華中科技大学

课 程 实 验 报 告

课程名称： 大数据分析

专业班级： **CS2011**

学　　号： **U202014774**

姓　　名： 王逸

指导教师： 崔金华

报告日期： **2022.12.28**

计算机科学与技术学院

目录

实验三 关系挖掘实验

3.1 实验内容

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

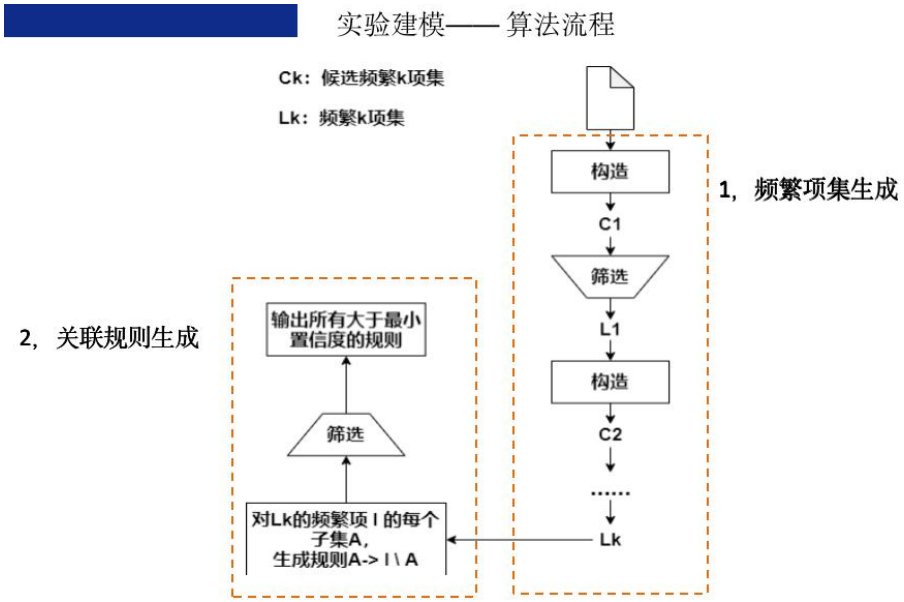
输出 1~3 阶频繁项集与关联规则，各个频繁项的支持度，各个规则的置信度，各阶频繁项集的数量以及关联规则的总数

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

3.2 实验过程

3.2.1 编程思路

本次实验的总体流程如下：



该实验的 apriori.py 文件中定义了以下几个函数：

1) read_data(data_file)：调用 read_csv()函数读取数据文件里的 item 列，返回一个 item 列表（集合）。

2) init_c1(data)：初始化构造 c1.将文件读取后获得的 items 总集合分成单项集，采用 frozenset 类型包装 item 并加入到 c1 中。
3) Calculate_lk(data, ck, min_sup, support_data)：对 ck 中的每个项进行计数，并根据最小支持度 min_sup 从中删除不满足的项，从而获得频繁集 Lk 及候选项集的支持度字典 support_data。
4) prune_apriori(ck_item, lk_1)：剪枝函数，若基于 k 阶频集生成的 k+1 阶候选项集中有子集不是 Lk 中的项集则返回 false，从中删去。
5) Concat_ck(lk, k)：基于 k 阶频集生成 Ck+1.根据前缀匹配进行连接生成 Ck+1 的候选集，并根据剪枝函数 prune_apriori（）判断是否满足条件，若满足则加入到 ck+1 的列表中。
6) get_rule(l, supoort_data, min_conf)：获得关联规则。利用最小置信度 min_conf 遍历最终生成的频集 l，并利用之前计算过程中得到的支持度字典计算 conf 生成满足条件的关联规则。

3.2.2 遇到的问题及解决方式

本次实验一开始最关注的是采用什么类型能更方便的处理集合以及子集判断的问题，查阅资料后最终采用 set 类型以及拥有的函数 issubset()来实现。并且由于最后需要计算规则之间的置信度，考虑到计算公式所需要的数据便初始化一个支持度字典并在计算 lk 时同时计算更新。

3.2.3 实验测试与结果分析

输出结果如图所示：

```
1 阶频繁项集数量为: 120.

2 阶频繁项集数量为: 605.

3 阶频繁项集数量为: 264.

关联规则的数量为: 99.

--------end--------

Process finished with exit code 0
```

