NLP 期末总结

王若琪

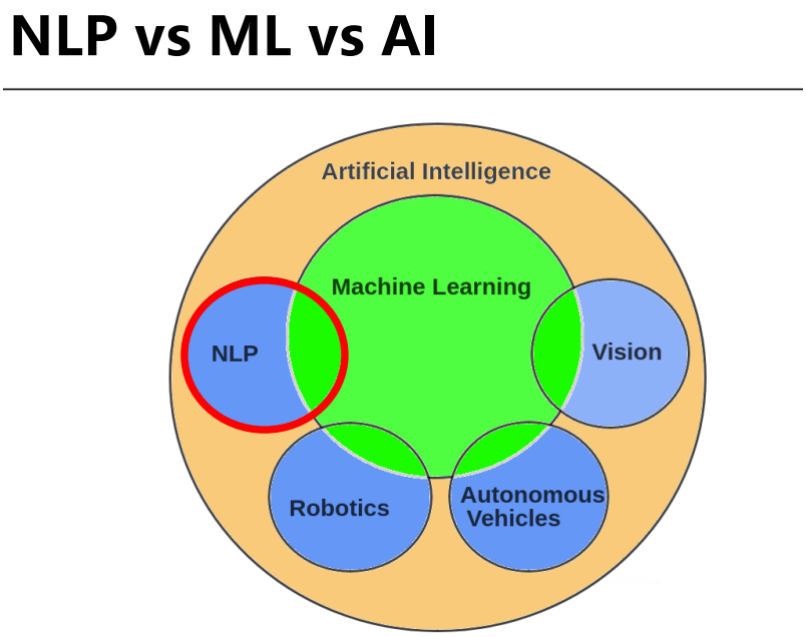
第 1 讲

概念

自然语言处理（自然语言理解），是计算机科学与人工智能领域中的一个重要方向。它研究能实现人与计算机之间通过自然语言进行交互的各种理论和方法。

让计算机能够自动或半自动地理解自然语言文本， 懂得人的意图和心声
让计算机实现海量语言文本的自动处理、挖掘和有效利用，满足不同用户的各种需求，实现个性化信息服务

自然语言处理又叫做计算语言学(computational linguistics)，涉及到计算、语言两方面的知识。



自然语言处理的研究方向包括：
• 中文自动分词
• 句法分析
• 信息抽取
• 情感计算
• 机器翻译
• 对话系统
• 信息检索
• 自动摘要

为什么NLP

语言是思维的载体， 是人类交流思想、 表达情感最自然、最直接、最方便的工具
人类历史上以语言文字形式记载和流传的知识占知识总量的80%以上
2008年1月中国互联网络信息中心(CNNIC) 发布的《第21次中国互联网络发展状况统计报告》 表明，中国互联网上有87.8%的网页内容是文本表示的

关于“理解”的标准

如何判断计算机系统的智能?
——计算机系统的表现(act)如何？反应(react)如何？相互作用(interact)如何？与有意识的个体（人）比较如何?

图灵测试

图灵测试（The Turing test）由艾伦·麦席森·图灵发明，指测试者与被测试者（一个人和一台机器）隔开的情况下，通过一些装置（如键盘）向被测试者随意提问。进行多次测试后，如果机器让平均每个参与者做出超过30%的误判，那么这台机器就通过了测试，并被认为具有人类智能。

课程目标

- 系统地学习自然语言处理（特别是中文语言）的基本概念、常用算法和重要应用。
- 重点掌握词汇、句法、语义分析等的基本知识，理解统计自然语言处理的关键算法；
- 在大规模语料库的支持下，掌握统计语言模型在语言知识自动学习中的应用；
- 结合文本分类和聚类、机器翻译、信息检索等热门应用技术；
- 了解深度学习技术在自然语言处理上的应用和进展.

基本问题

一：形态学 (Morphology)
• 研究词(word) 由有意义的基本单位 - 词素的构成问题。
• 单词的识别/ 汉语的分词问题。

二： 句法 (Syntax) 问题
• 研究句子结构成分之间的相互关系和组成句子序列的规则
• 为什么一句话可以这么说也可以那么说？ 如何建立快速有效的句子结构分析方法?

