网页文字处理

文本处理：信息检索基本组件，为后续应用提供支撑。将原始文档转化为词项以建立索引，使面向查询条件的精准的文档匹配成为可能。与查询解析相辅相成。

词条化处理

词条化：将给定字符序列拆分成一系列子序列的过程

英文分词(词组切分？标点符号？专有名词？)；**中文分词**(没有显式分隔符？虚词？分词歧义？**未登录词**（专有名词；专业术语；新词&变异词）？)；专业术语中文字与符号结合？新类型字符序列；多语言混杂。

分词方法 基于字符匹配（机械分词）：

ASM(d,a,m)，即Automatic Segmentation Model。其中：

- d，表示匹配方向，+1为正向，-1为逆向
- a：每次匹配失败后增/减字符数，+1为增字，-1为减字
- m：最大/最小匹配表示，+1为最大匹配，-1为最小匹配

例如，ASM（+，-，+）即正向减字最大匹配（即FMM方法）

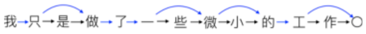
正向最大匹配 FMM：使用户满意->使用/户/满意

反向最大匹配 RMM：使用户满意->使/用户/满意

双向最大匹配：BM=FMM+RMM(选词数最少)

南京市长江大桥->南京市长/江/大桥(FMM)，南京市/长江大桥(RMM)->合并为词数最少的"南京市/长江大桥"

最少切分分词（搜索最短路径，边权可视作1，也可根据词频决定，N-最短路径法）；



优点：效率高、直观；**缺点**：依赖词典（维护词典开支大，难应对新生词汇，词频/重要性不影响结果）

基于统计：

$c = c_1c_2\ldots c_n$，c是待分词的句子（字串）。而$w = w_1w_2\ldots w_m$是切分的结果

设$P(w|c)$为c切分为w的某种估计概率。

$w_a, w_b, \ldots, w_k$为c的所有可能的分词方案。

那么，基于统计的分词模型就是找到目标词串w，使得w满足：

- $P(w|c) = \max\{P(w_a|c), P(w_b|c), \ldots, P(w_k|c)\}$
- 即估计概率最大所对应的词串。

(基于统计的分词方法不需要词典，但实际中可以用机械分词方法进行分词，以获得候选的分词集合：既发挥匹配分词切分速度快、效率高的特点，又利用了无词典分词结合上下文识别生词、自动消除歧义的优势)

N-gram 模型

N-gram指一个由N个单词组成的集合，各单词具有先后顺序。

N-gram模型的马尔可夫假设：

- 当前状态出现的概率仅同过去有限的历史状态有关，而与其他状态无关。
- 具体到分词任务，就是文本中第N个词出现的概率仅仅依赖于它前面的N-1个词，而与其他词无关。

常见的N-gram模型：

- N=1，一元文法模型（最大概率模型），$P(w) = P(w_1)P(w_2)\ldots P(w_n)$
- N=2，Bigram模型，$P(w) = P(w_1)P(w_2|w_1)\ldots P(w_n|w_{n-1})$
- N=3，Trigram模型，$P(w) = P(w_1)P(w_2|w_1)P(w_3|w_1w_2)\ldots P(w_n|w_{n-2}w_{n-1})$

以Bigram模型为例，基于最大似然估计进行推断

Bigram: $P(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)}{C(w_{n-1})}$

- 其中，$C(w_{n-1}w_n)$指词序列$w_{n-1}w_n$在语料库中出现的次数。
- 而$C(w_n)$指某个单词w_n在语料库中出现的次数。

首先，构造训练语料库，计算所有的$C(w_n)$与$C(w_{n-1}w_n)$。

其次，对于每一个可能的分词序列w，计算以下公式

- $P(w) = P(w_1)P(w_2|w_1)\ldots P(w_n|w_{n-1})$

最后，返回最大的$P(w)$所对应的分词序列作为结果。

当N=1时，N-gram模型退化为一元文法模型，此时词与词之间是独立的。

$P(W) = P(w_1, w_2, \ldots, w_n) \approx P(w_1) \times P(w_2) \times \ldots \times P(w_n)$ 独立性假设，一元文法

$P(w_i) = \frac{w_i\text{在语料库中的出现次数}n}{\text{语料库中的总词数}N}$

基于统计文法模型的**优缺点**：

优点：减轻了对于词典的依赖性

- 然而，这种依赖并非完全消除，取决于性能与效率的平衡
  - 如果深度结合机械分词（匹配分词），则效率提升但依赖词典
  - 如果减少对词典的依赖，则需要更多地遍历潜在的组合（解空间巨大！）

缺点：依赖已有数据中词频的统计，对于新生词汇或专业词汇不友好

- 冷门领域的稀有词汇往往难以准确切分
- 易受数据集先验偏差（Bias）的影响

基于序列标注：

四类标注：B(词的开始)、M(词的中间)、E(词的结束)、S(单字词)。例：中国科学技术大学是中国最好的大学：BMMMMMME S BE BME BE，中国科学技术大学/是/中国/最好的/大学

隐马尔可夫模型 HMM（初始状态概率矩阵，状态转移矩阵、观测状态概率矩阵；观测值集合（字符集合），隐藏状态值集合（BEMS））：维特比算法

1. 齐次假设：当前隐藏状态只与上一个状态有关 2. 观测独立性假设：观测值之间相互独立的，只与生成它的状态有关。

条件随机场 CRF：具有长距离依赖性和恶交叠性特征的能力；所有特征可以进行全局归一化，求得全局最优解

长短时记忆模型 LSTM：利用上下文关系 RNN

停用词处理

停用词：文档中频繁出现或对实际语义影响不大的词语

停用词识别方法：文本频率、词频统计、熵计算

去停用词隐患：特殊场景下意义/停用词组合有意义

趋势：**压缩技术**降低停用词表存储开支；**引入词项权重**降低高频词影响；**索引去除技术**排除低于权重的词项

规范化处理

归一化/词根化处理:词干提取 stemming(基于规则去除单词前后缀获得词根：复数/时态)->词性还原lemmatisation(基于词典将复杂形态转变成基础形态)

