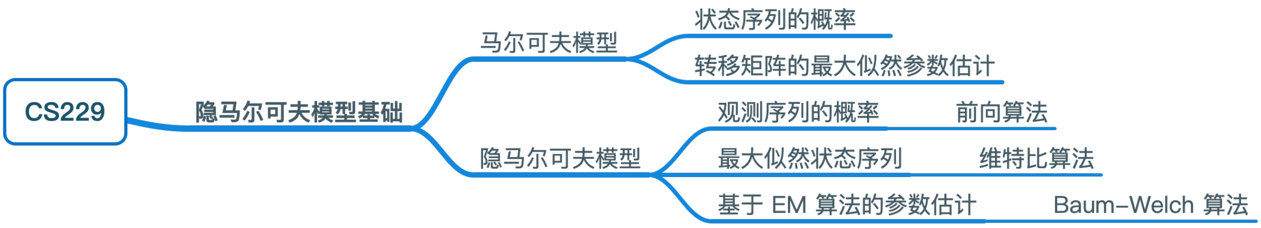
CS229：隐马尔可夫模型基础



马尔可夫模型

- 马尔可夫模型是一种推理时间序列上状态变化的形式
- 给定一个状态集 $S=\{s_1,s_2,\ldots s_{|S|}\}$
  - 我们可以观察出一个随时间变化的序列 $\vec{z}\in S^T$
  - 以一个天气系统的状态 $S=\{sun,cloud,rain\}$ 为例
    - 我们可以观察出一个 5 天（$T=5$）的天气变化序列 $\{z_1=s_{sun},z_2=s_{cloud},z_3=s_{cloud},z_4=s_{rain},z_5=s_{cloud}\}$
- 如果不进行某些限定，则时间 t 的状态 s_j 将会是任意数量变量的函数，将难以建模
  - 因此，我们会提出两个马尔可夫假设来便于我们建模
- 第一个假设是**有限地平线假设**（limited horizon assumption）
  - 该假设指出时间 t 的状态的概率分布只取决于 $t-1$ 时刻的状态
    - 直观的理解就是时刻 t 状态代表了对过去的“足够”总结，可以合理地预测未来

$$P(z_t|z_{t-1},z_{t-2},\ldots,z_1)=P(z_t|z_{t-1})$$

- 第二个假设是**平稳过程假设**（stationary process assumption）
  - 该假设指出给定当前状态，下一个状态的条件分布不随时间变化，即：

$$P(z_t|z_{t-1})=P(z_2|z_1);t\in 2\ldots T$$

- 为了方便，我们会假定存在一个初始状态和初始观察 $z_0\equiv s_0$
  - 其中 s_0 表示时刻 0 时状态的初始概率分布（可以理解为一个未知状态）
  - 这种符号定义可以允许我们将真实初始状态 z_1 的先验分布用 $P(z_1|z_0)$ 来表示
  - 因为对于任何状态序列都有 $z_0=s_0$，所以：

$$P(z_t|z_{t-1},\ldots,z_1)=P(z_t|z_{t-1},\ldots,z_1,z_0)$$

- 我们通过定义一个状态转移矩阵 $A\in\mathbb{R}^{(|S|+1)\times(|S|+1)}$ 来参数化这些转变
  - A_{ij} 的值表示在任意时刻 t 从状态 i 转移到状态 j 的概率
  - 下面给出关于天气状态的转换矩阵：

$$A=\begin{array}{c|cccc} & s_0 & s_{sun} & s_{cloud} & s_{rain} \\ \hline s_0 & 0 & .33 & .33 & .33 \\ s_{sun} & 0 & .8 & .1 & .1 \\ s_{cloud} & 0 & .2 & .6 & .2 \\ s_{rain} & 0 & .1 & .2 & .7 \end{array}$$

        - 从概率可以看出天气是自相关的
            - 即晴天趋向于保持晴天，多云趋向于保持多云
        - 这种模式出现在很多马尔可夫模型中，可以总结为转换矩阵的强对角性
        - 此外，在矩阵中，由初始状态转换为其他三个状态的概率是相同的

马尔可夫模型的两个问题

- 基于上述两个假设以及状态转移矩阵 A ，针对一个马尔科夫链中的状态序列，我们可以提出两个问题：
    1. 一个特定的状态序列 $\vec{z}$ 的概率是多少?
    2. 我们如何估计 A 的参数来最大化一个观测序列 $\vec{z}$ 的概率?

