计算语言学

第5讲 统计语言模型

刘群
中国科学院计算技术研究所
liuqun@ict.ac.cn

中国科学院研究生院2002~2003学年第二学期课程讲义

什么是统计语言模型

- 语言模型：$P(W=w_1w_2\ldots w_n)$
- 统计语言模型实际上就是一个概率分布，它给出了一种语言中所有可能的句子的出现概率
- 在统计语言模型看来，对于一种语言，任何一个句子都是可以接受的，只是接受的可能性（概率）不同

常用的统计语言模型

- N元语法
- 隐马尔科夫模型（HMM）
- 概率上下文无关语法（PCFG）
- 概率链语法（Probabilistic Link Grammar）

N元语法－定义

- N元语法（N-gram）

$$P(w) = \prod_{i=1}^{n} P(w_i \mid w_1 w_2 ... w_{i-1})$$

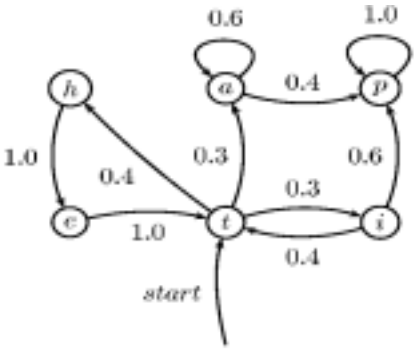
$$\approx \prod_{i=1}^{n} P(w_i \mid w_{i-N+1} w_{i-N+2} ... w_{i-1})$$

- 假设：单词w_i出现的概率只与其前面的N-1个单词有关

N元语法－举例

- N=1时：一元语法
  - 相当于词频表，给出所有词出现的频率
- N=2时：二元语法
  - 相当于一个转移矩阵，给出每一个词后面出现另一个词的概率
- N=3时：三元语法
  - 相当于一个三维转移矩阵，给出每一个词对儿后面出现另一个词的概率
- 在自然语言处理中，N元语法可以在汉字层面，也可以在单词层面，还可以在概念层面……

二元语法－图示



$P(t\text{-}i\text{-}p) = P(X_1 = t)P(X_2 = i|X_1 = t)P(X_3 = p|X_2 = i)$

$= 1.0 \times 0.3 \times 0.6 = 0.18$

袋子模型 Bag Model 1

- 将一个英语句子中所有的单词放入一个袋子中
- 用N元语法模型试图将其还原
  - 对于这些单词的任何一种排列顺序根据N元语法计算其出现概率
  - 取概率最大的排列方式

袋子模型 Bag Model 2

- 实验：取38个长度小于11个单词的英语句子，实验结果如下：

Exact reconstruction (24 of 38)

Please give me your response as soon as possible.
⇒ Please give me your response as soon as possible.

Reconstruction preserving meaning (8 of 38)

Now let me mention some of the disadvantages.
⇒ Let me mention some of the disadvantages now.

Garbage reconstruction (6 of 38)

In our organization research has two missions.
⇒ In our missions research organization has two.

代码识别问题 1

- 给出一段汉语文本，需要识别出其是GB码还是BIG5码

$$\max P(code \mid text)$$
$$= \max \frac{P(text \mid code)P(code)}{P(text)}$$
$$= \max P(text \mid code)P(code)$$
$$\approx \max P(text \mid code)$$

假设GB码的文本和BIG码的文本出现概率相同

代码识别问题 2

- 为GB码和BIG5码分别建立一元统计语言模型，也就是为两种代码分别建立字频表
- 将代码text按照GB码和BIG5码分别识别成不同的汉字串，计算其中所有汉字频率的乘积
- 算法的优点：简单、高效，通过很短的一段文本就可以识别出其代码类型

音字转换 1

- 给出一段拼音，要求转换成汉字
  pinyin＝woaini
  汉字＝我爱你、窝爱霓、我挨你……

$$\max P(text \mid pinyin)$$
$$= \max \frac{P(pinyin \mid text)P(text)}{P(pinyin)}$$
$$= \max P(pinyin \mid text)P(text)$$
$$= \max P(text)$$