L1:

```
frozenset({'liver loaf'}),  0.005083884087442806
frozenset({'citrus fruit'}),    0.08276563294356888
frozenset({'napkins'}), 0.05236400610066091
frozenset({'ham'}), 0.026029486527707167
frozenset({'beverages'}),   0.026029486527707167
frozenset({'jam'}), 0.005388917132689374
frozenset({'mayonnaise'}),  0.009150991357397052
frozenset({'dental care'}), 0.005795627859684799
frozenset({'frozen dessert'}),  0.010777834265378749
frozenset({'white bread'}), 0.042094560244026434
frozenset({'liquor'}),  0.011082867310625319
```

L2:

```
frozenset({'frozen vegetables', 'bottled water'}),  0
frozenset({'root vegetables', 'beef'}), 0.01738688357
frozenset({'soda', 'pip fruit'}),   0.013319776309100
frozenset({'beef', 'whole milk'}),  0.021250635485510
frozenset({'citrus fruit', 'newspapers'}),  0.0083375
frozenset({'yogurt', 'candy'}), 0.005490594814438231
frozenset({'napkins', 'newspapers'}),   0.00620233858
frozenset({'pasta', 'whole milk'}), 0.006100660904931
frozenset({'rolls/buns', 'long life bakery product'})
frozenset({'bottled beer', 'whole milk'}),  0.0204372
frozenset({'soda', 'coffee'}),  0.0099644128113879
```

L3:

```
frozenset({'citrus fruit', 'other vegetables', 'tropical fruit'}),  0.009049313675648195
frozenset({'rolls/buns', 'newspapers', 'whole milk'}),  0.007625826131164209
frozenset({'yogurt', 'curd', 'other vegetables'}),  0.006100660904931368
frozenset({'other vegetables', 'margarine', 'rolls/buns'}), 0.005185561769191663
frozenset({'other vegetables', 'soda', 'domestic eggs'}),   0.005083884087442806
frozenset({'other vegetables', 'butter', 'tropical fruit'}),    0.005490594814438231
frozenset({'root vegetables', 'other vegetables', 'rolls/buns'}),   0.012201321809862735
frozenset({'whole milk', 'whipped/sour cream', 'domestic eggs'}),   0.0056939501779359435
frozenset({'rolls/buns', 'frozen vegetables', 'whole milk'}),   0.005083884087442806
frozenset({'citrus fruit', 'other vegetables', 'root vegetables'}), 0.010371123538383325
frozenset({'other vegetables', 'bottled water', 'rolls/buns'}), 0.007320793085917641
```

规则：

```
frozenset({'baking powder'})    of  frozenset({'whole milk'})   : 0.5229885057471264
frozenset({'butter', 'tropical fruit'}) of  frozenset({'other vegetables'}) : 0.5510204081632654
frozenset({'root vegetables', 'rolls/buns'})    of  frozenset({'other vegetables'}) : 0.502092050209205
frozenset({'whipped/sour cream', 'domestic eggs'})  of  frozenset({'whole milk'})   : 0.5714285714285715
frozenset({'rolls/buns', 'frozen vegetables'})  of  frozenset({'whole milk'})   : 0.5
frozenset({'citrus fruit', 'root vegetables'})  of  frozenset({'other vegetables'}) : 0.5862068965517241
frozenset({'sausage', 'whipped/sour cream'})    of  frozenset({'whole milk'})   : 0.5617977528089887
frozenset({'brown bread', 'tropical fruit'})    of  frozenset({'whole milk'})   : 0.5333333333333333
frozenset({'root vegetables', 'butter'})    of  frozenset({'other vegetables'}) : 0.5118110236220472
frozenset({'curd', 'tropical fruit'})   of  frozenset({'other vegetables'}) : 0.5148514851485149
frozenset({'other vegetables', 'brown bread'})  of  frozenset({'whole milk'})   : 0.5
frozenset({'root vegetables', 'yogurt'})    of  frozenset({'other vegetables'}) : 0.5
frozenset({'yogurt', 'fruit/vegetable juice'})  of  frozenset({'whole milk'})   : 0.5054347826086957
```

3.3 实验总结

本次实验主要实现的是 apriori 算法，一开始就算对算法有初步的理解也不能很好的进行下来，对类型的选取及使用是本次实验一个比较关键的部分，后面也通过参考网上各种开源代码得以完成。