一个状态序列的概率

- 我们可以通过概率的链式法则来计算一个特定状态序列 $\vec{z}$ 的概率：

$$\begin{aligned} P(\vec{z}) &= P\left(z_t, z_{t-1}, \ldots, z_1; A\right) \\ &= P\left(z_t, z_{t-1}, \ldots, z_1, z_0; A\right) \\ &= P\left(z_t \mid z_{t-1}, z_{t-2}, \ldots, z_1; A\right) P\left(z_{t-1} \mid z_{t-2}, \ldots, z_1; A\right) \ldots P\left(z_1 \mid z_0; A\right) \\ &= P\left(z_t \mid z_{t-1}; A\right) P\left(z_{t-1} \mid z_{t-2}; A\right) \ldots P\left(z_2 \mid z_1; A\right) P\left(z_1 \mid z_0; A\right) \\ &= \prod_{t=1}^{T} P\left(z_t \mid z_{t-1}; A\right) \\ &= \prod_{t=1}^{T} A_{z_{t-1} z_t} \end{aligned}$$

    - 第三行使用了链式法则（也可以理解为贝叶斯定理的重复）
        - $P(z_1|z_0)$ 实际上是先验分布
    - 以之前的天气序列为例，我们可以将 $P\left(z_1 = s_{sun}, z_2 = s_{cloud}, z_3 = s_{rain}, z_4 = s_{rain}, z_5 = s_{cloud}\right)$ 表示为：

$$\begin{aligned} & P\left(s_{sun} \mid s_0\right) P\left(s_{cloud} \mid s_{sun}\right) P\left(s_{rain} \mid s_{cloud}\right) P\left(s_{rain} \mid s_{rain}\right) P\left(s_{cloud} \mid s_{rain}\right) \\ & = .33 \times .1 \times .2 \times .7 \times .2 \end{aligned}$$

最大似然参数赋值

- 从学习的观点来看，我们希望找到 A 的参数来最大化观测序列 $\vec{z}$ 的对数似然函数
    - 一个马尔科夫模型的对数似然函数定义如下：

$$\begin{aligned} l(A) &= \log P(\vec{z}; A) \\ &= \log \prod_{t=1}^{T} A_{z_{t-1} z_t} \\ &= \sum_{t=1}^{T} \log A_{z_{t-1} z_t} \\ &= \sum_{i=1}^{|S|} \sum_{j=1}^{|S|} \sum_{t=1}^{T} 1\left\{z_{t-1} = s_i \wedge z_t = s_j\right\} \log A_{ij} \end{aligned}$$

- 对于该优化问题，存在两个约束条件：
  - 每个状态向下一个状态转变的概率之和为 1
  - A 的所有元素都是非负的

- 基于以上两个约束，我们可以构建拉格朗日乘子：

$$
\begin{aligned}
\max_{A} \quad & l(A) \\
\text{s.t.} \quad & \sum_{j=1}^{|S|} A_{ij} = 1,\ i = 1..|S| \\
& A_{ij} \geq 0,\ i,j = 1..|S|
\end{aligned}
$$

  - 我们将等式约束用到乘子构建中，而不等式约束可以被忽略
    - 因为解总是为正数
  - 构建后的拉格朗日乘子为：

$$
\mathcal{L}(A,\alpha) = \sum_{i=1}^{|S|}\sum_{j=1}^{|S|}\sum_{t=1}^{T} 1\left\{z_{t-1} = s_i \wedge z_t = s_j\right\} \log A_{ij} + \sum_{i=1}^{|S|} \alpha_i \left(1 - \sum_{j=1}^{|S|} A_{ij}\right)
$$

