通信1701　　胡成成　　41724260

作业１： Apriori与FP-Growth算法比较

1. Apriori 与FP-Growth算法流程图

1.1． 概念回顾

- 支持度：P(A ∩ B)，既有A又有B的概率
- 置信度：P(B|A)，在A发生的事件中同时发生B的概率 p(AB)/P(A)
- 频繁k项集：如果事件A中包含k个元素，那么称这个事件A为k项集事件A满足最小支持度阈值的事件称为频繁k项集
- 强规则：同时满足最小支持度阈值和最小置信度阈值的规则称为强规则

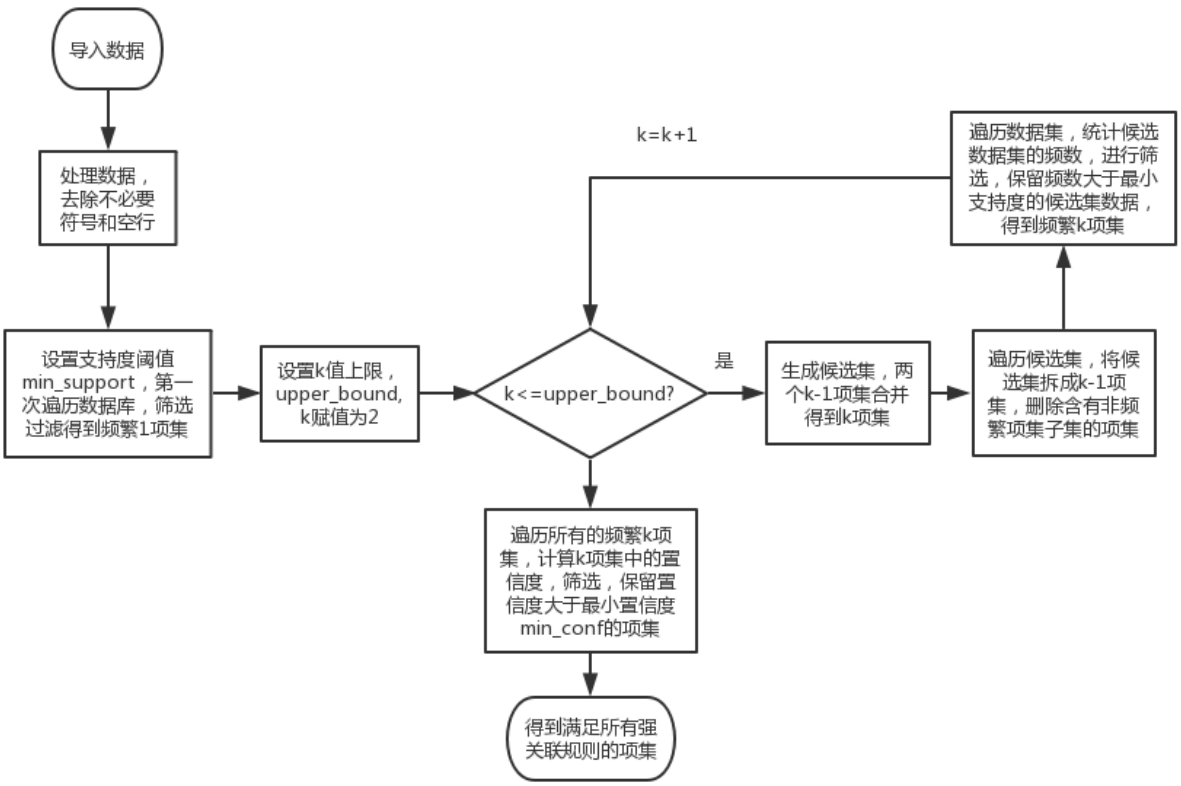
1.2． Apriori 算法过程

1.2.1 算法描述

第一步通过迭代，检索出事务数据库中的所有频繁项集，即支持度不低于用户设定的阈值的项集；

第二步利用频繁项集构造出满足用户最小信任度的规则。具体做法就是：首先找出频繁1-项集，记为L1；然后利用L1来产生候选项集C2，对C2中的项进行判定挖掘出L2，即频繁2-项集；不断如此循环下去直到无法发现更多的频繁k-项集为止。每挖掘一层Lk就需要扫描整个数据库一遍。

