Lec3
机械分词
按照一定的策略将待分析的汉字串与一个“充分大的”机器词典中的词条进行匹配，若在词典中找到某个字符串，则匹配成功，分为正向最大 FMM 反向最大 RMM 双向最大 BMM
最少切分（最短路径）使句子中切分的词数最少
将每个字视作节点，前后自然连接边，词形成一条边，即寻找有向图最短路径
N gram 概率模型 N = 2，Bigram 模型，
P(w)=P(w1)P(w2|w1)...P(wn|wn-1)
基于序列标注的分词方法
四类标注：B（词的开始）、M（词的中间）、E（词的结束）、S（单字词）
• 例子：中国科学技术大学是中国最好的大学
• 标注：BMMMMMME S BE BME BE
• 分词结果：中国科学技术大学 / 是 / 中国 / 最好的 / 大学

Lec4
布尔检索 文档被表示为关键词的集合 查询式被表示为关键词的布尔组合
优点：查询简单，易于理解
缺点：所有匹配文档均返回，不考虑权重和排序
实现布尔搜索：关联矩阵 词项向量
关联矩阵太大且稀疏 使用倒排索引
倒排索引：词汇表->倒排表
建立：
1.遍历文档 获得词项-文档 ID 对 写入临时索引
2.对临时索引按词项排序
3.遍历临时索引，合并相同词项
查询：查询的过程就是合并索引的过程
优化：
1.合并顺序 AND 短 OR 都行 先估计 OR 后大小再依次 AND
2.跳表指针 a<b 若 a 有跳表指针 则比较 a 的跳表指针与 b 否则直接跳到 a 的下一位
跳表指针设置：高内聚低耦合

索引存储：
顺序 简单，二分查找慢
哈希 快，可能冲突
B+树 相对快，维护复杂
Trie 树 快，用空间换时间近似完全 m 叉树

存储压缩：
词典压缩，放入内存
方法：视作单一字符串，词项指针指向开始位置
按块存储，增加一个字节在字符串开头，指示单词长度，k 个词项共用一个词项指针
索引压缩，减少占用，减少读取时间，更多移入内存
方法：
1.存储间距代替文档 ID 3 5 11 -> 3 2 6（效率问题？）
2.将文档 ID 变长编码表示间距 先写出 2 进制表示，从左向右，每 7 位插入一个 0，存放在一个字节中，最后一块高位补 1 存放在一个字节。5->10000101

Lec5
相关性反馈：
用户在查询后标记相关/不相关，然后迭代更新查询，以获得更好的结果
相关性反馈流程：
用户提出查询 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

Rocchio 算法动机：
使得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

Rocchio 算法：
new query vector= α *original query vector + β *positive feedback vector- γ *negative feedback vector
Typically, $\beta > \gamma$ because positive feedback is more important
feedback vector 表示所有反馈回来的文档的向量平均

情境感知的查询理解：
动机：用户查询存在歧义、用语精简，缺乏精确性，借助情境信息协助判断用户意图
情境信息：所有与人机交互相关，用于区分标定当前特殊场景的信息，例如搜索的上下文

Lec6
向量空间模型及前置方案
Jaccard 系数
两个集合 A、B Jaccard(A,B) = |A∩B| /|A∪B|
未考虑词项频率与文档长度，罕见词信息量更大
词项频率 TF
Tf(t,d)，指此项 t 在文档 d 中的出现次数
相关性与频率不是线性相关，为了抑制数量级影响，引入对数
对数词频 wf(t,d) = 1 + log10tf(t,d) if tf(t,d)>0 else 0
Wf 衡量词项 t 在文档 d 中的出现频率
文档频率 DF，Df(t)指出现词项 t 的文档数量
Idf(t) = log10(N/df(t)) N 是文档总数
Idf 表示词项 t 在所有文档中的罕见度
tf-idf：Wf(t,d) = (1 + log10tf(t,d)) * log10(N/df(t))
Wf*idf 用于衡量文档 d 与词项 t 的相关程度
向量空间模型 VSM
每个文档和查询都表达为 M 维度的向量，M 表示词项总量。其中每一维度的值等于该词项在此文档/查询中的 tf-idf 值。用余弦相似度衡量向量的相似程度 cos(q,d) = q · d/|q|*|d| 值越大越相似
优点：简洁直观，支持多种不同度量或权重方式，实现效果不错
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

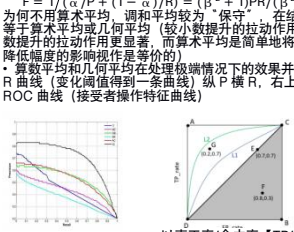
衡量网页权威性：PageRank 算法：
核心思想：将网页视作点，网页间的超链接视作有向边，从而形成一个巨大的有向图。网页入度越多，网页被引用和被推荐就越多，重要性就越大。
计算公式：PR(pi) = (1 − d)/N + d * $\sum_{pj \in M(pi)}$ PR(pj)/L(pj)
• PR(pi)为网页 pi 的 PageRank 值
• PR(pj)为指向网页 pi 的某个网页 pj 的 PageRank 值
• L(pj)为网页 pj 发出的链接数量
• d 为阻尼系数，取值在 0-1 之间
• N 为网页总数，M(pi)为链入 pi 的页面集合
计算过程：先给每个网页赋初始值，然后用公式迭代计算，得到近似结果

陷阱节点与终止节点会导致上面的过程收敛到平凡结果，孤立节点也有问题
为此加入重启动机制，即计算公式中的(1-d)/N 的部分，相当于以一定概率重新选择起点，跳出了陷阱与黑洞的干扰，d 一般选择 0.85 左右。
加入重启动后具体计算：
取 A = dM + [(1 − d) /N]$_{nn}$。其中 M 为跳转矩阵，跳转矩阵=邻接矩阵转置后，将各个数值除以所在列的非零元素数（即出边数量），e 为所有元素都为 1 的列向量
为 P0 赋初值，迭代过程变为P_{i+1} = A * P_n，收敛后即为 PageRank 值。
这样的 A 满足以下收敛性条件：
• 转移矩阵 A 为马尔科夫矩阵：A 矩阵所有元素都大于等于 0，并且每一列的元素和都为 1
• 转移矩阵 A 为不可约的：当图是强连通时，A 为不可约，而 Restart 保障了这一条件
• 转移矩阵 A 为非周期的
优点：1. 对于网页给出全局的重要性排序，并且可以离线完成以保障效率 2.基础的 PageRank 算法本身独立于检索主题，可以实现结果的通用
缺点：1.主题无关性，同时对于恶意链接，广告链接等缺乏区分 2.旧网页得分更高，因为新网页往往少有入链 3.一般不能单独用于排序，需要与相关性排序方法相结合 4.效率问题