    - 求偏导可以得到：

$$
\begin{aligned}
\frac{\partial \mathcal{L}(A,\alpha)}{\partial A_{ij}} &= \frac{\partial}{\partial A_{ij}}\left(\sum_{t=1}^{T} 1\left\{z_{t-1} = s_i \wedge z_t = s_j\right\}\log A_{ij}\right) + \frac{\partial}{\partial A_{ij}}\alpha_i\left(1 - \sum_{j=1}^{|S|} A_{ij}\right) \\
&= \frac{1}{A_{ij}}\sum_{t=1}^{T} 1\left\{z_{t-1} = s_i \wedge z_t = s_j\right\} - \alpha_i \equiv 0 \\
&\Rightarrow A_{ij} = \frac{1}{\alpha_i}\sum_{t=1}^{T} 1\left\{z_{t-1} = s_i \wedge z_t = s_j\right\}
\end{aligned}
$$

    - 将上式代回再对 α 求偏导可得：

$$
\begin{aligned}
\frac{\partial \mathcal{L}(A,\beta)}{\partial \alpha_i} &= 1 - \sum_{j=1}^{|S|} A_{ij} \\
&= 1 - \sum_{j=1}^{|S|}\frac{1}{\alpha_i}\sum_{t=1}^{T} 1\left\{z_{t-1} = s_i \wedge z_t = s_j\right\} \equiv 0 \\
\Rightarrow \alpha &= \sum_{j=1}^{|S|}\sum_{t=1}^{T} 1\left\{z_{t-1} = s_i \wedge z_t = s_j\right\} \\
&= \sum_{t=1}^{T} 1\left\{z_{t-1} = s_i\right\}
\end{aligned}
$$

    - 将上式代回 A_{ij} 的表达式可以得到最终的输出为：

$$
\hat{A}_{ij} = \frac{\sum_{t=1}^{T} 1\left\{z_{t-1} = s_i \wedge z_t = s_j\right\}}{\sum_{t=1}^{T} 1\left\{z_{t-1} = s_i\right\}}
$$

      - 对上式的直观理解：从状态 i 转换至状态 j 的最大似然概率即为 i 向 j 转移的实际

次数除以我们处于状态 i 的总次数

隐马尔可夫模型

- 马尔科夫模型是对时间序列数据的有力抽象
  - 但是如果我们无法观测到序列的状态，就无法进行抽象
- 隐马尔可夫模型可以用来解决这个问题
  - 我们无法观测到实际的状态序列，而是观测到由状态生成的某个输出序列
- 正式来说，在一个隐马尔可夫模型中，我们有如下序列：
  - 一个**观测输出序列**：

$$x = \{x_1, x_2, \ldots, x_T\}$$

    - 其取值自输出字典 $V = \{v_1, v_2, \ldots, v_{|V|}\}$
    - 即 $x_t \in V, t = 1..T$
  - 一个**状态序列**：

$$z = \{z_1, z_2, \ldots, z_T\}$$

    - 其取值自状态字典 $S = \{s_1, s_2, \ldots s_{|S|}\}$
    - 即 $z_t \in S, t = 1 \ldots T$
    - 该序列是**未知**的（无法观测）
- 在隐马尔可夫模型模型中，包含有两个矩阵：
  - 一个是之前提到的状态转移矩阵 A
    - A_{ij} 表示从状态 i 转移到状态 j 的概率
  - 另一个矩阵 B 用于对由隐藏状态生成观测输出的概率建模
    - 我们需要提出**输出独立性假设**：

$$P(x_t = v_k | z_t = s_j) = P(x_t = v_k | x_1, \ldots, x_T, z_1, \ldots, z_T) = B_{jk}$$

