作业题（客观题）

殷子凯

张哲

- HW1

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	213 312
kaleidoscope	87 009
marmalade	107 913
skies	271 658
tangerine	46 653
trees	316 812

- 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？

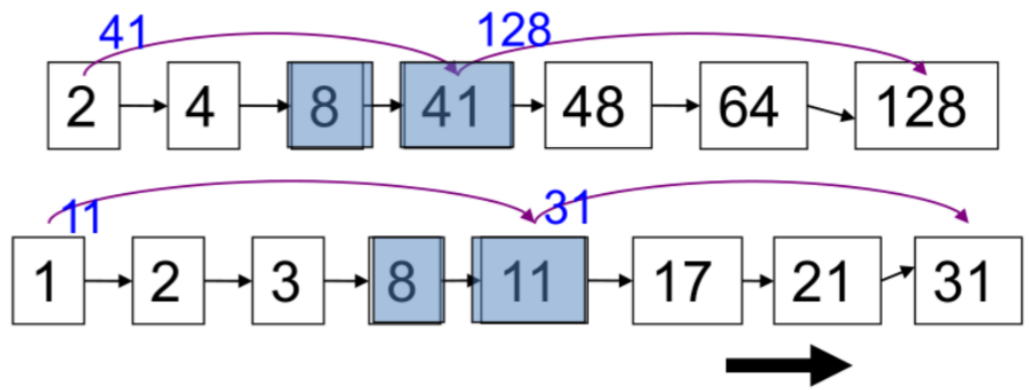
3) 如果不使用跳表指针，那么倒排记录之间的比较次数是多少？

- 考察知识点：倒排表合并的优化

```
INTERSECTWITHSKIPS(p1, p2)
 1  answer ← ⟨ ⟩
 2  while p1 ≠ NIL and p2 ≠ NIL
 3  do if docID(p1) = docID(p2)
 4        then ADD(answer, docID(p1))
 5             p1 ← next(p1)
 6             p2 ← next(p2)
 7        else if docID(p1) < docID(p2)
 8                then if hasSkip(p1) and (docID(skip(p1)) ≤ docID(p2))
 9                     then while hasSkip(p1) and (docID(skip(p1)) ≤ docID(p2))
10                          do p1 ← skip(p1)
11                     else p1 ← next(p1)
12                else if hasSkip(p2) and (docID(skip(p2)) ≤ docID(p1))
13                     then while hasSkip(p2) and (docID(skip(p2)) ≤ docID(p1))
14                          do p2 ← skip(p2)
15                     else p2 ← next(p2)
16  return answer
```

- **倒排表合并中的优化问题**

- 带有跳表指针的查询处理过程



- 首先，通过遍历发现了共同的记录8，继续移动指针
- 其次，表2在11的位置，我们发现跳表指针31小于表1的下一个数41
- 因此，我们直接将表2跳到31，而跳过其中的17、21两个数

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

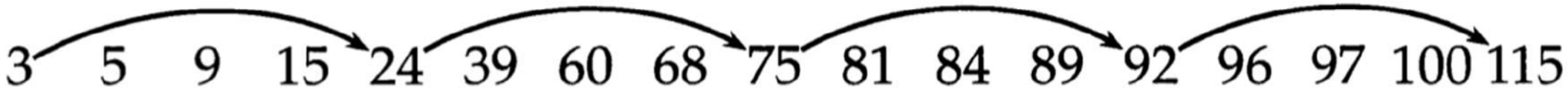
19次

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

- 常见错误：主要集中在题2)
  - 在3与3比较后，并没有将两个表的指针同时进一格（即5与5比较），而是将5与24比较。这样的做法并没有错，但是与算法的细节有偏差。
  - 75-89的比较需要几次。在原算法中，while中比较一次后，还需要比较一次。
- 这里其实本身就是有争议的。
- 暴论：考试不数



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

1.3 写出倒排记录表（777, 17743, 294068, 31251336）的可变字节编码。在可能的情况下对间距而不是文档 ID 编码。写出 8 位块的二元码。

- 考察知识点：索引压缩
  - 间距的数值必然小于文档ID的数值：采用间距ID代替文档ID
  - 可变长度编码

可变长度编码的基本流程大致如下：

- 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.

