

信息检索与数据挖掘

第8章 概率模型

书上第10章 XML检索[自学]

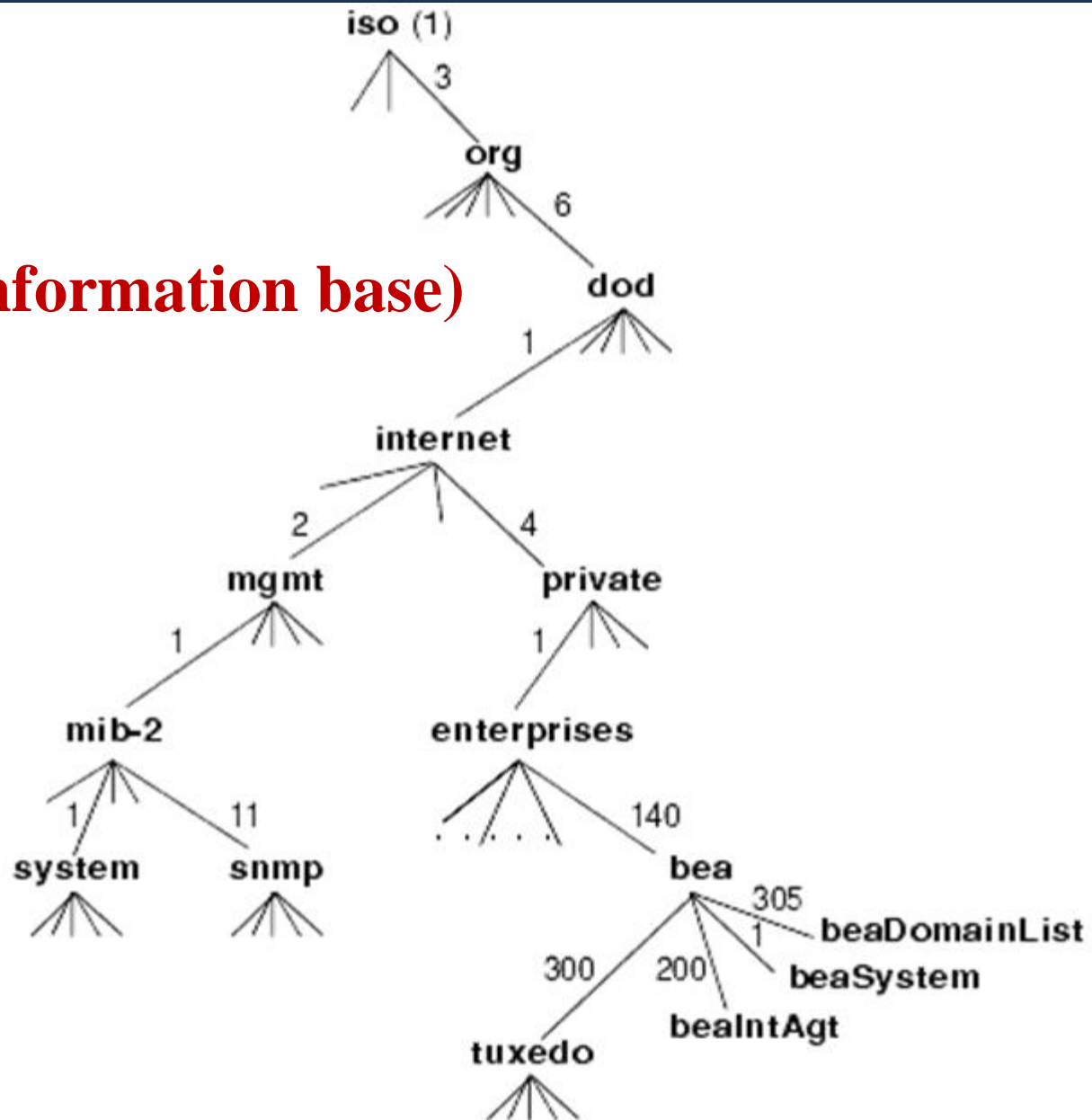
- 上世纪 90 年代末，使用不同的**数据管理系统**来存储和搜索他们的关键数据。
- 2001 年进入了 **XML 时代**。分析企业中的非结构化和半结构化数据的时代诞生。
- 如今，所有类型数据的激增。我们处于另一个演化方向的顶端，通常称为**大数据**。

表10-1 RDB搜索、非结构化IR及结构化IR。对于结构化检索来说，尽管很多学者都认为Xquery（10.5节）将会成为结构化查询的标准，但是关于这一点目前还没有最后定论

	RDB搜索	非结构化检索	结构化检索
对象	记录	非结构化文档	以文本为叶节点的树
模型	关系模型	向量空间或其他	?
主要数据结构	表格	倒排索引	?
查询语言	SQL查询	自由文本查询	?

MIB(management information base)

