



**数据挖掘**

**上机实验报告**

|  |  |
| --- | --- |
| 专 业： | 人工智能 |
| 班 级： | 192204 |
| 学 号： | 19220432 |
| 姓 名： | 陆昊宇 |

2024年11月11日

## 一、实验目的

1. 掌握使用Apriori算法和FP-growth算法挖掘频繁项集。

2. 熟悉使用提升度等度量评估关联规则。

## 二、实验内容

#### 1. 杂货店商品数据集挖掘关联规则实验

（1）了解Groceries\_dataset.csv中事务集的属性信息，根据每个用户每天的购买记录构造事务（比如用户1000在2015年3月15日对应的事务为{‘sausage’，’whole milk’，’semi-finished bread’，’yogurt’}），然后按照用户ID递增，时间递增的方式对事务进行排序；

（2）选择事务集中前100条事务，选择合适的支持度阈值，使用Apriori算法和FP-growth算法挖掘频繁项集（推荐使用mlxtend包的apriori和fpgrowth函数），并对比apriori算法和fpgrowth算法的时间复杂度；

（3）重复步骤（3），依次选择前500条，前1000条事务，对比apriori算法和fpgrowth算法的时间复杂度；

（4）使用前100条事务时，选择合适的置信度阈值，利用挖掘到的频繁项集生成强关联规则（推荐使用mlxtend包的association\_rules函数）。然后使用提升度、全置信度、最大置信度、Kluc度量和余弦度量评估这些强关联规则。

**提交入口：**

<https://send2me.cn/O2ZIY3Dk/QCSguG7mKBWtoQ>

## 三、实验报告评分标准

1. 完成实验（1）（5分）

2. 完成实验（2）（30分）

3. 完成实验（3）（20分）

4. 完成实验（4）（15分）

5. 实验过程与实验结论的记录完整（10分）

6. 表述逻辑清晰（10分）

7. 排版工整自洽，图表标题准确（10分）

## 四、实验过程

（1）了解Groceries\_dataset.csv中事务集的属性信息，根据每个用户每天的购买记录构造事务（比如用户1000在2015年3月15日对应的事务为{‘sausage’，’whole milk’，’semi-finished bread’，’yogurt’}），然后按照用户ID递增，时间递增的方式对事务进行排序；

首先使用pandas库导入Groceries\_dataset.csv文件，并查看是否导入成功（图1），然后输出导入表格文件的属性信息（图2）。

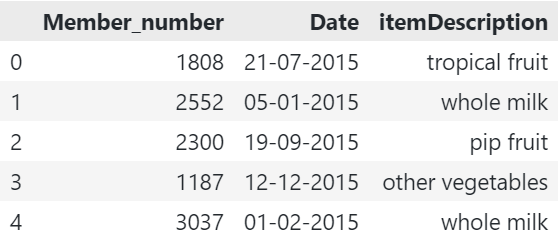


图1 导入Groceries\_dataset.csv文件

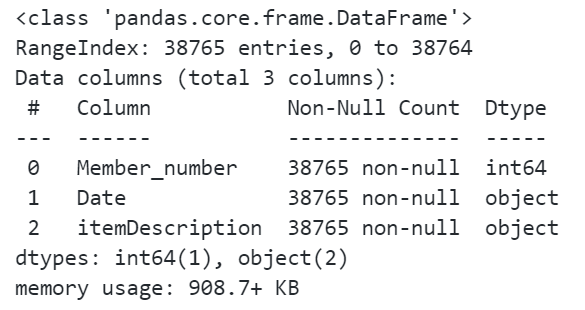


图2 表格文件的属性信息

根据每个用户每天的购买记录构造事务，然后按照用户ID递增，时间递增的方式对事务进行排序并得出结果（图3）。

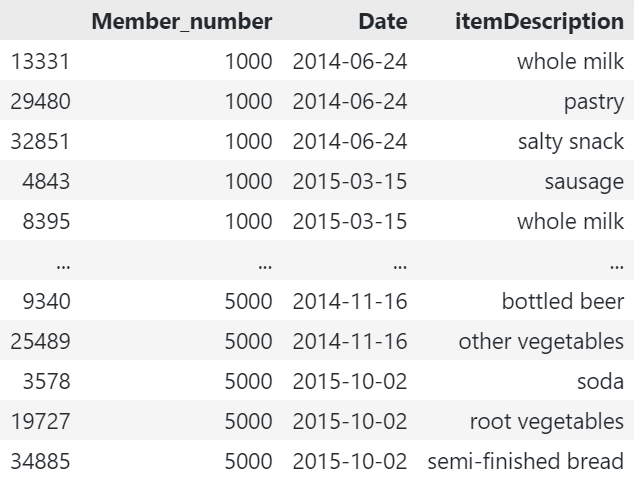


图3对事务进行排序的结果

1. 选择事务集中前100条事务，选择合适的支持度阈值，使用Apriori算法和FP-growth算法挖掘频繁项集（推荐使用mlxtend包的apriori和fpgrowth函数），并对比apriori算法和fpgrowth算法的时间复杂度；