文档ID	间距	VB编码
777		00000110 10001001
17743	16966	00000001 00000100 11000110
294068	276325	00010000 01101110 11100101
31251336	30957268	00001110 01100001 00111101 11010100

777 → 110 0001001

16966 → 1 0000100 1000110

276325 → 10000 1101110 1100101

30957268 → 1110 1100001 0111101 1010100

2.1 基于机械分词的常见方法中对于“最大匹配”的依赖，可能导致什么隐患？如何利用 N-最短路径缓解这一隐患？如何选择一个恰当的 N 值？

最大匹配的隐患：

1.有长度限制：预设词长过短时长词会切错；过长时效率较低

2.导致歧义：e.g. ~~南京市长春药店~~ 南京市长江大桥

N-最短路径：

基于Dijkstra算法，记录N条最短路径

原文中，N=2时，非统计粗切模型句子召回率达到99. 73%；N=8时，句子粗切召回率达到了99.90%。

参考文献：张华平, 刘群. 基于N—最短路径方法的中文词语粗分模型[J]. 中文信息学报, 2002, 16(005):3-9.

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

- HW2

1.1 给定以下词项的 idf 值，以及在三篇文档中的 tf，已知总文档数为 811,400，请完成如下计算任务：

	df	tf@Doc1	tf@Doc2	tf@Doc3
Car	18,871	34	8	32
Auto	3,597	3	24	0
Insurance	19,167	0	51	6
Best	40,014	18	0	13

1) 计算所有词项的 tf-idf 值。
2) 试采用欧式归一化方法（即向量各元素平方和为 1），得到处理后的各文档向量化表示，其中每个向量为 4 维，每一维对应 1 个词项。
3) 基于 2)中得到的向量化表示，对于查询“car insurance”，计算 3 篇文档的得分并进行排序。其中，查询中出现的词项权重为 1，否则为 0。

- 考察知识点：TF-IDF、余弦相似度

$$W_{t,d} = \left(1 + \log tf_{t,d}\right) \cdot \log \frac{N}{df_t}$$

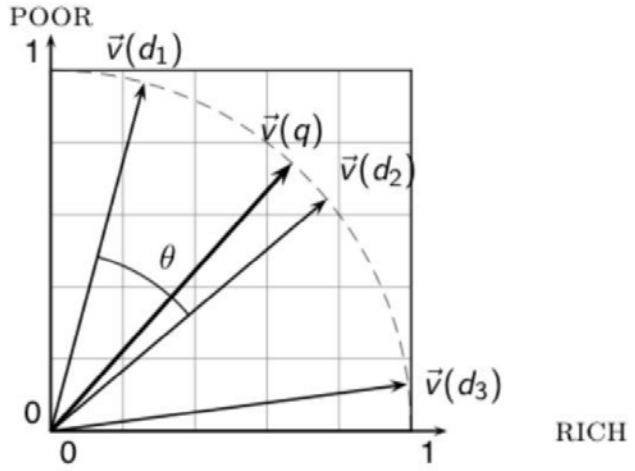
词项频率TF(t,d)，指词项 t 在文档 d 中出现的次数（Term Frequency）

dft，指出现词项 t 的文档数量

- 按照文档向量与查询向量的夹角大小来计算

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

- 显然，向量越一致，夹角越小，cosine值越高
  - 相应的，它们之间的相似度也就越高
  - 即使这种情况，它们的欧氏距离可能很大



1)

	tf-idf@doc1	Tf-idf@doc2	tf-idf@doc3
Car	4.135	3.109	4.092
Auto	3.476	5.601	0
Insurance	0	4.404	2.892
Best	2.947	0	2.763