不考虑同音字，即认为*P(pinyin|text)*为常量

音字转换 2

- 一元语法：
  - 空间：n（汉字总数）
  - 同音字中总是会选择最高频的汉字，不合适
- 二元语法：
  - 空间：n^2
  - 效果比一元语法有较大提高
- 估计对于汉语而言四元语法效果较好
- 实用系统：智能狂拼，微软拼音

N元语法的参数估计

- 最大似然估计

$$p(w_n \mid w_1 \ldots w_{n-1}) = \frac{f(w_1 \ldots w_n)}{f(w_1 \ldots w_{n-1})}$$

用实际样本中事件出现的频率来估计该事件的概率

数据平滑

- 数据稀疏问题
  - 如果$f(w_1 \ldots w_n)=0$，那么出现零概率，导致整个文本的出现概率为零
- 解决办法：劫富济贫
- 约束：概率的归一性

$$\sum_{w_n} p(w_n \mid w_1 \ldots w_{n-1}) = 1$$

平滑算法－加1法

- 给每个事件出现的词数加1

$$p(w_n \mid w_1 \ldots w_{n-1}) = \frac{f(w_1 \ldots w_n) + 1}{f(w_1 \ldots w_{n-1}) + m}$$

m为单词的个数

- 会导致未出现的事件概率过高
- 可以将1改成一个更小的常数ε

平滑算法－Good-Turing法

- 样本数据的大小为N
- 样本中出现r次的样本数为n_r
- 估计：

$$P_r = \begin{cases} \frac{r^*}{N}, \text{当} r > 0, \mathrm{r}^* = (r+1)\frac{n_{r+1}}{n_r} \\ \frac{n_1}{N}, \text{当} \mathrm{r} = 0 \end{cases}$$

平滑算法－绝对减值法

- 绝对减值法

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

K为所有可能的事件数目

- 参数b的上限 $b \le \dfrac{n_1}{n_1 + 2n_2}$

平滑算法－线性减值法

- 线性减值法

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

自由参数α的优化值为：n_1/N

平滑算法－回退（Back-off）法

当某一事件的频率小于K时，用n-1元语法来代替n元语法

$$P(x_n \mid x_1 \ldots x_{n-1}) = \begin{cases} (1-\alpha(f(x_1 \ldots x_n)))\dfrac{f(x_1 \ldots x_n)}{f(x_1 \ldots x_{n-1})}, \text{当} f(x_1 \ldots x_n) > K \\ \alpha(f(x_1 \ldots x_{n-1}))P(x_n \mid x_2 \ldots x_{n-1}), \text{当} f(x_1 \ldots x_n) \le K \end{cases}$$

α是归一化因子

平滑算法－删除插值法（Deleted Interpolation）

- 用低阶的语法来估计高阶的语法

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

- 参数$\lambda_1, \lambda_2, \lambda_3$的估计
  - 将训练语料分成两部分
  - 第一部分用于估计P'（不做平滑）
  - 第二部分语料用于估计参数$\lambda_1, \lambda_2, \lambda_3$，使得其出现的概率最大