三： 语义(Semantics) 问题
• 研究如何从一个语句中推导出词的意义， 以及这些词在该语句句法结构中的作用来推导出该语句的意义。

四：语用学(Pragmatics) 问题
• 研究在不同上下文中语句的应用， 以及上下文对语句理解所产生的影响。

主要困难

一：大量歧义(ambiguity)现象
I. 词法歧义 II. 词性歧义 III. 结构歧义 IV. 语义歧义 V. 语音歧义

二：大量未知语言现象
❖ 新词、人名、 地名、 术语等 ❖ 新含义 ❖ 新用法和新句型等，尤其在口语中或部分网络语言中，不断出现一些“非规范的”新的语句结构

第二讲 数学和信息论基础

概率论基础

基本概念 概率(probability) 极大似然估计(maximum likelihood estimation) 条件概率(conditional probability) 全概率公式(full probability) 贝叶斯法则(Bayes' theorem) 二项式分布(binomial distribution) 期望(expectation) 方差(variance)

极大似然估计 MLE

一个试验的样本空间是$\{s_1, s_2, \ldots, s_n\}$,在相同情况下重复试验 N 次，观察到样本$s_k (1 \le k \le n)$的次数为：$n_N(s_k)$，则$s_k$的相对频率为：

$$q_N(s_k) = \frac{n_N(s_k)}{N},$$

$$\because \sum_{k=1}^{n} n_N(s_k) = N, \quad \therefore \sum_{k=1}^{n} q_N(s_k) = 1$$

当 N 越来越大时，相对频率$q_N(s_k)$就越来越接近s_k的概率$P(s_k)$：$\lim_{N\to\infty} q_N(s_k) = P(s_k)$，因此相对频率常被用作概率的估计值，这种估计方法称为最大似然估计。

条件概率

如果 A 和 B 是样本空间 Ω 上的两个事件，$P(B) > 0$，那么在给定 B 时 A 的条件概率 $P(A \mid B)$ 为：

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)},$$

一般地，$P(A \mid B) \neq P(A)$。

全概率公式

设Ω为实验的样本空间，$B_1, B_2, \ldots, B_n$为Ω的一组两两互斥的事件，且每次试验中至少发生一个，则称$B_1, B_2, \ldots, B_n$为样本空间Ω的一个划分。

如果A为样本空间Ω的事件，$B_1, B_2, \ldots, B_n$为样本空间Ω的一个划分，且$P(B_i) > 0$（$i = 1, 2, \ldots, n$），则全概率公式为：

$$P(A) = \sum_{i=1}^{n} P(AB_i) = \sum_{i=1}^{n} P(B_i) P(A \mid B_i)$$

贝叶斯定理

如果 A 为样本空间 Ω 的事件，$B_1, B_2, \ldots, B_n$ 为样本空间 Ω 的一个划分，且 $P(A) > 0$，$P(B_i) > 0$（$i = 1, 2, \ldots, n$），则：

$$P(B_i \mid A) = \frac{P(B_i)P(A \mid B_i)}{\sum_{j=1}^{n} P(B_j)P(A \mid B_j)},$$

当 $n = 1$ 时，$P(B \mid A) = \frac{P(A \mid B)P(B)}{P(A)}$

先验概率、后验概率

先验概率(Prior probability):不考虑先决条件（信息或者知识） 而得到的该事件的概率：一般在试验前已知， 常常是以往经验的总结

后验概率(Posterior probability):在具备该事件出现的信息或者知识的条件下得到的该事件的概率：反映了试验之后对各种原因发生的可能性大小的新知识

假设某一种特殊的句法结构很少出现，平均大约每100,000个句子中才可能出现一次。我们开发了一个程序来判断某个句子中是否存在这种特殊的句法结构。如果句子中确实含有该特殊句法结构时，程序判断结果为“存在” 的概率为0.95。如果句子中实际上不存在该句法结构时，程序错误地判断为“存在” 的概率为0.005。 那么，这个程序测得句子含有该特殊句法结构的结论是正确的概率有多大?

假设 G 表示事件“句子确实存在该特殊句法结构”，T 表示事件“程序判断的结论是存在该特殊句法结构”。那么：

$$P(G) = \frac{1}{100000} = 0.00001, \quad P(\overline{G}) = \frac{100000 - 1}{100000} = 0.99999,$$

$$P(T \mid G) = 0.95, \quad P(T \mid \overline{G}) = 0.005$$

求解： $P(G \mid T) = ?$

$$P(G \mid T) = \frac{P(T \mid G)P(G)}{P(T \mid G)P(G) + P(T \mid \overline{G})P(\overline{G})}$$

$$= \frac{0.95 \times 0.00001}{0.95 \times 0.00001 + 0.005 \times 0.99999} \approx 0.002$$

二项式分布(binomial distribution)