      - B_{jk} 表示给定当前时间的状态 s_j，由该状态生成输出 v_k 的概率

关于隐马尔可夫模型的三个问题

- 对于隐马尔可夫模型，我们可以提出三个基本问题：
  1. 观测序列的概率是多少？
  2. 最可能生成该观测序列的状态序列是什么？
  3. 给定一些数据，我们如何学习出矩阵 A 和 B 的参数？

观测序列的概率：前向算法

- 在 HMM 中，我们假设观测序列是通过如下流程生成的：
  - 假设存在一个基于时间序列的状态序列 $\vec{z}$
    - 该序列由马尔可夫模型生成，以状态转移矩阵 A 为参数
  - 在每个时间步 t，我们选择一个输出 x_t 作为状态 z_t 的函数
- 因此，为了计算观测序列的概率，我们需要将给定所有可能状态序列的 $\vec{x}$ 的似然概率相加：

$$
\begin{aligned}
P(\vec{x}; A, B) &= \sum_{\vec{z}} P(\vec{x}, \vec{z}; A, B) \\
&= \sum_{\vec{z}} P(\vec{x} | \vec{z}; A, B) P(\vec{z}; A, B)
\end{aligned}
$$

- 上述公式对任何概率分布均成立
- HMM 假设可以让我们对上述表达式进行简化：

$$
\begin{aligned}
P(\vec{x}; A, B) &= \sum_{\vec{z}} P(\vec{x} | \vec{z}; A, B) P(\vec{z}; A, B) \\
&= \sum_{\vec{z}} \left(\prod_{t=1}^{T} P\left(x_t | z_t; B\right) \right) \left(\prod_{t=1}^{T} P\left(z_t | z_{t-1}; A\right) \right) \\
&= \sum_{\vec{z}} \left(\prod_{t=1}^{T} B_{z_t x_t} \right) \left(\prod_{t=1}^{T} A_{z_{t-1} z_t} \right)
\end{aligned}
$$

  - 上述推导基于输出独立性假设及马尔可夫模型中提到的两个假设
  - 然而，该求和是基于所有可能的状态序列，而 z_t 有 $|S|$ 个可能的取值
    - 所以直接求和的时间复杂度为 $O(|S|^T)$（T 是总时间步数）

- 幸运的是，我们可以通过一种动态规划算法：**前向算法**来更快地计算 $P(\vec{x}; A, B)$
  - 首先我们定义一个量：$\alpha_i(t) = P\left(x_1, x_2, \ldots, x_t, z_t = s_i; A, B\right)$
    - 其代表时间长度为 t 的所有观测值（状态不限）以及在时刻 t 状态为 s_i 的联合概率
    - 通过这样一个量，我们可以将之前的公式表示为：

$$
\begin{aligned}
P(\vec{x}; A, B) &= P\left(x_1, x_2, \ldots, x_T; A, B\right) \\
&= \sum_{i=1}^{|S|} P\left(x_1, x_2, \ldots, x_T, z_T = s_i; A, B\right) \\
&= \sum_{i=1}^{|S|} \alpha_i(T)
\end{aligned}
$$

  - 我们可以通过如下算法来快速地进行求解

Algorithm 1 Forward Procedure for computing $\alpha_i(t)$
1. Base case: $\alpha_i(0) = A_{0\,i},\ i = 1..|S|$
2. Recursion: $\alpha_j(t) = \sum_{i=1}^{|S|} \alpha_i(t-1) A_{ij} B_{j\,x_t},\quad j = 1..|S|,\ t = 1..T$