拼写错误处理：基于词典和**编辑距离**（两个字符串之间转换所最少需要的编辑操作步数，Dis("Kitten","Sitting")=3）进行检查和校对

同义词/相关词处理:基于人工维护知识库获取词项关系

网页索引

布尔检索：文档被表示为关键词集合，所有的查询式都被表示为关键词的布尔组合，一个文档当且仅当满足布尔查询式时才能被检索出来，检索策略是二值匹配

关联矩阵：文档的表示每列都是0/1向量，每个0/1对应一个词项，1(0)表示(不)包含该词。将给定的查询条件转化成行向量计算。高度稀疏

倒排索引

词汇表：词项的集合；倒排表 PostingList：文档 ID 列表，列举词项在哪些文档中出现。

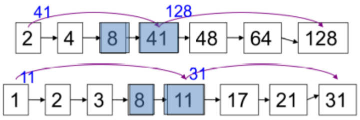
优点：搜索快；**缺点**：构建和维护成本高；**Key**：Term。

建立 PL 流程：检索每篇文档获得<词项，文档 ID>对，并写入临时索引->对临时索引中的词项进行排序->遍历临时索引，对相同词项的文档 ID 进行合并

基于倒排表的索引：查询是 PL 的"合并"

动态索引问题：主从索引(大的主索引，新文档存储在小的辅助索引，检索时同时遍历两个索引且合并，利用新的无效位向量删除，定期将辅助索引合并到主索引中)

优化(1)(A OR B)AND(C OR D)，先估计(A OR B)和(C OR D)的文档频率并排序，从小到大执行 AND **(2)**跳表(改进合并的 O(x+y))：利用**跳表指针**快速合并



扩展：二元词索引/位置信息索引

索引存储

词典与倒排表分开存储（可以支持并行、分布式查询）

词汇表存储：顺序、哈希、B/B+树、Trie 树（前缀树）

索引压缩

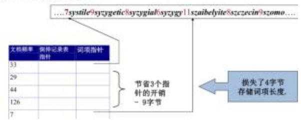
词典:1.将词典视作单一字符串（词项间用指针分割）

2.按块存储（每 k 个词项存储一个指针，需要额外 1 个字节表示词项长度）

单一字符串中词项指针上需要占用较多额外空间

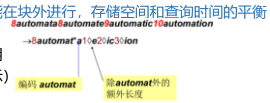
通过为每k个词项存储一个指针，来减少指针的总数量

- 需要额外1个字节用于表示词项长度。
- 例如，下图中的例子为k=4



二分查找只能在块外进行，存储空间和查询时间的平衡

3. 前端编码（公共前缀用特殊字符表示）



倒排表：**可变长度编码**（第一个存原值，其它存间距！）

先存储G，并分配1bit作为延续位

如果G<128，则采用第一位延续位为1 + 7位有效二进制编码的格式

如果G>=128，则先对低阶的7位编码，然后采取相同算法对高阶位进行编码。最后一个字节（8bit）的延续位为1，其他字节延续位为0.

214577 的二进制为 1101/0001100/0110001，VB 编码为 00001101/00001100/10110001

查询

查询表达的难点：用户表达的精简性和歧义性；用户表述方式差异性(同义词/方言，信息错漏的误导)；侧重点。

理解用户查询：基于查询的自然语言处理；相关性反馈；引导用户表达真实查询意图；借助间接反馈和情景信息

相关性反馈：用户在查询后标记文档的相关/不相关，然后迭代更新查询，以获得更好的结果；**流程**如下：

首先，用户提出一个查询条件（Query）

对于返回的文档，用户标出相关与不相关的部分

系统根据用户反馈，获得用户信息需求更为准确的描述

a) 基于相关性信息，更新查询条件，如为不同词项添加不同权重，或在查询条件中添加新的词项

b) 基于新查询条件，获取新的结果文档并再次提交用户进行评估

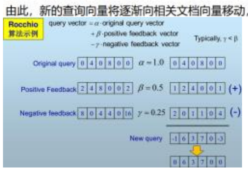
技术：**Rocchio 算法**（使查询尽可能离与之相关的文档更近，离与之不相关的文档更远）

$$\vec{q}_m = \alpha\vec{q}_0 + \beta\frac{1}{|D_r|}\sum_{\vec{d}_j\in D_r}\vec{d}_j - \gamma\frac{1}{|D_{nr}|}\sum_{\vec{d}_j\in D_{nr}}\vec{d}_j$$

其中，D_r为已知相关文档的向量集合，D_{nr}为已知不相关文档的向量集合

q_0为初始查询向量。α、β、γ为权重，根据手工调节或经验设定

由此，新的查询向量将逐渐向相关文档向量移动，远离不相关文档向量



（负分量->0）

正反馈（价值更大）vs 负反馈（收集难度高）

反馈分类 显式反馈：用户显式参与反馈过程（用户点击记录只有正样本、用户评论）；**隐式反馈**：系统追踪用户行为来推测返回文档的相关性（鼠标键盘/眼球动作，判断不一定很准确，分析难度高；但省却了用户的显式参与行为，减轻用户负担，提升体验感）；**伪反馈**：在没有用户参与的前提下，直接假设返回结果是相关的并进行反馈（结果未经用户判断，难以保证其准确性，可能出现查询漂移）。

查询扩展 用户针对词项的合适程度给出反馈，这些反馈将被用来构建更完整的查询条件

(1)同义词辞典：有助于提升查询召回率，但影响准确率

(2)基于搜索日志进行优化：查询日志，挖掘查询等价类

上下文感知：查询重组/查询特化/查询泛化/一般关联

排序

排序的难点:Web 网页的信息组织与内容质量参差不齐，用户庞杂且缺乏知识和经验，用户意图多样差异巨大

好的排序：网页内容匹配程度+网页内容质量

信息检索模型：文档与查询的表示形式与相关性的框架，实质是对文档基于相关性进行排序，在理解用户的基础之上产生近似用户决策的结果从而在顶部返回最相关的信息。[D,Q,F,R(Di,q)] **布尔模型** D:文档表达-词项的组合；Q:查询表达-布尔表达式；F:完全(二值)匹配；R:满足布尔表达式,相关性为 1(否则 0)