Hits 算法核心概念：
• 权威（Authority）网页与枢纽（Hub）网页的区分
• 权威网页：指某个领域或某个话题相关的高质量网页
• 中心网页：类似中介，指向了很多高质量的权威网页
HITS 的目的即在海量网页中找到并区分这些与用户查询主题相关的高 质量 “Authority” 与 “Hub” 网页，尤其是 “Authority”
两个基本假设：
• 基本假设 1：好的 Authority 会被很多好的 Hub 指向
权威值 a(p) = $\sum$ h(i)
• 基本假设 2：好的 Hub 会指向很多个好的 Authority
枢纽值 h(p) = $\sum$ a(i)
• 因此，在 HITS 算法中，每个网页 p 需要计算两个值
计算过程：
• 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：$a_{k+1} = M^T h_k$，$h_{k+1} = M a_{k+1}$
• 其中，a0，h0 为 Authority/Hub 向量的初始值，可设为全 1 向，每一步注意归一化

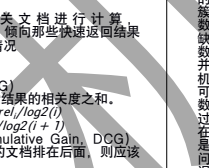
优点：1.更好地描述互联网组合特点 2.主题相关，因此可以单独用于网页排序
缺点：1.需要在统计算，时间代价较大 2.对链接结构变化敏感，且依然可能受到“链接作弊”的影响
其他衍生算法：个性化 PageRank（为了体现用户偏好） 主题敏感 PageRank（个性化代价太大，为了减少计算） Hilltop 算法（避免 PageRank 被滥用）

Lec7,评估
面向单个查询的评价指标：
• 无序/二元结果：Precision、Recall、F-value…
• 有序/多元结果：P@N、R@N、AP、NDCG…
TP 真正，FN 假负，FP 假正，TN 真负
准确率，查准率（precision）TP/(TP+FP)
召回率，查全率（recall）TP/(TP+FN)
Accuracy (TP+TN)/(TP+TN+FP+FN)为什么被抛弃----不返回任何结果 accuracy 很高
难以得到相关文档准确数目，用缓冲池方法进行召回率的近似计算，针对某一检索问题，各算法给出结果中的 Top N 文档，汇集起来人工标注得到相关文档池，假设大多数相关文档在这个文档池中
F 值，即准确率与召回率的加权调和和平均数，取 α =0.5 或 β =1 时，F1=2PR/(P+R)。
$F = 1/(\alpha/P + (1-\alpha)/R) = (\beta^2+1)PR/(\beta^2 P + R)$
为何不用算术平均，调和平均较为“保守”，在结果上小于等于算术平均或几何平均（较小数提升的拉动作用应比较大数提升的拉动作用更显著，而算术平均是简单地将相同提升/降低幅度的影响视作是等价的）
• 算数平均和几何平均在处理极端情况下的效果并不够合理
R 曲线（变化阈值得到一条曲线）纵 P 横 R，右上凸起下降
ROC 曲线（接受者操作特征曲线）



• 以真正率/命中率【TP/(TP+FN)】和假正率/误报率【FP/(FP+TN)】作为两条轴线，
• 通过选定不同的相关性阈值得到不同的真正率-假正率点并连接成线，
• 对角线表示区分能力为 0，即随机猜测，在对角线上端越远，效果越好，低于对角线的结果无意义

基于 P-R 曲线或 ROC 曲线判别算法好坏
• 如果线 A 将线 B 完全包住，显然线 A 对应的算法效果更好
• 如果两条线发生重合，则可依据以下规则判别：
• 计算 AUC（曲线下面积），更高者效果更好
• 当使用 P-R 曲线时，可使用 Precision = Recall 的点，值越高越好
P-R 和 ROC 比较：
• ROC 曲线兼顾正负样例，更为全面，而 P-R 曲线则只考虑正例
• 正负样本比例失调时，P-R 曲线更合适。当负样本比重过高时，负例的数目众多致使 FPR 的增长不明显，导致 ROC 曲线呈现一个过分乐观的效果估计，从而难以体现出性能的差异性
• P-R 曲线受分布影响大，多份数据且正负比例不一时 ROC 曲线更合适



如果相关文档数小于 N，P@N 的理论上限必定小于 1
由于返回结果有限，Recall@N 值，甚至其理论上限往往都远小于 1
R-Precision：N 取相关文档总数的 P@N
有序结果情况下，变化 N 产生 P-R 折线，新的不相关文档被检索时，Recall 不变，Precision 下降
平均准确率（Average Precision，AP）
• 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 0.0,1.0,2.0,…,1.0 的十一个点上的正确率来平均
• 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

相关度分级
1.累计增益（Cumulative Gain，CG）
• 用于衡量位于位置 1 到 p 的检索结果的相关度之和。
$CG_p = \sum_{i=1}^{p} rel_i$　$DCG_p = rel_1 + \sum_{i=2}^{p} rel_i/log2(i)$
$DCG_p = \sum_{i=1}^{p} (2^{rel_i} - 1)/log2(i+1)$
2.折损累计增益（Discounted Cumulative Gain，DCG）
基本思想：若搜索算法把相关度高的文档排在后面，则应该给予惩罚。
DCG 与具体查询和结果列表的长度 p 有关
归一化折损累计增益（Normalized DCG，NDCG）
基本思路：将 DCG 除以完美结果下得到的理想结果，iDCG（Ideal DCG） 即：NDCG = DCG / iDCG
多样性的两种衡量方法：
隐式模型：只计算文档之间的差异性
显式模型：更加具体地考量文档所对应的用户意图

Lec9 知识图谱
信息抽取，从语料中抽取指定的事件、事实等信息，形成结构化的数据
信息抽取：从文本中获取用户感兴趣的重要信息，借助于自然语言处理技术，通常领域相关（借助领域知识辅助抽取）
信息检索：从文档集合中找文档子集，通常利用统计与关键词等技术，通常领域无关
抽取三要素：确定关系
• 实体：即命名实体，指文本中的基本构成块，如人、机构等
• 属性：实体的特征，如人的年龄、机构的类型等
• 关系：实体之间存在的联系，也称事实，如公司和地址之间的位置关系
• 事件：实体的行为或实体参与的活动
5 类基本的信息抽取任务
命名实体 NE（实体抽取），模板元素 TE（属性抽取），共指关系 CR，模板关系 TR（关系抽取），场景模板 ST（事件抽取）

