大数据分析复习笔记

一些概念

- 相似项：共同元素比例较高的集合对
- $TF-IDF=\frac{f_i}{max_k f_k}\times \log_2 \frac{N}{n}$，词项在某一篇文档中的得分，得分越大，词项越重要
- 哈希函数：将某种数据类型的哈希键映射为整型的桶编号
- 集群计算：由计算节点集群而成
- 分布式文件系统：面向大规模文件系统的架构，文件由大小为 $64MB$ 的文件块组成

两个相似度的计算

- **余弦相似度**

  $sim(X,Y)=cos(X,Y)=\frac{X\cdot Y}{|X|\cdot|Y|}$

  余弦距离：$1-cos(X,Y)$
- **Jaccard 相似度**

  $sim(X,Y)=\frac{|A\bigcap B|}{|A\bigcup B|}$，其中$A$和$B$都是集合

  $Jaccard$ 距离：$1-sim(X,Y)$

聚类算法

- 层次聚类的过程
  - 开始时，每个点看成一个簇
  - 选择具有最短簇距离的两个簇进行合并，并计算簇新的质心
  - 算法终止前重复上述步骤
- $kmeans$ 算法
  - 选择k个初始点，这k个点尽可能在k个不同的簇中 (**关于如何选择这k个点例题有解释**)
  - 遍历每一个还未加入簇的点，将它加入离它最近的质心所在的簇，更新该簇的质心
  - 一直到遍历完所有的点，算法终止
- BFR算法中几个重要概念及其计算（例题计算）
  - 簇中点的数目N，每个点的维度为 d
  - 簇中所有点在每一个维度的分量之和，可以组成一个长度为 d 的向量 SUM
  - 簇中所有点在每一个维度的分量平方和，可以组成一个长度为 d 的向量 $SUMSQ$
  - 在第 i 维的方差：$SUMSQ_i/N-(SUM_i/N)^2$，方差也是一个长度为 d 的向量，每一维标准差则是相应方差的开方

数据流挖掘：数据以流的方式到来，如果不及时处理或存储数据会永久流失

- 流当中的数据抽样（概念有点抽象）

  这里需要了解一下如何利用哈希桶对用户进行抽样。假设我们想从所有用户中得到 $\frac{a}{b}$ 比例的样本，需要将用户名映射到 b 个编号为 0到$b-1$ 的哈希桶中。我们需要通过 $Hash(username)\%b<a$ 来判断用户名是否保留在样本中（为真则保留在样本中，为假则从样本中舍弃）。
- 流过滤
  - 假阳率：本来不能过滤的元素中通过过滤的比例，计算公式为：$(1-e^{\frac{-km}{n}})^k$

    - 上式使用了布隆过滤器，n 表示数组的位数，k 表示哈希函数的数目，m 表示集合 S 中元素的数目。这代表说，即使某一元素不在集合 S 中，它仍有一定的概率通过布隆过滤器，这就是假阳率。
    - 布隆过滤器的工作机制：位数组所有位初始值为0，对 S 中每一个元素利用每一个哈希函数进行处理。对于一些哈希函数 h_i 和 S 中的元素 K 而言，每一个 h_i（K）对应的位置值为1。当 K 元素到达时，若$h_i(K)$对应的位置为$1, \forall i$，允许该流元素通过。否则只要有一位为0，不允许通过。
- 流中独立元素计数(流当中从某个时刻开始出现的不同元素的数目)
    - FM 算法 : 流中看到的不同元素越多, 看到的不同哈希值也会越多, 看到一个异常值的可能性越大, 一个异常值(二进制)尾部通常以多个0结束
    - 尾长 : 对流元素 a 运用哈希函数时, 二进制串 $h(a)$ 尾部连续0的数目
    - 假设数据流中所有已有元素中 a 的尾长 R 最大, 则用 $\frac{2^R}{\phi}, \phi = 0.77351$ 来估计目前为止看到的独立元素的数目
- 矩的计算与二阶矩估计：AMS 算法（$X_i.element$ 与 $X_i.value$）
  $n \times (2 \times X_i.value - 1)$
- 窗口计数
    - 所谓窗口，就是二进制数据流，如 01110001010111
    - 使用 $DGIM$ 算法估计窗口中最后 k 位中 1 的个数（看例题）。窗口最右部的流元素对应的时间戳为 t。$DGIM$ 算法能够使用 $O(\log^2 N)$ 位来表示大小 N 位的窗口（N 位二进制数据流），同时保证窗口内 1 的数目估计错误率不高于 50%。算法把流划分成多个桶，满足下述条件：
        - 桶最右部位置上总为1
        - 桶的大小是所包含的1的数目，且必须为2的幂
        - 从右到左，桶的大小不会减小
        - 相同大小的桶的数目为1或2