当重复一个只有两种输出（假定为 $\overline{A}$ 和 A）的试验（伯努利试验），A 在一次实验中发生的概率为 p，现将实验独立地重复 n 次，如果用 X 表示 A 在这 n 次实验中发生的次数，那么，$X = 0, 1, \cdots, n$。则 n 次独立实验中成功的次数为 r 的概率为：$p_r = C_n^r p^r (1-p)^{n-r}$，其中，$C_n^r = \frac{n!}{(n-r)!r!}$，$0 \le r \le n$。此时 X 所遵从的概率分布称为二项式分布，并记为：$X \sim B(n, p)$。

期望

期望值是一个随机变量所取值的概率平均。设 X 为一随机变量，其分布为 $P(X=x_k)=p_k,\quad k=1,2,\ldots,$

若级数 $\sum_{k=1}^{\infty}x_k p_k$ 绝对收敛，那么随机变量 X 的数学期望或概率平均值为：$E(X)=\sum_{k=1}^{\infty}x_k p_k$ 。

方差(Variance)

一个随机变量的方差描述的是该随机变量的值偏离其期望值的程度。设 X 为一随机变量，其方差为：

$$Var(X)=E((X-E(X))^2)=E(X^2)-E^2(X)$$

信息论基础

熵(entropy)

❖如果 X 是一个离散型随机变量，其概率分布为：$p(x)=P(X=x),\ x\in X$。X 的熵 $H(X)$ 为：

$$H(X)=-\sum_{x\in X}p(x)\log_2 p(x) \qquad (1)$$

其中，约定 $0\log 0=0$

通常熵的单位为二进制位比特(bit)

熵又称为自信息(self-information)， 表示信源X 每发一个符号所提供的平均信息量。 **熵也可以被视为描述一个随机变量的不确定性的数量。一个随机变量的熵越大， 它的不确定性越大。 那么， 正确估计其值的可能性就越小。** 越不确定的随机变量越需要大的信息量用以确定其值。

例2-1： 计算下列两种情况下英文(26个字母和1个空格，共27个字符)信息源的熵：
1) 假设27个字符等概率出现；
2) 假设英文字母的概率分布如下：

解：（1）等概率出现情况：

$$H(X)=-\sum_{x\in X}p(x)\log_2 p(x)$$
$$=27\times\{-\frac{1}{27}\log_2\frac{1}{27}\}=\log_2 27=4.75\quad(\text{bits})$$

（2）实际情况：

$$H(X)=-\sum_{i=1}^{27}p(x_i)\log_2 p(x_i)=4.02\quad(\text{bits})$$

说明：考虑了英文字母和空格实际出现的概率后， 英文信源的平均不确定性，比把字母和空格看作等概率出现时英文信源的平均不确定性要小。

联合熵(joint entropy)

如果 X, Y 是一对离散型随机变量 $X, Y\sim p(x, y)$， X, Y 的联合熵 $H(X, Y)$ 为：

$$H(X,Y)=-\sum_{x\in X}\sum_{y\in Y}p(x,y)\log_2 p(x,y)\qquad(2)$$

联合熵是描述一对随机变量平均所需要的信息量 。

条件熵(conditional entropy)

给定随机变量 X，随机变量 Y 的条件熵定义为：

$$H(Y\mid X)=\sum_{x\in X}p(x)H(Y\mid X=x)$$
$$=\sum_{x\in X}p(x)[-\sum_{y\in Y}p(y\mid x)\log_2 p(y\mid x)]$$
$$=-\sum_{x\in X}\sum_{y\in Y}p(x,y)\log_2 p(y\mid x)\qquad(3)$$

相对熵(relative entropy, 或称 Kullback-Leibler divergence, KL 距离)

两个概率分布 $p(x)$ 和 $q(x)$ 的相对熵定义为：

$$D(p\parallel q)=\sum_{x\in X}p(x)\log\frac{p(x)}{q(x)}\qquad(4)$$

该定义中约定 $0\log(0/q)=0, p\log(p/0)=\infty$

相对熵常被用以衡量两个随机分布的差距。当两个随机分布相同时， 其相对熵为0。 当两个随机分布的差别增加时， 其相对熵也增加。

交叉熵(cross entropy)

如果一个随机变量 $X \sim p(x)$，$q(x)$ 为用于近似 $p(x)$ 的概率分布，那么，随机变量 X 和模型 q 之间的交叉熵定义为：

$$H(X,q) = H(X) + D(p \| q)$$
$$= -\sum_{x} p(x) \log q(x) \quad (5)$$

交叉熵用以衡量估计模型与真实概率分布之间的差异 。

互信息

如果 $(X, Y) \sim p(x, y)$，X, Y 之间的互信息 $I(X; Y)$ 定义为：

$$I(X; Y) = H(X) - H(X|Y) \quad (6)$$

根据 $H(X)$ 和 $H(X|Y)$ 的定义：

$$H(X) = -\sum_{x \in X} p(x) \log_2 p(x)$$
$$H(X|Y) = -\sum_{x \in X} \sum_{y \in Y} p(x,y) \log_2 p(x|y)$$

$$I(X;Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \log_2 \frac{p(x,y)}{p(x)p(y)}$$