Apriori算法通过生成频繁项集的候选集，并使用支持度评估它们的重要性。Apriori算法的时间复杂度主要取决于候选项集的生成和支持度计算的开销。生成候选项集的复杂度为O(k \* 2^k)，其中k是频繁项集的大小。支持度计算的复杂度为O(n \* m)，其中n是事务数据集的大小，m是候选项集的数量。因此，Apriori算法的总体时间复杂度可以表示为O(k \* 2^k \* n \* m)。

FP-growth算法是一种基于前缀树的关联规则挖掘算法，它通过构建频繁模式树（FP-tree）来挖掘频繁项集。FP-growth算法的时间复杂度主要取决于构建FP-tree和递归挖掘频繁项集的开销。构建FP-tree的复杂度为O(n \* log m)，其中n是事务数据集的大小，m是不同项的数量。递归挖掘频繁项集的复杂度取决于FP-tree的大小和递归的深度，通常是O(m \* (2^k - 1))，其中k是频繁项集的大小。因此，FP-growth算法的总体时间复杂度可以表示为O(n \* log m + m \* (2^k - 1))。

选择事务集中前100条事务，选择支持度阈值 [0.01, 0.05, 0.1] ，计算各阈值下的运行时间以对比apriori算法和fpgrowth算法的时间复杂度（图4），并进行可视化（图5）。

