第五章 概率语法

5.1 概述

大规模语料库的出现为自然语言统计处理方法的实现提供了可能，统计方法的成功使用推动了语料库语言学的发展。

隐马尔柯夫模型（Hidden Markov Model, HMM）在语音识别中的成功运用以及统计模型在自然语言处理研究中的应用，为自然语言处理研究增添了新的活力。

5.1 概述

基于大规模语料库和统计方法，我们可以

- 发现语言使用的普遍规律
- 进行机器学习、自动获取语言知识
- 对未知语言现象进行推测

5.1 概述

- 概率语法通常指
  - *n* 阶马尔柯夫链语言模型（*n*元文法）
  - 隐马尔柯夫模型（HMM）
  - 概率上下文无关文法 (probabilistic CFG, PCFG)
  - 概率链接语法（probabilistic link grammar）
  - … …

5.2 N元文法的定义

计算语句 $s = w_1\ w_2 \ldots w_m$ 的先验概率：

$$P(s) = P(w_1) \times P(w_2|w_1) \times P(w_3|w_1 w_2) \times \ldots \times P(w_m|w_1 \ldots w_{m-1})$$

$$= \prod_{i=1}^{m} P(w_i \mid w_1 \cdots w_{i-1}) \qquad \ldots (5.1)$$

(当i=1时，$P(w_1|w_0) = P(w_1)$。)

— 语言模型

5.2 N元文法的定义

说明：

(1) w_i 可以是字、词、短语或词类等等，称为统计基元。

(2) w_i 的概率由$w_1, \ldots, w_{i-1}$ 决定， 由特定的一组$w_1, \ldots, w_{i-1}$ 构成的一个序列，称为w_i 的历史。

5.2 N元文法的定义

问题：随着历史基元数量的增加，不同的“历史”（路径）按指数级增长。对于第 i（$i>1$）个统计基元，历史基元的个数为 i-1，如果共有 L 个不同的基元，如词汇表，理论上每一个单词都有可能出现在1到 i-1的每一个位置上，那么，i 基元就有 L^{i-1} 种不同的历史情况。我们必须考虑在所有的 L^{i-1} 种不同历史情况下产生第 i 个基元的概率。那么，模型中有L^m个自由参数 $P(w_m|w_1\ldots w_{m-1})$。

如果 L=5000, m = 3, 自由参数的数目为 1250 亿！

5.2 N元文法的定义

❒ 问题解决方法

设法减少历史基元的个数，将 $w_1\ w_2 \ldots w_{i\text{-}1}$ 映射到等价类 $S(w_1\ w_2 \ldots w_{i\text{-}1})$，使等价类的数目远远小于原来不同历史基元的数目。则有：

$$P(w_i \mid w_1, \cdots, w_{i-1}) = P(w_i \mid S(w_1, \cdots, w_{i-1})) \qquad \ldots (5.2)$$

5.2 N元文法的定义

❐ 如何划分等价类

将两个历史情况映射到同一个等价类，当且仅当这两个历史情况中的最近 n-1 个基元相同，即：

$$H_1:\ w_1\, w_2 \ldots\ldots \underbrace{w_{i\text{-}n+2}\, w_{i\text{-}n+3} \ldots w_{i\text{-}1}}_{n\text{-}1}\ w_i \ldots\ldots$$

$$H_2:\ \ v_1\, v_2 \ldots\ldots \overbrace{v_{k\text{-}n+2}\, v_{k\text{-}n+3} \ldots v_{k\text{-}1}}\ v_k \ldots\ldots$$

$$S(w_1, w_2, \cdots, w_i) = S(v_1, v_2 \cdots, v_k)\ \ iff\ \ H_1:(w_{i-n+2}, \cdots, w_i) = H_2:(v_{k-n+2}, \cdots, v_k) \quad \ldots(5.3)$$

5.2 N元文法的定义

这种情况下的语言模型称为 n 元文法(n-gram)。

<u>通常地</u>,

➢ 当 $n = 1$时，即出现在第 i 位上的基元 w_i 独立于历史，n-gram 被称为一阶马尔柯夫链（uni-gram 或 monogram）

➢ 当 $n = 2$时, n-gram 被称为2阶马尔柯夫链(bi-gram)

➢ 当 $n = 3$时, n-gram 被称为3阶马尔柯夫链(tri-gram)

5.2 N元文法的定义

为了保证条件概率在 i=1时有意义，同时为了保证句子内所有字符串的概率和为 1，即$\sum_s p(s)=1$ ，可以在句子首尾两端增加两个标志: <BOS> w_1 w_2 … w_m <EOS>。不失一般性，对于n > 2 的 n-gram，$P(s)$ 可以分解为：

$$P(s)=\prod_{i=1}^{m+1}P(w_i\mid w_{i-n+1}^{i-1}) \qquad \ldots (5.4)$$

其中，w_i^j 表示词 $w_i \ldots w_j$，$w_{2\text{-}n}$ 从 w_0 开始，w_0 为 <BOS>，w_{m+1} 为 <EOS>。

5.2 N元文法的定义

- 举例：

  给定句子：John read a book

  增加标记：<BOS> John read a book <EOS>

<u>2元文法的概率为</u>：

$$P(\text{John read a book}) = P(\text{John}|\text{<BOS>}) \times P(\text{read}|\text{John}) \times P(\text{a}|\text{read}) \times P(\text{book}|\text{a}) \times P(\text{<EOS>}|\text{book})$$

5.2 N元文法的定义

- 应用－1：音字转换问题

给定拼音串：ta shi yan jiu sheng wu de

可能的汉字串：踏实研究生物的

他实验救生物的

他使烟酒生物的

他是研究生物的

… …

5.2 N元文法的定义

$$
\begin{aligned}
\overset{\frown}{C}String &= \arg\max_{CString} P(CString \mid Pinyin) \\
&= \arg\max_{CString} \frac{P(Pinyin \mid CString)P(CString)}{P(Pingyin)} \\
&= \arg\max_{CString} P(Pinyin \mid CString)P(CString) \\
&= \arg\max_{CString} P(CString)
\end{aligned}
$$

5.2 N元文法的定义

$CString$ = {踏实研究生物的, 他实验救生物的, 他使烟酒生物的, 他是研究生物的, … …}

如果使用2-gram：

$$P(CString_1) = P(\text{踏实}|<BOS>) \times P(\text{研究}|\text{踏实}) \times P(\text{生物}|\text{研究}) \times P(\text{的}|\text{生物}) \times P(<EOS>|\text{的})$$

$$P(CString_2) = P(\text{他}|<BOS>) \times P(\text{实验}|\text{他}) \times P(\text{救}|\text{实验}) \times P(\text{生物}|\text{救}) \times P(\text{的}|\text{生物}) \times P(<EOS>|\text{的})$$

……

5.2 N元文法的定义

如果汉字的总数为：N

- 一元语法：1）样本空间为 N

  2）只选择使用频率最高的汉字

- 二元语法：1）样本空间为 N^2

  2）效果比一元语法明显提高

- 估计对汉字而言四元语法效果会好一些
- 智能狂拼、微软拼音输入法基于 n-gram.