树形结构的数据随处可见，人们习惯于有序地组织所有的数据。这类数据的检索既不同于传统的**RDMS**，也不同于**自由文本**检索。



.1.3.6.1.4.1.140.300 = absolute OID for "tuxedo" MIB

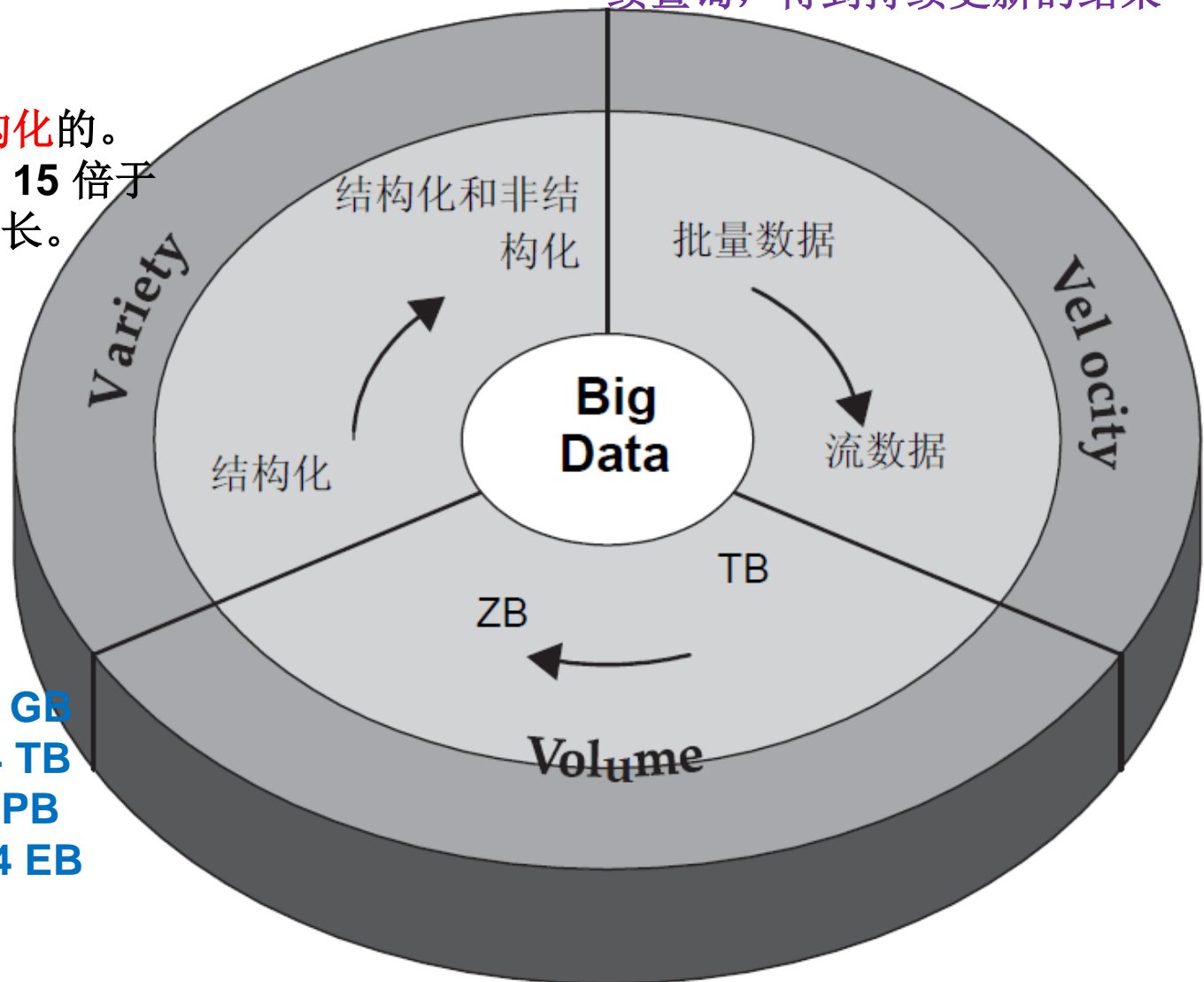
$V^3 \rightarrow V^4$

80% 的信息是非结构化的。
非结构化信息正在以 15 倍于
结构化信息的速率增长。

Value

1 Terabyte (TB) = 1024 GB
1 Petabyte (PB) = 1024 TB
1 Exabyte (EB) = 1024 PB
1 Zettabyte (ZB) = 1024 EB

使用流计算，执行一种类似于持续查询，得到持续更新的结果



可用 3 个特征来定义大数据：数量、种类和速度

信息检索与数据挖掘

第8章 概率模型

回顾：词项-文档关联矩阵

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	1	1	0	0	0	1
Brutus	1	1	0	1	0	0
Caesar	1	1	0	1	1	1
Calpurnia	0	1	0	0	0	0
Cleopatra	1	0	0	0	0	0
mercy	1	0	1	1	1	1
worser	1	0	1	1	1	0

- 每个文档用一个二维向量表示 $\in \{0,1\}^{|V|}$
- 布尔检索的本质
 - 将查询 q 中出现的词项对应行取出做布尔运算

词项-文档计数矩阵

- 考虑词项在文档中出现的次数
 - 将每个文档看成是一个计数向量：矩阵中的一列
 - 查询 q 对应的向量与文档对应的列向量求相似度

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	157	73	0	0	0	0
Brutus	4	157	0	1	0	0
Caesar	232	227	0	2	1	1
Calpurnia	0	10	0	0	0	0
Cleopatra	57	0	0	0	0	0
mercy	2	0	3	5	5	1
worser	2	0	1	1	1	0

tf, 查询 q 中词项在文档中出现的频度 → 词项的概率表征相关性?

二值→计数→权重矩阵 (tf-idf值)

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	5.25	3.18	0	0	0	0.35
Brutus	1.21	6.1	0	1	0	0
Caesar	8.59	2.54	0	1.51	0.25	0
Calpurnia	0	1.54	0	0	0	0
Cleopatra	2.85	0	0	0	0	0
mercy	1.51	0	1.9	0.12	5.25	0.88
worser	1.37	0	0.11	4.15	0.25	1.95

- 每个文档可看成一个向量，其中每个分量对于词典中一个词项，分量值为对于词项的tf-idf值

tf, 查询q中词项在文档中出现的频度→词项的概率表征相关性?
idf, 罕见词的idf高而高频词的idf低→根据语言学修正词项的概率

The diagram illustrates the differences between the Boolean model and the Vector Space model. At the top, a red title states: **d和q的相关性是0或1** (The relevance of d and q is 0 or 1). Below this, the **布尔模型** (Boolean model) is shown on the left, and the **向量空间模型** (Vector Space model) is shown on the right.

布尔模型 (Boolean Model):

- Input: **布尔检索结果太少或太多** (Boolean retrieval results are too few or too many).
- Process: A vertical flow of boxes: **布尔** (Boolean) → **词项频率TF** (Term Frequency TF) → **TF-IDF**.
- Output: **对结果进行排序** (Sort the results).
- Annotations: To the left of the TF and TF-IDF boxes are the labels **文档** (Document) and **评分** (Rating) respectively, with arrows pointing to the boxes.

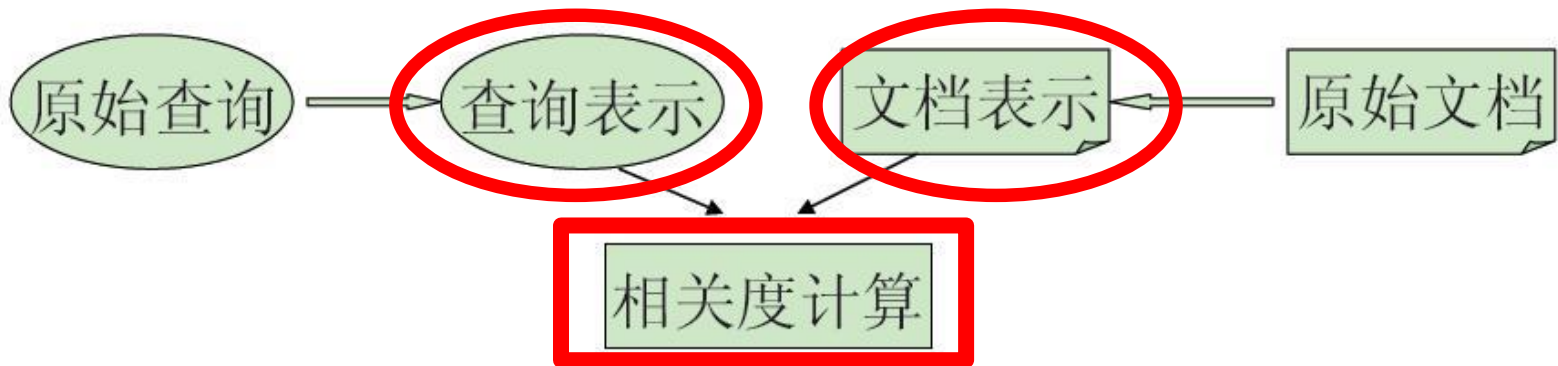
向量空间模型 (Vector Space Model):

- Input: **布尔模型** (Boolean model) is shown above the main flow.
- Process: A vertical flow of boxes: **词项-文档关联矩阵** (Term-Document Association Matrix) → **词项-文档计数矩阵** (Term-Document Count Matrix) → **词项-文档权重矩阵** (Term-Document Weight Matrix).
- Output: **文档和查询均表示成向量, 计算余弦相似度** (Documents and queries are both represented as vectors, and cosine similarity is calculated).
- Annotation: To the right of the count matrix box is the label **文档** (Document) with an arrow pointing to it.

At the bottom, a red title states: **d和q的相关性是0-1之间的一个数值** (The relevance of d and q is a numerical value between 0 and 1).