相关度计算：

Jaccard 系数：$\text{JACCARD}(A,B) = \frac{|A\cap B|}{|A\cup B|}$

词项频率 TF(t,d)：词项 t 在文档 d 中出现的次数

对数词频：$wf_{t,d} = \begin{cases}1+\log_{10}tf_{t,d} & tf_{t,d}>0\\ 0 & otherwise\end{cases}$

文档频率 DF(t)：出现词项 t 的文档数量

逆文档频率 IDF：$idf_t = log_{10}\frac{N}{df_t}$

TF-IDF: $W_{t,d} = (1+\log tf_{t,d})\cdot\log\frac{N}{df_t}$

向量空间模型 VSM 文档和查询视作一个词项权重构成的向量->计算向量之间相似性->按相似度大小排序返回 Top-K 文档给用户。

D:文档表达，每个文档可视作一个向量，其中每一维对应词项的 tf-idf 值；Q:查询表达，可视作一个向量，其中每一维对应词项的 tf-idf 值；F:非完全匹配方式；R:使用两个向量之间的相似程度来度量文档与查询之间的相关性

优点：简洁直观，可支持多种不同度量和权重方式，实用效果不错；**缺点**：缺乏语义层面的理解和匹配，同时依赖 tf-idf 值也可能造成干扰，用户无法描述词项之间的关系，词项之间的独立性假设实际上不成立。

VSM 下相似度：欧式距离；**余弦相似度**：

$$\cos(\vec{q},\vec{d}) = \frac{\sum_{i=1}^{|V|}q_id_i}{\sqrt{\sum_{i=1}^{|V|}q_i^2}\sqrt{\sum_{i=1}^{|V|}d_i^2}}$$

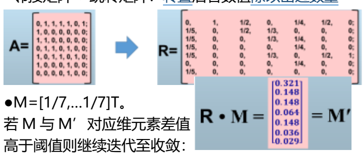
网页权威性计算

PageRank：优质网页引用或推荐的网页一定也优质。

$$PR(p_i) = \frac{1-d}{N} + d\sum_{p_j\in M(p_i)}\frac{PR(p_j)}{L(p_j)}$$

PR(pi)为网页 pi 的 PageRank 值，初值为 1/N；PR(pi)为指向网页 pi 的某个网页 pj 的 PageRank 值；L(pj)为网页 pj 发出的链接数量；d 为阻尼系数，取值在 0-1 之间；N 为网页总数，M(pi)为链入 pi 的页面集合。

●邻接矩阵->跳转矩阵：转置后各数值除以出边数量



●M=[1/7,...1/7]T。若 M 与 M' 对应维元素差值高于阈值则继续迭代至收敛:

陷阱节点:只有一条指向自己的边，没有其它出边；**终止节点**:没有出边，如同黑洞；**孤立结点**：没有任何入边，单纯使用 markov 链无法跳转，仅有初始概率，不再更新也不影响其它结点。**解决**：d 或 alpha（随机跳转概率是 1-d）。**why?** 其中的(1-d)/N 的部分相当于以一定等概率被选中作为新起点，由此跳过了陷阱和黑洞的干扰（所有节点全联通）。**收敛?** 马尔可夫过程的三个收敛条件（转移矩阵 A 所有元素非负，且每一列元素和为 1，为马尔科夫矩阵；Restart 保障图是强连通，则 A 不可约；A 非周期）Pagerank 算法都满足。

取$A = dM + [(1-d)/N]ee^T$，其中M为跳转矩阵，e为所有元素都为1的列向量

则有$P_{n+1} = AP_n$，形成马尔科夫过程

解特征方程（A-I）x=0 且$\sum x_i$=1 得概率 x。(λ=1)

PageRank 拓展 1.个性化:

个性化PageRank中，用户视作网络中的一个虚拟节点

首先，任何跳转行为均从用户开始，即用户自身作为浏览行为的起点

其次，从用户出发的跳转概率不一定视作均等（如1/N）

- 可根据用户对不同网页的偏好决定其概率
- 基本公式形如 $r = (1-d)Mr + dv$ 的形式
  - v体现了用户偏好，可视作不同网站在用户偏好下的点击概率

2. 主题敏感：仅有同行的评价才最有价值

对于某个Topic来说，起点仅限于隶属于该Topic的网站

$(1-d)e/N$更改为了$(1-d)s/|s|$，s为一个N维向量

- 如果某个网站隶属于该主题，则 s 中的该维为1，反之为0
- |s| 表示 s 中为 1 的元素的个数。

3. Hiltop 算法：非从属组织网页(从属组织网页：IP 前三个字段相同，URL 中的主域名段相同),专家页面(包括 K 个"指向的所有页面相互非从属组织"的出边),依赖专家页面相关性得分

PageRank 优点：可离线完成；全局排序；结果通用。**缺点**：主题无关，难区分恶意链接和广告；旧网页得分高；不能单独用，需与相关性排序结合；效率问题。

HITS：**权威网页 Authority**：某个领域或话题相关的高质量网页。**中心网页（枢纽 Hub）**：指向很多高质量的权威网页。

$$\forall p, a(p) = \sum_{i=1}^{n}h(i) \quad \forall p, h(p) = \sum_{i=1}^{n}a(i)$$

假定邻接矩阵为M，Authority向量为a，Hub向量为h

则有如下迭代式：

- $a_{k+1} = M^Th_k$，$h_{k+1} = Ma_{k+1}$

或者，可采用如下迭代式：

- $a_{k+1} = (M^TM)^kM^Ta_0$，$h_{k+1} = (MM^T)^{k+1}h_0$
- 其中，a_0，h_0为Authority/Hub向量的初始值，可设置为全1向量