5.2 N元文法的定义

- 应用－2：汉语分词问题

给定汉字串：他是研究生物的。

可能的汉字串：

1）他|是|研究生|物|的

2）他|是|研究|生物|的

5.2 N元文法的定义

$$\widehat{Seg} = \arg\max_{Seg} P(Seg \mid Text)$$

$$= \arg\max_{Seg} \frac{P(Text \mid Seg)P(Seg)}{P(Text)}$$

$$= \arg\max_{Seg} P(Text \mid Seg)P(Seg)$$

$$= \arg\max_{Seg} P(Seg)$$

5.2 N元文法的定义

如果采用二元文法：

P(Seg1) =P(<BOS>|他)×P(是|他)×P(研究生|是)×
P(物|研究生)×P(的|物)×P(的|<EOS>)

P(Seg2) = P(<BOS>|他)×P(是|他)×P(研究|是)×
P(生物|研究)×P(的|生物)×
P(的|<EOS>)

问题：如何获得二元语法模型？

5.3 参数估计

- 两个概念
  - 训练语料(*training data*)：用于建立模型的给定语料。
  - 最大似然估计(*maximum likelihood Evaluation*, MLE)：用相对频率计算概率的方法。

5.3 参数估计

对于 n-gram，参数 $P(w_i \mid w_{i-n+1}^{i-1})$ 可由最大似然估计求得:

$$P(w_i \mid w_{i-n+1}^{i-1}) = f(w_i \mid w_{i-n+1}^{i-1}) = \frac{c(w_{i-n+1}^{i})}{\sum_{w_i} c(w_{i-n+1}^{i})} \quad \ldots(5.5)$$

其中，$\sum_{w_i} c(w_{i-n+1}^{i})$ 是历史串 w_{i-n+1}^{i-1} 在给定语料中出现的次数，即 $c(w_{i-n+1}^{i-1})$。

$f(w_i \mid w_{i-n+1}^{i-1})$ 是在给定 w_{i-n+1}^{i-1} 的条件下 w_i 出现的相对频度。

5.3 参数估计

例如，给定训练语料：

“John read Moby Dick”，

“Mary read a different book”,

“She read a book by Cher”

根据二元文法求句子的概率？

5.3 参数估计

$$P(John|<BOS>) = \frac{c(<BOS>\ John)}{\sum_w c(<BOS>\ w)} = \frac{1}{3}$$

$$P(read|John) = \frac{c(John\ \ read)}{\sum_w c(John\ \ w)} = \frac{1}{1}$$

$$P(a|read) = \frac{c(read\ \ a)}{\sum_w c(read\ \ w)} = \frac{2}{3}$$

John read Moby Dick

Mary read a different book

She read a book by Cher

$$P(book|a) = \frac{c(a\ \ book)}{\sum_w c(a\ \ w)} = \frac{1}{2}$$

$$P(<EOS>|book) = \frac{c(book\ \ <EOS>)}{\sum_w c(book\ \ w)} = \frac{1}{2}$$

$$P(John\ read\ a\ book) = \frac{1}{3} \times 1 \times \frac{2}{3} \times \frac{1}{2} \times \frac{1}{2} \approx 0.06$$

5.3 参数估计

$P(\textit{Cher read a book}) = ?$

$$=P(Cher|<BOS>)\times P(read|Cher)\times P(a|read)\times P(book|a)\times P(<EOS>|book)$$

$$P(Cher|<BOS>)=\frac{c(<BOS>\ Cher)}{\sum_w c(<BOS>\ w)}=\frac{0}{3}$$

$$P(read|Cher)=\frac{c(Cher\ \ read)}{\sum_w c(Cher\ \ w)}=\frac{0}{1}$$

于是，$P(\textit{Cher read a book}) = 0$

John read Moby Dick

Mary read a different book

She read a book by Cher

5.3 参数估计

问题：

1、数据匮乏(稀疏) (*Sparse Data*) 引起零概率问题。

2、如何解决数据匮乏问题？

5.4 数据平滑

- 数据平滑(*Data Smoothing*)的基本思想：

调整最大似然估计的概率值, 使零概率增值，使非零概率下调，“劫富济贫”，消除零概率，改进模型的整体正确率。

- 基本约束：$\sum_{w_i} P(w_i \mid w_1, w_2, \cdots, w_{i-1}) = 1$
- 基本目标：

测试样本语言模型的困惑度越小越好。

5.4 数据平滑

➢ 回顾－困惑度的定义：

对于一个平滑的 n-gram，其概率为 $P(w_i \mid w_{i-n+1}^{i-1})$，

可以计算句子的概率:

$$P(s) = \prod_{i=1}^{m+1} P(w_i \mid w_{i-n+1}^{i-1})$$

假定测试语料 T 由 l_T 个句子构成 $(t_1, \ldots, t_{lT})$，则整个测试集的概率为：

$$P(T) = \prod_{i=1}^{l_T} P(t_i)$$

5.4 数据平滑

模型 $P(w_i \mid w_{i-n+1}^{i-1})$ 对于测试语料的交叉熵定义为：

$$H_p(T) = -\frac{1}{W_T}\log_2 P(T)$$

其中，W_T 是测试文本 T 的词数。

模型 ***P*** 的困惑度 $\boldsymbol{PP_P(T)}$ 定义为：$PP_P(T) = 2^{H_P(T)}$

n-gram 对于英语文本的困惑度范围一般为 50－1000，对应于交叉熵范围为 6－10 bits/word。

5.4 数据平滑

❑加1法 (Lidstone, 1920; Johnson, 1932)

基本思想：每一种情况出现的次数加1。

例如，对于 *uni*-gram，设 w_1, w_2, w_3 三个词，概率分别为：1/3, 0, 2/3，加1后情况？

2/6，1/6，3/6

5.4 数据平滑

对于2-gram 有：

$$P(w_i \mid w_{i-1}) = \frac{1 + c(w_{i-1} w_i)}{\sum_{w_i} [1 + c(w_{i-1} w_i)]}$$

$$= \frac{1 + c(w_{i-1} w_i)}{|V| + \sum_{w_i} c(w_{i-1} w_i)}$$

其中，V 为被考虑语料的词汇量（全部可能的基元数）。

5.4 数据平滑

在前面的 3 个句子的例子中，

$$P(Cher\ read\ a\ book) = P(Cher|\text{<BOS>}) \times P(read|Cher) \times P(a|read) \times P(book|a) \times P(\text{<EOS>}|book)$$

John read Moby Dick

Mary read a different book

She read a book by Cher

<u>原来</u>：

$P(Cher|\text{<BOS>}) = 0/3$

$P(read|Cher) = 0/1$

$P(a|read) = 2/3$

$P(book|a) = 1/2$

$P(\text{<EOS>}|book) = 1/2$

5.4 数据平滑

词汇量：|V|＝11

John read Moby Dick

Mary read a different book

She read a book by Cher

<u>平滑以后</u>：

$P(Cher|\text{<BOS>}) = (0+1)/(11+3) = 1/14$

$P(read|Cher) = (0+1)/(11+1) = 1/12$

$P(a|read) = (1+2)/(11+3) = 3/14$

$P(book|a) = (1+1)/(11+2) = 2/13$

$P(\text{<EOS>}|book)= (1+1)/(11+2) = 2/13$

$$P(Cher\ read\ a\ book) = \frac{1}{14} \times \frac{1}{12} \times \frac{3}{14} \times \frac{2}{13} \times \frac{2}{13} \approx 0.00003$$

5.4 数据平滑

同理，平滑后：

$P(John|\text{<BOS>}) = 2/14$, $P(read|John) = 2/12$,

$P(a|read) = 3/14$, $P(book|a) = 2/13$, $P(\text{<EOS>}|book) = 2/13$

于是，

$$P(John\ read\ a\ book) = P(John|\text{<BOS>}) \times P(read|John) \times P(a|read) \times P(book|a) \times P(\text{<EOS>}|book)$$

John read Moby Dick

Mary read a different book

She read a book by Cher

$$= \frac{2}{14} \times \frac{2}{12} \times \frac{3}{14} \times \frac{2}{13} \times \frac{2}{13} \approx 0.0001$$

5.4 数据平滑

- 减值法(**Discounting**)

基本思想：修改训练样本中的事件的实际计数，使样本中不同事件的概率之和小于1，剩余的概率量分配给未见概率。

5.4 数据平滑

(1) Good-Turing 估计

I. J. Good 1953年引用 Turing 的方法来估计概率分布。

假设 N 是样本数据的大小， n_r 是在样本中正好出现 r 次的事件的数目（在这里，事件为 n-gram $w_1, w_2, \ldots, w_n$）即：出现1次的 n_1个，出现2次的 n_2个, ……

5.4 数据平滑

那么，$N=\sum_{r=1}^{\infty} n_r r$ … (5.6)

由于，$N=\sum_{r=0}^{\infty} n_r r^* = \sum_{r=0}^{\infty} (r+1) n_{r+1}$ 所以，$r^* = (r+1)\frac{n_{r+1}}{n_r}$

那么，Good-Turing 估计在样本中出现 r 次的事件的概率为：

$$P_r = \frac{r^*}{N}$$ … (5.7) (证明略)