回顾：信息检索模型的作用

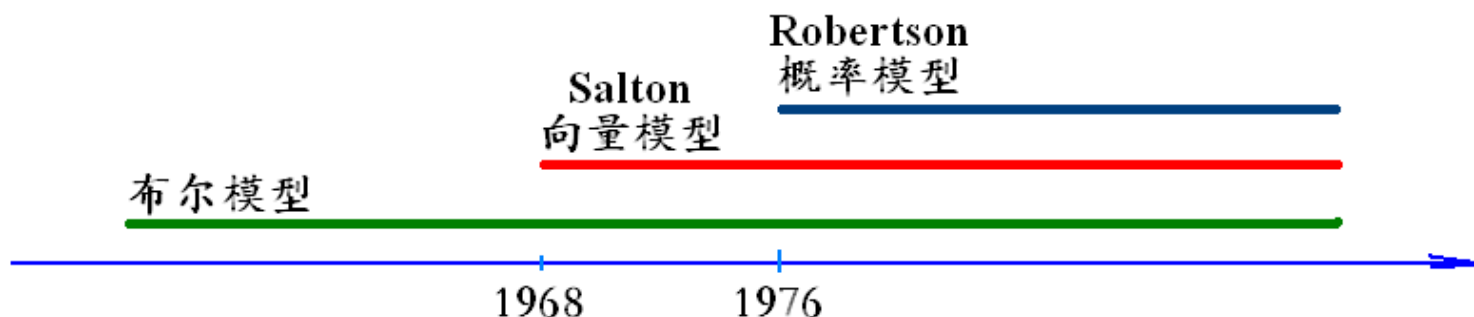
- 信息检索模型是指如何对查询和文档进行表示，然后对它们进行相似度计算的框架和方法
 - 本质上是对相关度建模
- 信息检索模型是IR中的核心内容之一



- 相关度的表示：0或1 → 0-1间的数值 → 概率

回顾：信息检索模型之经典模型

- **集合论模型 (Set Theoretic models)**
 - 布尔模型 (Boolean Model, BM)、模糊集合模型、扩展布尔模型
- **代数模型 (Algebraic models)**
 - 向量空间模型 (Vector Space Model, VSM)、广义向量空间模型、潜在语义标引模型、神经网络模型
- **概率模型 (Probabilistic models)**
 - 经典概率论模型 (PM)、推理网络模型、置信网络模型



概率检索模型是通过概率的方法将查询和文档联系起来

课程内容

- 第1章 绪论
- 第2章 布尔检索及倒排索引
- 第3章 词项词典和倒排记录表
- 第4章 索引构建和索引压缩
- 第5章 向量模型及检索系统
- 第6章 检索的评价
- 第7章 相关反馈和查询扩展
- **第8章 概率模型**
- 第9章 基于语言建模的检索模型
- 第10章 文本分类
- 第11章 文本聚类
- 第12章 Web搜索
- 第13章 多媒体信息检索
- 第14章 其他应用简介

本讲内容

- 概率基础知识
- 概率排序原理
 - 二元假设检验与概率排序原理
 - 概率排序的实现方式
- **BIM模型**
 - 二值独立概率模型BIM
 - BIM排序函数的推导
 - RSV的估算方法
- **BM25模型**

古之所谓善战者，胜于易胜者也

- **随机试验**：可在相同条件下重复进行；试验可能结果不止一个，但能确定所有的可能结果；一次试验之前无法确定具体是哪种结果出现。
 - 掷一颗骰子，考虑可能出现的点数
- **随机事件**：随机试验中可能出现或可能不出现的情况
 - 掷一颗骰子，4点朝上
- **概率**：事件A的概率是指事件A发生的可能性，记为 $P(A)$
 - 掷一颗骰子，出现6点的概率为多少？
- **条件概率**：已知事件A发生的条件下，事件B发生的概率称为A条件下B的条件概率，记作 $P(B|A)$
 - 30颗红球和40颗黑球放在一块，请问第一次抽取为红球的情况下第二次抽取黑球的概率？

关于事件

- 必然事件

例 1 下列成语所描述的事件是必然事件的是().
A.水中捞月 B.揠苗助长 C.守株待兔 D.瓮中捉鳖

- 随机事件

例 2 下列成语所描述的事件是随机事件的是().
A.长生不老 B.树倒猢猻散 C.八九不离十 D.海枯石烂

- 不可能事件

例 3 有下列成语:十拿九稳,刻舟求剑,三头六臂,大海捞针.其中描述的是不可能事件的有_____.

从概率论角度看“胜者表”

- ① 对于词典中的每个词项 t ，预先计算出 r 个最高权重的文档（ t 的胜者表）
- ② 给定查询 q ，对查询 q 中所有词项的胜者表求并集，并可以生成集合 A
- ③ 根据余弦相似度大小从 A 中选取前 $\text{top } K$ 个文档

查询 q 中出现的词项如果在文档 d 中频度高，
则文档 d 与查询 q 相关的概率可能大些

词袋模型与事件独立性

- 词袋模型：不考虑词在文档中出现的顺序。
“John is quicker than Mary” 和 “Mary is quicker than John” 的表示结果一样

词袋模型是向量空间模型的假设

- 两事件**独立**：事件A、B，若 $P(AB)=P(A)P(B)$ ，则称A、B独立
- 三事件**独立**：事件A B C，若满足 $P(AB)=P(A)P(B)$ ， $P(AC)=P(A)P(C)$ ， $P(BC)=P(B)P(C)$ ， $P(ABC)=P(A)P(B)P(C)$ ，则称A、B、C独立
- 多事件**独立**：两两独立、三三独立、四四独立....

键盘布局与条件概率

字母	A	B	C	D	E	F	G	H	I
频率	0.063	0.0105	0.023	0.035	0.105	0.0221	0.011	0.047	0.054
字母	J	K	L	M	N	O	P	Q	R
频率	0.001	0.003	0.029	0.021	0.059	0.0644	0.0175	0.001	0.053
字母	S	T	U	V	W	X	Y	Z	空格
频率	0.052	0.071	0.0215	0.008	0.012	0.002	0.012	0.001	0.2



乘法公式、全概率公式和贝叶斯公式

- 乘法公式:

- $P(AB) = P(A)P(B|A)$

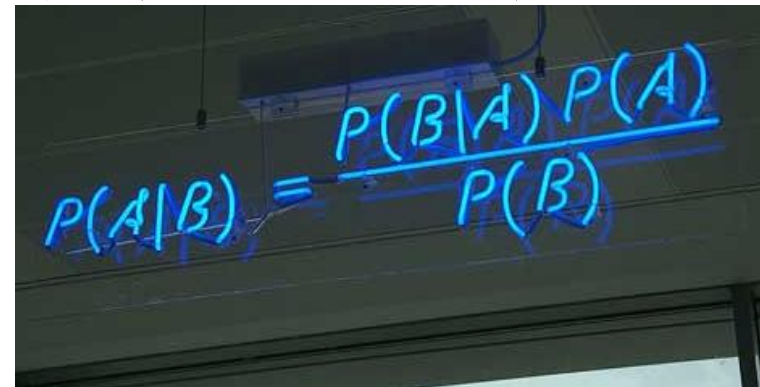
- $P(A_1A_2\dots A_n) = P(A_1)P(A_2|A_1)\dots P(A_n|A_1\dots A_{n-1})$

- 全概率公式: $A_1A_2\dots A_n$ 是整个样本空间的一个划分

$$P(B) = \sum_{i=1}^n P(A_i)P(B|A_i)$$

- 贝叶斯公式: $A_1A_2\dots A_n$ 是整个样本空间的一个划分

$$P(A_j | B) = \frac{P(A_j)P(B|A_j)}{\sum_{i=1}^n P(A_i)P(B|A_i)}, (j = 1, \dots, n)$$



A photograph of a chalkboard with the formula for Bayes' theorem written in blue chalk. The formula is $P(A|B) = \frac{P(B|A)P(A)}{P(B)}$. The text is written in a slightly messy, handwritten style.