1.2.2 算法流程图

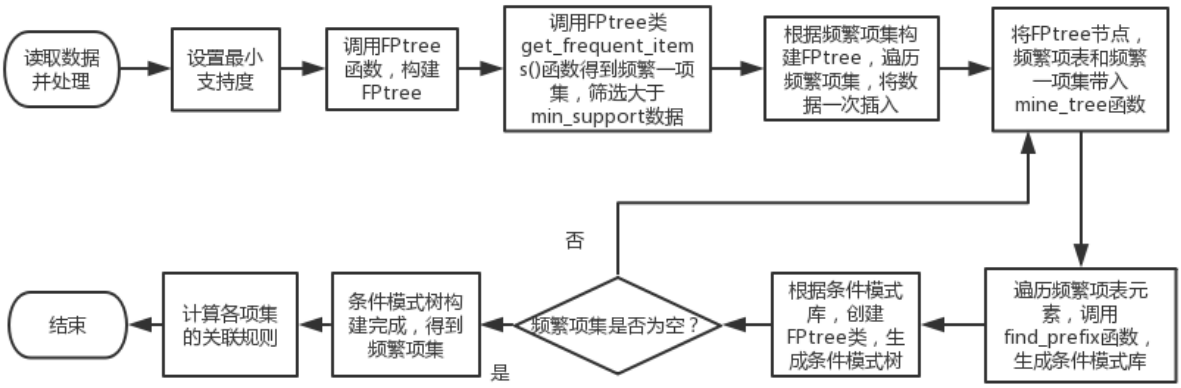


1.2． FP-Growth 算法过程

1.3.1 算法步骤

- 先扫描数据库，统计所有属性的出现次数(频数)，然后按照频数递减排序，删除频数小于min_suppt（最小支持度）的属性。
- 对每一条数据记录，重新排序（从大到小），并删除小于min_suppt的商品。并插入到FP-tree中。
- 从FP-Tree中划分出条件模式库。
- 构建条件频繁模式树。
- 挖掘频繁项集。

1.3.2 算法流程图



2. Apriori与FP-Growth算法效率对比 通过导入time库计算运行时间

```
import time
start = time.time()
...
end=time.time()
print('运行时间：',str(end-start))
```

- Apriori算法运行结果：

```
运行时间：  22.43546152114868秒
```

- FP-Growth算法运行结果：

```
运行时间：  0.8532192707061768秒
```

对比发现，**FP-Growth算法明显比Apriori算法效率高。**

3. FP-Growth算法后加入关联规则

3.1 关联规则引入

- 将mine_tree()调整 **说明：**
  - 将mine_tree()获取的频繁项集用定义的frequent_all_list = []与frequent_all_key_value_set = {}进行装载并将其按一定格式处理，便于后期计算置信度遍历这两个项集。
  - frequent_all_list = []与frequent_all_key_value_set = {}的区别在于 frequent_all_list = []只是把每个出现的自己汇总了，没有统计子集出现的频数，也就是没有除去总数的支持度。而requent_all_key_value_set = {}是一个字典，键表示每个子集的字符串表示形式，值就是这个子集出现的频数。

```
#频繁项集列表
frequent_all_list = []
#频繁项集字典，ｋｅｙ为频繁项集字符串形式，ｖａｌｕｅ为其出现次数
frequent_all_key_value_set = {}

def mine_tree(frequent_items, headerTable, min_support, frequent, item_list):
    # 频繁项表中的元素降序排列
    candidates = [v[0] for v in sorted(frequent_items.items(), key=lambda kv: (-kv[1],
list(kv[0])[0]))]
    # print(candidates)

    global frequent_all_key_value_set
    for item in candidates[::-1]:

        freq_set = frequent.copy()
        freq_set.add(item)
        freq_set_new = set()

        for i in freq_set:
            freq_set_new.add(i)


        fre_newlist = list(freq_set_new)

        frequent_all_list.append(freq_set_new)

        b = str(fre_newlist)

        frequent_all_key_value_set[b] = frequent_items[item]

        cpbs = find_prefix(headerTable[item][1])
        # print( 'its cpbs: ', cpbs)
        # 创建条件FP树
        cTree = FPTree(cpbs, min_support, 'root', 1, 'cfptree')

        cTree.build_tree()
        # cTree.show()
        # print ('-----headerTable: ', cTree.get_headertable())
        # 判断条件：频繁项表为空
        if len(cTree.get_headertable()) != 0:
            # print ('condtional tree for: ', freq_set)
            # cTree.show(1)
            mine_tree(cTree.get_frequent_items(), cTree.get_headertable(), min_support,
freq_set, item_list)
```