A. Nadas. On Turing's Formula for Word Probabilities. In IEEE Trans. On ASSP-33, Dec. 1985. Pages 1414-1416.

5.4 数据平滑

实际应用中，一般直接用 n_{r+1} 代替 $E(n_{r+1})$，n_r 代替 $E(n_r)$。这样，样本中所有事件的概率之和为：

$$\sum_{r>0} n_r \times P_r = 1 - \frac{n_1}{N} < 1 \qquad \ldots (5.8)$$

因此，有 $\frac{n_1}{N}$ 的剩余的概率量就可以均分给所有的未见事件 $(r = 0)$。

Good-Turing 估计适用于大词汇集产生的符合多项式分布的大量的观察数据。

5.4 数据平滑

(2) Back-off （后备/后退）方法

S. M. Katz 1987年提出，所以又称 Katz 后退法。

<u>基本思想</u>：当某一事件在样本中出现的频率大于 K (通常取 K 为0 或1)时，运用最大似然估计减值来估计其概率，否则，使用低阶的，即 (n-1)gram 的概率替代 n-gram 概率。而这种替代必须受归一化因子 α的作用。

5.4 数据平滑

Back-off 方法的另一种理解:

对每个计数 $r > 0$ 的减值，把减值而节省下来的剩余概率根据低阶的 $(n\text{-}1)$gram 分配给未见事件。

5.4 数据平滑

$f(w)$ 是指 w 的频率

最大似然估计方法求概率

$$P(w_n \mid w_1 \cdots w_{n-1}) = \begin{cases} (1-\alpha(f(w_1 \cdots w_n)))\dfrac{f(w_1 \cdots w_n)}{f(w_1 \cdots w_{n-1})} & \text{当} f(w_1 \cdots w_n) > K \\ \alpha(f(w_1 \cdots w_{n-1}))P(w_n \mid w_2 \cdots w_{n-1}) & \text{当} f(w_1 \cdots w_n) \le K \end{cases} \quad \ldots (5.9)$$

α 是归一化因子，为 f 的函数。

(n-1)gram 概率

5.4 数据平滑

(3) 绝对减值法

H. Ney 和 U. Essen 提出。

<u>基本思想</u>：从每个计数 r 中减去同样的量，剩余的概率量由未见事件均分。

设 K 为所有可能事件的数目（当事件为 n-gram 时，如果统计基元为词，且词汇集的大小为 L，则 $K=L^n$）。

5.4 数据平滑

那么，样本出现了 r 次的事件的概率可以由如下公式估计：

$$P_r = \begin{cases} \dfrac{r-b}{N} & \text{当 } r > 0 \\ \dfrac{b(K-n_0)}{Nn_0} & \text{当 } r = 0 \end{cases} \qquad \ldots (5.10)$$

其中，n_0 为样本中未出现的事件的数目。b 为减去的常量。

5.4 数据平滑

$b(K - n_0)/N$ 是由于减值而产生的剩余概率量。

b 为自由参数，可以通过留存数据(heldout data)法求得, 利用留一法(leave one out) 可以求得b的上限为：

$$b \le \frac{n_1}{n_1 + 2n_2} < 1 \qquad \dots (5.11)$$

实际运用中，常用上限代替优化的b。

H. Ney and U. Essen. Estimating Small Probabilities by Leaving-one-Out. In Proc. Eurospeech 1993. Pages 2239-2242.

5.4 数据平滑

(4) 线性减值法

基本思想：从每个计数 r 中减去与该计数成正比的量（减值函数为线性的），剩余概率量 α 被 n_0 个未见事件均分。

$$P_r = \begin{cases} \dfrac{(1-\alpha)r}{N} & \text{当 } r > 0 \\ \dfrac{\alpha}{n_0} & \text{当 } r = 0 \end{cases} \qquad \ldots (5.12)$$

自由参数 α 的优化值为：$\dfrac{n_1}{N}$

绝对减值法产生的n-gram 通常优于线性减值法。

5.4 数据平滑

❑ 删除插值法 (Deleted Interpolation)

<u>基本思想</u>：用低阶语法估计高阶语法，即当 tri-gram 的值不能从训练数据中准确估计时，用 bi-gram 来替代，同样，当 bi-gram 的值不能从训练语料中准确估计时，可以用 uni-gram 的值来代替。插值公式：

$$P(w_3 \mid w_1 w_2) = \lambda_3 P'(w_3 \mid w_1 w_2) + \lambda_2 P'(w_3 \mid w_2) + \lambda_1 P'(w_3) \quad \ldots (5.13)$$

其中， $\lambda_1 + \lambda_2 + \lambda_3 = 1$

5.4 数据平滑

- λ_1，λ_2，λ_3 的确定：

将训练语料分为两部分，即从原始语料中删除一部分作为留存数据（heldout data）。

第一部分用于估计 $P'(w_3 \mid w_1 w_2)$，$P'(w_3 \mid w_2)$ 和 $P'(w_3)$。

第二部分用于计算$\lambda_1, \lambda_2, \lambda_3$：使语言模型对留存数据的困惑度（Perplexity）最小。

5.4 数据平滑

- 关于各种平滑方法的比较请参阅：

  Chen, Stanley F. and Joshua Goodman. 1998. An Empirical Study of Smoothing Techniques for Language Model. Available from the website:
  http://www-2.cs.cmu.edu/~sfc/html/publications.html.

- **SRI** 语言模型工具：
  http://www.speech.sri.com/projects/srilm/

- **CMU-Cambridge** 语言模型工具：

  http://mi.eng.cam.ac.uk/~prc14/toolkit.html

5.5 马尔柯夫模型

❑马尔可夫模型描述

存在一类重要的随机过程：如果一个系统有 N 个状态 $S_1, S_2, \ldots, S_N$, 随时间的推移，该系统从某一状态转移到另一状态。系统在时间 t 的状态记为 q_t。系统在时间 t 处于状态 S_j $(1\leq j\leq N)$ 的概率取决于其在时间 1, 2, …, t-1 的状态，该概率为：

$$P(q_t = S_j \mid q_{t-1} = S_i, q_{t-2} = S_k, \cdots)$$

5.5 马尔柯夫模型

◆ 假设1：

如果在特定情况下，系统在时间 t 的状态只与其在时间 t-1 的状态相关，则该系统构成一个离散的一阶马尔柯夫链：

$$P(q_t = S_j \mid q_{t-1} = S_i, q_{t-2} = S_k, \cdots) = P(q_t = S_j \mid q_{t-1} = S_i) \qquad \ldots (5.15)$$

5.5 马尔柯夫模型

◆ 假设2：

如果只考虑公式(5.15)独立于时间 t 的随机过程，即所谓的不动性假设，状态与时间无关，那么：

$$P(q_t = S_j \mid q_{t-1} = S_i) = a_{ij}, \quad 1 \le i, j \le N \qquad \ldots (5.16)$$

该随机过程称为<u>马尔柯夫模型 **(Markov Model)**</u>。

5.5 马尔柯夫模型

在马尔柯夫模型中，状态转移概率 a_{ij} 必须满足下列条件：

$$a_{ij} \geq 0 \qquad \ldots (5.17)$$

$$\sum_{j=1}^{N} a_{ij} = 1 \qquad \ldots (5.18)$$

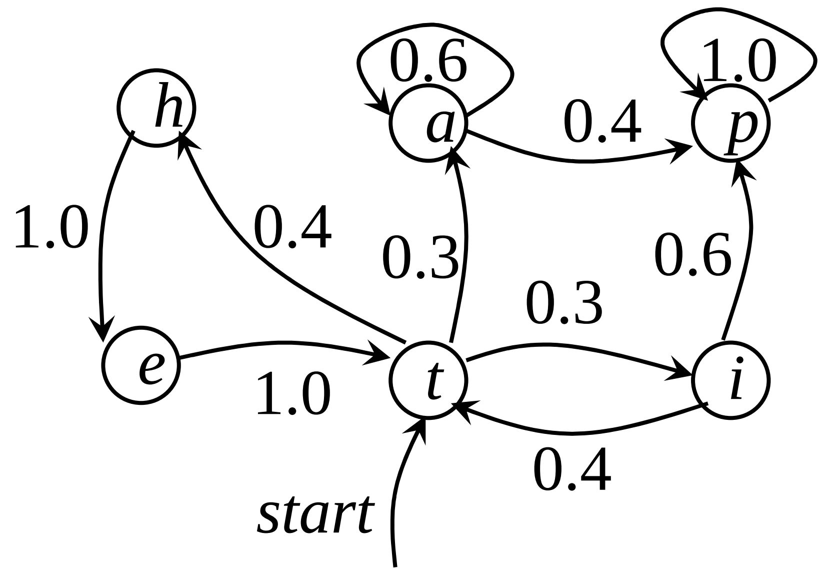
马尔柯夫模型又可视为随机有限状态自动机，该有限状态自动机的每一个状态转换过程都有一个相应的概率，该概率表示自动机采用这一状态转换的可能性。