知识图谱的优点
知识图谱至少可从以下三个层面提升搜索的效果：
• 找到最想要的信息：将信息直接呈现，无需用户劳动
• 提供最全面的摘要：对搜索对象进行总结，提供更完整的信息和关联
• 让搜索更有深度和广度：构建完整知识体系，使用户获得意想不到的新发现
知识图谱的基本形式
• 由结点和结点之间的边组成，结点表示概念（或实体），边表示关系（或属性）数学表现为一个有向图。点和边组成知识图谱的基本单位：三元组（实体-关系-实体）

利用知识图谱提高推荐系统的多样性，可解释性，推荐性能（基于路径推荐？）
事理图谱，描述逻辑社会，研究对象是谓词性事件及其内外联系（区别），借助事理逻辑链接形成对于事件的推理（应用）
多模态图谱，实体和属性可能是多模态的（区别），表示与整合多模态知识（应用）

命名实体识别（NER）识别出文本中的人名、地名等专有名称，和有意义的时间、日期等数量短语等，并加以归类
两个子任务 1.判别实体边界 2.判别实体类型
与分词的难点非常相似：新实体，歧义，构成结构复杂，类型多样
性能评价采用 Precision / Recall / F-value
识别方法
：基于词典
• 优点：方法简单快速，与具体语境无关，容易部署和更新（只需更新词典）
• 缺点：难以覆盖全部实体，构建维护词典代价大，难以处理歧义
：基于规则（以模式和字符串相匹配为主要手段）
• 优点：当提取的规则能较精确地反映语言现象时，性能较好
• 缺点：1. 规则依赖具体语言、领域和文本风格 2. 代价大大，系统建设周期长、移植性差，需要建立不同领域知识库
：基于统计
• 四类标注：B（词的开始）、M（词的中间）、E（词的结束）、S（单字词）
分支一：基于分类的命名实体识别方法
• 将 NER 视作一个多分类问题，通过设计特征训练分类器的方法加以解决
分支二：基于序列模型的命名实体识别方法
• 与分词中的序列标注方法思路类似，区别在于标注的不同

相关的问题：实体对齐（多词一义，如药物不同名称），实体消歧（一词多义，如苹果）

Lec10 关系抽取
概念：从文本中识别出两个实体（或多个实体）之间存在的事实上的关系
意义：搜索引擎发现和关联知识的重要渠道，知识库构建与知识关联的基础性手段，是支持问答系统、推荐系统等应用的有力工具
关系抽取方法 1.基于模板 2.基于模式 3.基于机器学习

基于模板的优缺点：
• 针对特定问题可设置针对性规则，但可能比较困难
• 需要专家和知识库，代价大
• 特定领域效果好，移植性差

基于模式的关系抽取：
首先由种子关系生成关系模式，然后基于此抽取新关系，从中选择可信度高的作为新种子，不断迭代至无新关系或模式

代表方法 1：DIPRE，大致思路如上
基本元素：元组，如<Foundation, Isaac Asimov> —<Title, Author>
模式，如 ?x，by ?y 的形式（可表示“title” by “author”）
基本假设，元组广泛存在各网页中；各部分往往位置接近，表示时，存在重复“模式”

代表性方法 2：Snowball，在于对 DIPRE 算法的提升
• 仅信任支持度和置信度较高的模式，从而保证模式质量
支持度（Support），即满足每个模式的元组的数量，删除少于一定数量元组支持的模式
置信度（Confidence），符合该模式的元组，确实符合相应关系的概率

基于模式方法的优缺点
• 适合某种特定具体关系抽取，如校长、首都关系。
• 基于字面匹配，没有引入深层信息，如词性、句法、语义等。
• 移值性差，必须为每一个具体的关系生成自己的识别模式。

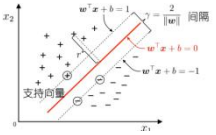
Lec 11 数据准备
数据挖掘（Data Mining）
基本含义，从海量数据中提取或挖掘潜在的知识和规律，用于支持当前的判断或未来的决策
过程：数据准备，数据建模，知识表示
目的：揭示满足以下条件模式或模型：有效性，实用性，解释性，意外性

数据挖掘方法
分类：给定一组有标签记录作为训练集，目标在于训练一个合适的模型，使其能够有效地区分无标签的新数据，将其归为合适的类别，有监督学习，面向预定义的类别 lec12
聚类：作为无监督学习的代表，没有预先定义的类别，而是借助相似性度量自动生成 lec13
关联规则：事务型数据是一类特殊的数据记录，一条记录往往对应着一个项目的集合（无序）。用户需要的信息往往并非孤立存在，而是彼此关联
关联规则，旨在分析事务型数据，从而根据一部分项目的存在记录，来判断另一部分项目是否同时存在于事务中
3. 关联规则的基本形式：A → B，A、B 均为集合形式
支持度：{A+B}在全体事务中的比重 s(X → Y) = σ (X ∪ Y)/ σ (N) 置信度：{A+B}占 A 出现的事务中的比重 s(X → Y) = σ (X ∪ Y)/ σ (X)
频繁项集：即支持度高于阈值的项目集合 A
6. 判断频繁项集：穷举所有的可能集合，并计算其支持度
d 个项目对应着 2^d 个（d+1）个潜在的频繁项集。复杂度过高！
7. Apriori 方法：先验原理，如果一个项集是频繁的，那么它的所有子集也是频繁的，非频繁的项集，其所有超集也是非频繁的
算法过程：从单个元素的项集开始，不断删除非频繁项集，逐渐组建含有 2 个、3 个元素的项集（树状）（需要理解）
FP-Growth 方法：输入数据的压缩表示，本质是一种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
异常检测，旨在发现与大部分其他对象不同的数据。离群检测主要方法：
1. 基于分布，最基本方法，一元正态分布离群点判定
2. 基于度量，K 近邻或者例数或半径内点个数
3. 基于聚类，先聚类所有对象，再评估对象属于簇的程度，往往采用以下两个指标 1.点到簇中心的距离 2. 点到簇中心的相对距离（与所有点到中心距离的中位数之比）从而调整簇密度对距离造成的影响
数据预处理，数据质量问题：测量采集错误，噪声、离群、缺失、重复
数据聚合：将两个或多个对象合并成为单个对象。目的：归并多个数据源的数据到统一格式下，解决部分数据重复，动机：减少空间和时间开支，对象群属性比个体更稳定，问题：可能会有不同属性名称，单位尺度，统计方式，统计不一
数据采样：选择一部分数据对象的子集进行分析，动机：通过采取小规模样本，起到近似效果同时降低大规模数据开支，在要求精确的场合，小规模样本初步分析了解数据特性，也是有效的。
问题：采样缺乏代表性将影响对原数据集的还原程度，产生误导
采样数据应至少在均值和方差等统计指标上近似原数据集，方法：例如随机采用：无放回、有放回、分层采样
样本容量：大有代表性但降低收益，小收益高但损失大
维度归约：删除不具有区分度的特征，可能降低噪声，避免维度灾难，同时模型更易理解和可视化
维度灾难/维度诅咒，指随着数据维度的增加，数据分析困难程度大幅上升的现象
可能原因：计算量指数级增长；数据稀疏，没有足够数据建模

