web data mining

- 词袋表示：文档由文档中出现的集合所表示。*词袋表示形成需要哪些步骤？这种表示的优缺点是什么？* 优点简单，缺点无序集合，句法信息丢失。
  - 符号化：识别词边界，英文大小写转换。（也叫分词）
  - 词形还原：删除词语的时态、单复数。
  - 取词根：删除后缀。
  - 过滤停用词：不具有内容的词。
- 余弦相似度：查询向量 q，文档向量 d，长度 n。
  - $similarity(q,d) = \frac{q \cdot d}{||q|| \cdot ||d||}$
- 倒排索引：关键词表头，链表后续为包含关键词的文档标号，及频率、位置等。*查询多个关键词，如何匹配？优势是什么？* 优势为关键词个数少于文档个数，检索效率高。
- 布尔检索模型：优点简单，对查询严格掌握。缺点，一般用户很难构建，检索结果文档无法排序。严格匹配导致过少、过多的结果。
- Web 搜索架构：以下两步在线下完成，线上完成用户的查询。
  - web 页面采集，目标为有效收集到更多有用的 web 页面，包括链接结构。
  - web 页面排序，比仅返回相关页面更重要。与查询相似度最高的网页不一定是最好的。排序基于相关度（查询与页面内容的相似程度）或重要性（基于链接分析，A指向B的链接很重要），综合排序考虑以上两个因素。
- PageRank 算法：用户随机游走，随即点击链接到达某网页的可能性。
  - 转移矩阵 $P = \{p_{ij}\}$
  - $if\ outlink[v_i] \neq 0, p_{ij} = \frac{M(i,j)}{\sum_{v_k \in outlink[v_i]} M(i,k)}$ *看上去十分复杂，实际上就是对于点 i 的所有出边，等概率地流动。如果没有出边当然是0.*
  - 最开始 $\pi(v_i) = \frac{1}{n}$，有 n 个点的话每个点都是等概率。$\pi = P^T\pi$ 直到收敛，得到 rank 值。*有一些特殊情况导致这样搞不行，所以我们添加了自环，以及连向每个顶点的边。这些不在原图里的边按照 $1-\alpha$ 的概率转移。John 上课讲过的那个。*
  - 优化的乘法：$\pi = \alpha P^T\pi + (1-\alpha)\frac{1}{n}e, e = (1,1,\dots,1)^T$
- HIT 算法：
  - authority, 权威值，到达它的点 hub 的和。
  - $a^{(k+1)}(v_i) = \sum_{v_j \in inlink[v_i]} h^{(k)}(v_j), or\ a = M^T h$
  - hub, 枢纽值，它到达的点 authority 的和。
  - $h^{(k+1)}(v_i) = \sum_{v_j \in outlink[v_i]} a^{(k)}(v_j), or\ h = Ma$
- 信息检索评价指标MAP的计算
  - REL_q：*relevant documents for **q** 说的是查询 q 的正确答案*
  - recall level：正确答案里对了多少个，precision 值是找到 n% 相关文档时的 precision 值。标准 recall level：0%, 10%, … , 90%, 100% 这 11 个
  - interpolation：检索结果不对应标准 recall level，例如 $REL_q = \{d8, d56, d89\}$ 则 recall level 分别是 0.33, 0.66, 1.0。r_j 表示标准 recall level，$P(r_j) = max_{r_j \le r \le r_{j+1}} P(r)$. *一种规范性措施，数字不整，强行规范到 10 为倍数，并且搞成单调递减* 。范围应该是从当前到最右边 100% 中的最大值。
  - $MAP = \sum_{q \in Q} \frac{\sum_{r=1}^{|REL_q|} \frac{P_q(r/|REL_q|)}{|REL_q|}}{|Q|}$