隐马尔科夫模型－假设

对于一个随机事件，有一个观察值序列：$O_1,...,O_T$

该事件隐含着一个状态序列：$X_1,...,X_T$

假设1：马尔可夫假设（状态构成一阶马尔可夫链）

$p(X_i|X_{i-1}\ldots X_1) = p(X_i|X_{i-1})$

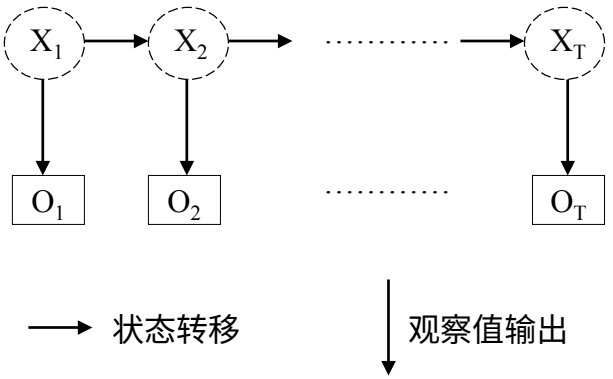
假设2：不动性假设（状态与具体时间无关）

$p(X_{i+1}|X_i) = p(X_{j+1}|X_j)$，对任意i,j成立

假设3：输出独立性假设（输出仅与当前状态有关）

$p(O_1,...,O_T \mid X_1,...,X_T) = \Pi\, p(O_t \mid X_t)$

隐马尔科夫模型－图示



隐马尔科夫模型－定义

一个隐马尔可夫模型 (HMM) 是一个五元组：

$$(\Omega_X, \Omega_O, A, B, \pi)$$

其中：

$\Omega_X = \{q_1,...q_N\}$：状态的有限集合

$\Omega_O = \{v_1,...,v_M\}$：观察值的有限集合

$A = \{a_{ij}\}$，$a_{ij} = p(X_{t+1} = q_j \mid X_t = q_i)$：转移概率

$B = \{b_{ik}\}$，$b_{ik} = p(O_t = v_k \mid X_t = q_i)$：输出概率

$\pi = \{\pi_i\}$，$\pi_i = p(X_1 = q_i)$：初始状态分布

隐马尔科夫模型－问题

令 $\lambda = \{A,B,\pi\}$ 为给定HMM的参数，

令 $\sigma = O_1,...,O_T$ 为观察值序列，

隐马尔可夫模型（HMM）的三个基本问题：

1. 评估问题：对于给定模型，求某个观察值序列的概率$p(\sigma|\lambda)$；（语言模型）
2. 解码问题：对于给定模型和观察值序列，求可能性最大的状态序列；
3. 学习问题：对于给定的一个观察值序列，调整参数λ，使得观察值出现的概率$p(\sigma|\lambda)$最大。

隐马尔科夫模型－算法

- 评估问题：向前算法
  - 定义向前变量
  - 采用动态规划算法，复杂度$O(N^2T)$
- 解码问题：韦特比（Viterbi）算法
  - 采用动态规划算法，复杂度$O(N^2T)$
- 学习问题：向前向后算法
  - EM算法

隐马尔科夫模型－例子

- 假设：某一时刻只有一种疾病，且只依赖于上一时刻疾病
  一种疾病只有一种症状，且只依赖于当时的疾病
- 症状(观察值)：发烧，咳嗽，咽喉肿痛，流涕
- 疾病(状态值)：感冒，肺炎，扁桃体炎
- 转移概率：从一种疾病转变到另一种疾病的概率
- 输出概率：某一疾病呈现出某一症状的概率
- 初始分布：初始疾病的概率
- 解码问题：某人症状为：咳嗽→咽喉痛→流涕→发烧
  请问：其疾病转化的最大可能性如何？

隐马尔科夫模型－例子（续）

- 转移概率

	感冒	肺炎	扁桃体炎
感冒	0.4	0.3	0.3
肺炎	0.2	0.6	0.2
扁桃体炎	0.1	0.1	0.8

- 输出概率

	发烧	咳嗽	咽喉痛	流涕
感冒	0.4	0.3	0.1	0.2
肺炎	0.3	0.5	0.1	0.1
扁桃体炎	0.2	0.1	0.6	0.1

- 初始分布

感冒	肺炎	扁桃体炎
0.5	0.2	0.3

HMM评估问题

评估问题：对于给定模型，求某个观察值序列的概率$p(\sigma|\lambda)$；（语言模型）

$$
\begin{aligned}
P(O \mid \lambda) &= \sum_X P(O, X \mid \lambda) \\
&= \sum_X P(X \mid \lambda) P(O \mid X, \lambda) \\
&= \sum_X (\pi_{X_1} \prod_{i=2}^{T} a_{X_{i-1} X_i}) (\prod_{i=1}^{T} b_{X_i O_i}) \\
&= \sum_X (\pi_{X_1} b_{X_1 O_1} \prod_{i=2}^{T} a_{X_{i-1} X_i} b_{X_i O_i})
\end{aligned}
$$