    - 每个时间步地时间复杂度仅为 $O(|S|)$
      - 算法总体时间复杂度为 $O(|S| \cdot T)$

- 除了前向算法之外，还有一个类似的**后向算法**用来计算如下概率：

$$
\beta_i(t) = P\left(x_T, x_{T-1}, \ldots, x_{t+1}, z_t = s_i; A, B\right)
$$

最大似然状态序列：维特比算法

- HMM 最常见的一个问题是：给定一个观测序列输出 $\vec{x} \in V^T$，最可能的状态序列 $\vec{z} \in S^T$ 是什么?
  - 该问题可以用如下公式表达：

$$\arg\max_{\vec{z}} P(\vec{z}|\vec{x};A,B) = \arg\max_{\vec{z}} \frac{P(\vec{x},\vec{z};A,B)}{\sum_{\vec{z}} P(\vec{x},\vec{z};A,B)} = \arg\max_{\vec{z}} P(\vec{x},\vec{z};A,B)$$

    - 第一步运用了贝叶斯法则
    - 第二步的依据是分母不与 $\vec{z}$ 直接相关
  - 我们可以通过枚举法进行求解，但时间复杂度为 $O(|S|^T)$
    - 与之前类似，我们可以使用动态规划的方法来简化运算
- 用于求解上述问题的动态规划方法被称为**维特比算法**（Viterbi algorithm）
  - 其与前向算法十分类似
  - 区别在于这里不需要追踪概率之和，而是追踪最大概率并记录其对应的状态序列

参数学习：基于 EM 算法的 HMM

- 关于 HMM 的最后一个问题是：给定一个状态序列集，如何求解矩阵 A 和 B 中的参数？
  - 本节将使用 [EM 算法]来进行求解，下图给出了算法的基本流程（针对单个样本）

Algorithm 2 Naive application of EM to HMMs

Repeat until convergence {
(E-Step) For every possible labeling $\vec{z} \in S^T$, set

$$Q(\vec{z}) \quad := \quad p(\vec{z}|\vec{x};A,B)$$

(M-Step) Set

$$\begin{aligned} A,B \quad &:= \quad \arg\max_{A,B} \sum_{\vec{z}} Q(\vec{z}) \log \frac{P(\vec{x},\vec{z};A,B)}{Q(\vec{z})} \\ s.t. \quad & \sum_{j=1}^{|S|} A_{ij} = 1,\ i = 1..|S|;\ A_{ij} \geq 0,\ \ i,j = 1..|S| \\ & \sum_{k=1}^{|V|} B_{ik} = 1,\ i = 1..|S|;\ B_{ik} \geq 0,\ \ i = 1..|S|, k = 1..|V| \end{aligned}$$

}

    - 注意 M-step 中包含了约束条件（因为 A 和 B 中含有概率）
      - 我们将使用拉格朗日乘子来求解上述优化问题
    - 此外，注意到在 E-step 和 M-step 中均需要枚举所有的状态序列，导致过大的时间复杂度
      - 因此我们将使用之前提到的前向和后向算法来简化计算
- 首先，使用马尔可夫假设来重写目标函数：

$$
\begin{aligned}
A, B &= \arg\max_{A,B} \sum_{\vec{z}} Q(\vec{z}) \log \frac{P(\vec{x}, \vec{z}; A, B)}{Q(\vec{z})} \\
&= \arg\max_{A,B} \sum_{\vec{z}} Q(\vec{z}) \log P(\vec{x}, \vec{z}; A, B) \\
&= \arg\max_{A,B} \sum_{\vec{z}} Q(\vec{z}) \log \left(\prod_{t=1}^{T} P\left(x_t | z_t; B\right) \right) \left(\prod_{t=1}^{T} P\left(z_t | z_{t-1}; A\right) \right) \\
&= \arg\max_{A,B} \sum_{\vec{z}} Q(\vec{z}) \sum_{t=1}^{T} \log B_{z_t x_t} + \log A_{z_{t-1} z_t} \\
&= \arg\max_{A,B} \sum_{\vec{z}} Q(\vec{z}) \sum_{i=1}^{|S|} \sum_{j=1}^{|S|} \sum_{k=1}^{|V|} \sum_{t=1}^{T} 1\left\{z_t = s_j \wedge x_t = v_k\right\} \log B_{jk} + 1\left\{z_{t-1} = s_i \wedge z_t = s_j\right\} \log A_{ij}
\end{aligned}
$$