5.5 马尔柯夫模型

◆ 马尔柯夫链可以表示成状态图（转移弧上有概率的非确定的有限状态自动机）

– 零概率的转移弧省略。

– 每个节点上所有发出弧的概率之和等于1。



5.5 马尔柯夫模型

状态序列 $S_1, \ldots, S_T$ 的概率：

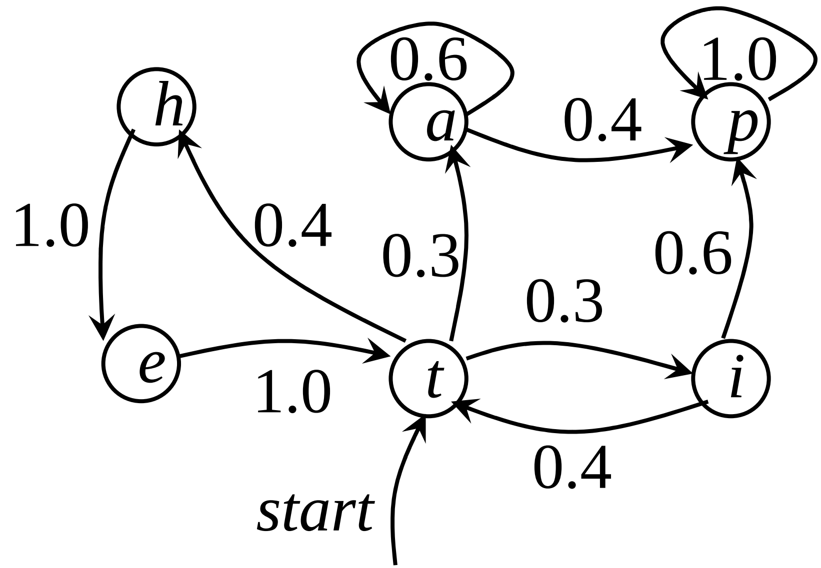
$$P(S_1,\cdots,S_T) = P(S_1)P(S_2 \mid S_1)P(S_3 \mid S_1,S_2)\cdots P(S_T \mid S_1,\cdots,S_{T-1})$$

$$= P(S_1)P(S_2 \mid S_1)P(S_3 \mid S_2)\cdots P(S_T \mid S_{T-1})$$

$$= \pi_{S_1} \prod_{t=1}^{T-1} a_{S_t S_{t+1}} \qquad \ldots (5.17)$$

其中，$\pi_i=P(q_1=S_i)$，为初始状态的概率。

5.5 马尔柯夫模型



$$P(t,i,p) = P(S_1 = t)P(S_2 = i \mid S_1 = t)P(S_3 = p \mid S_2 = i)$$

$$= 1.0 \times 0.3 \times 0.6$$

$$= 0.18$$

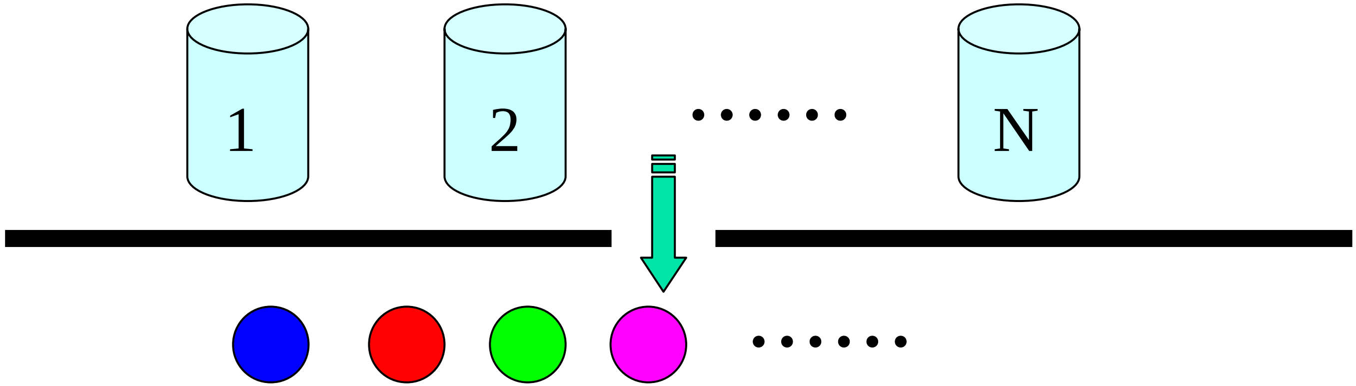
5.6 隐马尔柯夫模型

❑隐马尔柯夫模型(Hidden Markov Model, HMM)

描写：该模型是一个双重随机过程，我们不知道具体的状态序列，只知道状态转移的概率，即模型的状态转换过程是不可观察的（隐蔽的），而可观察的事件的随机过程是隐蔽的状态转换过程的随机函数。

5.6 隐马尔柯夫模型

例如：N 个袋子，每个袋子中有 M 种不同颜色的球。一实验员根据某一概率分布选择一个袋子，然后根据袋子中不同颜色球的概率分布随机取出一个球，并报告该球的颜色。对局外人：可观察的过程是不同颜色球的序列，而袋子的序列是不可观察的。每只袋子对应 HMM 中的状态；球的颜色对应于 HMM 中的状态的输出。



5.6 隐马尔柯夫模型

HMM 的组成

1. 模型中的状态数为 N (袋子的数量)

2. 从每一个状态可能输出的不同的符号数 M (不同颜色球的数目)

5.6 隐马尔柯夫模型

3. 状态转移概率矩阵 $A = a_{ij}$ （ a_{ij} 为实验员从一只袋子(状态 S_i) 转向另一只袋子(状态 S_j) 取球的概率)。其中

$$\begin{cases} a_{ij} = P(q_{t+1} = S_j \mid q_t = S_i), \quad 1 \le i, j \le N \\ a_{ij} \ge 0 \\ \sum_{j=1}^{N} a_{ij} = 1 \end{cases} \qquad \ldots (5.18)$$

5.6 隐马尔柯夫模型

4. 从状态 S_j 观察到某一特定符号 v_k 的概率分布矩阵为：

$$B=b_j(k)$$

其中，$b_j(k)$为实验员从第 j 个袋子中取出第 k 种颜色的球的概率。那么，

$$\begin{cases} b_j(k)=P(O_t=v_k \mid q_t=S_j), & 1\le j\le N,\ 1\le k\le M \\ b_j(k)\ge 0 \\ \sum_{k=1}^{M} b_j(k)=1 \end{cases} \qquad \ldots (5.19)$$

5.6 隐马尔柯夫模型

5. 初始状态的概率分布为：$\pi = \pi_i$，其中，

$$\begin{cases} \pi_i = P(q_1 = S_i), \quad 1 \le i \le N \\ \pi_i \ge 0 \\ \sum_{i=1}^{N} \pi_i = 1 \end{cases} \qquad \ldots (5.20)$$

为了方便，一般将 HMM 记为：$\mu = (A, B, \pi)$

或者 $\mu=(S,O,A,B,\pi)$ 用以指出模型的参数集合。

5.6 隐马尔柯夫模型

给定HMM求观察序列

给定模型 $\mu=(A,B,\pi)$，观察序列 $O=O_1,O_2,\cdots,O_T$ 可以由如下步骤产生：

(1) 令 $t=1$

(2) 根据初始状态概率分布 $\pi=\pi_i$ 选择一初始状态 $q_1=S_i$

(3) 根据状态 S_i 的输出概率分布 $b_i(k)$, 输出 $O_t=v_k$

(4) 根据状态转移概率分布 a_{ij} ，转移到新状态 $q_{t+1}=S_j$

(5) $t=t+1$, 如果 $t<T$, 重复步骤 (3) (4)，否则结束。

5.6 隐马尔柯夫模型

- HMM 中的三个问题

(1) 在给定模型 $\mu=(A,B,\pi)$ 和观察序列 $O=O_1,O_2,\cdots,O_T$ 情况下，怎样快速计算概率 $P(O\mid\mu)$ ？