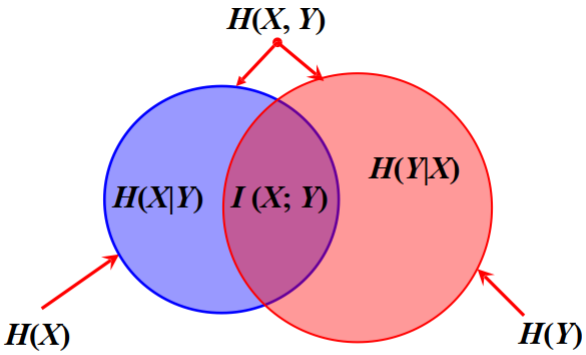


图. 互信息、条件熵与联合熵

在汉语分词问题中，用互信息来估计两个汉字的结合的强度，互信息值越大，表示两个汉字之间的结合越紧密，越可能成词。反之， 断开的可能性越大 。

应用实例

例一 词汇歧义消解

任何一种自然语言中，一词多义（歧义） 现象是普遍存在的。 如何区分不同上下文中的词汇语义， 就是词汇歧义消解问题，或称词义消歧(word sense disambiguation) .词义消歧是自然语言处理中的基本问题之一.

解决思路
每个词表达不同的含意时其上下文（语境）往往不同，也就是说，不同的词义对应不同的上下文， 因此，如果能够将多义词的上下文区别开， 其词义自然就明确了。 （基本的上下文信息：词、词性、位置 ）

- 基于上下文分类的消歧方法

(1) 基于贝叶斯分类器 (Gale *et al.*, 1992)

数学描述:

假设某个多义词 w 所处的上下文语境为 C,如果 w 的多个语义记作 s_i, 那么, 可通过计算 $\arg\max p(s_i|C)$ 确定 w 的词义.

- ➢消歧算法描述:

a) 对于多义词 w 的每个语义 s_i 执行如下循环: 对于词典中所有的词 v_k 利用训练语料计算

$$p(v_k|s_i)=\frac{N(v_k,s_i)}{N(s_i)}$$

b) 对于 w 的每个语义 s_i 计算:

$$p(s_i)=\frac{N(s_i)}{N(w)}$$

对于 w 的每个语义 s_i 计算 $p(s_i)$, 并根据上下文中的每个词 v_k 计算 $p(w|s_i)$, 选择:

$$\hat{s}_i=\arg\max_{s_i}\left[p(s_i)\prod_{v_k\in C}p(v_k|s_i)\right]$$

称测试过程或标注过程

说明: 在实际算法实现中, 通常将概率 $p(v_k|s_i)$ 和 $p(s_i)$ 的乘积运算转换为对数加法运算:

$$\hat{s_i}=\arg\max_{s_i}\left[\log p(s_i)+\sum_{v_k\in C}\log p(v_k|s_i)\right]$$

说明: 在实际算法实现中, 通常将概率 $p(v_k|s_i)$ 和 $p(s_i)$ 的乘积运算转换为对数加法运算:

$$\hat{s_i}=\arg\max_{s_i}\left[\log p(s_i)+\sum_{v_k\in C}\log p(v_k|s_i)\right]$$

第 3、4 讲 中文词法分析

内容

词法分析任务：从字符串到词串
中文词法分析的意义
中文文本分词面对的问题

3.1 词法分析任务：从字符串到词串

汉语的自然书面文本词与词之间无空格分开，因此，在汉语书面语的处理中（比如词频统计、句子结构分析、语义理解等），首先碰到的就是词的切分问题。

从字符串到词串（英文）

Tokenization： 把字符串变为词串

I'm a student -> I| 'm | a | student

Lemmatization Word Stemming：对词进行内部结构和形式分析
took -> take + ed (past tense)

3.2 中文词法分析的意义

文本分词是各个层次的自然语言处理任务的基础：

1. 文语转换Text-to-speech
2. 文本校对 Chinese Text Correction
3. 文本检索Information Retrieval
4. 词频统计、句法分析、机器翻译、 .....