随机变量

- 随机变量：若随机试验的各种可能的结果都能用一个变量的取值（或范围）来表示，则称这个变量为**随机变量**，常用 X 、 Y 、 Z 来表示
 - (离散型随机变量)：掷一颗骰子，可能出现的点数 X (可能取值1、2、3、4、5、6)
 - (连续型随机变量)：北京地区的温度(-15~45)

概率检索模型

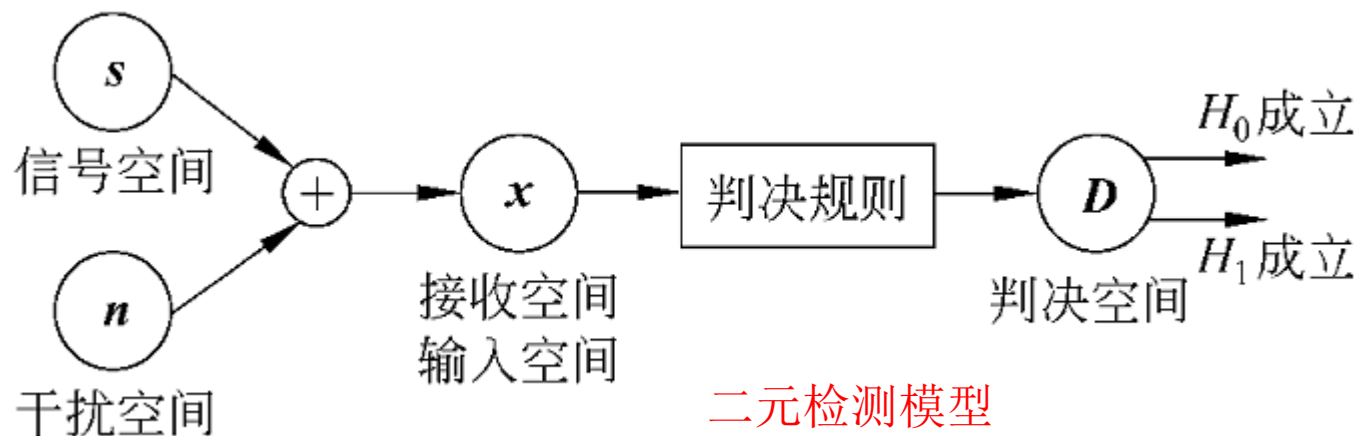
- 概率检索模型是**通过概率的方法将查询和文档联系起来**
 - 定义3个随机变量 R 、 Q 、 D ：相关度 $R=\{0,1\}$ ，查询 $Q=\{q_1, q_2, \dots\}$ ，文档 $D=\{d_1, d_2, \dots\}$ ，则可以通过计算条件概率 $P(R=1|Q=q, D=d)$ 来度量文档和查询的相关度。
- 概率模型包括**一系列模型**，如Logistic Regression(回归)模型及最经典的二值独立概率模型BIM、BM25模型等等(还有贝叶斯网络模型)。
- 1998出现的基于**统计语言建模**的信息检索模型本质上也是概率模型的一种。

本讲内容

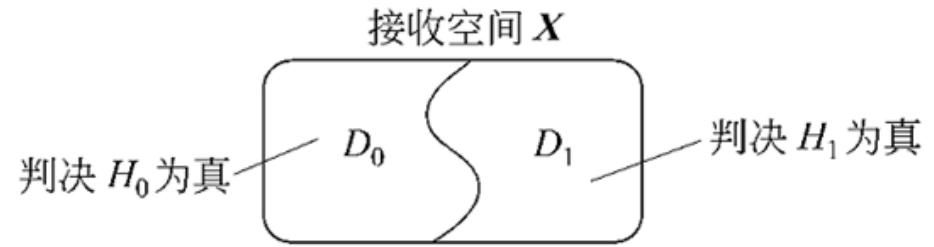
- 概率基础知识
- 概率排序原理
 - 二元假设检验与概率排序原理
 - 概率排序的实现方式
- **BIM模型**
 - 二值独立概率模型BIM
 - BIM排序函数的推导
 - RSV的估算方法
- **BM25模型**

二元假设检验示例：二元信号检测

- 第一部分是信号空间 s ：信源输出的信号只有两种状态, $s_1(t)$ 或 $s_0(t)$
- 第二部分是干扰空间：信号传输时所叠加的噪声
- 第三部分是接收空间（观测空间） x ：接收端接收到的受到干扰的信号
- 第四部分是判决规则：对输入空间的受到噪声干扰的信号按照某种准则进行判决归类，判断发送端发送的是 $s_1(t)$ 或 $s_0(t)$
- 第五部分是判决空间 D ： D 分为 D_0 区域和 D_1 区域两部分
 - D_0 区域：判断发送端发送的信号是 $s_0(t)$
 - D_1 区域：判断发送端发送的信号是 $s_1(t)$



二元假设检验示例： 判决概率



- 在接收端，无法确定信源在某一时刻输出是那种信号，为了分析方便，把**信源的输出称为假设**。二元数字通信系统中，信源由符号“0”和“1”组成。当信源输出“0”时，用假设 H_0 表示；而当信源输出“1”时，就用假设 H_1 表示。
- 对应于每一种判决结果，有相应的判决概率 $P(D_j | H_i)$ ($i, j=0, 1$): 假设 H_i 为真的条件下，判决 H_j 成立的概率。
- 在假设 H_i 为真的条件下，观测量 $(x|H_i)$ 的概率密度函数为： $f(x|H_i)$ 。由于观测量 $(x|H_i)$ 落在判决空间 D_i ，则判决 H_i 成立，所以判决概率有：

$$P(D_j | H_i) = \int_{D_j} f(x | H_i) dx, \quad i, j = 0, 1$$

就判决概率而言，我们希望正确的判决概率尽可能大，而错误判决概率尽可能小。

代价函数

- **代价函数 C_{ij}** :表示实际是 H_j 假设为真, 而判决为 H_i 假设为真所付出的代价。也称为风险函数。
- **检测概率**: 正确判决的概率 $P(D_1|H_1)$ 和 $P(D_0|H_0)$
- **虚警**: 实际 H_0 假设为真, 而判决为 H_1 假设为真。又称为第一类错误。
- 虚警引入的代价称为虚警代价 C_{01} 。
- 虚警发生的概率为: $P(D_1|H_0)$ 称为虚警概率。
- **漏报**: 实际 H_1 假设为真, 而判决为 H_0 假设为真。又称为第二类错误。
- 漏报引入的代价称为漏报代价 C_{10} 。
- 漏报发生的概率为: $P(D_0|H_1)$ 称为漏报概率。

二元Bayes判决

- 在判决规则已经确定的前提下，记 $P_0(\mathbf{x})$ 为收到 \mathbf{x} 时判决为 H_0 的概率； $P_1(\mathbf{x})$ 为收到 \mathbf{x} 时判决为 H_1 的概率
- 收到 \mathbf{x} ，判决为 H_0 的代价
 - $\omega_0 = C_{00} P_0(\mathbf{x}) P(H_0) + C_{01} P_0(\mathbf{x}) P(H_1)$
- 收到 \mathbf{x} ，判决为 H_1 的代价
 - $\omega_1 = C_{11} P_1(\mathbf{x}) P(H_1) + C_{10} P_1(\mathbf{x}) P(H_0)$
- $\omega_0 > \omega_1$ 判 H_0 真； $\omega_0 < \omega_1$ 判 H_1 真