- 第二行去除了分母，因为其与 A 和 B 无关
- 第三行应用了马尔可夫假设
- 第五行使用指示函数来按状态索引 A 和 B

- 下面构建拉格朗日乘子：
  - 与之前一样，因为解必为非负，所以不等约束可以忽略

$$
\begin{aligned}
\mathcal{L}(A, B, \delta, \epsilon) = &\sum_{\vec{z}} Q(\vec{z}) \sum_{i=1}^{|S|} \sum_{j=1}^{|S|} \sum_{k=1}^{|V|} \sum_{t=1}^{T} 1\left\{z_t = s_j \wedge x_t = v_k\right\} \log B_{jk} + 1\left\{z_{t-1} = s_i \wedge z_t = s_j\right\} \log A_{ij} \\
&+ \sum_{j=1}^{|S|} \epsilon_j \left(1 - \sum_{k=1}^{|V|} B_{jk}\right) + \sum_{i=1}^{|S|} \delta_i \left(1 - \sum_{j=1}^{|S|} A_{ij}\right)
\end{aligned}
$$

  - 求偏导并将其设为 0 可得：

$$
\begin{aligned}
\frac{\partial \mathcal{L}(A, B, \delta, \epsilon)}{\partial A_{ij}} &= \sum_{\vec{z}} Q(\vec{z}) \frac{1}{A_{ij}} \sum_{t=1}^{T} 1\left\{z_{t-1} = s_i \wedge z_t = s_j\right\} - \delta_i \equiv 0 \\
A_{ij} &= \frac{1}{\delta_i} \sum_{\vec{z}} Q(\vec{z}) \sum_{t=1}^{T} 1\left\{z_{t-1} = s_i \wedge z_t = s_j\right\} \\
\frac{\partial \mathcal{L}(A, B, \delta, \epsilon)}{\partial B_{jk}} &= \sum_{\vec{z}} Q(\vec{z}) \frac{1}{B_{jk}} \sum_{t=1}^{T} 1\left\{z_t = s_j \wedge x_t = v_k\right\} - \epsilon_j \equiv 0 \\
B_{jk} &= \frac{1}{\epsilon_j} \sum_{\vec{z}} Q(\vec{z}) \sum_{t=1}^{T} 1\left\{z_t = s_j \wedge x_t = v_k\right\}
\end{aligned}
$$

  - 将上述结果代回并关于拉格朗日乘子求偏导可得：

$$
\begin{aligned}
\frac{\partial \mathcal{L}(A,B,\delta,\epsilon)}{\partial \delta_i} &= 1-\sum_{j=1}^{|S|} A_{ij} \\
&= 1-\sum_{j=1}^{|S|} \frac{1}{\delta_i} \sum_{\vec{z}} Q(\vec{z}) \sum_{t=1}^{T} 1\left\{z_{t-1}=s_i \wedge z_t=s_j\right\} \equiv 0 \\
\delta_i &= \sum_{j=1}^{|S|} \sum_{\vec{z}} Q(\vec{z}) \sum_{t=1}^{T} 1\left\{z_{t-1}=s_i \wedge z_t=s_j\right\} \\
&= \sum_{\vec{z}} Q(\vec{z}) \sum_{t=1}^{T} 1\left\{z_{t-1}=s_i\right\} \\
\frac{\partial \mathcal{L}(A,B,\delta,\epsilon)}{\partial \epsilon_j} &= 1-\sum_{k=1}^{|V|} B_{jk} \\
&= 1-\sum_{k=1}^{|V|} \frac{1}{\epsilon_j} \sum_{\vec{z}} Q(\vec{z}) \sum_{t=1}^{T} 1\left\{z_t=s_j \wedge x_t=v_k\right\} \equiv 0 \\
\epsilon_j &= \sum_{k=1}^{|V|} \sum_{\vec{z}} Q(\vec{z}) \sum_{t=1}^{T} 1\left\{z_t=s_j \wedge x_t=v_k\right\} \\
&= \sum_{\vec{z}} Q(\vec{z}) \sum_{t=1}^{T} 1\left\{z_t=s_j\right\}
\end{aligned}
$$