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。通过这种方式，可以采用较少的综合指标综合先前存在于各个属性（且相关）中的各类信息，而综合指标之间彼此不相关
最大特征值对应的特征向量可以最大化投影方差。要求得数据样本的最大的 K 个特征值，其特征向量所对应的线性组合就可以形成 K 个新的综合指标。K 个特征值的比重反应了主成分的信息量，一般应大于 0.85。
归约会导致损失
数据离散将连续属性变换为分类属性
最基本：二元化，目的在于将连续或离散属性转化为一个或多个二元属性
非监督：不用类别信息而采用数据本身特性进行分类，等宽等频率等深等
有监督：定义某个区间的熵，总熵为加权和，判定区间内熵最小的分割
Lec 12 分类
基于规则：使用一组 if-then，互斥原理，穷举原理
具有较好可解释性和直观性。生成较为简便，分类较为迅速，如果面临连反互斥原则的情况，可能涉及多余规则投票，相对复杂
基于有监督学习分类
决策树（Decision Tree）
过程：
特征选择：选取有较强区分能力的特征
生成决策树：基于选定特征逐步生成完整的决策树
决策树剪枝：简化部分枝干，避免过拟合
信息熵（Entropy）表示随机变量不确定性的度量，不确定性越高，熵越高
$$Ent(D) = -\sum_{k=1}^{|y|} p_k \log 2 p_k$$
信息增益 Gain(D, a)
$$Gain(D,a) = Ent(D) - \sum_{k=1}^{K} (|D^v|/|D|) Ent(D^v)$$
$$Ent(D|a) = \sum (|D^v|/|D|) Ent(D^v)$$
特征选择准则：选择最大信息增益、最大信息增益率或基尼指数
信息增益偏向取值更多，增益率选取值少，折中后先找信息增益高于平均的，再从中找信息增益高的
基尼指数表示样本集合中一个随机样本被分错的概率，基尼指数越低，表明被分错的概率越低，相应的信息纯度也就越高。选择基尼指数最低的特征
$$Gini(p) = \sum_{k=1}^{K} p_k(1-p_k) = 1 - \sum_{k=1}^{K} p_k^2$$
决策树生成过程
计算当前节点类别信息熵 • 计算当前节点各个属性的信息熵，进而计算得到该属性对应的信息增益率 • 基于最大信息增益率的属性，对节点样本集合进行分类 • 重复上述过程，直至节点对应的样本类别统一，可以采用信息增益或者基尼指数
剪枝：剪枝防止过拟合和欠拟合，
预剪枝：生成过程中剪枝，判断划分后的节点能否提高泛化能力（验证集）
后剪枝：生成后自下而上，将节点换为叶子后能否提高泛化性能（验证集）

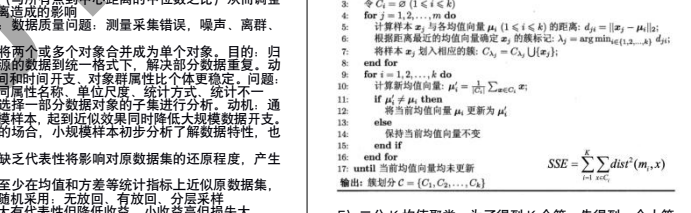
最近邻分类
K-最近邻分类：
输入：训练样本集合与距离度量和 K 值，用于限定最近邻的数量
流程：给定未知样本，首先计算与其他样本的距离，找到 K-最近邻，基于 K-最近邻的类别确定分类结果
距离度量：欧式距离，汉明距离（0/1 向量），余弦相似度，马氏距离，无穷范数等
$$D(x,x') = \sqrt{\sum_i \sigma_i^{\wedge}2 (x_i - x_i')^{\wedge}2}$$

K 取值：太小容易受噪声干扰，太大可能导致错误涵盖其他类别样本
消极惰学习：不需要模型但分类开销大；受噪声影响大；需要根据选择处理数据
支持向量机（SVM，support vector machine）
仅考虑二分类问题，那么分类问题可以转化为寻找一个超平面，实现对于样本空间中的节点进行有效分割
选择“正中”的最大间隔超平面，容忍性好，泛化能力强。
符合这样条件的超平面，在线性可分的条件下“存在且唯一”
其中，w,b 为参数，w 向量方向垂直于超平面。离超平面最近的节点（带圈的部分）被称作“支持向量” 有超平面方程如右：$w^T x + b = 0$



支持向量机的目的在于找到这样一个最大间隔超平面，使得超平面对应的间隔 γ 最大化
借助拉格朗日对偶性，通过求解对偶问题加以求解。
具体而言，求解方式可采用序列最小优化算法（SMO）进行求解
线性数据，某些样本在低维空间时线性不可分，通过非线性映射将其映射到高维空间的时候则线性可分，核函数的目的在于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，从而避免高维空间可能遇到的“维度灾难”问题。穷举法或混合使用挑选核函数
软间隔问题：引入“软间隔”允许少数样本不满足超平面约束，防止过拟合或无法确定超平面或核函数问题
不平衡分类问题
解决方案（1）代价敏感学习，引入代价矩阵，衡量将一个类错分到另一个类的代价，最终优化目标由原先的准确/召回变更为加权后的代价
解决方案（2）抽样方法，基于采样的方法，通过改变样本分布来缓解不平衡问题
过采样：提升少数类的样本比例，例如从少数类中进行重复随机采样
欠采样：降低多数类的样本比例，例如用多数类的采样样本而非全样本
混合采样：同时采用前两种策略
也可以利用已有的少数类样本，通过 K-最近邻生成新样本，来样中更注意噪声问题，噪声也可能被复制多次
Lec 13 聚类
簇，将样本表征为高维向量，在向量空间中，相似样本将自发地形成“簇”结构，簇内相似（距离较近），簇间相异（距离较远）
聚类目的：将样本分为若干个簇，每个簇都由若干相似样本所组成，无监督学习