1/0风险（1/0损失）： $C_{00}=C_{11}=0$

概率检索模型与二元假设检验

- 概率检索模型

- 定义3个随机变量 R 、 Q 、 D ：相关度 $R=\{0,1\}$ ，查询 $Q=\{q_1, q_2, \dots\}$ ，文档 $D=\{d_1, d_2, \dots\}$ ，则可以通过计算条件概率 $P(R=1|Q=q, D=d)$ 来度量文档和查询的相关度。

- \rightarrow 二元假设检验： $R=0$ ，假设 H_0 ； $R=1$ ，假设 H_1

- 1/0风险情形 C_{ij} :表示实际是 H_j ，而判决为 H_i 所付出的代价

- 判决为 H_0 的代价： $\omega_0 = C_{01} P(R=0|q, d) P(H_1)$

- 判决为 H_1 的代价： $\omega_1 = C_{10} P(R=1|q, d) P(H_0)$

- 贝叶斯最优决策原理

- 文档集很大时，给定 q_i ， d ， $P(H_1) \approx P(H_0)$ ； $C_{01}=C_{10}$

- \rightarrow 当且仅当 $P(R=1|q, d) > P(R=0|q, d)$ 时， d 相关

概率排序原理

PRP (probability ranking principle)

- 利用概率模型来估计每篇文档和需求的相关概率 $P(R=1|d,q)$ ，然后对结果进行排序。
- 最简单的PRP 情况是，检索没有任何代价因子，或者说不会对不同行为或错误采用不同的权重因子。在返回一篇不相关文档或者返回一篇相关文档不成功的情况下，将失去1分（在计算精确率时这种基于二值的情形也往往称为1/0 风险）。而检索的目标是对于用户任意给定的k值，返回可能性最高的文档前k 篇作为结果输出。也就是说，PRP 希望可以按照 $P(R=1|d,q)$ 值的降序来排列所有文档。
- **定理 11-1 在1/0 损失的情况下，PRP 对于最小化期望损失（也称为贝叶斯风险）而言是最优的。**

基于检索代价的概率排序原理

- C_1 表示一篇相关文档未返回所发生的代价
- C_0 表示返回一篇不相关文档发生的代价
- **PRP**认为，如果对于一篇特定的文档 d 及所有其他未返回的文档 d' 都满足

$$C_0 \cdot P(R=1|d) - C_1 \cdot P(R=0|d) \leq C_0 \cdot P(R=1|d') - C_1 \cdot P(R=0|d')$$

- 那么 d 就应该是下一篇被返回的文档。

即二元Bayes判决


相比于信号检测，IR使用了最简单的模型

本讲内容

- 概率基础知识
- 概率排序原理
 - 二元假设检验与概率排序原理
 - 概率排序的实现方式
- **BIM模型**
 - 二值独立概率模型BIM
 - BIM排序函数的推导
 - RSV的估算方法
- **BM25模型**

二元假设检验示例：

MAP准则(maximum a posterior probability criterion)

- 收到信号 x ，若此时 H_0 为真的概率大则判 H_0 为真，反之 H_1 为真
 - 若 $P(x, H_0) > P(x, H_1)$ 则判 H_0 为真
 - 若 $P(x, H_1) > P(x, H_0)$ 则判 H_1 为真 ← $P(x, H_0)$ 和 $P(x, H_1)$ 是未知的
 - 由乘法公式： $P(x, H_0) = P(H_0|x) P(x)$; $P(x, H_1) = P(H_1|x) P(x)$
- 
- 最大后验准则
 - 若 $P(H_0|x) > P(H_1|x)$ 则判 H_0 为真
 - 若 $P(H_1|x) > P(H_0|x)$ 则判 H_1 为真

$$P(H_1 | x) \underset{H_0}{\overset{H_1}{\gtrless}} p(H_0 | x)$$

二元假设检验示例：后验概率的计算

• 后验概率的计算

$$P(H_i | x) = \frac{P(H_i)P(x | H_i)}{P(x)}$$



$$P(H_i | x) = \frac{P(H_i)f(x | H_i)}{f(x)}$$

$$P(x | H_i) = P(x \leq X \leq x + dx | H_i) \approx f(x | H_i)dx$$

$$P(x) = P(x \leq X \leq x + dx) \approx f(x)dx$$

• 最终判决准则为

$$\frac{P(H_1 | x)}{P(H_0 | x)} = \frac{\frac{P(H_1)f(x | H_1)}{f(x)}}{\frac{P(H_0)f(x | H_0)}{f(x)}} = \frac{f(x | H_1)P(H_1)}{f(x | H_0)P(H_0)} \underset{H_0}{\overset{H_1}{\geq}} 1$$

$$l_0 = \frac{P(H_0)}{P(H_1)} = \frac{P(H_0)}{1 - P(H_0)}$$

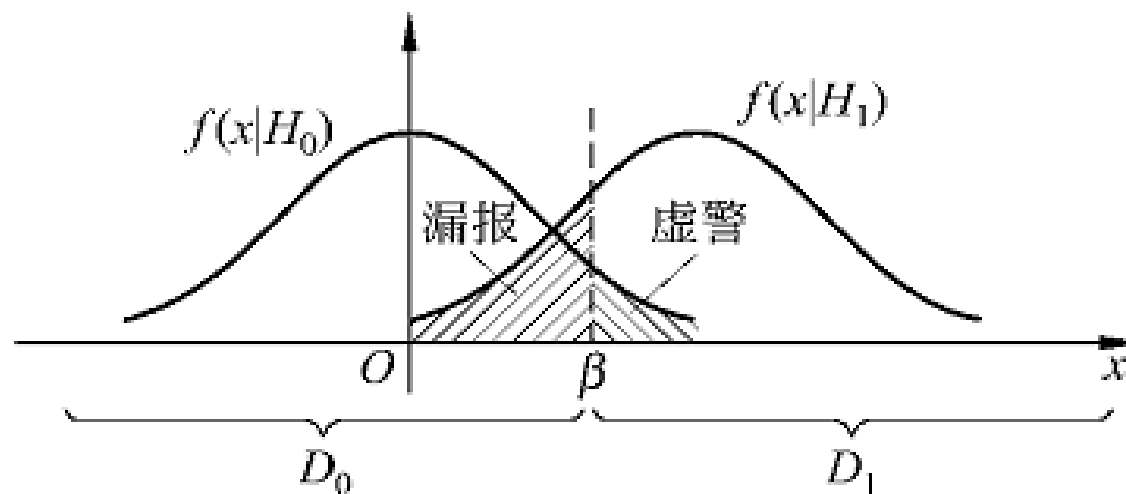
称为似然比门限值

$$l(x) = \frac{f(x | H_1)}{f(x | H_0)} \underset{H_0}{\overset{H_1}{\geq}} l_0$$

称为似然比

二元假设检验示例小结

- 二元假设检验的本质：如何决定判决区间的划分，使判决在某种意义上为最佳。
- 如果我们把 β 降低，则正确判决概率 $P(D_1|H_1)$ 将增大，但同时另一个正确判决概率 $P(D_0|H_0)$ 将减小。判决域的划分不仅影响判决概率，而且有最佳的划分方法。