2)

$$\mathbf{doc1} = \frac{\overrightarrow{W}_{t,d}}{|W_{t,d}|} = (\frac{4.135}{\sqrt{37.86561}},\ \frac{3.476}{\sqrt{37.86561}},\ 0,\ \frac{2.947}{\sqrt{37.86561}}) = (0.672,\ 0.565,\ 0,\ 0.478)$$

$$\mathbf{doc2} = (0.400,\ 0.720,\ 0.567,\ 0)$$

$$\mathbf{doc3} = (0.715,\ 0,\ 0.505,\ 0.483)$$

3)

$$\vec{Q} = (1,\ 0,\ 1,\ 0)$$

$$Cosine(\overrightarrow{d_1},\ \vec{Q}) = 0.475$$

$$Cosine(\overrightarrow{d_2},\ \vec{Q}) = 0.683$$

$$Cosine(\overrightarrow{d_3},\ \vec{Q}) = 0.863$$

$$\therefore Doc3 > Doc2 > Doc1$$

1.2 考虑如下邻接矩阵表示的图（1 代表出边，0 表示无连接）

	节点 0	节点 1	节点 2	节点 3	节点 4	节点 5	节点 6
节点 0	0	0	1	0	0	0	0
节点 1	0	1	1	0	0	0	0
节点 2	1	0	1	1	0	0	0
节点 3	0	0	0	1	1	0	0
节点 4	0	0	0	0	0	0	1
节点 5	0	0	0	0	0	1	1
节点 6	0	0	0	1	1	0	1

1) 当 Restart 部分的随机跳转概率为 0.15 时，写出 PageRank 的（随机）转移概率矩阵。
2) 计算该矩阵所对应的 PageRank 向量（每一维表示一个节点的 PageRank 值）。
3) 将邻接矩阵中“节点 2 指向节点 3”和“节点 6 指向节点 3”的两条边权重设为 0，其它不变。请计算该矩阵所对应的 Hub 值和 Authority 值向量。

- 考察知识点：PageRank、HITS算法

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

- 基于先前的基本假设， HITS的计算过程如下：
  - 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：
    - $a_{k+1} = M^T h_k$， $h_{k+1} = M a_{k+1}$
  - 或者，可采用如下迭代式：
    - $a_{k+1} = (M^T M)^k M^T a_0$， $h_{k+1} = (MM^T)^{k+1} h_0$
    - 其中，a_0, h_0为Authority/Hub向量的初始值，可设为全1向量

(1)

将矩阵转置并除以出边数量得跳转矩阵R：

$$R = \begin{bmatrix} 0 & 0 & \frac{1}{3} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\ 1 & \frac{1}{2} & \frac{1}{3} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{3} & \frac{1}{2} & 0 & 0 & \frac{1}{3} \\ 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & \frac{1}{3} \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & 1 & \frac{1}{2} & \frac{1}{3} \end{bmatrix}$$

转移概率矩阵：

$$A = dR + [(1-d)/N]ee^T = 0.85 * \begin{bmatrix} 0 & 0 & \frac{1}{3} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\ 1 & \frac{1}{2} & \frac{1}{3} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{3} & \frac{1}{2} & 0 & 0 & \frac{1}{3} \\ 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & \frac{1}{3} \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & 1 & \frac{1}{2} & \frac{1}{3} \end{bmatrix} + \frac{3}{140} * \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0.021 & 0.021 & 0.305 & 0.021 & 0.021 & 0.021 & 0.021 \\ 0.021 & 0.446 & 0.021 & 0.021 & 0.021 & 0.021 & 0.021 \\ 0.871 & 0.446 & 0.305 & 0.021 & 0.021 & 0.021 & 0.021 \\ 0.021 & 0.021 & 0.305 & 0.446 & 0.021 & 0.021 & 0.305 \\ 0.021 & 0.021 & 0.021 & 0.446 & 0.021 & 0.021 & 0.305 \\ 0.021 & 0.021 & 0.021 & 0.021 & 0.021 & 0.446 & 0.021 \\ 0.021 & 0.021 & 0.021 & 0.021 & 0.871 & 0.446 & 0.305 \end{bmatrix}$$