每次迭代都要归一化！

$a_{k+1} = M^TMa_k, h_{k+1} = MM^Th_k$。

解特征方程（A-λI）x=0 且$\sum x_i^2$=1 得概率 x。

HITS 优点:更好地描述互联网组织特点；主题相关，可单独使用；**缺点**：需要在线计算，时间代价大；链接敏感。

评价

指标：性能/效率/结果多样性/权威性/时新性与更新频率

单查询评价 (1)无序结果评价

P/N:算法对样本的判断;**T/F**:算法判断的正确与否

准确率 Precision: TP/(TP+FP)

召回率 Recall: TP/(TP+FN)

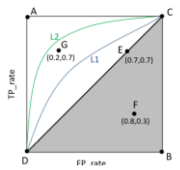
召回率的近似 缓冲池 Pooling:将各算法给出检索结果中的 TopN 文档汇集并人工标注，得到相关的文档池

F-measure：准确率与召回率的调和平均

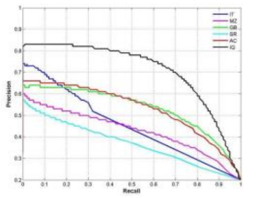
$F = \frac{1}{\alpha\frac{1}{P}+(1-\alpha)\frac{1}{R}} = \frac{(\beta^2+1)PR}{\beta^2P+R}$ 取 alpha=0.5 或 belta=1 得 F=2PR/(P+R)

通过绘制不同阈值下的指标变化曲线可选择合适阈值。

ROC 曲线：真正率 TP/(TP+FN)和假正率 FP/(FP+TN)作为轴线，兼顾正负 对角线上无区分能力（即随机猜测） 对角线上端越远越好 低于对角线无意义（无区分度）



P-R 曲线：x 轴召回率，y 轴准确率，不同阈值不同曲线。（只考虑正例）



判别算法好坏：(1)计算曲线下面积 AUC，越高效果越好 (2)PR 曲线平衡点（P=R）值越高越好

(2)有序结果评价

P@N:前 N 个检索结果文档的准确率；**R@N**:前 N 个检索结果找回的相关文档比例；**R-Precision**=P@相关文档总数；**平均准确率 AP**：不同召回率点上的正确率平均

- 未插值AP：某个查询Q共有6个相关结果，排序返回了5篇相关文档，其位置分别是第1，第2，第5，第10，第20位，则AP=(1/1+2/2+3/5+4/10+5/20+0)/6
- 插值AP：事先选定插值点数并进行插值。例如，当我们计算11点平均时，计算在召回率分别为0,0.1,0.2,...,1.0的十一个点上的正确率平均值
- 简化AP：只对返回的相关文档进行计算，AP=(1/1+2/2+3/5+4/10+5/20)/5，倾向那些快速返回结果的系统，没有考虑召回率和补零的情况

（召回率为 0 点的正确率为 1）

(3)相关度分级

累计增益 CG：位置 1 到 p 的检索结果相关度之和；**折损累计增益 DCG**:若把相关度高的文档排在后面给惩罚；$DCG_p = rel_1 + \sum_{i=2}^{p}\frac{rel_i}{log_2(i)}$（ppt 用第一个）

$DCG_p = \sum_{i=1}^{p}\frac{2^{rel_i}-1}{log_2(i+1)}$

或

归一化折损累计增益 NDCG=DCG/iDCG，其中 iDCG 是根据文档相关性从大到小排序得到理想化的最优序列，并对此序列计算 DCG 值所得到的。

假设有10个文档，相关度为0-3之间，10 个文档的得分依次如下:

- 3, 2, 3, 0, 0, 1, 2, 2, 3, 0

理想的输出结果序列为：3, 3, 3, 2, 2, 2, 1, 0, 0, 0

- 由此计算iDCG依次为：3, 6, 7.89, 8.89, 9.75, 10.52, 10.88, 10.88, 10.88, 10.88

而与此同时，基本的DCG结果如下：

- 3, 5, 6.89, 6.89, 6.89, 7.28, 7.99, 8.66, 9.61, 9.61（单调非减特性）

由此可得 NDCG结果如下：1, 0.83, 0.87, 0.76, 0.71, 0.69, 0.73, 0.8, 0.88, 0.88

- 可以看到任何查询结果位置的NDCG 值都规范化为[0,1]之间的值

多查询评价

MAP：对所有查询的（简易）AP 求算数平均（可以反映全部查询的综合效果，但在查询难度不平衡的条件下有误导）

GMAP: $GMAP = \sqrt[n]{\prod_{i=1}^{n}AP_i} = \exp(\frac{1}{n}\sum_{i=1}^{n}\ln AP_i)$

MRR：多个查询所得的倒数(第一相关文档)排序求平均 例：两个查询,第一个的第一个相关文档在位置 2,第二个的第一个相关查询在位置 4,则 MRR=(1/2+1/4)/2=3/8 即平均在 8/3 的位置上找到第一个相关文档

ERR：表示用户需求被满足时停止的位置的倒数的期望

$$ERR = \sum_{r=1}^{n}\frac{1}{r}PP_r = \sum_{r=1}^{n}\frac{1}{r}R_r\prod_{i=1}^{r-1}(1-R_i)$$

其中 Rr 表示位置为 r 的文档的相关度

方差：一个检索系统对不同查询的方差通常大于多个检索系统对相同查询的方差，说明查询难度差异大。

实体识别

信息抽取基本任务：命名实体 NE(实体抽取)、模板元素 TE(属性抽取)、共指关系 CR(等价)、模板关系 TR(关系抽取)、背景模板 ST(事件抽取)

知识图谱 基本单位：三元组（实体-关系-实体）

节点表示概念和实体；边表示关系和属性

应用：语义搜索、问答系统、推荐系统

拓展类型：事理图谱（描绘逻辑社会，研究对象是谓词性事件及其内外联系），多模态图谱（属性/实体多模态）基于路径的可解释图谱：提升多样性和可解释性!