3.3 文本分词面对的问题

什么是中文的“词”
分词歧义
未登录词识别

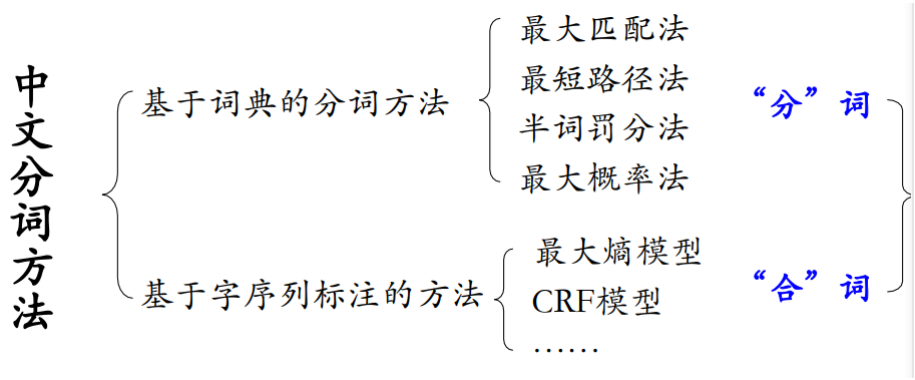
什么是词

分词规范

不同的人对“词”的认识有差异

文本分词中的歧义 组合型歧义 交集型歧义 混合型歧义 真歧义 伪歧义

4.1 中文分词基本方法概述



4.2 基于词典

最大匹配法

从左向右取待切分汉语句的前m 个字符（m为词典里最长的词字符数）；若这m个字符属于词典里面的词，则称匹配成功，然后将这m个字符切分出来，剩下的词语作为新的待切分汉语句；若这m个字符不属于词典里面的词，则去除这m个字符的**最后一个字符**，剩余的m-1个字符继续匹配，直到匹配或剩下一个字符为止；重复以上步骤，直到汉语句切完；

缺点

无法发现分词歧义 → 单向最大匹配改为双向

双向最大匹配法可以发现链长为奇数的交集型歧义，但无法发现链长为偶数的交集型歧义。

无法发现组合型歧义 。

在最大匹配法的基础上进行修改，如何给出“改错” 的触发条件带有一定的主观性

最优路径法

看待汉语词语切分问题的新视角：词图上的最优路径求解问题

- 词数最少的路径最优
  - 优点： 好于单向的最大匹配方法
  - 缺点：同样无法解决大部分交集型歧义
- 半词法 ——词数最少且半词最少

  在词图的路径优劣评判中引入罚分机制

  ■ 罚分规则：

  1) 每个词对应的边罚1分。

  2) 每个半词对应的边加罚1分。

  3) 一个分词方案的评分为它所对应的路径上所有边的罚分之和。

  4) 最优路径就是罚分最低的分词路径
  - 缺点：仍然无法解决“有意见分歧”的问题!
- 最大概率法分词

  字串成词概率最大的路径最优

  基本思想：在词图上选择词串概率最大的分词路经作为最优结果

  - 动态规划算法：最优路径中的第 i 个词 W_i 的累积概率等于它的左邻词 W_{i-1} 的累积概率乘以 W_i 自身的概率。

$$P'(w_i) = P'(w_{i-1}) \times P(w_i)$$

  - 为方便计算，一般把概率转化为路径代价

$$C = -\log(P)$$

$$C'(w_i) = C'(w_{i-1}) + C(w_i) \quad \text{公式1}$$

$C'(w_i)$：最小累积代价　$C'(w_{i-1})$：最佳左邻词

1) 对一个待分词的字串 S，按照从左到右的顺序取出全部候选词$w_1, w_2, \cdots, w_i, \cdots, w_n$；
2) 到词典中查出每个候选词的概率值$P(w_i)$，转换为代价$C(w_i)$，并记录每个候选词的全部左邻词；
3) 按照公式1计算每个候选词的累计代价，同时比较得到每个候选词的最佳左邻词；
4) 如果当前词w_n是字串S的尾词，且累计代价$C'(w_n)$最小，则w_n就是S的终点词；
5) 从w_n开始，按照从右到左顺序，依次将每个词的最佳左邻词输出，即为S的分词结果。