(2) 在给定模型 $\mu=(A,B,\pi)$ 和观察序列 $O=O_1,O_2,\cdots,O_T$ 情况下，如何选择在一定意义下“最优”的状态序列 $Q=q_1q_2\cdots q_T$，使得该状态序列“最好地解释”观察序列？

5.6 隐马尔柯夫模型

(3) 给定一个观察序列 $O=O_1,O_2,\cdots,O_T$，如何根据最大似然估计来求模型的参数值？即如何调节模型 $\mu=(A,B,\pi)$ 的参数，使得 $P(O|\mu)$ 最大？

5.7 向前算法

- 问题1：快速计算观察序列概率 $P(O|\mu)$

(1) 在给定模型 $\mu=(A,B,\pi)$ 和观察序列 $O=O_1,O_2,\cdots,O_T$ 的情况下，计算 $P(O|\mu)$ ：

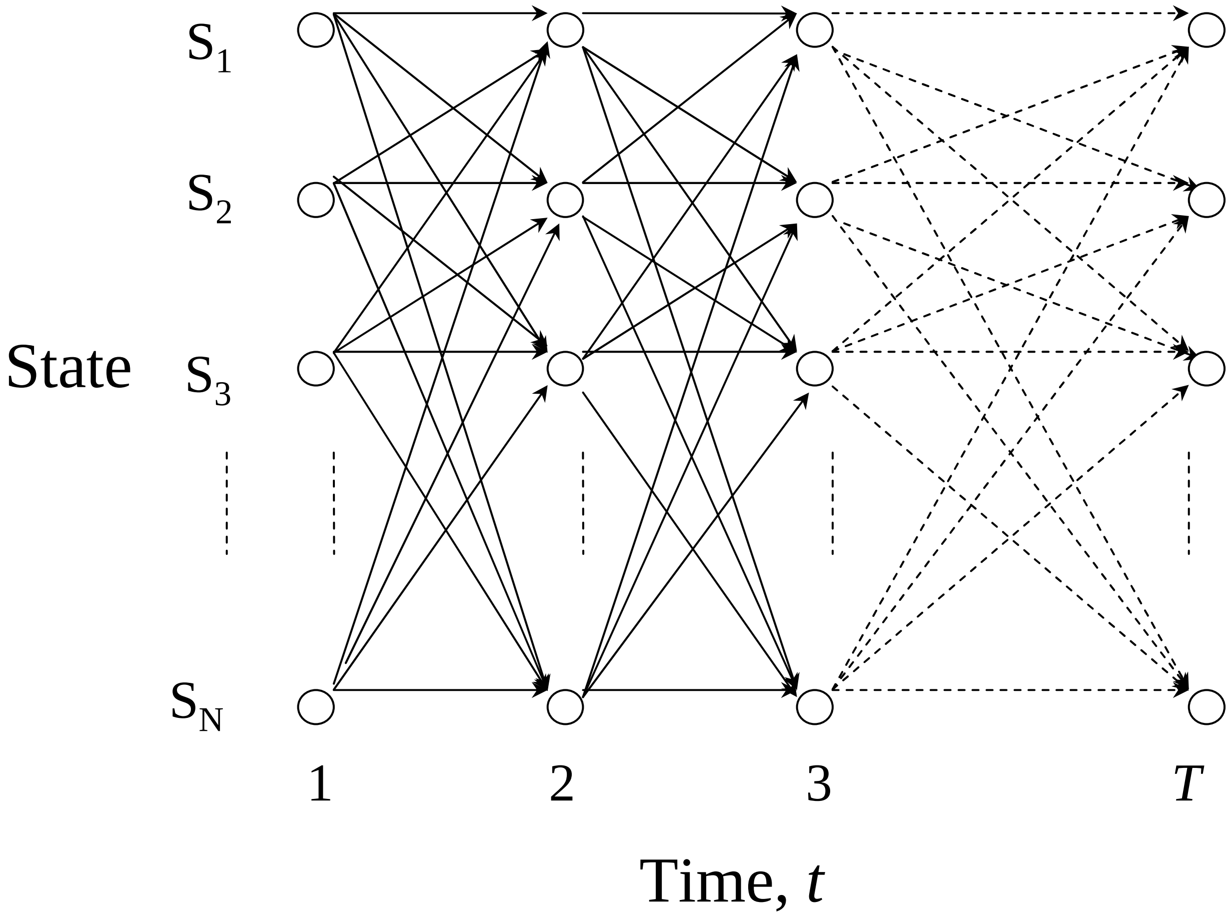
对于给定的状态序列 $Q=q_1q_2\cdots q_T$

$$P(O|\mu)=\sum_Q P(O,Q|\mu)=\sum_Q P(Q|\mu)P(O|Q,\mu) \qquad \ldots (5.21)$$

$$P(Q|\mu)=\pi_{q_1}a_{q_1q_2}a_{q_2q_3}\cdots a_{q_{t-1}q_T} \qquad \ldots (5.22)$$

$$P(O|Q,\mu)=b_{q_1}(O_1)\cdot b_{q_2}(O_2)\cdots b_{q_T}(O_T) \qquad \ldots (5.23)$$

5.7 向前算法



- <u>困难</u>：

  如果模型 $\mu=(A, B, \pi)$ 有 N 个不同的状态，时间长度为 T，那么有 N^T 个可能的状态序列，搜索路径成指数级组合爆炸。

5.7 向前算法

◆ 解决办法：动态规划，向前算法 (The forward procedure)

◆ 基本思想：定义向前变量$\alpha_t(i)$：

$$\alpha_t(i) = P(O_1 O_2 \cdots O_t, q_t = S_i \mid \mu) \qquad \ldots(5.24)$$

如果可以高效地计算$\alpha_t(i)$，就可以高效地求得 $P(O|\mu)$。

5.7 向前算法

因为 $P(O|\mu)$ 是在所有状态 q_T 下观察到序列 $O = O_1, O_2, \ldots, O_T$ 的概率：

$$P(O|\mu) = \sum_{S_i} P(O_1 O_2 \cdots O_T, q_T = S_i | \mu)$$

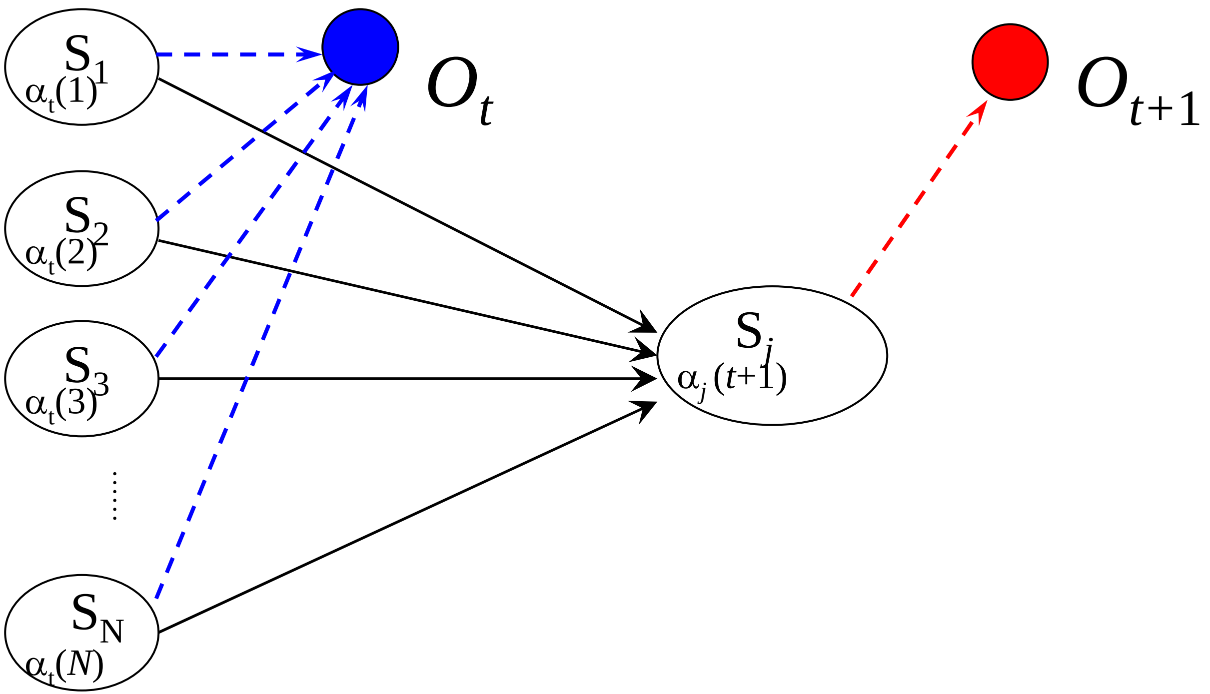
$$= \sum_{i=1}^{N} \alpha_T(i) \qquad \ldots (5.25)$$