- 将上述结果代回，可以解得：

$$
\begin{aligned}
\hat{A}_{ij} &= \frac{\sum_{\vec{z}} Q(\vec{z}) \sum_{t=1}^{T} 1\left\{z_{t-1}=s_i \wedge z_t=s_j\right\}}{\sum_{\vec{z}} Q(\vec{z}) \sum_{t=1}^{T} 1\left\{z_{t-1}=s_i\right\}} \\
\hat{B}_{jk} &= \frac{\sum_{\vec{z}} Q(\vec{z}) \sum_{t=1}^{T} 1\left\{z_t=s_j \wedge x_t=v_k\right\}}{\sum_{\vec{z}} Q(\vec{z}) \sum_{t=1}^{T} 1\left\{z_t=s_j\right\}}
\end{aligned}
$$

- 上述公式需要遍历所有状态序列，我们可以使用前向和后向概率来进行化简
  - 首先对 $\hat{A}_{ij}$ 的分子进行化简：

$$
\begin{aligned}
&\sum_{\vec{z}} Q(\vec{z}) \sum_{t=1}^{T} 1\left\{z_{t-1}=s_{i} \wedge z_{t}=s_{j}\right\} \\
=&\sum_{t=1}^{T} \sum_{\vec{z}} 1\left\{z_{t-1}=s_{i} \wedge z_{t}=s_{j}\right\} Q(\vec{z}) \\
=&\sum_{t=1}^{T} \sum_{\vec{z}} 1\left\{z_{t-1}=s_{i} \wedge z_{t}=s_{j}\right\} P(\vec{z} \mid \vec{x} ; A, B) \\
=&\frac{1}{P(\vec{x} ; A, B)} \sum_{t=1}^{T} \sum_{\vec{z}} 1\left\{z_{t-1}=s_{i} \wedge z_{t}=s_{j}\right\} P(\vec{z}, \vec{x} ; A, B) \\
=&\frac{1}{P(\vec{x} ; A, B)} \sum_{t=1}^{T} \alpha_{i}(t) A_{i j} B_{j x_{t}} \beta_{j}(t+1)
\end{aligned}
$$

- 前两步进行了重新排列并引入了 Q 的定义
- 第三步使用了贝叶斯法则
- 第四步使用了各元素的定义

- 类似地，分母也可以进行化简：

$$
\begin{aligned}
&\sum_{\vec{z}} Q(\vec{z}) \sum_{t=1}^{T} 1\left\{z_{t-1}=s_{i}\right\} \\
=&\sum_{j=1}^{|S|} \sum_{\vec{z}} Q(\vec{z}) \sum_{t=1}^{T} 1\left\{z_{t-1}=s_{i} \wedge z_{t}=s_{j}\right\} \\
=&\frac{1}{P(\vec{x} ; A, B)} \sum_{j=1}^{|S|} \sum_{t=1}^{T} \alpha_{i}(t) A_{i j} B_{j x_{t}} \beta_{j}(t+1)
\end{aligned}
$$

- 将上述结果综合，可以得到：

$$
\hat{A}_{i j}=\frac{\sum_{t=1}^{T} \alpha_{i}(t) A_{i j} B_{j x_{t}} \beta_{j}(t+1)}{\sum_{j=1}^{|S|} \sum_{t=1}^{T} \alpha_{i}(t) A_{i j} B_{j x_{t}} \beta_{j}(t+1)}
$$