(2) 用python实现PageRank并计算$P_{n+1}=AP_n$，大约迭代28次后前后两次PageRank值差值小于10^{-5}：

令$P_0=\begin{bmatrix}\frac{1}{7}\\ \frac{1}{7}\\ \frac{1}{7}\\ \frac{1}{7}\\ \frac{1}{7}\\ \frac{1}{7}\\ \frac{1}{7}\end{bmatrix}$

$$P=\begin{bmatrix}0.0545\\ 0.0373\\ 0.1166\\ 0.2431\\ 0.2101\\ 0.0373\\ 0.3012\end{bmatrix}$$

3)更新后的邻接矩阵为M

$$M = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}$$

采用迭代式$a_{k+1} = M^T h_k$，$h_{k+1} = M a_{k+1}$，每次迭代后进行欧式归一化。

计算得：$\vec{a} = \begin{bmatrix} 0.0008 \\ 0.0008 \\ 0.0022 \\ 0.1685 \\ 0.4980 \\ 0.2726 \\ 0.8058 \end{bmatrix}$，$\vec{h} = \begin{bmatrix} 0.0011 \\ 0.0015 \\ 0.0015 \\ 0.3351 \\ 0.4051 \\ 0.5422 \\ 0.6555 \end{bmatrix}$

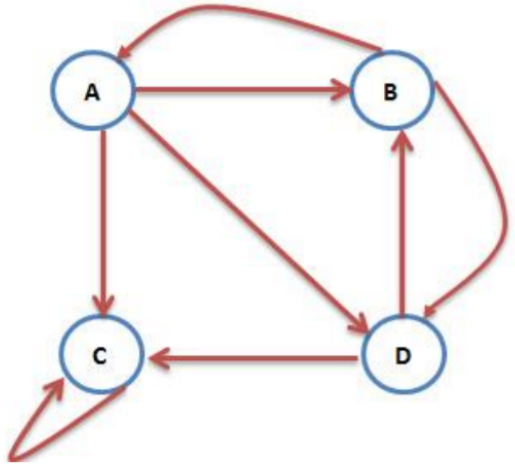
- 常见错误:
  - 1)中，分不清 随机跳转概率 的含义为d还是1-d

    随机跳转概率的意义是随机选择一个网页进行跳转的概率，结合公式应该知道代表的是1-d。

- 几类特殊情况的解决：Restart机制
  - 回顾PageRank的公式

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

  - 其中的(1-d)/N的部分，相当于以一定概率重新选择起点
    - 此时，所有节点以一定等概率被选中
    - 由此，跳出了陷阱和黑洞的干扰
    - d一般选择为0.85左右



- 常见错误:
  - 3)中，计算hub和authority时，没有每一步进行归一化，导致无法收敛

2.2 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。

2.3 如何在网页排序的同时提升结果的多样化水平？如何在实现这一目的的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

1.3 在由10,000篇文档构成的文档集中，某个查询的相关文档总数为10，
下面给出了针对该查询的前20个有序结果，其中R表示相关，N表示不相关。
R R N N R N N N R N R N N N R N N N N R

请计算：
a)该查询的P@10和P@20分别是多少？
b)该查询前10篇文档和前20篇文档的F1值分别是多少？
c)当该算法只返回前20个结果是，其简易AP值为多少？
假定该算法返回了全部10,000篇文档，上述20篇文档只是最开始的20个结果，那么
d)该算法可能的最大AP值是多少？
e)该算法可能的最小AP值是多少？