可能的状态序列有N^T种可能性，计算复杂度极高

HMM评估问题－向前算法 1

定义前向变量为HMM在时间t输出序列$O_1 \ldots O_t$，并且位于状态X_t的概率：

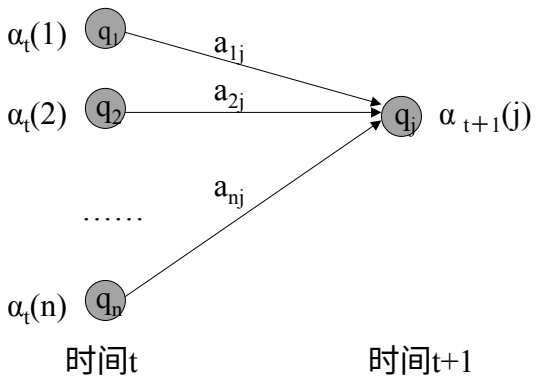
$$\alpha_t(i) = P(O_1 \cdots O_t, X_t = q_i \mid \lambda)$$

初始化：$\alpha_1(i) = \pi_i b_{iO_1}$

迭代公式为：$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_{jO_{t+1}}$

最终结果为：$P(O \mid \lambda) = \sum_{i=1}^{N} \alpha_T(i)$

HMM评估问题－向前算法 2



向前算法的时间复杂度：$O(N^2T)$

HMM评估问题－向后算法 1

定义后向变量为HMM在时间t并且位于状态X_t的情况下，输出序列$O_{t+1}\cdots O_T$，：

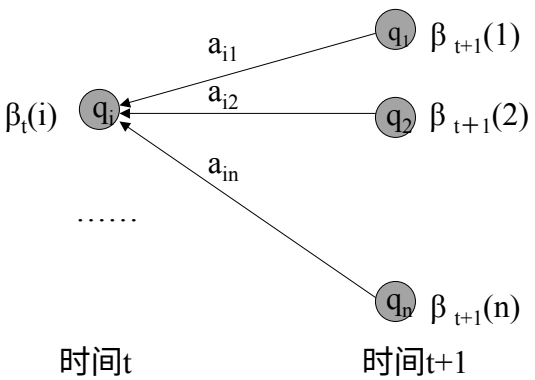
$$\beta_t(i) = P(O_{t+1}\cdots O_T, X_t = q_i \mid \lambda)$$

初始化：　$\beta_T(i) = 1$

迭代公式为：　$\beta_t(i) = \sum_{i=1}^{N}\left[a_{ij} b_{jO_{t+1}} \beta_{t+1}(j)\right]$

最终结果为：　$P(O \mid \lambda) = \sum_{i=1}^{N} \pi_i \beta_1(i)$

HMM评估问题－向后算法 2



向后算法的时间复杂度：$O(N^2T)$

HMM解码问题

解码问题：对于给定模型λ和观察值序列O，
求可能性最大的状态序列X；

$$X^* = \arg\max_X P(X \mid O, \lambda)$$

如果要枚举所有的状态序列，时间复杂度是$O(N^T)$

HMM解码问题－Viterbi算法 1

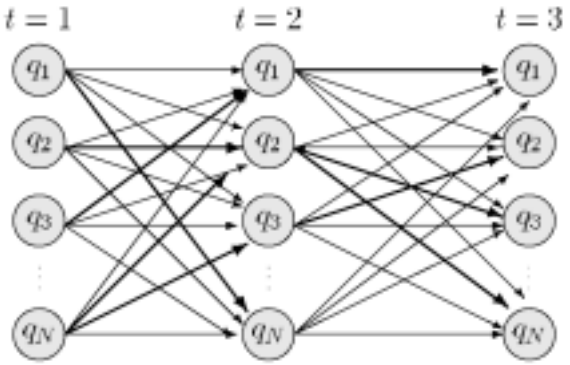
定义Viterbi变量为HMM在时间t沿着某一条路径到达状态q_i，且输出观察值$O_1O_2\ldots O_i$的最大概率

$$\delta_t(i) = \max_{X_1X_2\ldots X_{i-1}} P(X_1X_2\ldots X_t = q_i, O_1O_2\ldots O_t \mid \lambda)$$