常见方法：
将聚类方法分为层次的（嵌套）与划分的（不重复）两种
1. K 均值聚类
$$\vec{\mu}(C) = \frac{1}{|C|}\sum_{\vec{x} \in C} \vec{x}$$
A) “质心”就是一系列点（文档）的重心
设定 K 个中心，来形成 K 个簇，不断更新簇中心的向量，来更新聚类的结果，直至收敛。簇中心的更新，依赖于对当前簇中样本的算术平均；簇中心更新后，根据距离将样本重新分配至不同的簇；收敛：所有样本的聚类结果不再更新，停止迭代
B) 初始中心：可以随机，不同中心可能导致不同结果；也可以选少数样本层次聚类（开支大，k 要小）
C) 复杂度：K 均值聚类的复杂度为 O(n * K * I * d) 与样本数量 (n)，簇数量 (K)，迭代次数 (I)，向量维度 (d) 相关
D) 效果：平方误差和（SSE），K 增加时 SSE 一般下降，尽量同等 K 比较 SSE，某个 K 和 SSE 都较小的聚类，显然优于 K 和 SSE 都较大的聚类
其中 m 为簇中心，x 为簇中样本
E) K 均值的后处理：提升质量，方法包括：
1) 清除较小的，可能代表离群点的簇 2) 对较为“松弛”（如 SSE 较高）的簇进行拆分 3) 对较为“紧凑”（如 SSE 较低）的簇进行合并 4) 也可引进一个新的中心，或将一个簇完全打散（重新划分）通常，重新选择的簇中心是距离所有簇中心最远的点


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 $C_i = \varnothing$ $(1 \le i \le k)$
4: **for** $j = 1, 2, \ldots, m$ **do**
5: 计算样本 x_j 与各均值向量 μ_i $(1 \le i \le k)$ 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 $\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$

$$SSE = \sum_{i=1}^{k} \sum_{x \in C_i} dist^2(m_i, x)$$

F) 二分 K 均值聚类：为了得到 K 个簇，先得到一个大簇，然后不断选择所有簇其中一个使用 2-均值分裂，每次分裂出一个新簇，可以选样本数较大的簇或 SSE 较高的簇，直到分裂到指定数的簇为止。受初始中心影响小，因为是逐步求精过程
G) 处理空簇：以样本为初始中心，或选择最远样本新生一簇，或拆分最大 SSE 的簇
H) 增量更新：某个样本重新聚类时就更新中心，好处是无空簇，但是导致次序依赖和大开支
I) 局限性：1) 簇不同规模、密度、不规则形状情况下效果差 2) 容易受到离群点干扰，可以采用偏大的 K，然后进行合并

2. 层次聚类
A) 优势：不需要预设簇的数量；结果往往可以对应到具有一定意义的分类学目录上
B) 两种：凝聚式聚类（自下而上）分裂式聚类（自上而下）前者更常见，将所有样本视作个体簇，逐步合并最接近的两个簇；后者从包含所有样本的完整簇开始，每一步分裂一个簇
D) 凝聚式流程：
合并邻近度最高的两个簇
基于更新的簇重新计算邻近度，更新邻近度矩阵
取决于邻近度定义
单链（SL，MIN），指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
全链（CL，MAX），指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
组平均（Group Average），指所有来自不同簇的两点之间的平均邻近度，是前面两种方法的折中产物
中心距离，指来自不同簇的两个簇中心之间的邻近度，也可使用合并两个簇导致的 SSE 增加值等度量方式来衡量
沃德法（Ward’s method），指合并后簇中各点到新中心的距离平方和，计算复杂度较高 $d(C_i, C_j) = \sum_{x \in C_i \cup C_j} \|a - \mu_{ij}\|$
F) 局限性：1) 每一步的合并决策都是最终的，一旦做出合并两个簇的决策，就无法撤销 2) 没有全局的优化目标函数，每一步都是局部最优 3) 不同的聚类方法（邻近度定义），或多或少有问题，例如对噪声敏感，或难以保留较大簇等
基于密度聚类
基本假设：只有达到一定密度，才足以成为一个簇
密度：指定样本一定半径的样本数量
DBSCAN 算法：密度聚类的代表性算法
核心要素：三类不同的样本点
核心点：稠密部分内部的点（即指定半径内样本数超过阈值的点）
边界点：非核心点，但是处于稠密区域边界内/上的点
噪声点：处于稀疏区域的点
基本流程
将所有节点区分为核心点、边界点或噪声点
删除噪声点
将所有距离在预定半径内的核心点之间连一条边
连同的核心点形成一个簇
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，且可处理不规则形状
局限：如果簇的密度变化很大，效果可能会受到影响
先指定样本数阈值，之后挑选阈值最近邻距离出现拐点的位置作为半径
模糊聚类
$$SSE = \sum_{j=1}^{N}\sum_{i=1}^{N} w_{ij}^p (x_i - c_j)^2, \qquad \sum_{j=1}^{N} w_{ij} = 1$$
以一定从属度同时属于多个簇，模糊聚类时，目标函数可修正为 从属度 W 在[0, 1]. 每次更新样本从属度而非样本归属
对聚类问题评估：
有监督学习评估 Lec 7，此处无监督学习评估
动机之一：各类聚类方法得到的结果往往很不相同
非监督评估（或内部评估）仅使用数据本身的特性，不考虑任何外部标签信息
基于邻近度矩阵，理想结果：簇内点邻近度全为 1，簇间邻近度全为 0。实际邻近度矩阵与理想化矩阵的相似度，可用

于评估聚类结果。对于部分基于密度的聚类方法不适用，通过观察相似度矩阵是否体现出对角模式，可以大致判断结果好坏
凝聚度与分离度，可将簇内的邻近度定义为“凝聚度”，簇间的邻近度定义为“分离度”。凝聚度越高，分离度越低，聚类效果就越好。
基于图，凝聚度由簇内各点邻近度之和定义，分离度由簇间各点邻近度之和定义
基于原型，凝聚度由簇内各点到中心邻近度之和定义，分离度由簇中心到其他簇内中心（或点）的邻近度之和定义
有监督评估（或外部评估）引入外部信息，衡量与外部结果的匹配程度
面向分类的度量，有外部标签的情况下，可以借助分类手段进行度量，比如熵、纯度、准确率召回率 F 值等
面向相似性的度量，基于分类标签，也可以获得分类对应的“理想”矩阵。同一个类中的样本对应的矩阵元素为 1；不同类中的样本对应的矩阵元素为 0 。通过比较两个“理想”矩阵之间的相关性，可以近似估计聚类结果