- 类似地，对 $\hat{B}_{jk}$ 的分子进行如下化简：

$$
\begin{aligned}
&\sum_{\vec{z}} Q(\vec{z}) \sum_{t=1}^{T} 1\left\{z_{t}=s_{j} \wedge x_{t}=v_{k}\right\} \\
=&\frac{1}{P(\vec{x} ; A, B)} \sum_{t=1}^{T} \sum_{\vec{z}} 1\left\{z_{t}=s_{j} \wedge x_{t}=v_{k}\right\} P(\vec{z}, \vec{x} ; A, B) \\
=&\frac{1}{P(\vec{x} ; A, B)} \sum_{i=1}^{|S|} \sum_{t=1}^{T} \sum_{\vec{z}} 1\left\{z_{t-1}=s_{i} \wedge z_{t}=s_{j} \wedge x_{t}=v_{k}\right\} P(\vec{z}, \vec{x} ; A, B) \\
=&\frac{1}{P(\vec{x} ; A, B)} \sum_{i=1}^{|S|} \sum_{t=1}^{T} 1\left\{x_{t}=v_{k}\right\} \alpha_{i}(t) A_{i j} B_{j x_{t}} \beta_{j}(t+1)
\end{aligned}
$$

- 同理，其分母可以表示为：

$$
\begin{aligned}
&\sum_{\vec{z}} Q(\vec{z}) \sum_{t=1}^{T} 1\left\{z_t=s_j\right\} \\
=&\frac{1}{P(\vec{x} ; A, B)} \sum_{i=1}^{|S|} \sum_{t=1}^{T} \sum_{\vec{z}} 1\left\{z_{t-1}=s_i \wedge z_t=s_j\right\} P(\vec{z}, \vec{x} ; A, B) \\
=&\frac{1}{P(\vec{x} ; A, B)} \sum_{i=1}^{|S|} \sum_{t=1}^{T} \alpha_i(t) A_{ij} B_{j x_t} \beta_j(t+1)
\end{aligned}
$$

- 将上述结果综合，可以得到：

$$
\hat{B}_{jk}=\frac{\sum_{i=1}^{|S|} \sum_{t=1}^{T} 1\left\{x_t=v_k\right\} \alpha_i(t) A_{ij} B_{j x_t} \beta_j(t+1)}{\sum_{i=1}^{|S|} \sum_{t=1}^{T} \alpha_i(t) A_{ij} B_{j x_t} \beta_j(t+1)}
$$

- 基于上述结果，可以提出用于 HMM 参数学习的前向-后向算法
  - 该算法也被称为 Baum-Welch 算法

Algorithm 3 Forward-Backward algorithm for HMM parameter learning

Initialization: Set A and B as random valid probability matrices
where $A_{i0}=0$ and $B_{0k}=0$ for $i=1..|S|$ and $k=1..|V|$.
Repeat until convergence {
(E-Step) Run the Forward and Backward algorithms to compute α_i and β_i for $i=1..|S|$. Then set:

$$
\gamma_t(i,j) := \alpha_i(t) A_{ij} B_{j\,x_t} \beta_j(t+1)
$$

(M-Step) Re-estimate the maximum likelihood parameters as:

$$
\begin{aligned}
A_{ij} &:= \frac{\sum_{t=1}^{T} \gamma_t(i,j)}{\sum_{j=1}^{|S|} \sum_{t=1}^{T} \gamma_t(i,j)} \\
B_{jk} &:= \frac{\sum_{i=1}^{|S|} \sum_{t=1}^{T} 1\{x_t=v_k\} \gamma_t(i,j)}{\sum_{i=1}^{|S|} \sum_{t=1}^{T} \gamma_t(i,j)}
\end{aligned}
$$

}

  - 与许多 EM 算法的应用类似，该算法是一个非凸优化问题，存在许多局部最优解
    - EM 算法将基于初始值收敛至最大值
      - 因此可以考虑多次运行算法
    - 此外，对由 A 和 B 表示的概率分布进行平滑处理也十分重要
      - 即没有转移或生成为 0 概率（除去初始情况）