- 添加关联规则

```
i = 0
#最小置信度
min_conf=0.4

association_rules = []

# print(frequent_all_list)
# print(frequent_all_key_value_set)

for item_set in frequent_all_list:
    #print(item_set)
    for conclusion in frequent_all_list:
        #print(conclusion)
        if conclusion > item_set:
            confidence = float(frequent_all_key_value_set[str(list(conclusion))] /
frequent_all_key_value_set[str(list(item_set))])
            #print(confidence)
            #print(conclusion)
            if confidence > min_conf:
                i += 1
                association_rules.append([[item_set, conclusion-item_set], confidence])
                #print('conclusion:', conclusion-item_set, 'condition:', item_set,
'confidence:', confidence)
print(i)
print(association_rules)
```

3.2 结果对比

注：置信度阈值0.4

- Apriori算法关联规则结果：

[[[frozenset({'chicken'}), frozenset({'whole milk'})], 0.4099526066350711],
[[frozenset({'oil'}), frozenset({'whole milk'})], 0.40217391304347827],
[[frozenset({'whipped/sour cream'}), frozenset({'other vegetables'})],
0.40283687943262414], [[frozenset({'hamburger meat'}), frozenset({'whole milk'})],
0.4434250764525994], [[frozenset({'sugar'}), frozenset({'whole milk'})],
0.4444444444444444], [[frozenset({'beef'}), frozenset({'whole milk'})],
0.4050387596899225], [[frozenset({'frozen vegetables'}), frozenset({'whole milk'})],
0.4249471458773784], [[frozenset({'cream cheese'}), frozenset({'whole milk'})],
0.4153846153846154], [[frozenset({'margarine'}), frozenset({'whole milk'})],
0.4131944444444444], [[frozenset({'domestic eggs'}), frozenset({'whole milk'})],
0.47275641025641024], [[frozenset({'yogurt'}), frozenset({'whole milk'})],
0.40160349854227406], [[frozenset({'root vegetables'}), frozenset({'other vegetables'})],
0.43470149253731344], [[frozenset({'curd'}), frozenset({'whole milk'})],
0.4904580152671756], [[frozenset({'tropical fruit'}), frozenset({'whole milk'})],
0.40310077519379844], [[frozenset({'whipped/sour cream'}), frozenset({'whole milk'})],
0.44964539007092197], [[frozenset({'chicken'}), frozenset({'other vegetables'})],
0.41706161137440756], [[frozenset({'white bread'}), frozenset({'whole milk'})],
0.4057971014492754], [[frozenset({'root vegetables'}), frozenset({'whole milk'})],
0.44869402985074625], [[frozenset({'butter milk'}), frozenset({'whole milk'})],
0.41454545454545455], [[frozenset({'ham'}), frozenset({'whole milk'})], 0.44140625],
[[frozenset({'sliced cheese'}), frozenset({'whole milk'})], 0.43983402489626555],
[[frozenset({'hamburger meat'}), frozenset({'other vegetables'})], 0.41590214067278286],
[[frozenset({'butter'}), frozenset({'whole milk'})], 0.4972477064220184],
[[frozenset({'onions'}), frozenset({'other vegetables'})], 0.45901639344262296],
[[frozenset({'yogurt', 'tropical fruit'}), frozenset({'other vegetables'})],
0.42013888888888889], [[frozenset({'whole milk', 'butter'}), frozenset({'other
vegetables'})], 0.41697416974169743], [[frozenset({'other vegetables', 'butter'}),
frozenset({'whole milk'})], 0.5736040609137056], [[frozenset({'root vegetables',
'rolls/buns'}), frozenset({'whole milk'})], 0.5230125523012552], [[frozenset({'whole milk',
'citrus fruit'}), frozenset({'other vegetables'})], 0.4266666666666667],
[[frozenset({'citrus fruit', 'other vegetables'}), frozenset({'whole milk'})],
0.4507042253521127], [[frozenset({'yogurt', 'rolls/buns'}), frozenset({'whole milk'})],
0.4526627218934911], [[frozenset({'root vegetables', 'citrus fruit'}), frozenset({'other
vegetables'})], 0.5862068965517241], [[frozenset({'yogurt', 'other vegetables'}),
frozenset({'whole milk'})], 0.5128805620608899], [[frozenset({'whole milk', 'pork'}),
frozenset({'other vegetables'})], 0.45871559633027525], [[frozenset({'other vegetables',
'pork'}), frozenset({'whole milk'})], 0.4694835680751174], [[frozenset({'yogurt', 'root
vegetables'}), frozenset({'other vegetables'})], 0.5], [[frozenset({'tropical fruit',
'rolls/buns'}), frozenset({'whole milk'})], 0.4462809917355372], [[frozenset({'other
vegetables', 'pastry'}), frozenset({'whole milk'})], 0.46846846846846846],
[[frozenset({'whipped/sour cream', 'whole milk'}), frozenset({'other vegetables'})],
0.45425867507886436], [[frozenset({'whipped/sour cream', 'other vegetables'}),
frozenset({'whole milk'})], 0.5070422535211268], [[frozenset({'root vegetables', 'tropical
fruit'}), frozenset({'other vegetables'})], 0.5845410628019324], [[frozenset({'whole milk',
'tropical fruit'}), frozenset({'other vegetables'})], 0.40384615384615385],
[[frozenset({'other vegetables', 'tropical fruit'}), frozenset({'whole milk'})],
0.47592067988668557], [[frozenset({'yogurt', 'whipped/sour cream'}), frozenset({'whole
milk'})], 0.5245098039215687], [[frozenset({'root vegetables', 'rolls/buns'}),
frozenset({'other vegetables'})], 0.502092050209205], [[frozenset({'other vegetables',
'soda'}), frozenset({'whole milk'})], 0.4254658385093168], [[frozenset({'root vegetables',
'tropical fruit'}), frozenset({'whole milk'})], 0.5700483091787439],
[[frozenset({'fruit/vegetable juice', 'other vegetables'}), frozenset({'whole milk'})],
0.4975845410628019], [[frozenset({'yogurt', 'root vegetables'}), frozenset({'whole

```
milk'})], 0.562992125984252], [[frozenset({'whipped/sour cream', 'yogurt'}),
frozenset({'other vegetables'})], 0.49019607843137253], [[frozenset({'yogurt', 'tropical
fruit'}), frozenset({'whole milk'})], 0.5173611111111112], [[frozenset({'root vegetables',
'whole milk'}), frozenset({'other vegetables'})], 0.47401247401247404], [[frozenset({'root
vegetables', 'other vegetables'}), frozenset({'whole milk'})], 0.4892703862660944],
[[frozenset({'pip fruit', 'whole milk'}), frozenset({'other vegetables'})],
0.44932432432432434], [[frozenset({'pip fruit', 'other vegetables'}), frozenset({'whole
milk'})], 0.5175097276264592], [[frozenset({'yogurt', 'citrus fruit'}), frozenset({'whole
milk'})], 0.47417840375586856], [[frozenset({'other vegetables', 'bottled water'}),
frozenset({'whole milk'})], 0.4344262295081967], [[frozenset({'domestic eggs', 'whole
milk'}), frozenset({'other vegetables'})], 0.4101694915254237], [[frozenset({'domestic
eggs', 'other vegetables'}), frozenset({'whole milk'})], 0.5525114155251142],
[[frozenset({'other vegetables', 'rolls/buns'}), frozenset({'whole milk'})],
0.4200477326968974]]
```