命名实体识别 NER 子任务：判别实体边界；判别实体类型 **识别内容**：实体类（人名/地名/机构名），时间类（日期/时间），数值类（货币/百分比）

难点：新实体，歧义，别名/缩略词/音译，实体类型多样

方法(1)基于词典:简单快速，与具体语境无关，容易部署和更新；难枚举所有命名实体名，构建和维护词典代价大，难有效处理实体歧义**(2)基于规则**:手工构造规则模板，选用特征包括统计信息/标点符号/关键字/指示词和方向词/位置词/中心词(依赖具体语言、领域和文本风格，代价大)**(3)基于统计**:分类（词性标注）、序列模型

评价方式：Precision/Recall/F-value，分子为"返回的正确答案数量（+1/2 的部分正确答案数量）"

实体对齐、**实体消歧**<-语义表征

数据准备

数据挖掘方法：(1)分类：有监督学习，面向预定义的类别 **(2)聚类**：没有预先定义的类别，借助相似度度量自动生成 **(3)关联规则 (4)异常检测**

事务型数据 Transaction data：一条记录对应一个项目(Item)集合(无序)

关联规则：分析事务型数据，从而根据一部分项的存在记录来判断另一部分项目是否同时存在于事务中

基本形式：A->B，A、B 均为集合

支持度为{A+B}在全体事务中的比重 s(A->B)=(|A ∪ B|)/N，**置信度**{A+B}占 A 出现的事务中的比重 c(A->B)= (|A ∪ B|)/|A|。

例：考虑{Diaper,Milk->Beer}关联规则

$$s=\frac{(\text{Milk, Diaper, Beer})}{|\text{T}|}=\frac{2}{5}=0.4$$

$$c=\frac{(\text{Milk, Diaper, Beer})}{(\text{Milk, Diaper})}=\frac{2}{3}=0.67$$

TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

频繁项集：支持度高于**阈值**的项目集合 A

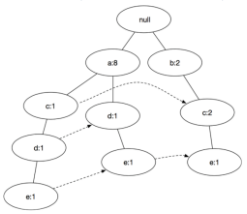
频繁项集生成方法（1)最基本的：穷举所有可能集合，计算支持度，繁杂度过高!

(2)Apriori（剪枝思想，但仍是"生成-测试"范式）：逐步减去所有的非频繁项集，然后基于频繁项集生成其超集。（先验原理：如果一个项目集是频繁的，那么它的所有子集也是频繁的。则非频繁项集的所有超集也是非频繁的。）

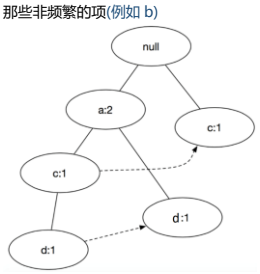
(3)FP-Growth：本质是输入数据的压缩表示，通过逐个读入事务，并将事务映射到 FP 树中的某条路径来构造。

●建树：对各个项按支持度排序，将排序后的项集逐步读入并建立树状结构，对相同项节点采用指针连接，方便快速访问。

●基于 FP-树，生成频繁项集：◇FP-Growth 本质上是自底向上的探索。首先查找以 e 为结尾的频繁项集，其次是 d/c/b，最后是 a，包含 e 的路径如下：



◇判定 e 本身是否为频繁项集（此处设阈值为 2，高于阈值），将 e 的前缀路径转化为条件 FP 树（需要更新路径上的支持度计数，只有包含 e 的事务会被统计），删除那些非频繁的项(例如 b)



◇考虑更长结尾的频繁项集的子问题，以{de}为结尾的频繁项集判定为例，在前一张图上统计与 d 相关的支持度求和为 2，为频繁项集；以 de 为结尾，得到前缀路径如下，通过其条件 FP 树发现{ade}支持度为 2，也频繁。

异常检测：异常数据≠错误数据，而是包含不同寻常规律的数据。

(1)基于分布：前提是识别数据集的具体分布，错误识别会导致错误检测；基于一元正态分布的**离群点判定**:已知参数的前提下根据正态分布判定离群的概率，舆情监测。

(2)基于度量：基于距离或密度的方式，检测空间中远离大多数数据点的离群点。●求数据点到 K 最近邻的平均距离：高于阈值则判定为异常点 ●基于密度：采用 K 近邻距离的倒数作为密度或给定半径内点的个数

(3)基于聚类●抛弃远离其他簇的小簇，但簇的个数将影响结果。●先聚类所有对象，再评估对象属于簇的程度（点到簇中心的距离或相对距离）

数据预处理 数据质量问题：(1)数据测量、采集等过程中出现的错误(2)噪声、离群点 (3)缺失数据(->删除和填补并重) (4)重复数据(多源数据归并:实体歧义,多马甲账号)

数据聚合 多源数据归并时的问题：不同属性名称、单位尺度、属性统计方式、不同数据源的统计不一致性

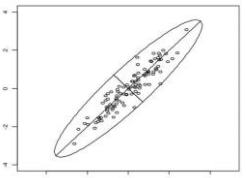
解决：(1)换算和汇总 (2)实体对齐

数据采样 采样代表性；启发式采样规模确定方法（分组采样，组内数据高度相似，不同组对象差异大，每组至少取一个）

数据规约

维度规约：删除不具有区分度的特征，可能降低噪声，避免维度灾难的同时使模型更容易理解，也易可视化

降维方法 1 主成分分析 PCA：通过正交变换将一组可能存在相关性的变量转换为一组线性不相关的变量，转换后的这组变量叫主成分。多维椭球：找出主轴与几个最长的轴作为新维度。选择投影方差最大的轴。



最大特征值对应的特征向量可以最大化投影方差。

完成第一个方向基选择后，第二个方向基应选择使其不存在相关性，故优化目标为协方差为0，即 $\frac{1}{m}\sum_{i=1}^{m}a_ib_i=0$。等价于协方差矩阵$C=\frac{1}{m}XX^T$对角化，即除对角线（方差）外的其它元素化为0。故优化目标转变为寻找矩阵P，满足PCP^T是一个对角矩阵。

最大 K 个特征值的特征向量对应的线性组合是 K 个新指标（K 个特征值的比重反映了主成分的信息量，一般>0.85）

输入：n维样本集$X=(x_1,x_2,\cdots,x_m)^T$，要降维到n'