本题考查评价指标的计算方式。

R R N N R N N N R N　　R N N N R N N N N R　　相关文档总数为10

- a) $P@10 = 4/10 = 0.4$

  $P@10 = 7/20 = 0.35$

- b) $R@10 = \frac{4}{10} = 0.4$

  $R@20 = \frac{7}{10} = 0.7$

  由$F1 = \frac{2PR}{P+R}$得

  $F1@10 = 0.4$

  $F1@20 = 0.467$

- c）简化AP：不插值，分母等于positive结果的数量，即7。

  $AP = (1/1 + 2/2 + 3/5 + 4/9 + 5/11 + 6/15 + 7/20)/7 = 0.607$

- d）ground truth为共有10篇相关文档，而目前已检索到7篇，还剩下3篇未返回，所以最佳情形为：第21、22、23就返回相关文档。

  $AP_{max} = (1/1 + 2/2 + 3/5 + 4/9 + 5/11 + 6/15 + 7/20 + 8/21 + 9/22 + 10/23)/10 = 0.547$

- e) 最差情形为：直到第9998、9999、10000才返回相关文档。

  $AP_{min} = (1/1 + 2/2 + 3/5 + 4/9 + 5/11 + 6/15 + 7/20 + 8/9998 + 9/9999 + 10/10000)/10 = 0.425$

- HW3

1.1考虑下表中的事务性数据集:

Customer ID	Transaction ID	Items Bought
1	0001	$\{a, d, e\}$
1	0024	$\{a, b, c, e\}$
2	0012	$\{a, b, d, e\}$
2	0031	$\{a, c, d, e\}$
3	0015	$\{b, c, e\}$
3	0022	$\{b, d, e\}$
4	0029	$\{c, d\}$
4	0040	$\{a, b, c\}$
5	0033	$\{a, d, e\}$
5	0038	$\{a, b, e\}$

1)每个事务ID对应一条事务，计算{e}，{b, c}，{b, c, e}的支持度。
2)使用（1）的计算结果，计算关联规则{b, c}→{e}和{e}→{b, c}的置信度。
3)从（2）的结果看，置信度是对称的吗？请根据计算公式分析其对称性。

1) 支持度：$\{A+B\}$在全体事务中的比重　　$s(X \to Y) = \dfrac{\sigma(X \cup Y)}{N}$

$\{e\} : \dfrac{8}{10} = 0.8$

$\{b,c\} : \dfrac{3}{10} = 0.3$

2) $\{b,c,e\} : \dfrac{2}{10} = 0.2$

置信度：$\{A+B\}$占A出现的事务中的比重　　$c(X \to Y) = \dfrac{\sigma(X \cup Y)}{\sigma(X)}$

$\{b,c\} \to \{e\} : \dfrac{2}{3} = 0.67$

$\{e\} \to \{b,c\} : \dfrac{2}{8} = 0.25$

3) 根据(2)中结果，可以得知置信度是不对称的；另一方面，根据计算公式可知，当且仅当分母相同时置信度才相等。

注意：

1.数数要认真

1.2 考虑如下二元分类的数据集：

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， $\sum_v \frac{|D^v|}{|D|}Ent(D^v) = \frac{3}{7}(-\frac{1}{3}\log_2(\frac{1}{3}) - \frac{2}{3}\log_2(\frac{2}{3})) + \frac{2}{7} * 0 + \frac{2}{7} = 0.679$ ， 增益为0.306

以occupation划分时， $\sum_v \frac{|D^v|}{|D|}Ent(D^v) = \frac{3}{7}(-\frac{1}{3}\log_2(\frac{1}{3}) - \frac{2}{3}\log_2(\frac{2}{3})) * 2 = 0.787$ ， 增益为0.198

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：

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

假设以特征A把数据集D划分成N个子集 $D_1, D_2, \ldots D_N$ ，

则针对特征A，集合D的基尼指数为：

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

2)

