。
- 例如，像 $z-iy+q$ 这样的组合在英语中可能永远不会出现。
- **如果一个三音素在训练数据中没有出现，我们就无法为它训练模型！**
- 即使只出现几次，这点数据也完全不足以训练出一个稳定、可靠的 GMM 参数。

问题二：数据稀疏 (Data Sparsity)

训练数据覆盖不全，导致大量三音素模型无法被有效训练。这在测试时会遇到严重的“未登录三音素”问题。

我们需要一个 trade-off!

核心思想

将声学特性相似的三音素的对应状态绑定，共享一套 GMM 参数。

例：p-ah+b 与 t-ah+b

两者中心音素和右邻音素相同，左邻 /p/ 与 /t/ 均为清塞音。可依据语音学属性与数据中的似然增益，判断是否共享对应状态的参数。

- 减少参数量：多个状态使用相同的发射分布。
- 缓解数据稀疏：合并各状态的训练数据，提高参数估计的稳定性。

关键问题：如何决定哪些状态应该被绑定？



我们不能凭感觉去绑定状态。我们需要一个**自动的、数据驱动**的方法来决定。

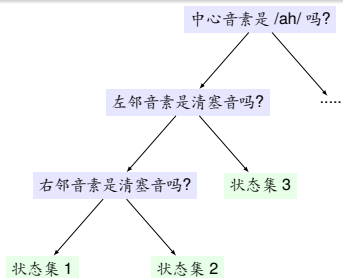
目标

找到一个最优的聚类方案，使得聚类后的模型在训练集上的似然概率 (Likelihood) 最大。

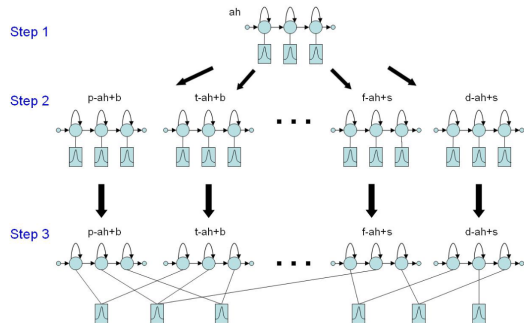
- **方法**: 对每个单音素 (如 /ah/) 的每个 HMM 状态, 分别进行聚类。
- **例如**: 我们把所有中心音素是 /ah/ 的三音素 (如来自 "pub" 的 p-ah+b, 来自 "tub" 的 t-ah+b, 来自 "fuss" 的 f-ah+s 等) 的**第 1 个状态**都拿出来, 放在一个集合里, 然后试图对它们进行聚类。
- **如何聚类**? 通过提出一系列关于上下文音素的“是/非”问题, 来逐步地、层层地划分这些状态。这自然就引出了——**决策树 (Decision Tree)**。

核心思想

将声学特性相似的三音素状态“绑定”在一起，共享 GMM 参数。



- 决策树问题: 基于音素的语音学属性提问。
- 分裂准则: 选择能带来最大似然增益 (Likelihood Gain) 的问题来分裂节点。



- 决策树根据上下文音素的属性逐层划分状态。
- 同一叶子节点中的状态共享 GMM 参数。
- 以 $p-ah+b$ 和 $t-ah+b$ 为例，相似状态有机会进入同一叶子。
- 叶子聚合的数据用于估计共享参数。

- **分布表达能力**: 有限个高斯分量可能难以充分拟合复杂声学分布。
- **特征相关性**: 常用对角协方差简化计算, 同时限制对维度相关性的建模。完整协方差 GMM 可以表示相关性。
- **上下文利用**: 逐帧的发射模型通常需要额外设计动态特征, 才能利用相邻帧信息。

DNN-HMM 的思路

保留 HMM 的状态转移与时间对齐, 用神经网络估计当前帧的状态后验, 再转换为解码分数。

核心思想

各取所长：保留 HMM 强大的时序建模能力（状态转移），同时用一个强大的**深度神经网络 (DNN)** 来替换掉相对较弱的 GMM，用以计算发射概率。

$$\text{GMM-HMM: } \text{HMM}(\pi, A) + \text{GMM}(B)$$



$$\text{DNN-HMM: } \text{HMM}(\pi, A) + \text{DNN}(B)$$

- 在这个混合系统中，DNN 的任务是：给定一个声学特征向量 $\mathbf{x}_t$ ，计算出它属于每一个 HMM 状态 s_j 的概率。
- HMM 的任务保持不变：使用维特比算法，在给定发射概率（由 DNN 提供）和转移概率的情况下，寻找最优的状态序列。

- **HMM 需要什么？**

HMM 解码器需要的是**似然概率** $p(\mathbf{x}_t|s_j)$ ，即在状态 s_j 下生成观测 $\mathbf{x}_t$ 的概率。

- **DNN 提供什么？**

DNN 本质上是一个分类器。它的输出（经过 Softmax 层后）是**后验概率** $P(s_j|\mathbf{x}_t)$ ，即给定观测 $\mathbf{x}_t$ 后，当前帧属于状态 s_j 的概率。