HMM解码问题－Viterbi算法 2

- 初始化 $\delta_1(i) = \pi_i b_{iO_1}$
- 迭代计算 $\delta_t(j) = \max_{1\le i\le N}[\delta_{t-1}(i)a_{ij}]b_{jO_t}$
  $2\le t\le T$
  $1\le j\le N$
  $\Psi_t(j) = \arg\max_{1\le i\le N}[\delta_{t-1}(i)a_{ij}]b_{jO_t}$
- 取最优 $p^* = \max_{1\le i\le N}[\delta_T(i)]$
  $q_T^* = \arg\max_{1\le i\le N}[\delta_T(i)]$
- 路径回溯 $q_t^* = \Psi_{t+1}(q_{t+1}^*)$
  $2\le t\le T$

HMM解码问题－Viterbi算法 3



Viterbi算法的时间复杂度：$O(N^2T)$

HMM学习问题

- 已知观察序列O=$O_1O_2\ldots O_T$
- 估计λ的参数：π_i, a_{ij}, b_{ik}

HMM学习问题－最大似然估计

已知观察序列O对应的状态序列为（有指导学习）：

$X=X_1X_2\ldots X_T$

采用最大似然估计：

$$\overline{\pi}_i = \delta(X_1, q_i)$$

其中 $\delta(x,y)=\begin{cases}1, \text{如果} x=y\\ 0, \text{如果} x\neq y\end{cases}$

$$\overline{a}_{ij} = \frac{X\text{中从状态}q_i\text{转移到状态}q_j\text{的次数}}{X\text{中从状态}q_i\text{转移另一状态（含}q_j\text{）的次数}} = \frac{\sum_{t=1}^{T-1}\delta(X_t,q_i)\times\delta(X_{t+1},q_j)}{\sum_{t=1}^{T-1}\delta(X_t,q_i)}$$

$$\overline{b}_{jk} = \frac{X\text{中从状态}q_j\text{输出到状态}v_k\text{的次数}}{X\text{中到达状态}q_j\text{次数}} = \frac{\sum_{t=1}^{T}\delta(X_t,q_j)\times\delta(O_t,v_k)}{\sum_{t=1}^{T}\delta(X_t,q_j)}$$

HMM学习问题－Baum-Welch算法 1

- 不知道O对应的状态序列：无指导学习
- 采用Baum-Welch（又称向前向后算法）
- 是EM（Expectation-Maximization)算法的一种实现
  - 初试化：λ_0
  - EM步骤：循环执行以下步骤，直到λ_i收敛
    - E-步骤：根据λ_i计算所有可能的状态序列
    - M-步骤：根据状态序列和输出序列估计参数λ_{i+1}