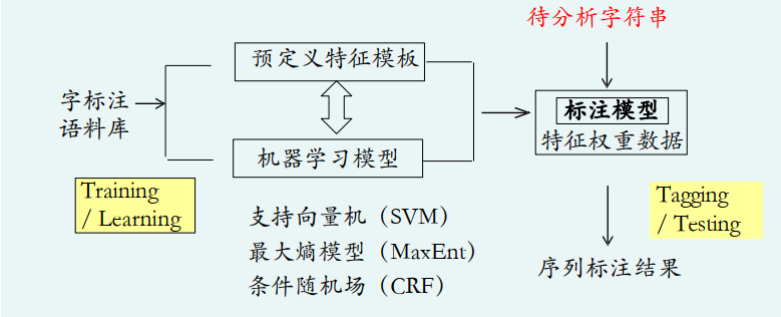
缺点：

- 并不能解决所有的交集型歧义问题
- 一般也无法解决组合型歧义问题

基于字序列标注的分词方法

字位标注法

分词可以看做是对字加“词位标记”的过程 。字位标注的原理：根据字本身及其上下文的特征，来决定当前字的词位标注



基于字序列标注的方法的优点：

- 能够平衡地看待词表词和未登录词的识别问题。 文本中的词表词和未登录词都是用统一的字标注来实现的
- 在学习架构上， 既可以不必专门强调词表词信息， 也不用专门设计特定的未登录词(如人名、 地名、 机构名)识别模块， 这使得分词系统的设计大大简化
- 在字标注过程中， 所有的字根据预定义的特征进行词位特性的学习， 获得一个概率模型。 然后，在待分字串上， 根据字与字之间的结合紧密程度， 得到一个词位的标注结果
- 在这样一个分词过程中， 分词成为字重组的简单过程， 结果令人满意的
- **总之——简单、鲁棒性强、效果好**

4.4 中文分词技术的评测

- 计算分词正确率的不同标准
  1) 以词数算
  2) 以句数算

- 分词质量对NLP应用系统的影响
  1) 分词质量对MT的影响
  2) 分词质量对IR的影响
  ……

- 准确率(precision)

$$准确率（P）= \frac{切分结果中正确分词数}{切分结果中所有分词数} * 100\%$$

- 召回率(recall)

$$召回率（R）= \frac{切分结果中正确分词数}{标准答案中所有分词数} * 100\%$$

- F-评价(F-measure 综合准确率和召回率的评价指标)

$$F1 = \frac{2*P*R}{P+R}$$

第 5 讲 马尔可夫模型

5.1 马尔可夫模型

●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可夫链：

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

●假设2：

如果只考虑公式(1)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：

$$p(q_t=S_j \mid q_{t-1}=S_i)=a_{ij}, \quad 1\le i,j\le N \quad \ldots(2)$$

该随机过程称为马尔可夫模型(Markov Model)。

在马尔可夫模型中，状态转移概率 a_{ij} 必须满足下列条件：

$$a_{ij} \geq 0 \qquad \ldots (3)$$

$$\sum_{j=1}^{N} a_{ij} = 1 \qquad \ldots (4)$$

状态序列 $S_1, \ldots, S_T$ 的概率：

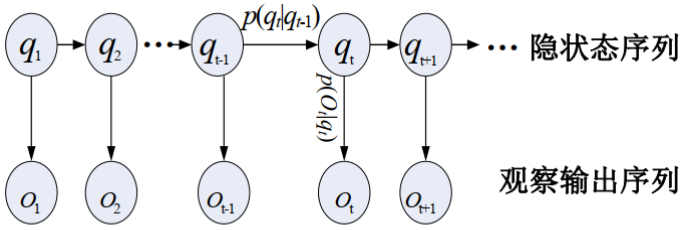
$$\begin{aligned} p(S_1,\cdots,S_T) &= p(S_1)\times p(S_2 \mid S_1)\times p(S_3 \mid S_1,S_2)\times\cdots\times p(S_T \mid S_1,\cdots,S_{T-1}) \\ &= p(S_1)\times p(S_2 \mid S_1)\times p(S_3 \mid S_2)\times\cdots\times p(S_T \mid S_{T-1}) \\ &= \pi_{S_1}\prod_{t=1}^{T-1} a_{S_t S_{t+1}} \qquad \ldots (5) \end{aligned}$$

其中，$\pi_i = p(q_1 = S_i)$为初始状态的概率。

5.2 隐马尔可夫模型

隐马尔可夫模型(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） ， 而可观察事件的随机过程是隐蔽状态转换过程的随机函数。



HMM 图解

◆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)

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，

$$\begin{cases} a_{ij} = p(q_{t+1} = S_j \mid q_t = S_i), \quad 1 \leq i, j \leq N \\ a_{ij} \geq 0 \\ \sum_{j=1}^{N} a_{ij} = 1 \end{cases} \qquad \ldots (6)$$

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，

$$\begin{cases} b_j(k)=p(O_t=v_k \mid q_t=S_j), & 1\le j\le N,\ 1\le k\le M \\ b_j(k)\ge 0 \\ \sum_{k=1}^{M} b_j(k)=1 \end{cases} \quad \ldots(7)$$