Lec 14 推荐系统
信息爆炸，质量也良莠不齐
长尾现象：少数最热门条目所获得关注，远远高于尾部大批条目所获得的关注。跟风现象导致差距会进一步加大（马太效应）
本质上是矩阵补全问题
评估推荐效果：预测打分常用均方根误差（RMSE），还可以用准确率/召回率/F 值、Top N 结果采用 Pre@N 等。推荐物品受欢迎程度、推荐顺序、蓝房效应。RMSE 对于高分出色其他部分不佳的方法加以惩罚（没有关注到用户喜好）

$$\sqrt{\sum_{xi}(r_{xi}-r^*_{xi})^2}$$

基于内容推荐：
基本想法：用户的偏好一般相对稳定，推荐用户以前喜欢的物品
基本流程：根据用户喜欢物品，找到物品的特点，根据特点匹配物品推荐给用户
物品画像：为实现基于内容的推荐，对于每个备选物品，需要给出相应的画像，一般以向量存在
用户画像可由曾经评分过的商品画像所估计，一般采用加权平均的方式得到用户画像向量（基于评分进行加权）
基于用户与物品画像，可采用相似性度量进行评分，一般采用两个向量之间的余弦相似度（Lec6 有）
优点：
每个人的推荐过程相互独立，不需要其他用户的数据
可以为具有独特偏好的用户进行有效推荐，不受大众倾向性和热度的影响
可以推荐新物品或非热门物品
推荐结果有着较好的可解释性，可列举内容特征作为推荐的依据
缺点：
找到合适的特征是一件困难的事，对于非结构化信息尤其如此，部分特征的提取可能存在误导性
无法给新用户推荐物品
过度特化现象，永远只能给用户推荐局限于其画像中的内容——信息茧房问题，用户的多方面兴趣难以体现，难以通过他人的评价对推荐结果进行评估
衍生问题：
引入多样性评估解决过于集中推荐问题：最大边界相关性（MMR）
结构化知识图谱替代向量化画像，基于图谱上的游走实现推荐，路径可作为推荐的依据
推荐中的偏见问题，位置、模态、关键词（标题党）
双向选择问题，被选择对象存在名额限制，用户的偏好不一定能够得到满足，采用双向选择描述推荐过程更为合理，最后反应稳定匹配结果

基于协同过滤推荐：
基于内容的推荐方案只基于单一用户记录向该用户进行推荐，实际应用中其他用户的浏览、行为对当前用户有借鉴作用，协同过滤的思想在于基于矩阵的其他行，协助填补本行的空缺
基于内存
基于用户推荐，找到相似用户并基于历史行为推荐
基于共同评分的物品，衡量用户之间的相似性

基于物品，和用户基本一致，但是预测分时不用加平均
实践中往往基于物品的推荐效果更好：属性单一，受欢迎理由相对固定
优点：可适用任意种类；缺点：冷启动、稀疏性、热度偏差
冷启动解决方法：提供非个性化推荐收集数据、借助他处信息、诱导式推荐（代表性物品）、众包文本标注、知识图谱

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

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

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

Average rating of user b

基于模型
基于内存的推荐技术仅对数据进行简单处理，适用于各种数据，然而数据的稀疏性、计算最近邻的高复杂度，限制了其有效性
引入潜在因素表示用户偏好与物品属性
借鉴矩阵分解的思路，揭示潜在因子。评分矩阵 R 被近似视作物品属性矩阵 Q 与用户偏好矩阵 P 的乘积，P 与 Q 的维度，一方面与用户/物品的数量有关，另一方面体现了潜在因子的数量
当用户与物品的潜在因子已知，则任何缺失的评分，均可以通过对应的 P、Q 矩阵相应的行列运算估计得到
通过优化模型结果和真实之间的 SSE 获得潜在因子估计值
参数过多训练困难同时容易过拟合，引入正则项进行正则化避免过大参数值，将结果向原点拉。历史行为越少，越容易过拟合
部分情况添加非负约束保障非负性（文档主题属性有强弱无负分布）进行非负矩阵分解
概率矩阵分解在数据稀疏且有噪声的情况下，引入某些规律，可以实现对于参数更好的描述。比如参数符合高斯分布
扩展：加约束
社交约束：好友们在偏好与行为上十分相似。虽然效果好但不一定始终成立

Lec15.社会网络
将社会网络抽象表示为图结构
• 节点用于表示网络中的实体，如社会网络中的人
• 边用于描述网络中的关系，如人们之间的社交关系
网络中的边可能有向，也可能无向，各自表达不同含义
邻居集合 N(v)，度 dv

真实网络中的节点度数往往符合幂律分布：少数节点拥有大多数的边
连通性（强：双向可达）
连通组件，即一个连通的子图

节点角色：
1. “意见领袖” 2. “结构洞”，为组织引入外部的信息
一种结构洞判定方法：如果移除某节点会使网络变成多个连通组件，则该节点为一个结构洞
。假设可能过强，计算不便
另一种衡量方式：聚集系数
• 某个节点的聚集系数为：它的任意两个好友也互为好友的概率（比重）
• 聚集系数越低，该节点作为中介的作用越大
结构洞意义：各方沟通的桥梁，相应成为了“权力”

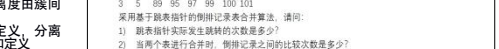
作业客观题：
1.1 请推荐如下查询的处理次序。
(tangerine OR trees) AND (marmalade OR skies) AND (kaleidoscope OR eyes)
其中，每个词项对应的倒排记录表的长度分别如下：词项 倒排记录表长度 eyes 213312 kaleidoscope 87009 marmalade 107913 skies 271658 tangerine 46653 trees 316812
考察知识点：倒排索引的优化
对于 OR 操作：顺序任意，考虑 O(x+y)的最坏情况
对于 AND 操作：先处理文档频率小的，再处理大的

OR操作的处理顺序可以任意，保守估计每个OR操作后的结果大小：

词项	最坏情况下的长度
tangerine OR trees	363465
marmalade OR skies	379571
kaleidoscope OR eyes	300321

因此对于AND操作，采用((kaleidoscope OR eyes) AND (tangerine OR trees)) AND (marmalade OR skies)的顺序处理。

1.2 考虑利用如下带有跳表指针的倒排记录表



和一个中间结果表（如下所示，不存在跳表指针）进行合并操作。
3 5 89 95 97 99 100 101
采用基于跳表指针的倒排记录表合并算法，请问：
1) 跳表指针实际发生跳转的次数是多少？
2) 当两个表进行合并时，倒排记录之间的比较次数是多少？
3) 如果不使用跳表指针，那么倒排记录之间的比较次数是多少？

1)
1 (24→75)

2)
3-3, 5-5, 9-89, 15-89, 24-89, 75-89, 75-89（while中），92-89, 81-89, 84-89, 89-89, 92-95, 115-95, 96-95, 96-97, 97-97, 100-99, 100-100, 115-101
19次