概率排序原理实际应用

- 贝叶斯最优决策原理

- 文档集很大时，给定 q_i , d , $P(H_1) \approx P(H_0)$; $C_{01}=C_{10}$
- \rightarrow 当且仅当 $P(R=1|q,d) > P(R=0|q,d)$ 时, d 相关

$\leftarrow P(R=1|q,d)$ 和 $P(R=0|q,d)$ 是未知的

- 由乘法公式

- $P(R,d,q)=P(q) \cdot P(d|q) \cdot P(R|d,q)$

- $P(R,d,q)=P(q) \cdot P(R|q) \cdot P(d|R,q)$

- $P(R|q)$: $P(R=1|q)$ 和 $P(R=0|q)$ 可根据不相关文档百分比估计
- $P(R,d,q)$ 的估计转化为估计 $P(d|R,q)$

小结：概率排序原理(PRP)

- 简单地说：如果文档按照与查询的相关概率大小返回，那么该返回结果是所有可能获得结果中效果最好的。
- 严格地说：如果文档按照与查询的**相关概率**大小返回，而这些相关概率又能够基于已知数据进行尽可能**精确的估计**，那么该返回结果是所有基于已知数据获得的可能的结果中效果最好的。

$$P(R,d,q)=P(q)\cdot P(R|q)\cdot P(d|R,q)$$

本讲内容

- 概率基础知识
- 概率排序原理
 - 二元假设检验与概率排序原理
 - 概率排序的实现方式
- **BIM模型**
 - 二值独立概率模型**BIM**
 - BIM排序函数的推导
 - RSV的估算方法
- **BM25模型**

二值独立概率模型BIM

- 二值独立概率模型(**B**inary **I**ndependence Model, 简称 BIM): 伦敦城市大学Robertson及剑桥大学Sparck Jones 1970年代提出, 代表系统OKAPI
- 为了能够在实际中对概率函数 $P(R|d,q)$ 进行估计, 该模型中引入了一些简单的假设。
 - “**二值**”等价于布尔值: 文档和查询都表示为词项出现与否的布尔向量。也就是说, 文档 d 表示为向量 $x=(x_1, \dots, x_M)$, 其中当词项 t 出现在文档 d 中时, $x_t=1$, 否则 $x_t=0$ 。由于不考虑词项出现的次数及顺序, 许多不同的文档可能都有相同的向量表示。类似地, 我们将查询 q 表示成词项出现向量 q (由于查询 q 通常就是采用一系列词的集合来表示, 所以 q 和 q 的之间的区别并不十分重要)。
 - “**独立性**”指的是词项在文档中的出现是互相独立的, BIM 并不识别词项之间的关联。

BIM模型的核心思想

- Bayes公式是理解BIM模型的关键

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

- BIM模型通过Bayes公式对所求条件概率 $P(R=1|q,d)$ 和 $P(R=0|q,d)$ 展开进行计算。

- 贝叶斯最优决策原理

- 文档集很大时，给定 q_i , d , $P(H_1) \approx P(H_0)$; $C_{01}=C_{10}$
- \rightarrow 当且仅当 $P(R=1|q,d) > P(R=0|q,d)$ 时, d 相关

$\leftarrow P(R=1|q,d)$ 和 $P(R=0|q,d)$ 是未知的

BIM中Bayes公式的使用

$$P(R,d,q)=P(q) \cdot P(d|q) \cdot P(R|d,q)$$

$$P(R,d,q)=P(q) \cdot P(R|q) \cdot P(d|R,q)$$

- 在BIM模型下，基于词项出现向量的概率 $P(R | \bar{x}, \bar{q})$ 对概率 $P(R|d,q)$ 建模，利用贝叶斯定理，有

$$P(R = 1 | \bar{x}, \bar{q}) = \frac{P(\bar{x} | R = 1, \bar{q}) P(R = 1 | \bar{q})}{P(\bar{x} | \bar{q})}$$

$$P(d|q) \cdot P(R|d,q) = P(R|q) \cdot P(d|R,q)$$

$$P(R = 0 | \bar{x}, \bar{q}) = \frac{P(\bar{x} | R = 0, \bar{q}) P(R = 0 | \bar{q})}{P(\bar{x} | \bar{q})}$$

$$P(R|d,q) = P(R|q) \cdot P(d|R,q) / P(d|q)$$

$P(\bar{x} | R = 1, \bar{q})$ 和 $P(\bar{x} | R = 0, \bar{q})$ 分别表示当返回一篇相关或不相关文档时文档表示为 \bar{x} 的概率

$P(R = 1 | \bar{q})$ 和 $P(R = 0 | \bar{q})$ 分别表示对于查询 \bar{q} 返回一篇相关和不相关文档的先验概率。

BIM排序函数的推导

优势率： $O(A) = \frac{P(A)}{P(\bar{A})} = \frac{P(A)}{1 - P(A)}$

- 文档相关性的优势率定义如下

$$O(R | \bar{x}, \bar{q}) = \frac{P(R = 1 | \bar{x}, \bar{q})}{P(R = 0 | \bar{x}, \bar{q})} = \frac{\frac{P(R = 1 | \bar{q})P(\bar{x} | R = 1, \bar{q})}{P(\bar{x} | \bar{q})}}{\frac{P(R = 0 | \bar{q})P(\bar{x} | R = 0, \bar{q})}{P(\bar{x} | \bar{q})}} = \frac{P(R = 1 | \bar{q})}{P(R = 0 | \bar{q})} \cdot \frac{P(\bar{x} | R = 1, \bar{q})}{P(\bar{x} | R = 0, \bar{q})}$$

对于给定查询是个常数

$$\frac{P(\bar{x} | R = 1, \bar{q})}{P(\bar{x} | R = 0, \bar{q})} = \prod_{t=1}^M \frac{P(x_t | R = 1, \bar{q})}{P(x_t | R = 0, \bar{q})}$$

独立性假设
M?

$$O(R | \bar{x}, \bar{q}) = O(R | \bar{q}) \cdot \prod_{t=1}^M \frac{P(x_t | R = 1, \bar{q})}{P(x_t | R = 0, \bar{q})}$$

本讲内容

- 概率基础知识
- 概率排序原理
 - 二元假设检验与概率排序原理
 - 概率排序的实现方式
- **BIM模型**
 - 二值独立概率模型BIM
 - **BIM排序函数的推导**
 - RSV的估算方法
- **BM25模型**

BIM排序函数的推导

$$O(R | \vec{x}, \vec{q}) = O(R | \vec{q}) \cdot \prod_{t=1}^M \frac{P(x_t | R=1, \vec{q})}{P(x_t | R=0, \vec{q})}$$

$\leftarrow x_t$ 取值要么为0要么为1

$$O(R | \vec{x}, \vec{q}) = O(R | \vec{q}) \cdot \prod_{t: x_t=1} \frac{P(x_t=1 | R=1, \vec{q})}{P(x_t=1 | R=0, \vec{q})} \cdot \prod_{t: x_t=0} \frac{P(x_t=0 | R=1, \vec{q})}{P(x_t=0 | R=0, \vec{q})}$$

p_t 词项出现在一篇相关文档中的概率

记 $p_t = P(x_t=1 | R=1, \vec{q})$

记 $u_t = P(x_t=1 | R=0, \vec{q})$

u_t 词项出现在一篇不相关文档中的概率

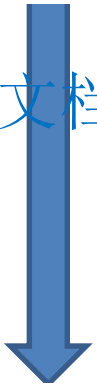
$$O(R | \vec{x}, \vec{q}) = O(R | \vec{q}) \cdot \prod_{t: x_t=q_t=1} \frac{p_t}{u_t} \cdot \prod_{t: x_t=0, q_t=1} \frac{1-p_t}{1-u_t}$$

