

**课 程 实 验 报 告**

**课程名称： 大数据分析**

**专业班级： 校交1902班**

**学 号： U201912633**

**姓 名： 张睿**

**指导教师： 王蔚**

**报告日期： 2021-12-23**

**计算机科学与技术学院**

**目录**

[实验三 关系挖掘实验 1](#_Toc58793746)

[**3.1 实验内容** 1](#_Toc58793747)

[**3.2 实验过程** 1](#_Toc58793748)

[3.2.1 编程思路 1](#_Toc58793749)

[3.2.2 遇到的问题及解决方式 1](#_Toc58793750)

[3.2.3 实验测试与结果分析 1](#_Toc58793751)

[**3.3 实验总结** 2](#_Toc58793752)

# 实验三 关系挖掘实验

## **3.1 实验内容**

**必做：**

1. 实验内容

编程实现Apriori算法，要求使用给定的数据文件进行实验，获得频繁项集以及关联规则。

2. 实验要求

以Groceries.csv作为输入文件

输出1~3阶频繁项集与关联规则，各个频繁项的支持度，各个规则的置信度，各阶频繁项集的数量以及关联规则的总数

固定参数以方便检查，频繁项集的最小支持度为0.005，关联规则的最小置信度为0.5

**加分项：**

1. 实验内容

在Apriori算法的基础上，要求使用pcy或pcy的几种变式multiHash、multiStage等算法对二阶频繁项集的计算阶段进行优化。

2. 实验要求

以Groceries.csv作为输入文件

输出1~4阶频繁项集与关联规则，各个频繁项的支持度，各个规则的置信度，各阶频繁项集的数量以及关联规则的总数

输出pcy或pcy变式算法中的vector的值，以bit位的形式输出

 参数不变，频繁项集的最小支持度为0.005，关联规则的最小置信度为0.5

## **3.2 实验过程**

### 3.2.1 编程思路

关系挖掘算法主要有如下步骤：根据支持度找出各阶频繁项集，根据置信度从各阶频繁项集中找出关联规则。接下来分别叙述。

对于频繁项集的生成，一共需要生成4阶的频繁项集，根据其规则，由于二阶频繁项的每一个数据一定是一阶频繁项，以此类推，所以可以用1阶生成2阶，2阶生成3阶，如此类推来避免对某些项的不必要计算。这里又可以细分为两步，首先根据前一阶频繁项生成下一阶待选频繁项，再算待选频繁项的支持度并剔除小于支持度的项获得当前阶的频繁项。

对于PCY算法对于频繁项集生成的加速思路为，对于上一阶，将其排列组合成多个二阶待选频繁项，将其哈希进哈希桶中，并用bitmap来标识当前桶是否为频繁桶，采用如下策略对待选频繁项进行筛选：

非频繁桶中的项一定为非频繁项；

在频繁桶中可以进入下一轮筛选，即算其支持度；

其之所可以加速，是因为计算支持度是一个比较费时的操作，而且排列组合的结果非常多导致空间浪费，所以希望能尽可能减少需要进行支持度计算的项。

针对于编程来说，需要建立双方的映射，一方面既要将项映射到哈希桶中，又需要建立桶和原来项的映射关系，以便通过桶找回相应的项。

对于关联规则生成，则是需要针对所有频繁项，通过每次抽出部分元素来计算条件概率，即置信度，来判断当前规则是否成立，这里需要从取一个开始，这样可以方便进行剪枝，因为如果A,B,C🡪D,不成立的话A,B🡪C,D也一定不成立，通过这种trick，可以大大提升计算速度。