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，

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

选择基尼指数最小的特征，即interest

注意：

1. 务必给出详细计算过程
2. 选择信息增益最大的，选择基尼指数最小的
3. 基尼指数的计算，牢记公式，避免以下错误：

$$Gini(p) = \sum_{k=1}^{K} p_k(1 - p_k) = 1 - \sum_{k=1}^{K} p_k^2$$

$$Gini(in) = 1 - \sum_{k=1} 3p_{1k}^2 = 1 - [\frac{3}{7}^2 + \frac{2}{7}^2 + \frac{2}{7}^2] = 0.653$$

$$Gini(oc) = 1 - \sum_{k=1} 3p_{2k}^2 = 1 - [\frac{3}{7}^2 + \frac{3}{7}^2 + \frac{1}{7}^2] = 0.612$$

1.3 已知正例点 $x_1 = (2.5, 2.5)^T$，$x_2 = (5, 2)^T$，和负例点 $x_3 = (1.5, 1.5)^T$，试用SVM对其进行分类，求最大间隔分离超平面，并指出所有的支持向量。

考察支持向量机的计算过程，务必牢记对偶问题的形式（Lagrange函数和边界条件）、分离超平面的向量和偏置的求解公式、支持向量的定义。

SVM的对偶问题：

$$\min_{\alpha} \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j (\vec{x_i} \bullet \vec{x_j}) - \sum_{i=1}^{N}\alpha_i$$

$$s.t. \quad \sum_{i=1}^{N} y_i\alpha_i = 0$$

$$\alpha_i \geq 0 \quad i = 1, \ldots, N$$

其中，N为样例点的个数，注意极值可能在边界上取。
求解对偶问题，得到最优解$\alpha_i^* (i = 1, \ldots, N)$，计算超平面方程：

$$\vec{w}^* = \sum_{i=1}^{N} \alpha_i^* y_i \vec{x_i}$$

$$b^* = y_j - \vec{w}^* \bullet \vec{x_j}, \forall \alpha_j^* \neq 0$$

支持向量是满足$\alpha_i^* \neq 0$的样例点。

对偶问题为：

$$\min_{\alpha} \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j(\vec{x_i}\bullet\vec{x_j})-\sum_{i=1}^{N}\alpha_i$$

$$=6.25\alpha_1^2+14.5\alpha_2^2+2.25\alpha_3^2+17.5\alpha_1\alpha_2-7.5\alpha_1\alpha_3-10.5\alpha_2\alpha_3-\alpha_1-\alpha_2-\alpha_3$$

$$s.t.\quad \alpha_1+\alpha_2-\alpha_3=0$$

$$\alpha_i\geq 0\quad i=1,2,3$$

由边界条件 $\alpha_1+\alpha_2-\alpha_3=0$ 得 $\alpha_3=\alpha_1+\alpha_2$ 代入目标函数得：

$$\min_{\alpha} \alpha_1^2+6.25\alpha_2^2+4\alpha_1\alpha_2-2\alpha_1-2\alpha_2$$

$$s.t.\quad \alpha_i\geq 0\quad i=1,2$$

求偏导，得 $\begin{cases}\alpha_1+2\alpha_2-1=0\\ 12.5\alpha_2+4\alpha_1-2=0\end{cases}\Rightarrow\begin{cases}\alpha_1=\frac{17}{9}\\ \alpha_2=-\frac{4}{9}\end{cases}$，不满足边界条件，故极值应在边界上取。

1) $\alpha_1=0$，目标函数为$6.25\alpha_2^2-2\alpha_2$，当$\alpha_2=\frac{4}{25}$时取极小值$-\frac{4}{25}$