相似项发现

- $k-shingle$ 的定义：一篇文档中任意长度为 k 的子串，这里是把一篇文档当作一个字符串。对于像邮件的文档，k 选择 5；而对于像论文一样的文档，k 选择 9 比较合适。（例3.2.1，答案在solution3）可以用文档的 $k-shingle$ 集合之间的 $Jaccaard$ 相似度来计算文档之间的文本相似度。
- 特征矩阵：例3.6，要素：全集和子集
- 两个集合（例如 S_i，S_j）经过随机排列转换之后得到的两个最小哈希值（S_i 列元素中第一个列值为1的元素的行号）相等的概率等于这两个集合的 $Jacard$ 相似度
- 学会计算最小哈希签名以及利用签名矩阵估计两个原始集合的 $Jaccard$ 相似度

推荐系统：这一章还是考察相似度计算比较多

- 效用矩阵：A、B、C是用户，a、b、c、d、e、f、g是项。推荐系统的目标是预测效用矩阵的空白项。
- 长尾效应：网络世界与实体世界推荐系统的差异
- 基于内容的推荐系统关注项的属性。项之间的相似度通过它们特征向量之间的相似度来确定

- 基于协同过滤的推荐系统关注两个项的用户评分之间的相似度（看例题）
- 如何表述两种推荐系统的推荐机制
  - 基于内容的系统：一个用户观看了多部科幻片，则系统会向该用户推荐数据库中属于“科幻”类的电影
  - 协同过滤系统：系统会把与用户A相似的用户B所喜欢的电影推荐给用户A

频繁项集

- 项、项集和频繁项集的概念 (后面的频繁桶的计算方式也是一样的)
- 支持度和支持度阈值
- 频繁项集的应用：零售商知道哪些商品通常会被顾客一起购买
- 关联规则：
  $任给一个项集I，对于I的一真子集A，可以生成一个规则：A \rightarrow I/A，其中I/A是I除去A的子集$
  。
  - 可信度计算方法：$confidence(A \rightarrow I/A) = \frac{support(I)}{support(A)}$ 。该关联规则的意义是：如果 A 中所有项出现在某个购物篮，I/A 中的所有项也很有可能出现这个购物篮中。
  - 兴趣度计算方法 : 可信度减去包含 I/A 的比率
- $Apriori$ 算法 : 就是不断生成候选项集集合并从中筛选出真正频繁的项集 (从1阶开始)
- PCY 算法 : 与 $Apriori$ 算法相比, 它主要是降低了候选项集集合 C_2 的规模。

  它主要通过哈希桶来优化算法。首先对于项对 $\{i,j\}$, i和j 肯定都是一阶频繁项集, 将 $\{i,j\}$ 映射到某个哈希桶中。如果该哈希桶是频繁桶, 那么这个二阶项集有可能是频繁项集 ; 如果这个哈希桶不频繁, 那这个二阶项集就不可能是频繁项集。

PageRank

- 一个网页的 $PageRank$ 越高，那么它越重要
- 计算转移矩阵 M（每一列的元素之和为1），参照例5.1
- $Dead\quad ends$: 这一点没有出链，最后秩向量的分量为0
- $Spider\quad trap$: 这一点的出链指向自己（或是某几个点组成的集合中的点都没有出链指向外部的点），最后秩向量除该点表示的分量为1外，其余分量都为0
- 抽税法计算 $PageRank$:
  $v^* = \beta Mv + (1-\beta)\frac{\vec{e}}{n},\quad \beta = 0.85,\quad \vec{e}是单位向量，v初始为\frac{\vec{e}}{n}$
- 有偏向的随机游走（面向主题），计算公式大体同抽税法，从 Web 图选出部分节点，组成集合 S ，则 n 为 S 中节点的个数，$\vec{e}$ 中对应 S 中节点的分量为1，其余为0
- Web（不可达网页、可达网页和自有网页）和 $Spam\quad farm$ 中 $PageRank$ 的计算
- $TrustRank$ 的计算（有偏随机游走计算法）
- $Spam\quad Mass = \frac{PageRank - TrustRank}{PageRank}$

本课程特色的奇异值分解

看懂教材例子即可，一般学会计算 3×2 的矩阵

Mapreduce

- map : 输出键-值对序列
- $reduce$: 每次作用于一个键及其对应关联值表，并以某种方式组合起来输出。

注意看教材每章节小结，完全不需要看PPT，把本笔记、教材章节小结与教师布置的课后练习吃透，期末即可保证90以上