- 关联规则挖掘过程与 Apriori 算法
  - 关联规则表示项之间的关系
  - 数据包含：`TID(transaction id) | Basket(subset of items)`，所有项的集合 $I=\{i_1,i_2,\ldots,i_m\}$，事务 $T\subset I$，关联规则 $A\to B, A\subset I, B\subset I, A\cap B=\emptyset$ *规则的意思是子集 A 和子集 B 共同出现在事务中可能性较高，可以根据该规则做预测*
  - 支持度，表示规则的有用性。D 中**同时包含 A B** 的事务数除以总事务数，*搞搞清楚这个集合的并集和平时的定义不一样。*$sup(A\to B)=\frac{||\{T\in D|A\cup B\subset T\}||}{||D||}$
  - 置信度，表示规则的确定性。$conf(A\to B)=\frac{||\{T\in D|A\cup B\subset T\}||}{||T\subset D|A\subset T||}$，同时包含 A B 与只包含 A 的比值。*希望 A 单独出现尽可能少，总是和 B 一起出现*
  - ppt4 p50，一道题，自己搞搞清楚。
  - 频繁项集：满足最小支持度的项集。强规则：满足最小 sup 和 conf 的规则。*搞清楚求频繁项集的时候，还没有筛选出蕴含式，只需要把频繁出现的子集搞出来。**从项集生成规则**后用 conf 筛选出强关联规则*
  - naive 的做法，m 个项，n 个事务，复杂度为 $O(2^m n)$。Apriori 算法的思想：定理，若 A 频繁项集，则其每个子集也是频繁项集。做法是从小往大生成项集，由k-项集生成 (k+1)-项集。通过**连接**产生候选，再用 Apriori 性质删去不频繁子集的候选（**剪枝**）。ppt4 p61
- 朴素贝叶斯分类算法
  - $h=argmax_{h\in H}P(X|h)P(h)$，给定数据 X，h 是类别，要预测 X 属于哪一类。$P(X|h)$ 是 h 条件下，发生 X 的概率。
  - $P(X|C_i)=\Pi_{k=1}^{n}P(x_k|C_i)$，将特征划分为 k 个属性。
  - 最后根据贝叶斯公式得到 $P(h|X)$ 的概率。
- K近邻分类算法
  - 用余弦相似度的方法找到新文档的**k 个近邻向量**，根据近邻向量的类别确定该文档的类别。基于**投票机制**，将新文档划分到 k 个近邻文档中多数文档所属类别。**加权投票机制**，按照相近程度赋予一定权重。*一个是谁多谁说了算，另一个加上相近这个权重。*
- 分类与回归的联系与区别：*ppt里面没具体写，大致一个是离散值，另一个是预测两变量关系的连续值，最小二乘法就是回归*
- K 均值聚类算法
  - **随机选择** k 个种子点作为初始中心点
  - 将每个文档**指派**到与其最相近的中心点类簇
  - 根据当前类簇文档重新计算中心点
- 凝聚式聚类算法
  - 初始时**每个文档形成一个类簇**，每次合并最相近的两个类簇，形成一个新类簇，循环执行，直到满足终止条件。
  - 评估两个类簇相近的方法是最大、最小或平均距离。
- 结合 K 均值与凝聚式聚类的 Buckshot 算法
  - 从原始 n 个中选**根号 n** 个数据点，运行凝聚式聚类，**用其结果作为初始种子点**，在原数据上基于初始种子点运行 K-means
- 半监督聚类之 COP K-means 算法
  - 用户提供了 must-link 和 cannot-link 约束。初始化时，类簇中心随机选择，但要保证 must-link 的两个数据点不能成为不同类簇的中心。算法在归属每个点的类簇时不能违反任何约束，归属到相邻的类簇。
- 自然语言处理领域的歧义现象
  - 自然语言本身歧义：句法歧义（组合歧义，咬死了猎人的狗；结构起义，句法成分不明，进口彩电），语义歧义（多义词），语用歧义（修辞、反语，该来的没来）
  - 中文分词的歧义：切分歧义（交集型，AJB 中 AJ JB 都是词。组合型，AB 中 A B AB 都是词）真歧义（乒乓球拍卖完了）

- 正向最大匹配分词与逆向最大匹配分词
  - 正向，不断找最长的词切分，找不到则 I--，接着找。
  - 逆向，从后往前找，比 FMM 更有效。1/245 错误率。
- 无向图度数中心性、中介中心性与亲近中心性的计算（未规范化与规范化）
  - 度数中心性，即度数，规范化则除以最大可能值，如 N-1
  - 中介中心性，多少点对的最短路径经过给定节点。$C_B(i) = \sum_{j<k} g_{jk}(i)/g_{jk}$，$g_{jk}$ 为节点 j k 间最短路径的数量。规范化，$C'_B(i) = C_B(i)/[(n-1)(n-2)/2]$。*排除掉 i 这点后的点对数目*
  - 亲性中心性，某节点与其他节点最短路径平均值的倒数。*衡量一个节点和其他节点的亲近程度，距离的平均越小权重越大，规范化需要取平均。* $C'_c(i) = [\frac{\sum_{j=1}^{N} d(i,j)}{N-1}]^{-1}$
- 基于图排序(PageRank)的文档摘要方法
  - 依赖于句子相似度，句子作为顶点，有关系则构建边，用 PageRank 算法获得每个顶点的权重，基于权重选择句子形成摘要。
- 基于句子分类的文档摘要方法
  - 二分类：句子是否属于摘要。实现用 SVM。*评价标准这块是不是得看一下*
- 基于PMI的情感词汇**获取方法**及文本情感**分类方法**
  - 具有倾向性（肯定，否定）的词倾向于在文档中共同出现；主观性形容词倾向于出现在其他主观性词周围。PMI 定义，$pmi(x,y) = log\frac{p(x,y)}{p(x)p(y)}$，或利用 NEAR 操作计算：$pmi(t,t_i) = log_2\frac{hits(t\ NEAR\ t_i)}{\#(t)\#(t_i)}$，$t$：目标词，$t_i$：种子词
  - 基于 PMI 预测倾向性：$\sum_{t_i=Pos} PMI(t,t_i) - \sum_{t_i=Neg} PMI(t,t_i)$
  - 基于 PMI 的情感分类：抽取包含 adj/adv 的两词短语，计算语义倾向，$SO(phrase) = PMI(phrase, pos) - PMI(phrase, neg)$，文档语义倾向为所有短语语义倾向的平均值
- 观点抽取的目的和主要步骤
  - 目的：给定观点文本，抽取五元组 o_j(目标对象), a_{jk}(对象o_j的特征), so_{ijkl}(观点表达的情感值), h_i(观点持有者), t_l(表达的时间)，将无结构文本结构化。
  - 两个子任务，特征抽取与聚类（抽取对象所有的特征表达，并将同义特征聚类。包括频繁特征抽取；非频繁特征抽取；有监督学习，采用序列标注模型三种）、特征情感分类（确定观点对于**每个特征**的情感倾向，对特征分别判断，这是它与情感分类的区别；输入 (f,s); f为产品特征，s为包含f的一个句子。输出 s中针对f的观点倾向）。
- 基于用户/物品的协同推荐算法
  - 基于用户：思想，过去对**物品购买、评分一致**的用户很可能再次一致。使用相似用户的意见预测特定用户对于一个物品的意见，相似性通过用户对其他物品的意见吻合程度衡量。
  - 用户相似性：a）相关系数，r_i, r_j为矩阵第i，j行向量，*注意 items 是 i 和 j 都有评价的，如果其中一个空着就不算它* $w_p(a,i) = \frac{cov(\mathbf{r}_i, \mathbf{r}_j)}{\mathbf{std}(\mathbf{r}_i)\mathbf{std}(\mathbf{r}_j)} = \frac{\sum_{k\in itmes}(r_{ik}-\overline{r_i})(r_{jk}-\overline{r_j})}{\sqrt{\sum_{k\in itmes}(r_{ik}-\overline{r_i})^2}\sqrt{\sum_{k\in itmes}(r_{jk}-\overline{r_j})^2}}$ b）余弦度量 $w_c(i,j) = \frac{\mathbf{r}_i \cdot \mathbf{r}_j}{||\mathbf{r}_i||_2 \times ||\mathbf{r}_j||_2}$
  - 预测算法：a）使用整个矩阵，预测用户 a 对物品 j 的评价。*我们参考其他用户对物品 j 的意见，如果其他用户和 a 越像，那么他意见占的权重越大*