2) $\alpha_2=0$，目标函数为$\alpha_1^2-2\alpha_1$，当$\alpha_1=1$时取极小值-1

故对偶问题的解为$\alpha_1=1,\alpha_2=0,\alpha_3=1$。

然后求解分离超平面：

$$\vec{w} = \alpha_1 y_1 \vec{x_1} + \alpha_3 y_3 \vec{x_3} = (2.5,2.5) - (1.5,1.5) = (1,1)$$

$$b = y_1 - \vec{w} \bullet \vec{x_1} = 1 - 5 = -4$$

超平面方程为 $\vec{w} \bullet \vec{x} + b = x^{(1)} + x^{(2)} - 4 = 0$

支持向量是 $\vec{x_1} = (2.5,2.5)$ 和 $\vec{x_3} = (1.5,1.5)$

注意：

1. 超平面容易目测得到，可以验证计算结果。但很多人基于目测的超平面方程，反推计算过程，这样只能把过程写得十分简单，不要犯这种小聪明。
2. 计算过程必须详细，目标函数的确定、求偏导、求极值点、求超平面方程的过程，必须要有。
3. 弄清楚支持向量的定义。

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。

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

b) 阐述 K-mediods 算法相比于 K-means 算法的优势

c) 阐述 K-mediods 算法相比于 K-means 算法的不足

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。

阐述K-mediods算法相比于K-means算法的优势

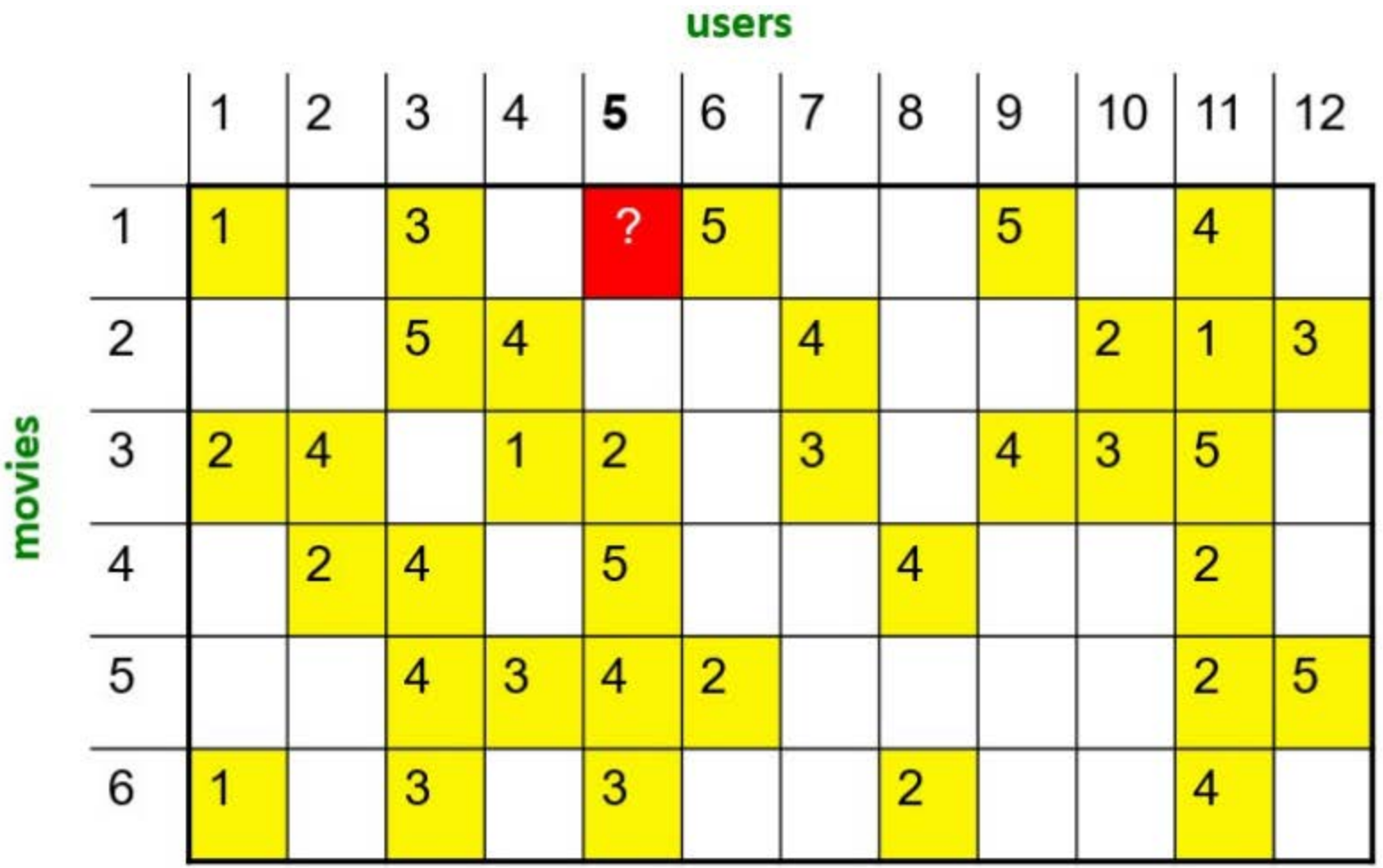
与K-means相比，K-mediods算法对于噪声不那么敏感，这样对于离群点和异常点就不会造成划分的结果偏差过大，异常数据不会造成重大影响