BIM排序函数的推导

$$O(R | \vec{x}, \vec{q}) = O(R | \vec{q}) \cdot \prod_{t: x_t = q_t = 1} \frac{p_t}{u_t} \cdot \prod_{t: x_t = 0, q_t = 1} \frac{1 - p_t}{1 - u_t}$$

出现在文档中的查询词项的概率乘积

不出现在文档中的查询词项的概率乘积


$$O(R | \vec{x}, \vec{q}) = O(R | \vec{q}) \cdot \prod_{t: x_t = q_t = 1} \frac{p_t(1 - u_t)}{u_t(1 - p_t)} \cdot \prod_{t: q_t = 1} \frac{1 - p_t}{1 - u_t}$$

基于出现在文档中的查询词项来计算

考虑的是所有查询词项，对于给定的查询而言是常数

RSV (retrieval status value, 检索状态值)

排序函数只需计算

$$\prod_{t:x_t=q_t=1} \frac{p_t(1-u_t)}{u_t(1-p_t)}$$

p_t 词项出现在一篇
相关文档中的概率

u_t 词项出现在一篇不
相关文档中的概率

最终用于排序的是

$$RSV_d = \log \prod_{t:x_t=q_t=1} \frac{p_t(1-u_t)}{u_t(1-p_t)} = \sum_{t:x_t=q_t=1} \log \frac{p_t(1-u_t)}{u_t(1-p_t)}$$



c_t 查询词项的优势率比率 (odds ratio) 的对数值

$$\text{定义 } c_t = \log \frac{p_t(1-u_t)}{u_t(1-p_t)} = \log \frac{p_t}{1-p_t} + \log \frac{1-u_t}{u_t}$$

$$RSV_d = \sum_{x_t=q_t=1} c_t$$

c_t 如何计算?

	文档	相关 (R=1)	不相关 (R=0)
词项出现	$x_t=1$	p_t	u_t
词项不出现	$x_t=0$	$1-p_t$	$1-u_t$

本讲内容

- 概率基础知识
- 概率排序原理
 - 二元假设检验与概率排序原理
 - 概率排序的实现方式
- **BIM模型**
 - 二值独立概率模型BIM
 - BIM排序函数的推导
 - **RSV**的估算方法
- **BM25模型**

求 c_t : 理论上的概率估计方法

表中 df_t 是包含 t 的文档数目

	文档	相关	不相关	总计
词项出现	$x_t=1$	s	df_t-s	df_t
词项不出现	$x_t=0$	$S-s$	$(N-df_t)-(S-s)$	$N-df_t$
	总计	S	$N-S$	N

p_t 词项出现在一篇相关文档中的概率

u_t 词项出现在一篇不相关文档中的概率

$$c_t = \log \frac{p_t(1-u_t)}{u_t(1-p_t)} = \log \frac{p_t}{1-p_t} + \log \frac{1-u_t}{u_t}$$



$$p_t = s/S, u_t = (df_t-s)/(N-S)$$

$$c_t = K(N, df_t, S, s) = \log \frac{s / (S - s)}{(df_t - s) / ((N - df_t) - (S - s))}$$


平滑

- 在减少出现事件的概率估计值的同时提高未出现事件的概率估计值的方法称为平滑（smoothing）

可能出现的0 概率？

$$c_t = K(N, df_t, S, s) = \log \frac{s / (S - s)}{(df_t - s) / ((N - df_t) - (S - s))}$$

一种最简单的平滑方法就是对每个观察到的事件的数目都加上一个数 α
相当于在所有词汇表上使用了均匀分布作为一个贝叶斯先验
 α 的大小表示我们对均匀分布的信心强度


$$\hat{c}_t = K(N, df_t, S, s) = \log \frac{(s + \frac{1}{2}) / (S - s + \frac{1}{2})}{(df_t - s + \frac{1}{2}) / (N - df_t - S + s + \frac{1}{2})}$$

求 c_t : 实际中的概率估计方法

$$c_t = \log \frac{p_t(1-u_t)}{u_t(1-p_t)} = \log \frac{p_t}{1-p_t} + \log \frac{1-u_t}{u_t}$$

• u_t 的估算

- 假设相关文档只占有所有文档的极小一部分，那么可通过整个文档集的统计数字来计算与不相关文档有关的量。

$$u_t = df_t / N \quad \log[(1-u_t) / u_t] = \log[(N-df_t) / df_t] \approx \log N / df_t$$

• 估算 p_t

p_t 词项出现在一篇相关文档中的概率

u_t 词项出现在一篇不相关文档中的概率

- 如果我们知道某些相关文档，那么可以利用这些已知相关文档中的词项出现频率来对 p_t 进行估计
- Croft 和Harper（1979）在组合匹配模型（combination match model）中提出了利用常数来估计 p_t 的方法。
- Greiff（1998）提出

$$p_t = \frac{1}{3} + \frac{2}{3} \cdot \frac{df_t}{N}$$

求 c_t : 利用相关反馈获取更精确的 p_t 估计 不断迭代估计过程来获得 p_t 的更精确的估计结果

- (1) 给出 p_t 和 u_t 的初始估计。如, 假设所有查询中的词项的 p_t 是个常数, 具体地可以取 $p_t=0.5$ 。
- (2) 利用当前 p_t 和 u_t 的估值对相关文档集合 $R = \{d : R_{d,q} = 1\}$ 进行最佳的猜测。用该模型返回候选相关文档集给用户。
- (3) 利用用户交互对上述模型进行修正, 这是通过用户对某个文档子集 V 的相关性判断来实现的。基于相关性判断结果, V 可以划分成两个子集: $VR = \{d \in V, R_{d,q} = 1\}$ 和 $VNR = \{d \in V, R_{d,q} = 0\}$, 后者与 R 没有交集。
- (4) 利用已知的相关文档和不相关文档对 p_t 和 u_t 进行重新估计。如果 VR 和 VNR 足够大的话, 可以直接通过集合中的文档数目来进行最大似然估计: $p_t = |VR_t|/|VR|$ 。

VR_t 是 VR 中包含词项 x^t 的文档子集

实际中往往要对上述估计进行平滑

一个例子

查询为：信息 检索 教程

所有词项的在相关、不相关情况下的概率 p_t 、 u_t 分别为：

词项	信息	检索	教材	教程	课件
R=1时的概率 p_t	0.8	0.9	0.3	0.32	0.15
R=0时的概率 u_t	0.3	0.1	0.35	0.33	0.10

