实验 2 Apriori 关联规则挖掘实验

1. 实验目的

通过本实验，学习使用 Apriori 算法来挖掘数据间的关联规则，并比较支持度和置信度对关联规则产生的影响。

1. 实验步骤

2.1 运行 main\_apr.py 示例代码；

2.2 阅读代码与 Apriori 算法对应，将伪代码描述对应到带代码，写清注释；

2.3 更换较大的数据集 Income.csv ，修改代码并执行以下操作：

a) 以最小支持度为 0.1 ，最小置信度为 0.5 建立 Apriori 关联规则；

b) 以最小支持度为 0.1 ，最小置信度为 0.6 建立 Apriori 关联规则；

c) 以最小支持度为 0.2 ，最小置信度为 0.5 建立 Apriori 关联规则；

2.4 比较 2.3 生成的三个关联规则的数目。

1. 实验结果展示

3.1 代码注释说明

def find\_rule(d, support, confidence, ms=u'--'):  
 """ 寻找关联规则的函数  
  
 算法流程  
 1. 获得候选一项集  
 2. 计算候选一项集中每个一项集的支持度  
 3. 根据 2 计算出的支持度与 support 参数来对候选一项集进行筛选  
 4. 拼接候选 k-项集（ k >= 2 ）  
 5. 计算候选 k-项集中每个项集的支持度  
 6. 根据 5 计算出的支持度与 support 参数来对候选 k-项集进行筛选  
 7. 根据 k-项集创建候选关联规则  
 8. 计算候选关联规则的置信度  
 9. 根据 8 计算出的置信度与 confidence 参数来对候选关联规则进行筛选  
 10. 重复步骤 4 ，直到 k-项集的数量 <= 1  
  
 ***:param*** d: 一个空余部分被填充了 float 0 的数据矩阵， shape=(事务数量, 事务的最大项数))  
 ***:param*** support: 用户定义的支持度  
 ***:param*** confidence: 用于定义的置信度  
 ***:param*** ms: 字符连接符  
 ***:return*** result: 符合条件的关联规则  
 """  
 # 定义输出结果  
 result = pd.DataFrame(index=['support', 'confidence'])  
 # 支持度序列，每一个元素都会做一次计算，即计算各个元素的支持度  
 support\_series = 1.0 \* d.sum() / len(d)  
 # 初步根据支持度筛选，获得计算支持度大于支持度阈值的点集  
 column = list(support\_series[support\_series > support].index)   
 # k 即为 k-项集 中的 k ，初始为 0  
 k = 0  
 # 随着 k 值的增加，column 中的元素会一个个被删除  
 while len(column) > 1:  
 k = k + 1  
 print(u'\n正在进行第%s次搜索...' % k)  
 column = connect\_string(column, ms)  
 print(u'数目：%s...' % len(column))  
 print(column)  
  
 # 新一批支持度的计算函数  
 sf = lambda i: d[i].prod(axis=1, numeric\_only=True)  
 # 创建连接数据，这一步耗时、耗内存最严重。当数据集较大时，可以考虑并行运算优化。  
 d\_2 = pd.DataFrame(list(map(sf, column)), index=[ms.join(i) for i in column]).T  
 # 计算连接后的支持度 每一列求和除以事务数量，再 \* 1.0 （这个 1.0 可以不用把）  
 support\_series\_2 = 1.0 \* d\_2[[ms.join(i) for i in column]].sum() / len(d)  
 # 新一轮支持度筛选  
 column = list(support\_series\_2[support\_series\_2 > support].index)  
 support\_series = support\_series.append(support\_series\_2)  
  
 column2 = []  
 # 遍历可能的推理，如 {A, B, C} 究竟是 A + B --> C 还是 B + C --> A 还是 C + A --> B ？  
 for i in column:  
 i = i.split(ms)  
 for j in range(len(i)):  
 column2.append(i[:j]+i[j+1:]+i[j:j+1])  
  
 # 以下三个部分是对置信度的处理  
 # 定义置信度序列  
 cofidence\_series = pd.Series(index=[ms.join(i) for i in column2])  
 # 计算置信度序列  
 for i in column2:  
 cofidence\_series[ms.join(i)] = support\_series[ms.join(sorted(i))] / support\_series[ms.join(i[:len(i)-1])]  
 # 置信度筛选  
 for i in cofidence\_series[cofidence\_series > confidence].index:  
 result[i] = 0.0  
 result[i]['confidence'] = cofidence\_series[i]  
 result[i]['support'] = support\_series[ms.join(sorted(i.split(ms)))]  
  