阐述K-mediods算法相比于K-means算法的不足

kmediod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。

- HW4

1.1 在课件中，我们给出了如下评分矩阵。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。



	1	2	3	4	**5**	6	7	8	9	10	11	12
1	1		3		?	5			5		4	
2			5	4			4			2	1	3
3	2	4		1	2		3		4	3	5	
4		2	4		5			4			2	
5			4	3	4	2					2	5
6	1		3		3			2			4	

第一步：计算各用户的平均打分。

$\overline{r_1}$=(1+2+1)/3=1.333

$\overline{r_2}$=(4+2)/2=3

$\overline{r_3}$=(3+5+4+4+3)/5=3.8

$\overline{r_4}$=(4+1+3)/3=2.667

$\overline{r_5}$=(2+5+4+3)/4=3.5

$\overline{r_6}$=(5+2)/2=3.5

$\overline{r_7}$=(4+3)/2=3.5

$\overline{r_8}$=(4+2)/2=3

$\overline{r_9}$=(5+4)/2=4.5

$\overline{r_{10}}$=(2+3)/2=2.5

$\overline{r_{11}}$=(4+1+5+2+2+4)/6=3

$\overline{r_{12}}$=(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）.

users

movies	1	2	3	4	**5**	6
1	1		3		?	5
2			5	4		
3	2	4		1	2	
4		2	4		5	
5			4	3	4	2
6	1		3		3	

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

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

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

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

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

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：

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

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。

2.2 试证明：在信息级联（Information Cascade）的定义下，信息传播最大化问题的目标函数具有“收益递减”特性，即给定两个集合 S、T 与集合外的节点 v，其中 S ⊆ T，满足：

$$f(S \cup \{v\}) - f(S) \quad \geq \quad f(T \cup \{v\}) - f(T)$$

因为$S \subseteq T$，所以有$f(T) \geq f(S)$，故对于新节点v，必有$f(T) \cap f(v) \geq f(S) \cap f(v)$

上式两边同时加上$f(T) - f(S)$，得：

$$f(T) - f(S) + f(T) \cap f(v) \geq f(T) - f(S) + f(S) \cap f(v)$$
$$\Rightarrow f(T) - f(S) \geq [f(T) - f(T) \cap f(v)] - [f(S) - f(S) \cap f(v)]$$
$$\Rightarrow f(T) - f(S) \geq [f(T) + f(v) - f(T) \cap f(v)] - [f(S) + f(v) - f(S) \cap f(v)]$$

注意到$\begin{cases} f(T \cup v) = f(T) + f(v) - f(T) \cap f(v) \\ f(S \cup v) = f(S) + f(v) - f(S) \cap f(v) \end{cases}$

$$\Rightarrow f(T) - f(S) \geq f(T \cup v) - f(S \cup v)$$
$$\Rightarrow f(S \cup v) - f(S) \geq f(T \cup v) - f(T)$$