输出：降维后的样本集Y

(1) 对所有样本进行中心化$x_i=x_i-\frac{1}{n}\sum_{j=1}^{n}x_j$

(2)计算样本的协方差矩阵$C=\quad XX^T$ 归一化

(3)求出协方差矩阵的特征值及其对应的特征向量

(4)将特征向量按特征值大小从上到下按行排列成矩阵，取前k行组成矩阵P →原数据中心化后

(5)$Y=PX$即为降维到k维后的数据

PCA 特点（1)依赖原始变量也只能反映原始变量 (2)PCA 内在假设之一是原始变量直接存在一定关联。若原始变量相互独立则降维失败，数据越相关，降维效果越好(3)PCA 的结果未必清晰可解释

降维方法 2 特征子集选择（而不是归纳新特征）去除冗余特征和不相关特征，或为特征赋予不同权值

数据离散（将连续属性变换为分类属性）

(1)二元化将连续或离散属性转化为多个二元属性 0/1

(2)非监督离散化 不用类别信息（等宽/等频率/等深）

(3)有监督离散化（基于熵）

先进行二分，选择熵最小的点进行分割。对其中具有较大熵（即纯度不高，信息混乱）的部分再下一轮分割。

分类

基于规则的分类　规则分类器的基本形式：

规则 condition（属性组合，前提）->标签

规则分类器的基本原理：

互斥原理（每条记录至多被一条规则所覆盖）&穷举原理（每条记录至少被一条规则所覆盖）

规则分类器的有序性 基于规则的排序方案：按照规则的质量（如准确性）进行排序；基于类的排序方案：同类规则排在一起，相对顺序被忽略。

如何制定规则分类器：

(1)直接从数据中自动学习规则：RIPPER、CN2

●**顺序覆盖**：贪心

算法开始时，决策表为空，即不包含任何规则

每一步针对某个类 y，提取覆盖当前训练集的最佳规则

- 什么是好的规则？覆盖的样本尽可能的多，同时样本类别尽可能一致

如果规则能覆盖大多数正例（即 y），而没有或覆盖极少负例（非 y），则保留

- 将该规则加入决策表的尾端，同时删除该规则覆盖的所有训练样本

重复上述过程，直至满足终止条件（例如：某个增益的阈值，如精或准确率）

规则增长策略：

"从一般到特殊"：初始规则条件为空，给定目标标签 y，逐步加入合取项(AND 相连)来提高规则质量

"从特殊到一般"：随机选择一个正样本作为初始种子，逐步删除规则中的合取项，来覆盖更多的同类别正例

(2)间接借助其他分类模型学习规则：决策树

基于监督学习的分类

决策树：从根节点到叶节点的每一条路径都对应一条分类规则。

特征选择（有较强区分能力的特征）->生成决策树->决策树剪枝，避免过拟合

信息熵$Ent(D)=-\sum_{k=1}^{|y|}p_klog_2p_k$

信息增益$Gain(D,a)=Ent(D)-Ent(D|a)$

$=Ent(D)-\sum_{v=1}^{V}\frac{|D^v|}{|D|}Ent(D^v)$

$Gain\_ratio(D,a)=\frac{Gain(D,a)}{IV(a)}$

$IV(a)=-\sum_{v=1}^{V}\frac{|D^v|}{|D|}log_2\frac{|D^v|}{|D|}$

●**特征选择准则**(1)信息增益 Gain(偏好取值较多的属性) (2)信息增益率 Gain_ratio(引入惩罚项，但倾向于选择取值较少的属性）先从候选特征中找到信息增益高于平均水平的集合，再从中找到信息增益率最大特征 (3)基尼指数（一个随机样本被分错的概率，Gini 指数低则信息纯度高）

K 个类，样本点属于第 k 类的概率为 pk:

$\text{Gini(p)}=\sum_{k=1}^{K}p_k(1-p_k)=1-\sum_{k=1}^{K}p_k^2$　$Gini(D,A)=\sum_{i=1}^{N}\frac{|D_i|}{|D|}Gini(D_i)$

例如：$Gini=\frac{3}{7}(1-\frac{1}{9}-\frac{4}{9})+\frac{2}{7}(1-\frac{1}{4}-\frac{1}{4})=0.333$

●**决策树生成** 计算当前结点各个属性的信息增益率，基于最大信息增益率属性，迭代式对节点进行分类，直到某个节点的样本类别统一

●**剪枝** 预剪枝：每个节点划分前衡量当前节点的划分能否提高决策树的泛化能力（验证集上精度）

后剪枝：自底向上考虑每个非叶子节点被替换成叶子节点后能否提高泛化性能

最近邻分类 计算与未知样本与其他样本的距离找到 K-最近邻，基于 K-最近邻的类别确定分类结果。**动机**：表征空间上相似的文档是相似的；基于实例的学习，不需要对数据进行抽象（如提取特征）；消极学习，不需要模型，但开销大；基于局部信息判别，受噪声影响大。

距离度量：欧式距离，汉明距离（0/1 向量 统计多少维数字不同），余弦相似度，马氏距离，无穷范数（分量最大值） **K** 小则易受噪声干扰，**K** 大则错误涵盖其他类别样本。

支持向量机 SVM：二分类转化为寻找最大间隔超平面，实现对高维空间中节点进行有效分割，使得**超平面**和**支持向量**（离该超平面最近的点）的间隔最大化

$\arg\max_{w,b}\frac{2}{\|w\|}$

s.t. $y_i(w^Tx_i+b)\geq 1,\ i=1,2,\ldots,m.$

第一步：引入拉格朗日乘子$\alpha_i\geq 0$得到拉格朗日函数

$L(w,b,\alpha)=\frac{1}{2}\|w\|^2-\sum_{i=1}^{m}\alpha_i\left(y_i(w^Tx_i+b)-1\right)$

第二步：令$L(w,b,\alpha)$对w和b的偏导为零可得　求解对偶问题得到最优解α^*

$w=\sum_{i=1}^{m}\alpha_iy_ix_i,\quad \sum_{i=1}^{m}\alpha_iy_i=0.$　计算超平面：$\vec{w}=\sum_{i=1}^{N}\alpha_i^*y_i\vec{x_i}$