如何从后验转换到似然？

我们可以使用贝叶斯定理：

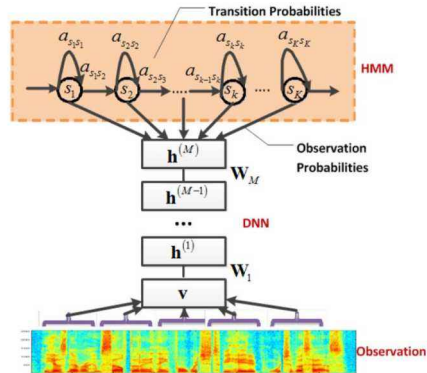
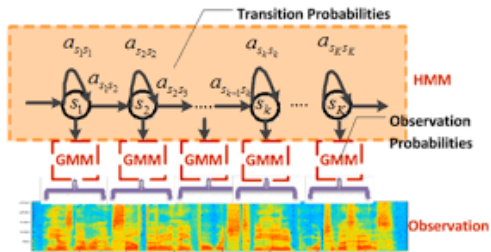
$$p(\mathbf{x}_t | s_j) = \frac{P(s_j | \mathbf{x}_t) p(\mathbf{x}_t)}{P(s_j)}$$

在解码的任意时刻 t ，对于所有状态 j 来说， $p(\mathbf{x}_t)$ 都是一个常数，可以在比较中被忽略。因此，我们实际使用的“伪似然”是：

$$\text{pseudo-likelihood} \propto \frac{P(s_j | \mathbf{x}_t)}{P(s_j)}$$

- $P(s_j | \mathbf{x}_t)$: 由 **DNN 网络输出**。
- $P(s_j)$: 状态 s_j 的**先验概率**，可以通过它在训练数据中出现的频率来估计。

GMM-HMM v.s. DNN-HMM



- ① **训练 GMM-HMM**: 用 Baum-Welch 等方法得到初始声学系统。
- ② **强制对齐**: 限定为正确文本对应的状态路径, 为每一帧生成 HMM 状态标签。
- ③ **训练 DNN**: 输入当前帧及相邻帧特征, 用交叉熵训练状态分类器。
- ④ **构建混合系统**: 将 DNN 后验除以状态先验作为声学分数, 结合 HMM、词典与语言模型解码。

注意

这里介绍的是依靠 GMM-HMM 对齐启动训练的经典流程。强制对齐时已知参考文本, 测试解码时需要搜索候选文字。

- 多层非线性变换可以学习复杂的声学状态判别边界。
- 拼接相邻帧作为输入，使分类器利用局部时间上下文。
- 可以直接使用 FBank 等声学特征，减少部分特征变换。
- HMM 继续负责状态转移，DNN 提供状态后验，解码结构保持模块化。

与 GMM 的区别

GMM 建模 $p(\mathbf{x}_t | s_j)$ ，DNN 通常估计 $P(s_j | \mathbf{x}_t)$ 。比较两者时必须区分似然与后验。

- 脆弱的“流水线”式架构：每一模块的错误都会累积和传播。
- 依赖大量人工设计的启发式规则：如决策树的问题集、HMM 拓扑结构等。
- 训练目标不一致：各模块分别构建或训练，局部目标与最终识别错误率可能不一致。

两个状态 s_1, s_2 ，两个离散观测 a, b ：

$$\pi = (0.6, 0.4), \quad A = \begin{pmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{pmatrix}, \quad B = \begin{pmatrix} 0.5 & 0.5 \\ 0.1 & 0.9 \end{pmatrix}.$$

B 的两列分别对应 a, b ，观测序列为 (a, b) 。

- 1 计算 α_1, α_2 ，求 $P(a, b \mid \lambda)$ 。
- 2 计算 δ_1, δ_2 ，回溯最优状态路径。
- 3 解释为什么观测概率与最佳路径的联合概率不同。

$$\begin{aligned}\alpha_1 &= (0.30, 0.04), & \alpha_2 &= (0.113, 0.1026), \\ P(a, b \mid \lambda) &= 0.2156, \\ \delta_1 &= (0.30, 0.04), & \delta_2 &= (0.105, 0.081).\end{aligned}$$

最佳路径为 (s_1, s_1) ，联合概率为 0.105。前向算法累加所有路径，Viterbi 只保留最大路径，因此两者不同。

进一步复习

- 隐藏状态为何不能直接从观测中读出？
- $\gamma_t(i)$ 与 $\xi_t(i, j)$ 分别怎样构成期望计数？
- 状态绑定怎样缓解数据稀疏？DNN 后验为何需要先验校正？

推荐阅读

- 斯坦福大学 HMM 教程
- Hidden Markov Model Clearly Explained
- 台大李宏毅老师课程

致谢

本补充材料参考了多位学者的资料。部分内容与图示的出处（包括但不限于）如下：西北工业大学谢磊教授的课程 Slides、上海交通大学俞凯教授的课程 Slides、剑桥大学 Jingzhou Yang 的博士论文，以及网络视频“Hidden Markov Model Clearly Explained”等。在此谨致谢意。