

西安电子科技大学

数据挖掘大作业



姓名	鲍超俊
学号	15020510059
指导老师	缙水萍

一、先验算法

1. 相关概念

关联规则可以描述成：项集 \rightarrow 项集。项集 X 出现的事务次数（亦称为support count）定义为：

$$\sigma(X) = |t_i|X \subseteq t_i, t_i \in T|$$

其中， t_i 表示某个事务， T 表示事务的集合。

支持度

关联规则 $X \rightarrow Y$ 的支持度：

$$s(X \rightarrow Y) = \frac{\sigma(X \cup Y)}{|T|}$$

支持度刻画了项集 $X \cup Y$ 的出现频次。

置信度

置信度定义如下：

$$s(X \rightarrow Y) = \frac{\sigma(X \cup Y)}{\sigma(X)}$$

2. 先验定理

定理1

如果一个项集是频繁的，那么其所有的子集也一定是频繁的。

定理2

如果一个项集是非频繁的，那么其所有的超集也一定是非频繁的。

3. 算法流程

输入

数据集 D ，支持度阈值 α

输出

最大的频繁 k 项集

伪代码

Algorithm 6.1 Frequent itemset generation of the *Apriori* algorithm.

```
1:  $k = 1$ .
2:  $F_k = \{ i \mid i \in I \wedge \sigma(\{i\}) \geq N \times \text{minsup} \}$ .    {Find all frequent 1-itemsets}
3: repeat
4:    $k = k + 1$ .
5:    $C_k = \text{apriori-gen}(F_{k-1})$ .    {Generate candidate itemsets}
6:   for each transaction  $t \in T$  do
7:      $C_t = \text{subset}(C_k, t)$ .    {Identify all candidates that belong to  $t$ }
8:     for each candidate itemset  $c \in C_t$  do
9:        $\sigma(c) = \sigma(c) + 1$ .    {Increment support count}
10:    end for
11:  end for
12:   $F_k = \{ c \mid c \in C_k \wedge \sigma(c) \geq N \times \text{minsup} \}$ .    {Extract the frequent  $k$ -itemsets}
13: until  $F_k = \emptyset$ 
14: Result =  $\bigcup F_k$ .
```

二、程序说明

1. 工具包清单

- numpy

2. 模块功能

Apriori 类

```
def __init__(self):  
    self.freq_set = [] # 频繁项集  
    return
```

```
def fit(self, database, threshold):  
    """  
    计算最大频繁项集  
    :param database: 数据库  
    :param threshold: 阈值  
    :return:  
    """  
    pass
```

```
def __concentrate__(self):  
    """  
    更新频繁项集  
    :return:  
    """  
    pass
```

```
def __update_sup__(self):  
    """  
    更新支持度  
    :return:  
    """  
    pass
```

```
def __cut__(self, threshold):  
    """  
    剪枝，删除支持度小于阈值的项集。  
    :param threshold: 阈值  
    :return:  
    """  
    pass
```

三、程序测试

1. 测试程序

```
if __name__ == '__main__':  
    # 数据库  
    database = [ ["面包", "牛奶", "啤酒", "尿布"],  
                  ["面包", "牛奶", "啤酒"],  
                  ["啤酒", "尿布"],  
                  ["面包", "牛奶", "花生"] ]  
    apr = Apriori()          # 声明Apriori类  
    apr.fit(database, 0.7)   #计算最大频繁项集，阈值为0.7.
```

2. 测试结果

```
/usr/local/bin/python3.6 /Users/Setsuna/Documents/GitRepo/Apriori/apriori.py  
{'面包', '牛奶'} 0.75  
  
Process finished with exit code 0
```