5.7 向前算法

动态规划计算 $\alpha_t(i)$：在时间 t+1 的向前变量可以根据时间 t 的向前变量 $\alpha_t(1),\cdots,\alpha_t(N)$ 的值递推计算：

$$\alpha_{t+1}(j) = [\sum_{i=1}^{N} \alpha_t(i) a_{ij}] b_j(O_{t+1}) \qquad \ldots (5.26)$$

5.7 向前算法



t $\quad\quad\quad\quad$ $t+1$

$\alpha_t(i) \longrightarrow \alpha_{t+1}(j)$

5.7 向前算法

算法8.1：向前算法

(1) 初始化：$\alpha_1(i)=\pi_i b_i(O_1)$，$1\le i\le N$

(2) 循环计算：

$$\alpha_{t+1}(j)=[\sum_{i=1}^{N}\alpha_t(i)a_{ij}]b_j(O_{t+1}) \qquad 1\le t\le T-1$$

(3) 结束，输出：

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

5.7 向前算法

- 算法的时间复杂性：

每计算一个 $\alpha_t(i)$ 必须考虑从 t-1 时的所有 N 个状态转移到状态 s_i 的可能性，时间复杂性为 $O(N)$，对应每个时刻 t，要计算 N 个向前变量：$\alpha_t(1), \alpha_t(2), \ldots, \alpha_t(N)$，所以时间复杂性为：$O(N)\times N = O(N^2)$。又因 $t = 1, 2, \ldots, T$，所以向前算法总的复杂性为：$O(N^2T)$。

5.8 向后算法

◆ 向后算法 (The backward procedure)

定义向后变量 $\beta_t(i)$ 是在给定了模型 $\mu=(A,B,\pi)$ 和假定在时间 t 状态为 S_i 的条件下，模型输出观察序列 $O_{t+1}O_{t+2}\cdots O_T$ 的概率：

$$\beta_t(i)=P(O_{t+1}O_{t+2}\cdots O_T \mid q_t=S_i,\mu) \qquad \ldots (5.27)$$

5.8 向后算法

与向前变量一样，运用动态规划计算向后变量：

(1) 从时刻 t 到 t+1，模型由状态 S_i 转移到状态 S_j，并从 S_j 输出 O_{t+1}；

(2) 在时间 t+1，状态为 S_j 的条件下，模型输出观察序列 $O_{t+1}O_{t+2}\cdots O_T$

5.8 向后算法

第一步的概率：$a_{ij} \times b_j(O_{t+1})$

第二步的概率按向后变量的定义为：$\beta_{t+1}(j)$

于是，有归纳关系：

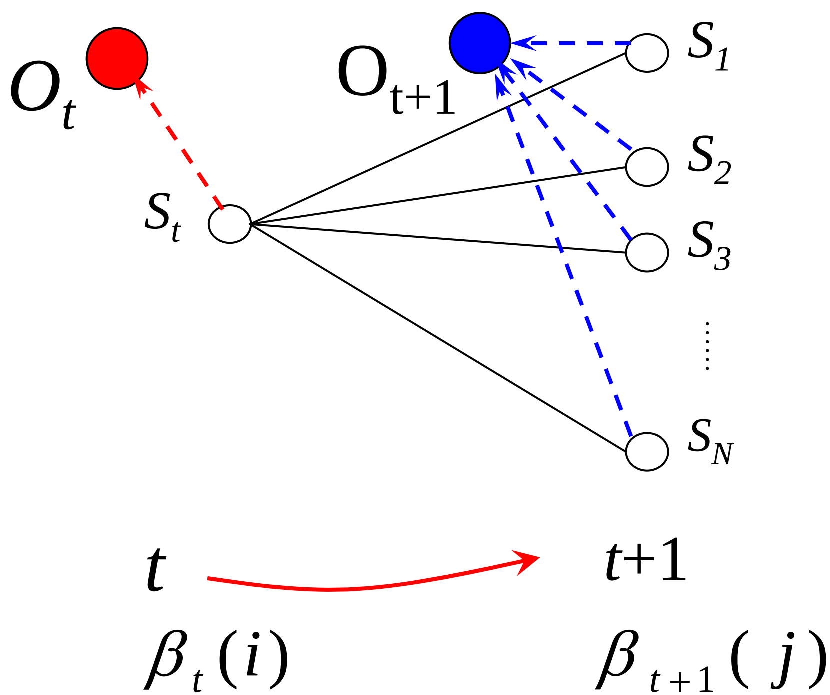
$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j) \qquad \ldots (5.28)$$

归纳顺序：$\beta_T(x), \beta_{T-1}(x), \cdots, \beta_1(x)$

(x 为 HMM的状态)

5.8 向后算法

算法的图形解释：



5.8 向后算法

◆ 算法描述：

(1) 初始化： $\beta_T(i)=1,\quad 1\le i\le N$

(2) 循环计算：

$$\beta_t(i)=\sum_{j=1}^{N} a_{ij} b_j(O_{t+1})\beta_{t+1}(j) \qquad T-1\ge t\ge 1,\ \ 1\le i\le N$$

(3) 输出结果： $P(O|\mu)=\sum_{i=1}^{N}\pi_i\beta_1(i)$

算法的时间复杂性： $O(N^2T)$

5.9 Viterbi 搜索算法

- 问题2－如何发现“最优”状态序列
  能够“最好地解释”观察序列

解释不唯一，关键是如何理解“最优”的状态序列？一种解释是：状态序列中的每个状态都单独地具有概率，即：对于每个 t ($1 \le t \le T$)，寻找 q_t 使得 $\gamma_t(i) = P(q_t = S_i \mid O, \mu)$ 最大。

5.9 Viterbi 搜索算法

$$\gamma_t(i) = P(q_t = S_i \mid O, \mu) = \frac{P(q_t = S_i, O \mid \mu)}{P(O \mid \mu)} \quad \ldots (5.29)$$

HMM 的输出序列 O，并且在时间 t 到达状态 i 的概率。

5.9 Viterbi 搜索算法

<u>分解过程</u>：

（1）HMM 在时间 t 到达状态 i, 并且输出 $O_1, O_2, \cdots, O_t$ 根据前向变量的定义，实现这一步的概率为：$\alpha_t(i)$。

（2）从时间 t，状态 S_i 出发，HMM 输出 $O_{t+1}, O_{t+2}, \cdots, O_T$，根据向后变量定义，实现这一步的概率为 $\beta_t(i)$。

于是： $P(q_t = S_i, O | \mu) = \alpha_t(i) \times \beta_t(i)$ … (5.30)

5.9 Viterbi 搜索算法

而 $P(O|\mu)$ 与时间 t 的状态无关，因此：

$$P(O|\mu)=\sum_{i=1}^{N}\alpha_t(i)\times\beta_t(i) \qquad \ldots (5.31)$$

$$\gamma_t(i)=\frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{N}\alpha_t(i)\times\beta_t(i)} \qquad \ldots (5.32)$$