HMM学习问题－Baum-Welch算法 2

- 初试化：随机给 π_i, a_{ij}, b_{jk} 赋初始值
  需要满足以下归一化约束：

$$\sum_{i=1}^{N} \pi_i = 1$$

$$\sum_{j=1}^{N} a_{ij} = 1, \text{对于 } 1 \le i \le N$$

$$\sum_{k=1}^{N} b_{jk} = 1, \text{对于 } 1 \le j \le N$$

HMM学习问题－Baum-Welch算法 3

E-步骤：已知观察序列$O_1O_2\ldots O_T$和模型参数 π_i, a_{ij}, b_{jk}，估计：

- 在时间t和t+1分别位于状态q_i，q_j的概率：

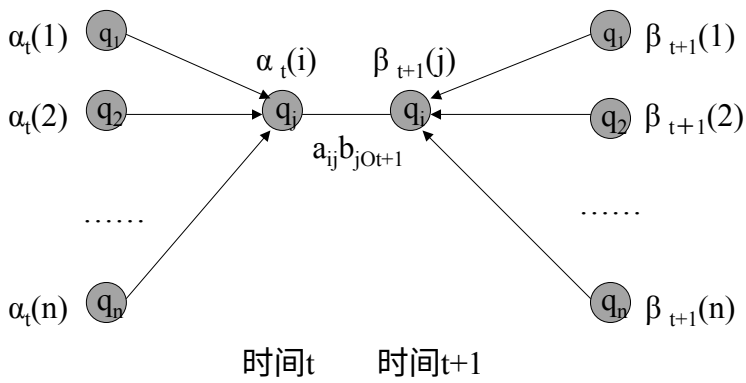
$$\xi_t(i,j) = \frac{P(X_t = q_i, X_{t+1} = q_j, O \mid \lambda)}{P(O \mid \lambda)}$$
$$= \frac{\alpha_t(i) a_{ij} b_{j O_{t+1}} \beta_{t+1}(j)}{P(O \mid \lambda)}$$
$$= \frac{\alpha_t(i) a_{ij} b_{j O_{t+1}} \beta_{t+1}(j)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i) a_{ij} b_{j O_{t+1}} \beta_{t+1}(j)}$$

- 在时间t位于状态q_i的概率：

$$\gamma_i = \sum_{j=1}^{N} \xi_t(i,j)$$

HMM学习问题－Baum-Welch算法 4

$\xi_t(i,j)$ 的计算使用了前向变量和后向变量：



HMM学习问题－Baum-Welch算法 5

- M-步骤：已知$\xi_t(i,j)$和$\gamma_t(i)$，估计模型λ：

$$\bar{\pi}_i = X_1\text{为}q_i\text{的概率} = \gamma_1(i)$$

$$\bar{a}_{ij} = \frac{X\text{中从状态}q_i\text{转移到状态}q_j\text{的次数}}{X\text{中从状态}q_i\text{转移另一状态（含}q_j\text{）的次数}} = \frac{\sum_{t=1}^{T-1}\xi_t(i,j)}{\sum_{t=1}^{T-1}\gamma(i)}$$

$$\bar{b}_{jk} = \frac{X\text{中从状态}q_j\text{输出到状态}v_k\text{的次数}}{X\text{中到达状态}q_j\text{次数}} = \frac{\sum_{t=1}^{T}\gamma(j)\times\delta(O_t,v_k)}{\sum_{t=1}^{T}\gamma(j)}$$

HMM学习问题－Baum-Welch算法 5

迭代终止条件：

$$\left|\log P(O\mid\lambda_{i+1}) - \log P(O\mid\lambda_i)\right| < \varepsilon$$

ε是事先给定的阈值

HMM学习问题－Baum-Welch算法 6

- Baum-Welch算法只能达到局部最优
- Baum-Welch算法的结果依赖于初始值的设定

隐马尔科夫模型－应用

- 语音识别
- 音字转换
- 词性标注（POS Tagging）
- 组块分析
- 基因分析

隐马尔科夫模型－资源

- Rabiner, L. R., A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, *Proceedings of the IEEE*, vol. 77, no. 2, Feb. 1989, pgs 257 - 285. There is a lot of notation but verbose explanations accompany.
- HTK：HMM Toolkit
- Hidden Markov Model (HMM) White Paper (GeneMatcher)
- ……

隐马尔科夫模型－总结

- HMM模型可以看作一种特定的Bayes Net
- HMM模型等价于概率正规语法或概率有限状态自动机
- HMM模型可以用一种特定的神经网络模型来模拟
- 优点：研究透彻，算法成熟，效率高，效果好，易于训练

复习思考题

- 做一个代码识别程序，能够识别以下种类的代码（给出系统详细设计及参数训练方法）：
  GBK，BIG5，Unicode简体，Unicode繁体，HZ码，
  UTF-7简体，UTF-7繁体，UTF-8简体，UTF-8繁体，
  Unicode日语，Unicode法语，Unicode德语，……
- 计算疾病模型的解码问题：假设某人症状为：咳嗽→咽喉痛→流涕→咽喉痛→发烧→发烧，请计算其疾病转化的最大可能性如何？
- 下载HTK，利用HTK实现一个拼音汉字转换程序
- 下载《人民日报》语料库，用汉语的二元语法实现一个袋子模型