文档D1： 检索 课件

$$P(\vec{x} | R=1, \vec{q}) = (1-0.8)*0.9*(1-0.3)*(1-0.32)*0.15$$

$$P(\vec{x} | R=0, \vec{q}) = (1-0.3)*0.1*(1-0.35)*(1-0.33)*0.10$$

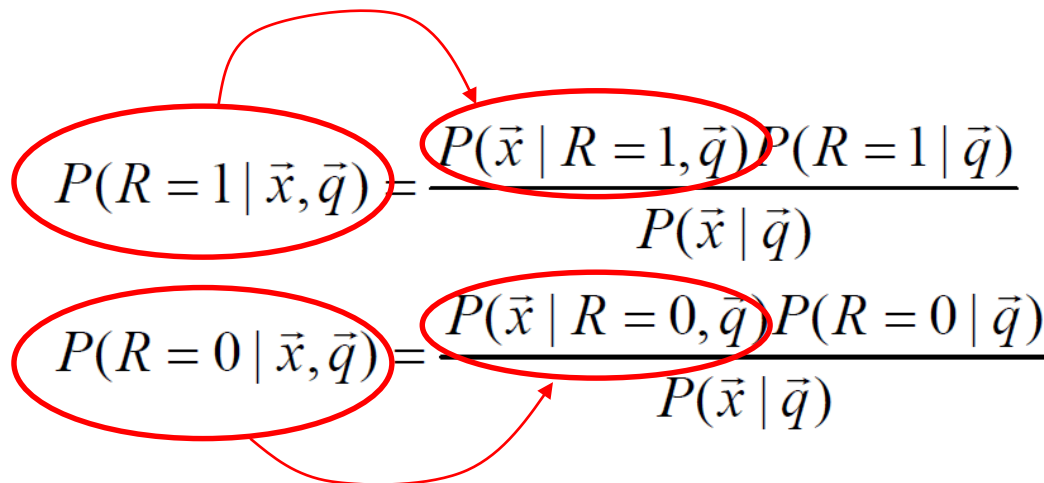
$$O(R | \vec{x}, \vec{q}) = 4.216$$

BIM模型小结

- 目标是求排序函数

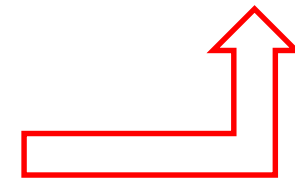
$$O(R | \vec{x}, \vec{q}) = \frac{P(R = 1 | \vec{x}, \vec{q})}{P(R = 0 | \vec{x}, \vec{q})}$$

- 转化为后验概率的估算


$$P(R = 1 | \vec{x}, \vec{q}) = \frac{P(\vec{x} | R = 1, \vec{q}) P(R = 1 | \vec{q})}{P(\vec{x} | \vec{q})}$$
$$P(R = 0 | \vec{x}, \vec{q}) = \frac{P(\vec{x} | R = 0, \vec{q}) P(R = 0 | \vec{q})}{P(\vec{x} | \vec{q})}$$

$$\prod_{t: x_t = q_t = 1} \frac{p_t(1 - u_t)}{u_t(1 - p_t)}$$

转化为 p_t 和 u_t 的估算



BIM模型的优缺点

- 优缺点：

- 优点：

- BIM模型建立在数学基础上，理论性较强

- 缺点：

- 需要估计参数
 - 原始的BIM没有考虑TF、文档长度因素
 - BIM中同样存在词项独立性假设

本讲内容

- 概率基础知识
- 概率排序原理
 - 二元假设检验与概率排序原理
 - 概率排序的实现方式
- **BIM模型**
 - 二值独立概率模型BIM
 - BIM排序函数的推导
 - RSV的估算方法
- **BM25模型**

BIM→BM25

Okapi BM25: 一个非二值的模型

- **BIM** 模型最初主要为较短的编目记录（**catalog record**）和长度大致相当的摘要所设计，在这些环境下它用起来也比较合适。但是对现在的全文搜索文档集来说，很显然模型应该重视**词项频率**和**文档长度**。一种称为**BM25** 权重计算机制（**BM25 weighting scheme**）或**Okapi** 权重计算机制（**Okapi weighting**）的方法第一次在某个检索系统实施之后，便发展成为基于词项频率、文档长度等因子来建立概率模型的一种方法，并且它不会引入过多的模型参数（**Spärck Jones** 等人2000）

Okapi BM25: 一个非二值模型

- **BIM**是最简单的文档评分方式:

$$RSV(Q, D) = \sum_{t_i \in D \cup Q} W_i^{IDF}$$

- 考虑词项在文档中的**tf权重**, 有:

$$RSV(Q, D) = \sum_{t_i \in D \cup Q} W_i^{IDF} \frac{(k_1 + 1)tf_{t_i, D}}{k_1((1 - b) + b \times (L_D / L_{ave})) + tf_{t_i, D}}$$

- $tf_{t_i, D}$: 词项 t_i 在文档 D 中的词项频率
- L_D (L_{ave}): 文档 D 的长度(整个文档集的平均长度)
- k_1 : 用于控制文档中词项频率比重的调节参数
- b : 用于控制文档长度比重的调节参数

Okapi BM25: 一个非二值模型

- 如果查询比较长，则加入**查询的tf**

$$RSV(Q, D) = \sum_{t_i \in D \cup Q} W_i^{IDF} \cdot \frac{(k_1 + 1)tf_{t_i, D}}{k_1((1 - b) + b \times (L_D / L_{ave})) + tf_{t_i, D}} \cdot \frac{(k_3 + 1)tf_{t_i, Q}}{k_3 + tf_{t_i, Q}}$$

- $tf_{t_i, Q}$: 词项 t_i 在 Q 中的词项频率
- k_3 : 用于控制查询中词项频率比重的调节参数
- 没有查询长度的归一化 (由于查询对于所有文档都是固定的)
- 理想情况下，上述参数都必须在开发测试集上调到最优。一般情况下，实验表明， k_1 和 k_3 应该设在 1.2到2之间， b 设成 0.75。

其它概率检索模型 应该看一下的综述（书后推荐的）

- **“Is this document relevant?...probably”: a survey of probabilistic models in information retrieval**
- ***F. Crestani et al. 1998,***
- ***ACM Computing Surveys, Vol. 30, No. 4, December 1998***

*F. Crestani*的综述

• 3. PROBABILISTIC RELEVANCE MODELS

- 3.1 Probabilistic Modeling as a Decision Strategy
- 3.2 The Binary Independence Retrieval Model ← Many Documents - One Query
- 3.3 The Binary Independence Indexing Model ← One Document - Many Queries
- 3.4 The Darmstadt Indexing Model ← relevance judgments
- 3.5 The Retrieval with Probabilistic Indexing Model ← d 中词项相对 q 的权重
- 3.6 The Probabilistic Inference Model ← An epistemic probability function P is defined on the concept space
- 3.7 The Staged Logistic Regression Model ← 为多个特征函数的线性组合
- 3.8 The N-Poisson Indexing Model ← tf 被建模为分布(Poisson distributions)

• 4. UNCERTAIN INFERENCE MODELS

- 4.1 A Nonclassical Logic for IR
- 4.2 The Inference Network Model

各种数学方法的应用

本讲要点小结

- 概率检索模型、概率排序原理

- 定义3个随机变量 R 、 Q 、 D ：相关度 $R=\{0,1\}$ ，查询 $Q=\{q_1, q_2, \dots\}$ ，文档 $D=\{d_1, d_2, \dots\}$ ，则可以通过计算条件概率 $P(R=1|Q=q, D=d)$ 来度量文档和查询的相关度。

- **BIM模型**

$$RSV_d = \log \prod_{t: x_t = q_t = 1} \frac{p_t(1-u_t)}{u_t(1-p_t)} = \sum_{t: x_t = q_t = 1} \log \frac{p_t(1-u_t)}{u_t(1-p_t)},$$

- 缺点：原始的BIM没有考虑TF、文档长度因素
- BIM中同样存在词项独立性假设
- **BIM模型→BM25模型**