### 3.2.2 遇到的问题及解决方式

1. 占用空间过大，计算缓慢

这个问题是关联关系挖掘的核心问题，因为其实算法思想非常简单，无非是算一个条件概率，为解决这个问题，从以下方面进行改进：

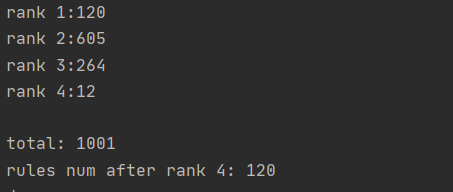
- 将物品数值化，不用物品本来的字符串，而是用一个id来进行接下来的计算

- 在计算过程中时刻进行排序，利用排序可以减少很多非必要的计算，比如123🡪4,其实和132🡪4是一样的

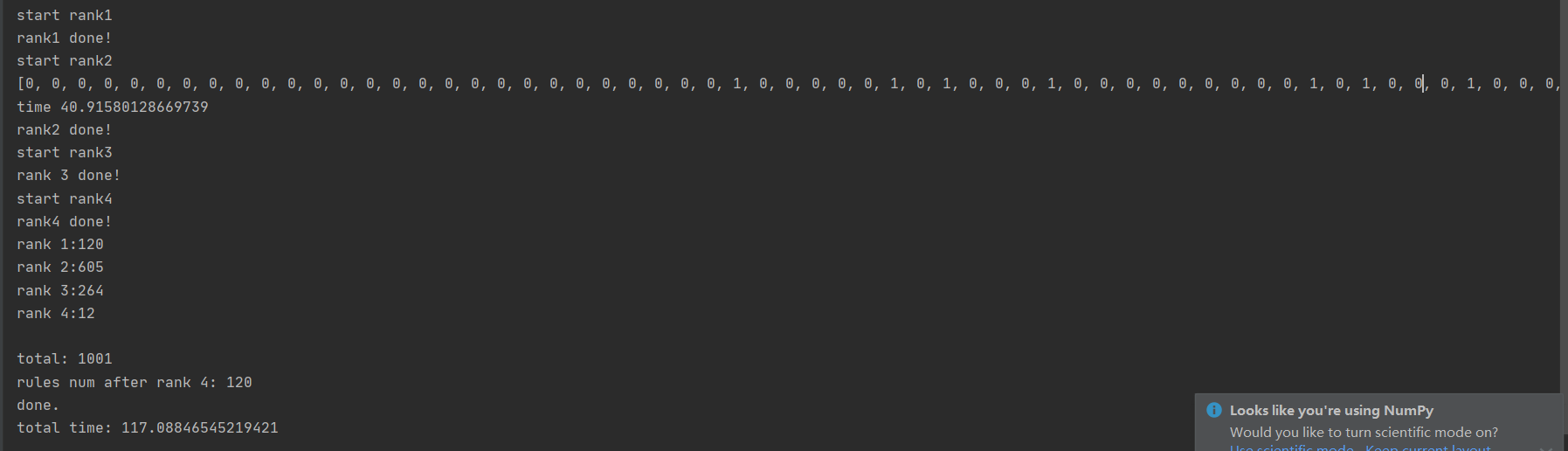
2. 桶的个数选择

桶的个数在这里是一个超参数，不同的个数对结果影响很大，这也很好理解，当桶的个数过小，会导致一个桶中有太多项，很容易成为频繁桶，也就没有了筛选功能，甚至还会因为多了这些操作拖慢时间；另一方面，当桶的个数过大，会导致计算开销上升。对此则是经过了一系列测试，最终将桶的个数定位1000。

### 3.2.3 实验测试与结果分析



通过和助教的答案进行对比，可知结果正确，相关规则在output.txt中已打印。



可以看到，总的时间用了117tick左右，算是比较快的，而对于PCY算法，其使用了约40tick左右。

值得注意的是，经过实验发现，在当前的数据规模下，使用PCY算法并没有很大地加速算法，这个跟数据内容和数据规模有一定地关系，也有可能是因为代码写地不好。

## **3.3 实验总结**

这次实验感觉是四个实验中最难的一个了，花费了许多时间理解算法原理并查阅了很多相关的资料，以及python的使用上也遇到了一些难题，不仅如此，调bug也调了很久，性能也较差，希望自己继续努力。