 # 结果整理，输出  
 result = result.T.sort\_values(['confidence','support'], ascending = False)  
 print(u'\n结果为：')  
 print(result)  
  
 return result

3.2 关联规则比较

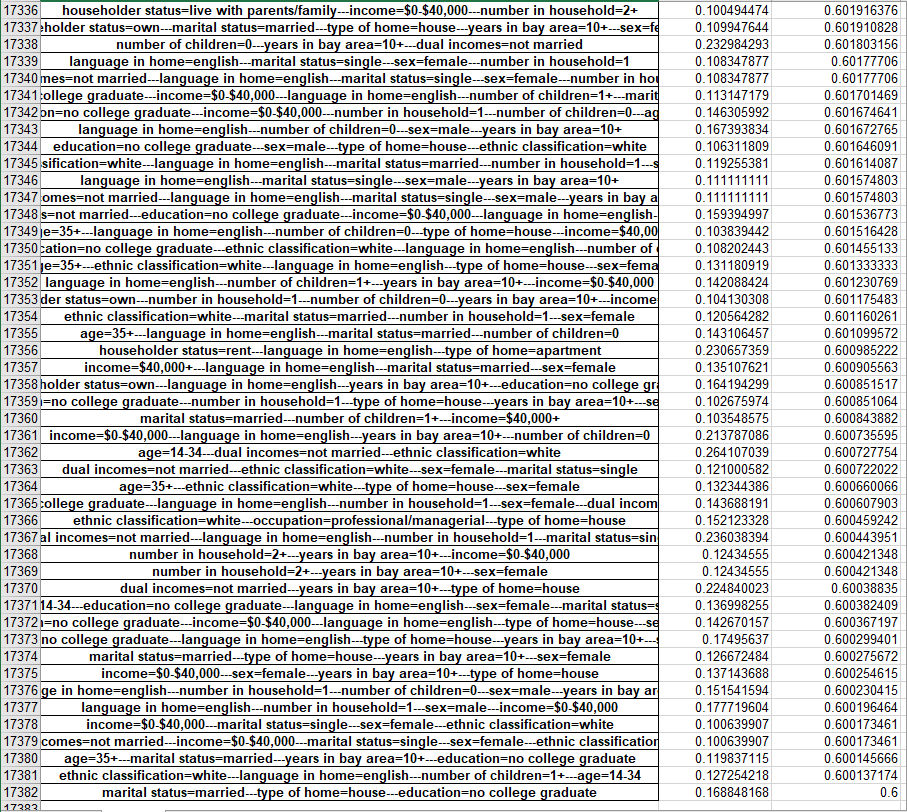
增加最小支持度与增大最小置信度可以获得相关性更好的联系，但是联系的数目会减少。

|  |  |  |
| --- | --- | --- |
| 最小支持度 | 最小置信度 | 关联规则数目 |
| 0.1 | 0.5 | 20410 |
| 0.1 | 0.6 | 17382 |
| 0.2 | 0.5 | 2065 |

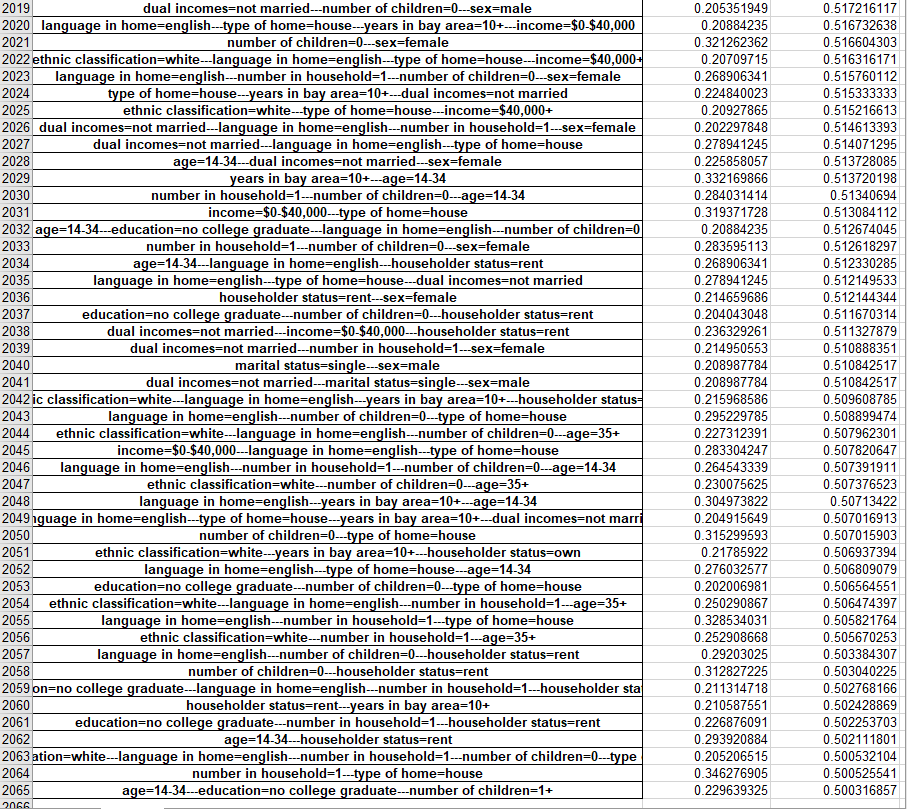
2.3-a 最小支持度为 0.1 ，最小置信度为 0.5 时，共计生成 20410 条关联规则；



2.3-b 最小支持度为 0.1 ，最小置信度为 0.6 时，共计生成 17382条关联规则；



2.3-c 最小支持度为 0.2 ，最小置信度为 0.5 时，共计生成2065条关联规则；



1. 实验总结与收获

通过这次实验，我学习了 Apriori 算法的基本思想，比较和思考了最小置信度和最小支持度的大小对关联规则生成数量的影响。