第三步：回代

$\min_{\alpha}\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_jy_iy_jx_i^Tx_j-\sum_{i=1}^{m}\alpha_i$

s.t. $\sum_{i=1}^{m}\alpha_iy_i=0,\ \alpha_i\geq 0,\ i=1,2,\ldots,m.$　$b^*=y_j-w^*\cdot x_j,\forall\alpha_j^*\neq 0$

支持向量是满足$\alpha_i^*\neq 0$的样例点

求解对偶问题即求极值时:(1)内部偏导均为 0,解方程组,检查解是否非负(2)边界也可取极值(比如单一分量为 0)

核函数：将高维空间下的内积运算转化为低维空间下的核函数计算。

软间隔:允许少数样本不满足超平面约束(引入惩罚项 C)

原始问题 $\min_{w,b}\frac{1}{2}\|w\|^2+C\sum_{i=1}^{m}\max\left(0,1-y_i(w^T\phi(x_i)+b)\right)$

对偶问题

$$\min_{\alpha}\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_jy_iy_j\phi(x_i)^T\phi(x_j)-\sum_{i=1}^{m}\alpha_i$$

$$\text{s.t. }\sum_{i=1}^{m}\alpha_iy_i=0,\ 0\leq\alpha_i\leq C,\ i=1,2,\ldots,m.$$

不平衡分类问题解决方案：(1)代价敏感学习：代价矩阵衡量将一个类错分到另一个类的代价，优化目标由准确/召回变为加权后的代价 **(2)抽样**：少数类过采样、多数类欠采样、K-最近邻用少数类样本生成新样本

聚类

基本问题：聚类依据？相似性度量？簇的数量？

方法：层次聚类/划分聚类

K 均值聚类(K-means)

输入：样本集 $D=\{x_1,x_2,\ldots,x_m\}$;
聚类簇数 k.

过程:
1: 从 D 中随机选择 k 个样本作为初始均值向量 $\{\mu_1,\mu_2,\ldots,\mu_k\}$
2: repeat
3: 令 $C_i=\varnothing\ (1\leqslant i\leqslant k)$
4: for $j=1,2,\ldots,m$ do
5: 计算样本 x_j 与各均值向量 $\mu_i\ (1\leqslant i\leqslant k)$ 的距离: $d_{ji}=\|x_j-\mu_i\|_2$;
6: 根据距离最近的均值向量确定 x_j 的簇标记: $\lambda_j=\arg\min_{i\in\{1,2,\ldots,k\}}d_{ji}$;
7: 将样本 x_j 划入相应的簇: $C_{\lambda_j}=C_{\lambda_j}\cup\{x_j\}$;
8: end for
9: for $i=1,2,\ldots,k$ do
10: 计算新均值向量: $\mu_i'=\frac{1}{|C_i|}\sum_{x\in C_i}x$;
11: if $\mu_i'\neq\mu_i$ then
12: 将当前均值向量 μ_i 更新为 μ_i'
13: else
14: 保持当前均值向量不变
15: end if
16: end for
17: until 当前均值向量均未更新
输出: 簇划分 $\mathcal{C}=\{C_1,C_2,\ldots,C_k\}$

$SSE=\sum_{i=1}^{K}\sum_{x\in C_i}dist^2(m_i,x)$　(mi 为簇 Ci 的中心)

初始中心的选择(1)K 均值的后处理：清除较小的可能代表离群点的簇，对 SSE 高的簇拆分，对 SSE 低的合并(2)**二分 K 均值聚类**：先分为 2 个簇，再不断选择一个分裂**空簇的处理**：将最大 SSE 的簇拆分，或选择一个最远样本点新生成一个簇。

缺点：易受离群点干扰，当簇存在不同规模、密度及不规则形状的情况下，K 均值聚类效果较差。

层次聚类

优点：不需预设簇的数量，结果有意义对应到分类学目录上；**缺点**：局部最优，每步合并决策都是最终的

凝聚式聚类：(1)合并邻近度最高的两个簇 (2)基于更新的簇重新计算邻近度，更新邻近度矩阵

邻近度(1)**单链 MIN**：不同簇最近的点之间的邻近度（擅长处理非椭圆形状的簇，但对噪声比较敏感） (2)**全链 MAX**：不同簇最远的点之间的邻近度（对噪声不太敏感，但可能使得较大的簇变得支离破碎） (3)组平均：所有来自不同簇的两点之间的平均邻近度 (4)中心距离：两个簇中心之间的邻近度 (5)沃德法：合并后簇中各点到新中心的距离平方和。

分裂式聚类：(1)二分 K 均值聚类 (2)最小生成树聚类：由差异矩阵生成一颗最小生成树(节点之间权值最小)，每步断开差异最大的一条边，从而创建一个新的簇。

MST 聚类结果与单链凝聚聚类的结果相同。

基于密度聚类

密度：样本一定半径的样本数量

DBSCAN 算法 (1)将所有点区分为核心点(半径内样本数超过阈值的点）、边界点(非核心点但处于稠密区域边界内/上)和噪声点(处于稀疏区域的点）(2)删除噪声点 (3)将所有距离在预定半径内的核心点之间连一条边 (4)连通的核心点形成一个簇 (5)将所有的边界点指派到一个与之关联的核心点所在的簇中

优点：对噪声鲁棒；**缺点**：密度变化大的簇受影响

模糊聚类（计算归属改为计算从属度）

$SSE=\sum_{j=1}^{k}\sum_{i=1}^{N}w_{ij}(x_i-c_j)^2,\sum_{j=1}^{k}w_{ij}=1$

从属度由$w_{ij}\in 0,1$改为$w_{ij}\in[0,1]$。

聚类问题的评估 非监督评估：(1)基于邻近度矩阵(理想结果为"簇内点邻近度全为 1，簇间为 0"，呈对角模式)；(2)凝聚度(簇内两两点的邻近度之和 or 簇内个点到簇中心的邻近度之和)与分离度(簇间两两点的邻近度之和 or 簇中心到其他簇中心的邻近度之和) **有监督评估：**(1)分类度量：熵、纯度（簇在多大程度上包含单个类的对象，以最多类的比例计算）、准确率、召回率、F 值 (2)相似性度量：分类对应的矩阵，同一类的样本对应的元素为 1,不同类为 0，比较两矩阵相关性