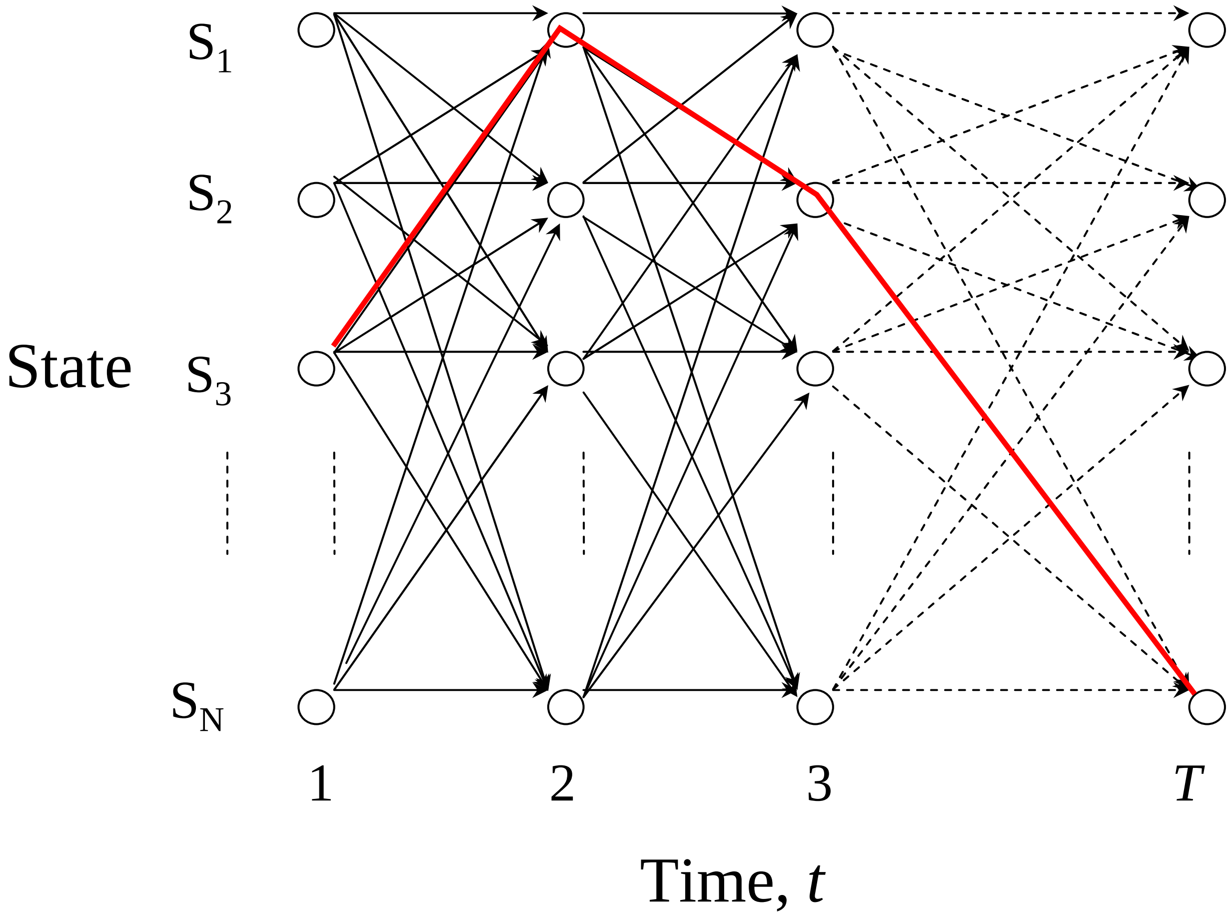
t 时刻的最优状态为：$\hat{q}_t=\arg\max_{1\le i\le N}(\gamma_t(i))$

5.9 Viterbi 搜索算法

问题：

每一个状态单独最优不一定使整体的状态序列最优，可能两个最优的状态 $\hat{q}_t$ 和 $\hat{q}_{t+1}$ 之间的转移概率为0，即 $a_{\hat{q}_t\hat{q}_{t+1}}=0$ 。

5.9 Viterbi 搜索算法



5.9 Viterbi 搜索算法

<u>另一种解释</u>：在给定模型 μ 和观察序列 O 的条件下求概率最大的状态序列：

$$\widehat{Q} = \arg\max_{Q} P(Q \mid O, \mu) \qquad \ldots (5.33)$$

<u>Viterbi algorithm</u>: 动态搜索最优状态序列。

<u>定义</u>：Viterbi 变量 $\delta_t(i)$ 是在时间 t 时，HMM 沿着某一条路径到达 S_i，并输出观察序列 $O_1, O_2, \cdots, O_t$ 的最大概率：

$$\delta_t(i) = \max_{q_1, q_2, \cdots, q_{t-1}} P(q_1, q_2, \cdots, q_t = S_i, O_1 O_2 \cdots O_t \mid \mu) \qquad \ldots (5.34)$$

5.9 Viterbi 搜索算法

递归计算： $\delta_{t+1}(i)=[\max_{j}\delta_t(j)a_{ji}]\cdot b_i(O_{t+1})$ … (5.35)

Viterbi 算法描述：

（1）初始化： $\delta_1(i)=\pi_i b_i(O_1)$， $1\le i\le N$

概率最大的路径变量： $\psi_1(i)=0$

（2）递推计算：

$$\delta_t(j)=\max_{1\le i\le N}[\delta_{t-1}(i)a_{ij}]\cdot b_j(O_t)，\quad 2\le t\le T，\quad 1\le j\le N$$

$$\psi_t(j)=\arg\max_{1\le i\le N}[\delta_{t-1}(i)a_{ij}]\cdot b_j(O_t)，\quad 2\le t\le T，\quad 1\le i\le N$$

5.9 Viterbi 搜索算法

（3）结束：

$$\widehat{Q}_T = \arg\max_{1\le i\le N}[\delta_T(i)] \qquad \widehat{P}(\widehat{Q}_T) = \max_{1\le i\le N}\delta_T(i)$$

（4）通过回溯得到路径（状态序列）：

$$\widehat{q}_t = \psi_{t+1}(\widehat{q}_{t+1}), \quad t = T-1, T-2, \cdots, 1$$

算法的时间复杂性：$O(N^2T)$

5.10 参数学习

- 问题3－模型参数学习

  给定一个观察序列 $O = O_1, O_2, \ldots, O_T$，如何根据最大似然估计来求模型的参数值？即如何调节模型 $\mu=(A, B, \pi)$ 的参数，使得$P(O|\mu)$ 最大？即估计模型中的π_i, a_{ij}, $b_j(k)$使得观察序列 O 的概率 $P(O|\mu)$ 最大。

向前向后算法（Baum-Welch or forward-backward procedure）

5.10 参数学习

如果产生观察序列 O 的状态 $Q=q_1q_2\ldots q_T$ 已知，可以用最大似然估计来计算 HMM 的参数：

$$\bar{\pi}_i = \delta(q_1, S_i)$$

$$\bar{a}_{ij} = \frac{Q\text{中从状态}q_i\text{转移到}q_j\text{的次数}}{Q\text{中所有从状态}q_i\text{转移到另一状态（包括}q_i\text{自身）的总次数}}$$

$$= \frac{\sum_{t=1}^{T-1}\delta(q_t, S_i)\times\delta(q_{t+1}, S_j)}{\sum_{t=1}^{T-1}\delta(q_t, S_i)}$$

其中，$\delta(x, y)$ 为克罗奈克(**Kronecker**)函数，当 $x=y$ 时, $\delta(x, y)=1$, 否则$\delta(x, y)=0$。v_k 是模型输出符号集中的第 k 个符号。

5.10 参数学习

类似地，

$$\overline{b}_j(k) = \frac{Q\text{中从状态 } q_j \text{输出符号 } v_k \text{的次数}}{Q\text{到达} q_j \text{的总次数}}$$

$$= \frac{\sum_{t=1}^{T} \delta\ (q_t, S_j) \times \delta(O_t, v_k)}{\sum_{t=1}^{T} \delta(q_t, S_j)} \qquad \ldots\ (5.36)$$

5.10 参数学习

期望值最大化算法（Expectation-Maximization, EM）

基本思想：初始化时随机地给模型的参数赋值(遵循限制规则，如：从某一状态出发的转移概率总和为1)，得到模型μ_0，然后可以从μ_0 得到从某一状态转移到另一状态的期望次数，然后以期望次数代替公式(5.36)中的实际次数，便可得到模型参数的新估计，由此得到新的模型μ_1，从 μ_1 又可得到模型中隐变量的期望值，由此可重新估计模型参数。循环这一过程，参数收敛于最大似然估计值。

5.10 参数学习

给定 HMM 模型 μ 和观察序列 $O=O_1, O_2, \cdots, O_T$，那么，在时间 t 位于状态 s_i，时间 t+1 位于状态 s_j 的概率：