3)
3-3, 5-5, 9-89, 15-89, 24-89, 39-89, 60-89, 68-89, 75-89, 81-89, 84-89, 89-89, 92-95, 96-95, 96-97, 97-97, 100-99, 100-100, 115-101
19次

User interest	User occupation	Click
Tech	Professional	1
Fashion	Student	0
Fashion	Professional	0
Sports	Student	0
Tech	Student	1
Tech	Retired	0
Sports	Professional	1

1) 计算分别以属性 User interest 和 User occupation 划分时的信息增益，构建决策树将会选择哪个属性？
2) 计算分别以属性 User interest 和 User occupation 划分时的 Gini 指数，构建决策树将会选择哪个属性？

1）计算全体熵：

$$Ent(D)=-\frac{3}{7}\log_2(\frac{3}{7})-\frac{4}{7}\log_2(\frac{4}{7})=0.985$$

以 interest 划分时，增益为 0.306

$$\sum_v\frac{|D^v|}{|D|}Ent(D^v)=\frac{3}{7}(-\frac{1}{3}\log_2(\frac{1}{3})-\frac{2}{3}\log_2(\frac{2}{3}))+\frac{2}{7}*0+\frac{2}{7}=0.679$$

以 occupation 划分时，增益为 0.198

$$\sum_v\frac{|D^v|}{|D|}Ent(D^v)=\frac{3}{7}(-\frac{1}{3}\log_2(\frac{1}{3})-\frac{2}{3}\log_2(\frac{2}{3}))*2=0.787$$

选择信息增益最大的特征，即 interest
2）基尼指数的定义：假设集合 D 共有 K 个类别，则集合 D 的基尼指数为：

$$Gini(D)=1-\sum_{k=1}^{K}(\frac{|C_k|}{|D|})^2$$

假设以特征 A 把数据集 D 划分成 N 个子集，则针对特征 A，集合 D 的基尼指数为

$$Gini(D,A)=\sum_{i=1}^{N}\frac{|D_i|}{|D|}Gini(D_i)$$

选择基尼指数最小的特征，即 interest。选择信息增益最大的，选择基尼指数最小的

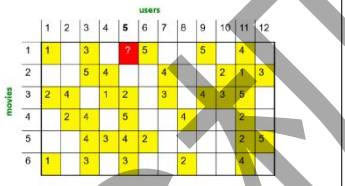
以 User Interests 划分时，

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

以 User Occupation 划分时，

$$Gini=\frac{3}{7}(1-\frac{1}{9}-\frac{4}{9})*2=0.381$$

1.4 在课件中，我们给出了如下评分矩阵：



4) 采用基于用户的评分预测方法（同样采用 2-最近邻），预测用户 5 对于电影 1 的评分，并与课件中给出的基于物品的评分结果进行比较。
5) 针对用户 5 对于电影 3 的评分，采用基于用户的评分预测方法，比较近邻数从 2 到 5 对于预测结果的影响，并简述选择合适近邻数的思路。
以上各题需要写出详细计算过程。

第一步：计算各用户的平均打分。

$\bar{r}_1$=(1+2+1)/3=1.333
$\bar{r}_2$=(4+2)/2=3
$\bar{r}_3$=(3+5+4+4+3)/5=3.8
$\bar{r}_4$=(4+1+3)/3=2.667
$\bar{r}_5$=(2+5+4+3)/4=3.5
$\bar{r}_6$=(5+2)/2=3.5
$\bar{r}_7$=(4+3)/2=3.5
$\bar{r}_8$=(4+2)/2=3
$\bar{r}_9$=(5+4)/2=4.5
$\bar{r}_{10}$=(2+3)/2=2.5
$\bar{r}_{11}$=(4+1+5+2+2+4)/6=3
$\bar{r}_{12}$=(3+5)/2=4

第二步：计算各用户与用户 5 之间的相似度，注意去中心化/个性化，只考虑两个用户都打过分的电影（忽略空值）。
以计算用户 1，用户 5 的相似度为例：用户 1 平均打分是 4/3，用户 5 平均打分是 7/2，二者都打过分的是电影 3 和电影 6。
所以计算相似度时，用户 1 的向量表示为（2/3, -1/3），用户 5 则是（-3/2，-1/2）

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

Average rating of user b

$$sim(1,5)=\frac{(2-\frac{4}{3})(2-3.5)+(1-\frac{4}{3})(3-3.5)}{\sqrt{(1/3)^2\times2+(2/3)^2}\sqrt{1.5^2+1.5^2+0.5^2+0.5^2}}=-0.456$$

$$sim(3,5)=\frac{(4-3.8)(5-3.5)+(4-3.8)(4-3.5)+(3-3.8)(3-3.5)}{\sqrt{0.8^2+1.2^2+0.2^2+0.2^2+0.8^2}\sqrt{1.5^2+1.5^2+0.5^2+0.5^2}}=0.214$$

$$sim(6,5)=\frac{(2-3.5)(4-3.5)}{\sqrt{1.5^2+1.5^2}\sqrt{1.5^2+1.5^2+0.5^2+0.5^2}}=-0.158$$

$$sim(9,5)=\frac{(4-4.5)(2-3.5)}{\sqrt{0.5^2+0.5^2}\sqrt{1.5^2+1.5^2+0.5^2+0.5^2}}=0.474$$

$$sim(11,5)=\frac{(5-3)(2-3.5)+(2-3)(5-3.5)+(2-3)(4-3.5)+(4-3)(3-3.5)}{\sqrt{1^2+2^2+2^2+1^2+1^2+1^2}\sqrt{1.5^2+1.5^2+0.5^2+0.5^2}}=-0.710$$

第三步：找到用户 5 的 2-最近邻，估计评分，注意去中心化，并加上用户 5 的平均打分。
最相似的两个用户为 3 和 9：

$$Pred(5,1)=3.5+\frac{0.214\times(3-3.8)+0.474\times(5-4.5)}{0.214+0.474}=3.5956$$

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

故预测用户 5 对电影 1 的评分为 3.5956。
课件中，基于物品的方法预测评分是 2.6，基于用户的方法预测评分更高。
注意：
1.平均值修正、去中心化。
2.基于用户的和基于物品的方法差异。

作业主观题：
2.1 在分布式爬虫中，往往通过对 URL 的哈希结果来进行任务分配。然而，往往因为服务器陷阱、节点崩溃等原因出现节点的减少或新增，如何设计更为有效的策略，在节点数量动态变更的情况下保障负载均衡？
可以采用一致性哈希，使用一个巨大的 hash key 空间，并组织成回路。URL 和目标网站 IP 都映射到一个 hash key 空间，每个 hash key 对应一台抓取计算机，如果某个爬虫节点失效，那么该机器中的 URL 都迁移到顺时针方向的下一个节点。