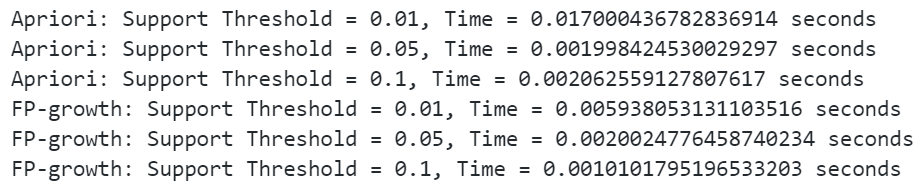


图4 Apriori算法和FP-growth在各阈值下的运行时间

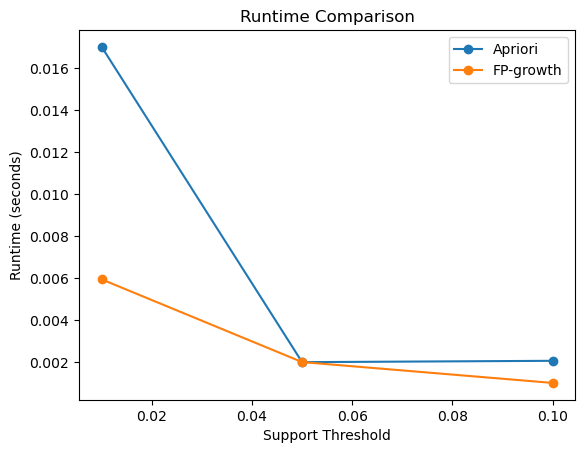


图5 算法运行时间可视化

（3）重复步骤（3），依次选择前500条，前1000条事务，对比apriori算法和fpgrowth算法的时间复杂度；

依次选择前500条，前1000条事务，计算apriori算法和fpgrowth算法的运行时间以对比两种算法的时间复杂度（图6），并可视化结果（图7）。

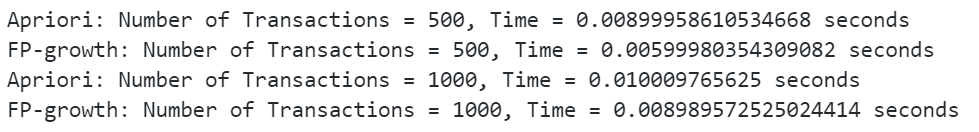


图6 apriori算法和fpgrowth算法的运行时间

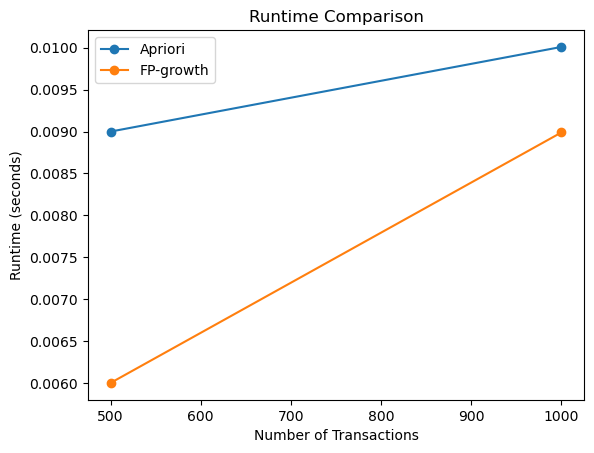
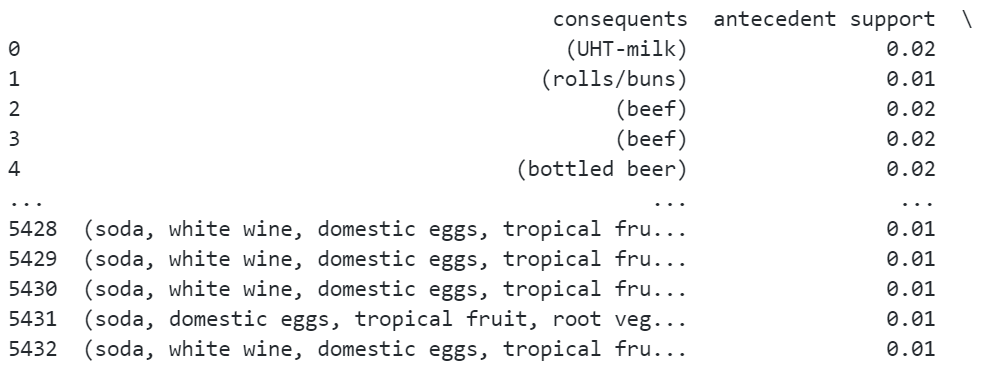
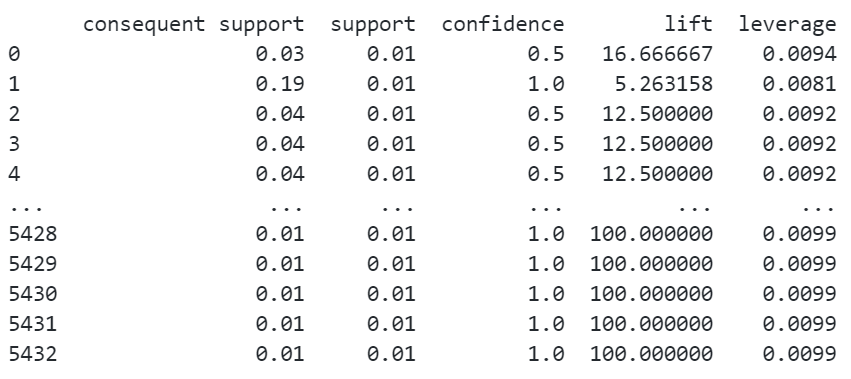


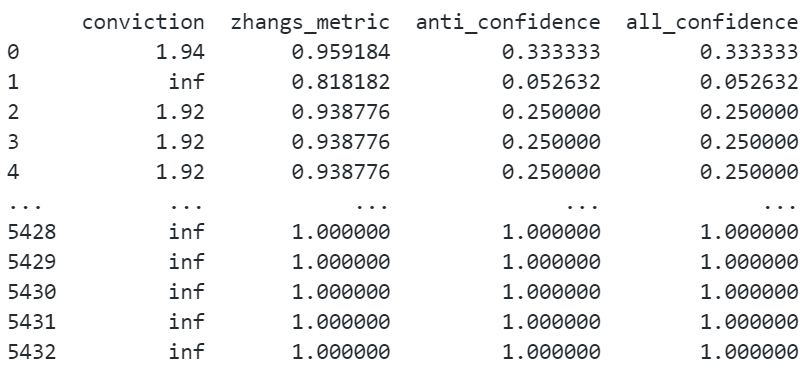
图7 算法运行时间可视化

（4）使用前100条事务时，选择合适的置信度阈值，利用挖掘到的频繁项集生成强关联规则（推荐使用mlxtend包的association\_rules函数）。然后使用提升度、全置信度、最大置信度、Kluc度量和余弦度量评估这些强关联规则。

选取前100条事务，使用Apriori算法挖掘频繁项集，设置最小支持度阈值为0.01。使用置信度作为评估关联规则的度量，设置最小置信度阈值为0.5，生成强关联规则。然后通过对关联规则进行计算和处理，生成提升度、全置信度、最大置信度、Kluc度量和余弦度量等度量指标，并输出查看（图8）。







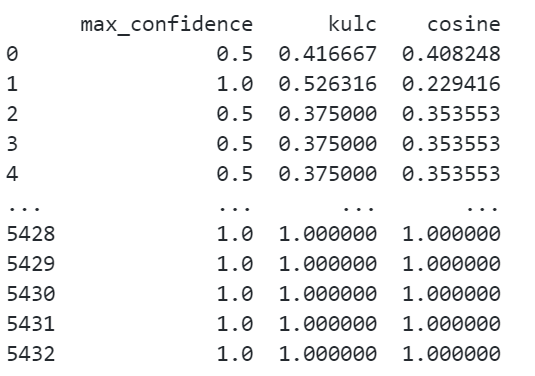


图8 关联规则输出

## 五、实验结论

Apriori算法和FP-growth算法都可以用于频繁项集挖掘，但在时间复杂度方面，FP-growth算法通常更高效。

随着事务数的增加，Apriori算法的时间复杂度呈指数增长，而FP-growth算法的时间复杂度相对较低。

选择合适的支持度阈值和置信度阈值是生成有效关联规则的关键，可以使用不同的评估指标来评估关联规则的质量和重要性。