初始状态的概率分布为：$\pi=\pi_i$，其中，

$$\begin{cases} \pi_i=p(q_1=S_i), & 1\le i\le N \\ \pi_i\ge 0 \\ \sum_{i=1}^{N} \pi_i=1 \end{cases} \quad \ldots(8)$$

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。

◆给定HMM求观察序列

给定模型$\mu=(A,B,\pi)$，产生观察序列 $O=O_1O_2\ldots O_T$：

(1) 令 $t=1$;

(2) 根据初始状态分布 $\pi=\pi_i$选择初始状态 $q_1=S_i$;

(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 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。

◆三个问题：

(1) 在给定模型$\mu=(A, B, \pi)$ 和观察序列 $O=O_1O_2\ldots O_T$ 的情况下，怎样快速计算概率 $p(O|\mu)$？

(2) 在给定模型 $\mu=(A, B, \pi)$ 和观察序列 $O=O_1O_2\ldots O_T$ 的情况下，如何选择在一定意义下“最优”的状态序列 $Q=q_1q_2\ldots q_T$，使得该状态序列“最好地解释”观察序列?

(3) 给定一个观察序列 $O=O_1O_2\ldots O_T$，如何根据极大似然估计来求模型的参数值？即如何调节模型的参数，使得$p(O|\mu)$ 最大？

5.3 前向算法

◆问题1：快速计算观察序列概率$p(O|\mu)$

给定模型 $\mu=(A, B, \pi)$ 和观察序列$O=O_1O_2\ldots O_T$，快速计算 $p(O|\mu)$：

●**解决办法**：动态规划
　　前向算法(The forward procedure)

●**基本思想**：定义前向变量 $\alpha_t(i)$：

$$\alpha_t(i)=p(O_1O_2\cdots O_t, q_t=S_i \mid \mu) \quad \ldots(12)$$

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

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

● **算法1：前向算法描述**

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

(2) **循环计算：**

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

(3) **结束，输出：**

$$p(O|\mu)=\sum_{i=1}^{N}\alpha_T(i)$$

算法的时间复杂性：$O(N^2T)$。

● **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 个前向变量：$\alpha_t(1), \alpha_t(2), \ldots, \alpha_t(N)$，所以，时间复杂性为：$O(N)\times N=O(N^2)$。又因 $t=1, 2, \ldots, T$，所以前向算法总的复杂性为：$O(N^2T)$。

5.4 后向算法

● **后向算法**（The backward procedure）

定义后向变量 $\beta_t(i)$ 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：

$$\beta_t(i)=p(O_{t+1}O_{t+2}\cdots O_T|q_t=S_i,\mu)\qquad\ldots(15)$$

● **<u>算法2</u>：后向算法描述**

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

(2) 循环计算：

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

(3) 输出结果：$p(O|\mu)=\sum_{i=1}^{N}\beta_1(i)\times\pi_i\times b_i(O_1)$

算法的时间复杂性：$O(N^2T)$

5.5 Viterbi 搜索算法

<u>Viterbi 算法</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，模型沿着某一条路径到达 S_i，输出观察序列 $O=O_1O_2\ldots O_t$ 的最大概率为：

$$\delta_t(i)=\max_{q_1,q_2,\cdots,q_{t-1}}p(q_1,q_2,\cdots,q_t=S_i,O_1O_2\cdots O_t|\mu)\qquad\ldots(22)$$

5.6 参数学习

给定一个观察序列 $O = O_1O_2 \ldots O_T$，如何根据最大似然估计来求模型的参数值？或者说如何调节模型 μ 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。

如果产生观察序列 O 的状态 $Q = q_1q_2 \ldots q_T$ 已知（存在大量标注的样本），可以用极大似然估计来计算 μ 的参数：

$$\pi_i = \delta(q_1, S_i)$$

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

… (24)

类似地，

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

… (6.25)

如果产生观察序列 O 的状态 $Q = q_1q_2 \ldots q_T$ 已知（存在大量标注的样本），可以用极大似然估计来计算 μ 的参数：

$$\pi_i = \delta(q_1, S_i)$$

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

… (24)

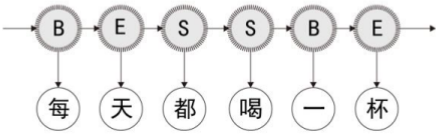
类似地，

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

… (6.25)

5.7 HMM应用举例

- 将状态值集合Q置为{B, E, M, S}，分别表示词的开始、结束、中间（begin、end、middle）及字符独立成词（single）；观测序列即为中文句子。
- 例如，“每天都喝一杯”通过HMM(Viterbi 算法)求解得到状态序列“B E S S B E”，则分词结果为“每天/都/喝/一杯”。