2.2 如何结合查询词项的分布细节，设计相对合理的跳表指针步长？
索引分布密集处使用较大的步长；索引分布稀疏的地方使用较短的步长

2.3 在信息检索系统中，如何同时使用位置索引（对倒排索引的位置信息扩展）和停用词表，潜在问题有哪些，如何解决？

可以先引入停用词表，在进行位置索引时，如果遇到非停用词，按照位置索引的方式进行记录，遇到停用词时则不进行记录。潜在问题是某些时候停用词有特殊含义无法被查询，对一些短语进行查询时没有相对位置信息进行检索（比如 to be or not to be 中 be 可能作为停用词无法被查询，从而难以从单词出现位置进行短语的判断。解决方案：用某一统一的符号代替所有停用词，这样可以进行较为模糊的停用词位置判断。

2.5 Trie 树的缺陷在于“以空间换效率”，对于存储空间的压力较大。如何结合英文/中文的语言特点，适当放宽限定，以节约 Trie 树的存储空间？同时，请分析这一改进下对于查询效率的影响

因为中/英文的词语/单词中，出现共同前缀的频率是比较高的。比如 t 开头的单词许多都是 tr 开头，所以其实在词汇表较小但需要动态变更的情况下，Trie 树的稀疏现象严重，空间利用率较低，所以考虑到上述原因，可以采用 double array trie 树的存储方式。如此一来，只需要一个加法一次比较即可完成一次状态转移，只需花费常数时间，极大地提高了单词搜索的效率

2.3 用户在浏览网页时，可能通过点击“后退”按钮回到上一次浏览的页面，用户的这种回退 行为 (包括连续回退行为) 能否用马尔科夫链进行建模？为什么？

不能。马尔科夫链的下一状态只与当前状态有关，回退行为需要记录之前的状态，两者矛盾。

2.4 在用户意图尚不明确的情况下，使搜索结局具有一定多样性，以确保可能具有不同意图 的用户都能够获得相应反馈，是一种常见的排序策略。请简要回答下列问题：
a) 如何在网页排序的同时提升结果的多样化水平？如何在此同时保障算法的效率？
修改优化目标，直接在将多样化指标写入目标函数，扩大召回率，增加精排、重排模块
b) 在用户通过点击行为等反馈方式表达了更为具体的意图之后，是否还需要保持结果的多样化？为什么？
两种都可以
需要，或许用户表明了更具体的意图但表达尚不完整，即用户还有部分内容尚未完整表达出来。
不需要，因为用户已经明确了要搜索的具体内容，而增加结果的多样化只会浪费用户的浏览时间，或者把用户引导到错误的内容中去

2.1 主成分分析的基本流程是什么？与特征值有何关系，为什么？
基本流程：
将坐标轴中心移到数据的中心，然后旋转坐标轴，使得数据在 C1 轴上的方差最大，即全部 n 个数据个体在该方向上的投影最为分散。意味着更多的信息被保留下来。C1 成为第一主成分。
找一个 C2，使得 C2 与 C1 的协方差（相关系数）为 0，以免与 C1 信息重叠，并且使数据在该方向的方差尽量最大。以此类推，找到第三主成分、第四主成分……第 p 个主成分。
p 个随机变量可以有 p 个主成分。
特征值是数据在旋转之后的坐标上对应维度上的方差。特征值越大，该方向投影数据越离散，信息量越大。

基本流程：

1.对所有样本进行中心化：$x_i\leftarrow x_i-\frac{1}{n}\sum_{i=1}^{n}x_i$

2.计算样本协方差矩阵XX^T

3.对协方差矩阵XX^T做特征值分解

4.取最大的m个特征值所对应的单位特征向量$w_1,w_2,\dots w_n$

5.输出投影矩阵$W=(w_1,w_2,\dots,w_m)$

2.3 无论是 K 最近邻分类还是 K 均值聚类，都涉及到 K 的取值问题。请简述两个问题各自选 取合适 K 值的思路，并比较两者在思路上有何不同？

K 最近邻分类：在训练集上，使用不同的 K 进行分类，选择分类效果最好的 K。
K 均值聚类：尝试使用不同的 K 值聚类，检验各自得到聚类结果的质量，选择聚类效果最优的 K。
基本思路本质上是一致的。
K 最邻近分类里 K 是为了找 K 个最相似样本，以确定待分类样本的类别，选择合适的 K 使得能准确划分待分类的样本，而 K 均值为了把一些样本分成 K 个簇，选择 K 使得尽可能按实际情况划分出样本的类别。

2.4 K-mediods 算法描述：
a) 首先随机选取一组聚类样本作为中心点集
b) 每个中心点对应一个簇
c) 计算各样本到各个中心点的距离(如欧几里得距离)，将样本点放入距离中心点 最短的那个簇中
d) 计算各簇种，据簇内各样本点距离的绝对误差最小的点，作为新的中心点
e) 如果新的中心点集和原中心点集相同，算法中止；如果新的中心点集与原中心 点集不完全相同，返回 b)
试着：
a) 阐述 K-mediods 算法和 K-means 算法相同的缺陷
必须事先确定类簇数和中心点，簇数和中心点的选择对结果影响很大，一般在获得一个局部最优的解后就停止了，对于除数值型以外的数据不适合；只适用于聚类结果为凸形的数据集等。
b) 阐述 K-mediods 算法相比于 K-means 算法的优势
与 K-means 相比，K-mediods 算法对于噪声不那么敏感，这样对于离群点和异常点就不会造成划分的结果偏差过大，异常数据不会造成重大影响
c) 阐述 K-mediods 算法相比于 K-means 算法的不足
K-mediods 需要不断的找出每个点到其他所有点的距离的最小值来修正聚类中心，这要求更高的计算复杂度，减缓了收敛的速度，因而对于大规模数据的聚类力不从心。

PCA 主成分分析
将 X 矩阵（m*n，m 条数据，n 维度）找到变换矩阵 P（n*k，k<n），相乘后降为 Y（m*k）矩阵，Y=XP
过程：
1. 原始数据作为 X 矩阵
2. X 的元素都减去该列均值(零均值中心化)得到 Xs
3. 求协方差矩阵 C=Xs^T*Xs / (m-1)，n*n 维
4. 求解 C 矩阵特征值和对应特征向量
5. 特征向量单位化（除以模）
6. 挑特征值最大的 k 个向量（列），组成 P 矩阵（n*k）
7. 得到 Y=XP m*k，成功从 n 降到 k
贡献：挑选的特征值占总特征值比例