$p_{a,j} = \overline{v}_a + k\sum_{i=1}^{n} w(a,i)(v_{i,j} - \overline{v}_i), k - normalizer$ b）K近邻算法。*a 和 b 的区别是一个用了整个矩阵的用户意见，另一个只用了最近的 k 个吗*

- 基于物品：一个用户对**相似的物品**有相同的评分。物品相似性用其他用户对物品的评分意见吻合程度衡量。r_i, r_j为矩阵第i，j列向量，*注意 users 是对 i 和 j 都有评价的用户，如果其中一个空着就不算它*$s_{ij} = \frac{\sum_{u \in users}(r_{ui} - \overline{r_i})(r_{uj} - \overline{r_j})}{\sqrt{\sum_{u \in users}(r_{ui} - \overline{r_i})^2}\sqrt{\sum_{u \in users}(r_{uj} - \overline{r_j})^2}}$
- 预测算法：为一个物品找到 k 个最相似的，用户对该物品的评价为**该用户**在相似物品评价的加权平均。$r_{aj} = \frac{\sum_{i \in similar} s_{ij} r_{ai}}{\sum_{i \in similar} s_{ij}}$

	基于用户	**基于物品**
优点		物品相似性比用户相似性稳定
用户冷启动	不足以确定新用户的相似用户	能够较好处理
物品冷启动	不能为新物品预测物品评分	
数据稀疏	物品数量多时，用户只评价少数物品，难以找到相似用户	
扩展性	规模变大时计算慢	

- 基于矩阵分解的协同推荐算法
  - 将评分矩阵分解为两个矩阵，基于分解结果可得到（原来矩阵中不存在的）用户对物品的评分。*目标函数这个自己百度搞清楚吧。*。目标函数为：$minimize_{p,q} \sum_{(u,i) \in S} (r_{ui} - < p_u, q_i >)^2 + \lambda[||p||^2_{Frob} + ||q||^2_{Frob}]$

- 智能问答系统架构
  - 基于语料库的自动问答为三步，问题处理，段落检索，答案提取。基于知识库的自动问答多一步知识库构建。
  - 问题处理：查询构建（从问题中抽取关键词项），问题分类（根据期待的答案类型分类）
  - 段落检索：从返回文档集中抽取**潜在的**包含候选答案的文本段集合。过滤（排除不包括候选答案的段），排序（根据包含答案的可能性）。
  - 答案提取：从段落中抽取特定答案，经典方法有二：N 元短语排列（罗列出段落或摘要中所有的 n-gram，n=1,2,3，根据与期待答案类型的匹配程度打分，选出分数较高的），模板匹配（选择符合答案类型模板的）

- 思考：
  - 对当前主流中文互联网搜索系统有所了解、比较与思考。
  - 思考社交媒体对搜索技术的挑战以及解决办法。
  - 思考移动搜索的挑战与关键技术。
  - 面向特定领域设计一个智能问答系统。
  - 面向特定领域的知识库构建及其应用。