- FP-Growth算法关联规则结果：

[[[{frozenset({'sliced cheese'})}, {frozenset({'whole milk'})}], 0.43983402489626555], [[{frozenset({'ham'})}, {frozenset({'whole milk'})}], 0.44140625], [[{frozenset({'butter milk'})}, {frozenset({'whole milk'})}], 0.41454545454545455], [[{frozenset({'oil'})}, {frozenset({'whole milk'})}], 0.40217391304347827], [[{frozenset({'onions'})}, {frozenset({'other vegetables'})}], 0.45901639344262296], [[{frozenset({'hamburger meat'})}, {frozenset({'other vegetables'})}], 0.41590214067278286], [[{frozenset({'hamburger meat'})}, {frozenset({'whole milk'})}], 0.4434250764525994], [[{frozenset({'sugar'})}, {frozenset({'whole milk'})}], 0.4444444444444444], [[{frozenset({'cream cheese'})}, {frozenset({'whole milk'})}], 0.4153846153846154], [[{frozenset({'white bread'})}, {frozenset({'whole milk'})}], 0.4057971014492754], [[{frozenset({'chicken'})}, {frozenset({'whole milk'})}], 0.4099526066350711], [[{frozenset({'chicken'})}, {frozenset({'other vegetables'})}], 0.41706161137440756], [[{frozenset({'frozen vegetables'})}, {frozenset({'whole milk'})}], 0.4249471458773784], [[{frozenset({'beef'})}, {frozenset({'whole milk'})}], 0.4050387596899225], [[{frozenset({'curd'})}, {frozenset({'whole milk'})}], 0.4904580152671756], [[{frozenset({'butter'})}, {frozenset({'whole milk'})}], 0.4972477064220184], [[{frozenset({'butter'}), frozenset({'other vegetables'})}, {frozenset({'whole milk'})}], 0.5736040609137056], [[{frozenset({'butter'}), frozenset({'whole milk'})}, {frozenset({'other vegetables'})}], 0.41697416974169743], [[{frozenset({'pork'}), frozenset({'other vegetables'})}, {frozenset({'whole milk'})}], 0.4694835680751174], [[{frozenset({'whole milk'}), frozenset({'pork'})}, {frozenset({'other vegetables'})}], 0.45871559633027525], [[{frozenset({'margarine'})}, {frozenset({'whole milk'})}], 0.4131944444444444], [[{frozenset({'domestic eggs'})}, {frozenset({'whole milk'})}], 0.47275641025641024], [[{frozenset({'other vegetables'}), frozenset({'domestic eggs'})}, {frozenset({'whole milk'})}], 0.5525114155251142], [[{frozenset({'whole milk'}), frozenset({'domestic eggs'})}, {frozenset({'other vegetables'})}], 0.4101694915254237], [[{frozenset({'whipped/sour cream'})}, {frozenset({'other vegetables'})}], 0.40283687943262414], [[{frozenset({'whipped/sour cream'})}, {frozenset({'whole milk'})}], 0.44964539007092197], [[{frozenset({'yogurt'}), frozenset({'whipped/sour cream'})}, {frozenset({'other vegetables'})}], 0.49019607843137253], [[{frozenset({'yogurt'}), frozenset({'whipped/sour cream'})}, {frozenset({'whole milk'})}], 0.5245098039215687], [[{frozenset({'whipped/sour cream'}), frozenset({'other vegetables'})}, {frozenset({'whole milk'})}], 0.5070422535211268], [[{frozenset({'whole milk'}), frozenset({'whipped/sour cream'})}, {frozenset({'other vegetables'})}], 0.45425867507886436], [[{frozenset({'fruit/vegetable juice'}), frozenset({'other vegetables'})}, {frozenset({'whole milk'})}], 0.4975845410628019], [[{frozenset({'pip fruit'}), frozenset({'other vegetables'})}, {frozenset({'whole milk'})}], 0.5175097276264592], [[{frozenset({'whole