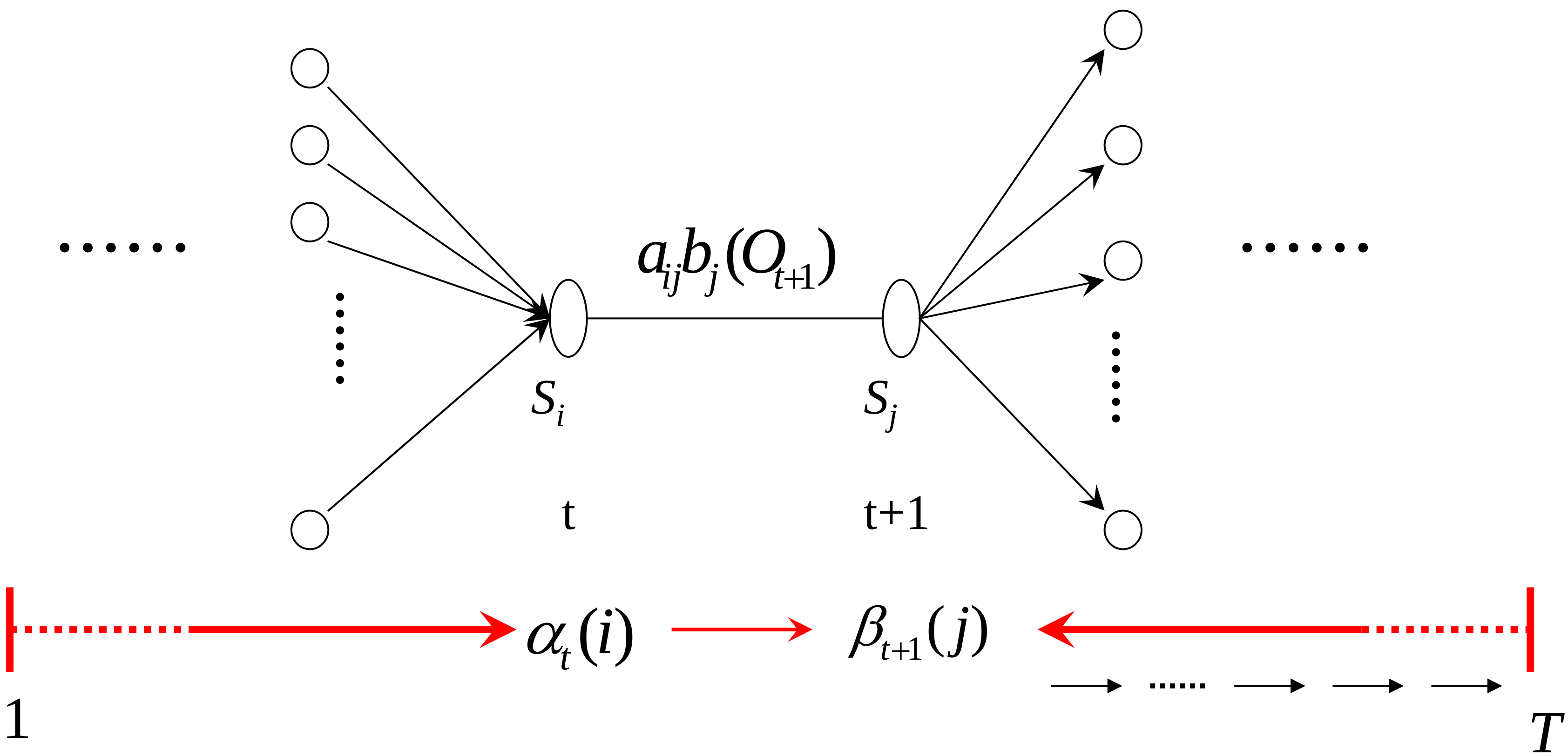
$$\xi_t(i,j) = P(q_t = S_i, q_{t+1} = S_j \mid O, \mu) = \frac{P(q_t = S_i, q_{t+1} = S_j, O \mid \mu)}{P(O \mid \mu)}$$

$$= \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{P(O \mid \mu)}$$

$$= \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)} \quad \ldots (5.37)$$

5.10 参数学习

图解搜索过程：



5.10 参数学习

那么，给定模型 μ 和观察序列 $O=O_1,O_2,\cdots,O_T$ ，在时间 t 位于状态 S_i 的概率为：

$$\gamma_t(i)=\sum_{j=1}^{N}\xi_t(i,j) \qquad \dots (5.38)$$

由此，模型 μ 的参数可由下面的公式重新估计：

(1) q_1 为 S_i 的概率：

$$\pi_i=\gamma_1(i) \qquad \dots (5.39)$$

5.10 参数学习

(2) $$\bar{a}_{ij} = \frac{Q\text{中从状态}q_i\text{转移到}q_j\text{的期望次数}}{Q\text{中所有从状态}q_i\text{转移到下一状态（包括}q_i\text{自身）的期望次数}}$$

$$= \frac{\sum_{i=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \qquad \ldots (5.40)$$

(3) $$\bar{b}_j(k) = \frac{Q\text{中从状态 } q_j\text{输出符号 } v_k\text{的期望次数}}{Q\text{到达 } q_j\text{的期望次数}}$$

$$= \frac{\sum_{t=1}^{T} \gamma_t(j) \times \delta(O_t, v_k)}{\sum_{t=1}^{T} \gamma_t(j)} \qquad \ldots (5.41)$$

5.10 参数学习

◆ **Baum-Welch** 算法描述:

(1) 初始化：随机地给 π_i, a_{ij}, $b_j(k)$ 赋值，使得

$$\begin{cases} \sum_{i=1}^{N} \pi_i = 1 & \\ \sum_{j=1}^{N} a_{ij} = 1 & 1 \le i \le N \\ \sum_{k=1}^{M} b_i(k) = 1 & 1 \le i \le N \end{cases} \qquad \ldots (5.42)$$

由此得到模型 μ_0，令 $i = 0$.

5.10 参数学习

(2) 执行 EM 算法：

i) 由模型 μ_i 根据公式 (5.37) 和 (5.38) 计算期望值 $\xi_t(i, j)$和 $\gamma_t(i)$。

ii) 用 i) 中所得到的期望值，根据公式 (5.38-5.41) 重新估计 π_i, a_{ij}, $b_j(k)$ 得到模型 μ_{i+1}。

iii) $i = i+1$，重复执行 i) 和 ii)，直至 π_i, a_{ij}, $b_j(k)$ 的值收敛：$|\log P(O|\mu_{i+1}) - \log P(O|\mu_i)| < \varepsilon$ 。

(3) 结束算法，获得相应参数。

5.10 参数学习

- 隐马尔柯夫模型使用中注意的问题
  - Viterbi 算法运算中的小数连乘，出现溢出－对数
  - Forward-Backward 算法的小数溢出－放大系数

－参阅[Rabiner and Juang, 1993: pp. 365-368]

－参阅 http://htk.eng.cam.ac.uk/

本章小结

- 语言模型和 N 元语法的定义及其应用
  - uni-gram, bi-gram, tri-gram
- 数据平滑方法：
  - 减值法：1) Good-Turing 2) Back-off 3) 绝对减值 4) 线性减值
  - 删除减值法：低阶代替高阶，参数由 Baum-Welch 算法估计
- 隐马尔柯夫模型：
  - 5个组成部分：状态数、输出符号数、… …

本章小结

- 隐马尔柯夫模型：
  - 3个问题：
    1) 快速计算给定模型的观察序列的概率
       - 向前算法或向后算法
    2) 求最优状态序列
       - Viterbi 算法
    3) HMM 中的参数估计
       - Baum-Welch (向前向后)算法
  - 模型实现中需要注意的问题：小数溢出

习题

1. 请阅读有关文献，了解除了本讲义介绍的数据平滑方法以外的其它方法；请对 Good-Turing 平滑方法进行简要的评价，阐述你个人的观点。
2. 从北大计算语言学研究所网站 (http://icl.pku.edu.cn/)上下载部分《人民日报》标注语料，用 bi-gram 实现一个汉语分词程序。
3. 下载 HTK (http://htk.eng.cam.ac.uk/)，了解相应工具的使用方法，编写程序实现一个简单的汉语音字转换程序。

Thanks

谢谢!