推荐系统

推荐评估：(1)评分均方根误差 RMSE $\sqrt{\sum_{xi}(r_{xi}-r_{xi}^*)^2}$ (2) 分类（推荐正确/错误 :Precision/Recall/F ；TopN:Pre@N,Rec@N) (3)排序评估

基于内容的推荐：物品画像&用户画像(基于用户评分进行加权)采用余弦相似性度量进行评分

优点：用户推荐过程独立，个性化，推荐结果可解释

缺点：难以提取物品特征，难建立新用户画像

多样化评估：最大边界相关性 MMR

基于路径推荐：知识图谱代替向量化画像，基于图谱上的游走实现推荐，路径可作为推荐的依据。

偏见 Bias 问题：位置(第一更受关注)，模态(与众不同的模态吸引关注)，关键词效应(标题党的威力)

双向选择问题：用户<->物品，稳定匹配

协同过滤 本质：矩阵补全

基于内存(Memory-based)：

User-based: (预测时忽略空值)

$$sim(a,b)=\frac{\sum_{p\in product(P)}(r_{a,p}-\bar{r}_a)(r_{b,p}-\bar{r}_b)}{\sqrt{\sum_{p\in product(P)}(r_{a,p}-\bar{r}_a)^2}\sqrt{\sum_{p\in product(P)}(r_{b,p}-\bar{r}_b)^2}}$$

$$pred(a,p)=\bar{r}_a+\frac{\sum_{b\in neighbors(n)}sim(a,b)\cdot(r_{b,p}-\bar{r}_b)}{\sum_{b\in neighbors(n)}sim(a,b)}$$

Average rating of user b

Item-based

$$r_{ix}=\frac{\sum_{j\in N(i;x)}s_{ij}\cdot r_{jx}}{\sum s_{ij}}$$

(无需平均分修正)

相似度计算时，如果未评分，直接设为 0，不用减平均数。

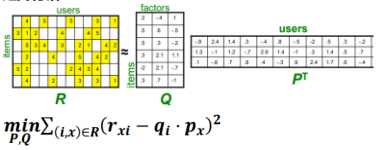
基于物品的推荐效果更好？物品属性单一，用户偏好多样。

基于内存**优点**：不受多模态非结构化信息表征与特征选取困扰；**缺点**：冷启动、稀疏性、热度偏差。

冷启动问题：(1)先非个性化推荐 (2)借助个人信息或其它网站浏览信息 (3)诱导式推荐迭代收集用户反馈 (4)与基于内容混合 (5)Side information：众包文本、知识图谱

基于模型(Model-based)：潜在因子

矩阵分解 Matrix Factorization:



$\min_{P,Q}\sum_{(i,x)\in R}(r_{xi}-q_i\cdot p_x)^2$

解决过高的 K 带来的过拟合？引入正则项，避免过大参数值。

$$\min_{P,Q}\underbrace{\sum_{training}(r_{xi}-q_ip_x)^2}_{\text{"error"}}+\underbrace{\left[\lambda_1\sum_x\|p_x\|^2+\lambda_2\sum_i\|q_i\|^2\right]}_{\text{"length"}}$$

(1)非负矩阵分解(2)概率矩阵分解(3)社交约束

社会网络

基本元素：节点（网络中实体），有向/无向边（关系），邻居/出入度，连通性/连通组件。

节点角色：意见领袖、结构洞（作用：为组织引入外部信息，衡量方式：聚集系数（任意两好友也互为好友的概率）低）

链接预测 三元闭包

(1)两节点间存在边的概率，与它们共同好友的个数呈正比（如何削弱好友个数影响？对指标进行正则化）

(2)共同好友作为中介的引荐力度：还要考虑好友的好友数

(3)考虑"共同好友"的好友：基于多条关系的链接预测

$s_{xy}^{Katz}=\sum_{l=1}^{\infty}\beta^l\cdot|paths_{xy}^{<l>}|=\beta A_{xy}+\beta^2(A^2)_{xy}+\beta^3(A^3)_{xy}+\cdots$

社团挖掘：

边介数：网络中任意两点的最短路径，有多少条会通过该边

(1)**基于层次聚类**

Girvan-Newman 算法：计算网络中所有边的边介数；去除边介数最高的边；重新计算去除边后网络中所有边的边介数；跳至步骤二，直至网络中没有边存在

(2)基于划分聚类：K-means，谱聚类(基于最小割，面向图)

关系抽取

关系抽取方法 (1)基于规则：纯手工定值规则，文本匹配"模板"和"槽"；难制定规则，领域专家构筑大规模知识库，领域移植困难。

(2)基于模式：从种子关系中获得模式，再由模式寻找更多种子，迭代优化。（基于字面匹配，移植性差，适合特定关系）

双重迭代模式关系提取 DIPRE:

模式的Order和Middle，即为 Occurrence 集合的Order和Middle

模式的URLPrefix、Prefix、Suffix，分别为Occurrence集合中最长的公共URL前缀与前、后缀。

- 其他部分采用通配符填充

Snowball:仅信任支持度(满足每个模式的元组数量)和置信度高(符合该模式的元组确实符合相应关系的概率)的模式

(3)基于机器学习：转化为分类问题，训练模型求解

开放关系抽取 (1)基于知识监督 (2)基于句法

远程监督 如果某个实体对之间具有某种关系，那么所有包含这个实体对的句子都是用于描述这种关系；目的：获取足够数量的、高质量的标注

语义漂移:不是所有包含该实体对的句子都表达该关系且错误不断放大 **优化:动态转移矩阵**描述各类相互标错的概率;**规则学习**设计相应否定模式列表去除错误的标签;**注意力机制**

事件抽取的模板：选定相应模板后，通过事件元素(事件参与者)与事件元素角色 (事件元素在事件中充当的角色) 的识别，将相应的元素填入模板合适的槽（描述命名实体基本信息，内容可包括名称/类别/种类）内