milk'}), frozenset({'pip fruit'})}, {frozenset({'other vegetables'})}], 0.44932432432432434], [[{frozenset({'citrus fruit'}), frozenset({'root vegetables'})}, {frozenset({'other vegetables'})}], 0.5862068965517241], [[{frozenset({'citrus fruit'}), frozenset({'yogurt'})}, {frozenset({'whole milk'})}], 0.47417840375586856], [[{frozenset({'citrus fruit'}), frozenset({'other vegetables'})}, {frozenset({'whole milk'})}], 0.45070422535211267], [[{frozenset({'citrus fruit'}), frozenset({'whole milk'})}, {frozenset({'other vegetables'})}], 0.4266666666666667], [[{frozenset({'pastry'}), frozenset({'other vegetables'})}, {frozenset({'whole milk'})}], 0.46846846846846846], [[{frozenset({'tropical fruit'})}, {frozenset({'whole milk'})}], 0.40310077519379844], [[{frozenset({'tropical fruit'}), frozenset({'root vegetables'})}, {frozenset({'whole milk'})}], 0.5700483091787439], [[{frozenset({'tropical fruit'}), frozenset({'root vegetables'})}, {frozenset({'other vegetables'})}], 0.5845410628019324], [[{frozenset({'tropical fruit'}), frozenset({'rolls/buns'})}, {frozenset({'whole milk'})}], 0.4462809917355372], [[{frozenset({'tropical fruit'}), frozenset({'yogurt'})}, {frozenset({'other vegetables'})}], 0.42013888888888889], [[{frozenset({'tropical fruit'}), frozenset({'yogurt'})}, {frozenset({'whole milk'})}], 0.5173611111111112],

```
[[{frozenset({'tropical fruit'}), frozenset({'other vegetables'})}, {frozenset({'whole
milk'})}], 0.47592067988668557], [[{frozenset({'tropical fruit'}), frozenset({'whole
milk'})}, {frozenset({'other vegetables'})}], 0.40384615384615385], [[{frozenset({'root
vegetables'})}, {frozenset({'other vegetables'})}], 0.43470149253731344],
[[{frozenset({'root vegetables'})}, {frozenset({'whole milk'})}], 0.44869402985074625],
[[{frozenset({'rolls/buns'}), frozenset({'root vegetables'})}, {frozenset({'other
vegetables'})}], 0.5020920502092055], [[{frozenset({'rolls/buns'}), frozenset({'root
vegetables'})}, {frozenset({'whole milk'})}], 0.5230125523012552],
[[{frozenset({'yogurt'}), frozenset({'root vegetables'})}, {frozenset({'other
vegetables'})}], 0.5], [[{frozenset({'yogurt'}), frozenset({'root vegetables'})},
{frozenset({'whole milk'})}], 0.562992125984252], [[{frozenset({'root vegetables'}),
frozenset({'other vegetables'})}, {frozenset({'whole milk'})}], 0.4892703862660944],
[[{frozenset({'root vegetables'}), frozenset({'whole milk'})}, {frozenset({'other
vegetables'})}], 0.47401247401247404], [[{frozenset({'bottled water'}), frozenset({'other
vegetables'})}, {frozenset({'whole milk'})}], 0.4344262295081967],
[[{frozenset({'yogurt'})}, {frozenset({'whole milk'})}], 0.40160349854227406],
[[{frozenset({'rolls/buns'}), frozenset({'yogurt'})}, {frozenset({'whole milk'})}],
0.4526627218934911], [[{frozenset({'yogurt'}), frozenset({'other vegetables'})},
{frozenset({'whole milk'})}], 0.5128805620608899], [[{frozenset({'soda'}),
frozenset({'other vegetables'})}, {frozenset({'whole milk'})}], 0.4254658385093168],
[[{frozenset({'rolls/buns'}), frozenset({'other vegetables'})}, {frozenset({'whole
milk'})}], 0.4200477326968974]]
```