第 6 章 条件随机场

6.1 概述

随机场是由若干个位置组成的整体，当给每一个位置中按照某种分布随机赋予一个值之后，其全体就叫做随机场。以词性标注为例： 假如有一个十个词组成的句子需要做词性标注。这十个词每个词的词性可以在已知的词性集合（名词， 动词...)中去选择。当我们为每个词选择完词性后，这就形成了一个随机场。

马尔科夫随机场是随机场的特例，它假设随机场中某一个位置的赋值仅仅与和它相邻的位置的赋值有关，和与其不相邻的位置的赋值无关。

条件随机场(conditional random field, CRF)是马尔科夫随机场的特例，它假设马尔科夫随机场中只有X和Y两种变量，X一般是给定的， 而Y一般是在给定X的条件下的输出。
在词性标注的例子中， X是词， Y是词性。如果我们假设它是一个马尔科夫随机场，那么它也就是一个条件随机场 。

条件随机场是用于标注和划分序列结构数据的概率化结构模型， 在NLP和图像处理中得到了广泛应用。
基本思路：给定观察序列X， 输出标记序列Y， 通过计算 P(Y|X) 求解最优标记序列。

自然语言处理中的词性标注和中文分词就是适合CRF使用的任务，因为它们往往和上下文有关。

参数化形式

我们如何将条件随机场转化为机器学习模型呢？

答案：通过特征函数和其权重系数。

什么是特征函数呢？

这里的特征函数分为两类，一类是定义在Y节点上的节点特征函数，称作状态函数，只和当前节点有关，记为：

$$s_l(y_i, X, i) \quad l = 1, 2, 3, \ldots, L$$

其中，L是定义在该节点的节点特征函数的总个数，i是当前节点在序列的位置。

什么是特征函数呢？

第二类是定义在Y上下文的转移特征函数，这类特征函数只和当前节点和上一个节点有关，记为：

$$t_k(y_{i-1}, y_i, X, i) \quad k = 1, 2, 3, \ldots, K$$

其中，K 是定义在该节点的转移特征函数的总个数，i是当前节点在序列的位置。之所以只有上下文相关的转移特征函数，没有不相邻节点之间的特征函数，是因为CRF满足马尔科夫性质.

6.2 模型训练

CRFs及其应用

应用举例： 基于字标注的分词方法

基本思想： 将分词过程看作是字的分类问题， 每个字在构造一个特定的词语时都占据着一个确定的构词位置(即词位)。一般情况下， 每个字只有4个词位：词首(B)、 词中(M)、词尾(E)和单独成词(S) 。

在字标注过程中，对所有的字根据预定义的特征进行词位特征学习，获得一个概率模型，然后在待切分字串上，根据字与字之间的结合紧密程度，得到一个词位的分类结果， 最后根据词位定义直接获得最终的分词结果。

实现 CRFs 也需要解决如下三个问题：
□特征选取
□参数训练
□解码

- **①特征选取**

  对应转移函数的特征：

$$t_1(y_{i-1}, y_i, X, i) = \begin{cases} 1 & \text{如果前一个字的标记} y_{i-1} \text{是B，当前字的标记} y_i \text{是M} \\ 0 & \text{否则} \end{cases}$$

$$t_2(y_{i-1}, y_i, X, i) = \begin{cases} 1 & \text{如果前一个字的标记} y_{i-1} \text{是M，当前字的标记} y_i \text{是M} \\ 0 & \text{否则} \end{cases}$$

- **②参数训练**

  通过训练语料估计特征权重λ_j使其在给定一个观察序列X的条件下，找到一个最有可能的标记序列Y，即条件概率$P(Y|X)$ 最大。

  条件概率已由上文给出：

$$P(Y|X) = \frac{1}{Z(X)} exp\left(\sum_{i,k} \lambda_k t_k(Y_{i-1}, Y_i, X, i) + \sum_{i,l} \mu_l s_l(Y_i, X, i) \right)$$

为了训练特征权重λ_j，需要计算模型的损失和梯度。由梯度更新λ_j，直到λ_j收敛。

- 损失函数定义为负对数似然函数：

$$L(\boldsymbol{\lambda}) = -\log p(Y \mid X, \boldsymbol{\lambda}) + \frac{\varepsilon}{2}\boldsymbol{\lambda}^2$$

采用随机梯度下降优化该损失函数！

- **③解码**

条件随机场解码的过程就是根据模型求解的过程,可以由维特比(Viterbi)算法完成。维特比算法是一个动态规划算法，动态规划要求局部路径也是最优路径的一部分。