两种算法的出结果一致，都有60组置信度超过0.4的关联规则。

4. 总结与收获

4.1收获

- 最大的收获就是对关联度规则有了透彻的了解，对置信度这个参数深入学习掌握，经过几番折腾，对比 Apriori 算法的关联规则书写FP-Growth算法的置信度求解和关联规则。
- 通过对frozenset（）这不可变集合与可变集合set（）以及列表和字典的转换使用，对这几种类型的增删改操作得到熟练锻炼。
- 在运用集合遍历求解置信度过程中，学到了一个颠覆认知的观念，就是集合可以比较大小。例如｛1，2｝>｛1｝是True，也就是一个集合是一个集合的子集，子集集合比父集合要小。而不是通过元素数量比较大小。例如｛1，2｝>｛3｝是不对的。
- 由于 Apriori算法与FP-Growth算法对数据处理的类型不同，不能局限于一种数据类型，要学会变通，客服惯性思维影响，也正是因为惯性思维的限制导致对问题的突破困难。

4.2总结

- 与枚举所有的项集相比， Apriori算法利用频繁项集的单性，大大减少了候选集的数量，从而提高了关联规则挖掘的效率。然而这种方法仍然可能构造大量无用的候选项集。
- FP-Growth算法首先通过遍历两次原始数据集，将原始数据集表示成一个压缩的树形数据结构 FP-tree。后续的频繁项集挖掘直接利用FP-tree, 而不再依赖于原始数据集。
- FP-tree通常比原始数据集更小, 因此与需要多次遍历原始数据集的 Apriori算法相比, FP-Growth往往能够获得更高的性能。同时，与 Apriori进行相比, FP-Growth没有生成无用的候选项集，运行相对快一个数量级。