FP-Growth 关联规则：原理、公式、应用与实战

2025 年 9 月 21 日

1 引言

FP-Growth 通过构建频繁模式树（FP-tree）高效挖掘频繁项集，避免 Apriori 对候选集的海量生成。算法仅需一次数据库扫描构建 FP-tree，然后利用条件模式基递归挖掘，使其在大规模或稠密数据集上表现优于 Apriori。

2 原理与公式

2.1 FP-tree 构建

给定事务集合 $\mathcal{D}$ 及最低支持度 `min_supp`，FP-Growth 首先统计单项频率并删除不频繁项，然后根据频率对事务内项排序，再逐一插入 FP-tree。树的公共前缀会合并，节点计数累加；表头（header table）记录各项在树中的链表位置，支持后续遍历。

2.2 条件模式基

对表头中的每个项 i，收集从根到该项所有路径，形成条件模式基。利用这些带权路径可构建条件 FP-tree，并递归挖掘包含 i 的频繁模式。项集的支持度为到达该项节点计数的总和。由于仅扩展频繁前缀，搜索空间大幅缩减。

2.3 规则生成

得到频繁项集后，可按支持度、置信度与提升度生成关联规则 $X \Rightarrow Y$：

$$\mathrm{supp}(X) = \frac{|\{T \mid X \subseteq T,\, T \in \mathcal{D}\}|}{|\mathcal{D}|}, \tag{1}$$

$$\mathrm{conf}(X \Rightarrow Y) = \frac{\mathrm{supp}(X \cup Y)}{\mathrm{supp}(X)}, \tag{2}$$

$$\mathrm{lift}(X \Rightarrow Y) = \frac{\mathrm{conf}(X \Rightarrow Y)}{\mathrm{supp}(Y)}. \tag{3}$$

亦可结合确信度、杠杆率等指标筛选价值较高的规则。

3 应用与技巧

- **零售分析**：在海量 POS 数据中识别商品组合并规划促销方案。
- **网页行为挖掘**：从点击流中发现常见访问路径或资源组合。
- **工业监控**：定位报警或事件的共现模式，辅助故障诊断。
- **实用建议**：调节 `min_supp` 控制树规模，对事务内项按频率排序保证结果稳定，为稠密数据设置递归深度上限，并结合业务约束验证规则。

4 Python 实战

脚本 `gen_fp_growth_figures.py` 生成模拟交易数据，实现精简 FP-Growth 算法，并输出支持度-置信度散点与提升度分布图，帮助评估挖掘结果。

Listing 1: 脚本 $\mathrm{gen}_f p_g rowth_f igures.py$

```
frequent_itemsets = fpgrowth(transactions, min_support=0.06)
rules = derive_rules(frequent_itemsets, min_confidence=0.5)

for lhs, rhs, support, confidence, lift in rules:
    support_vals.append(support)
    confidence_vals.append(confidence)
    lift_vals.append(lift)
```

5 实验结果

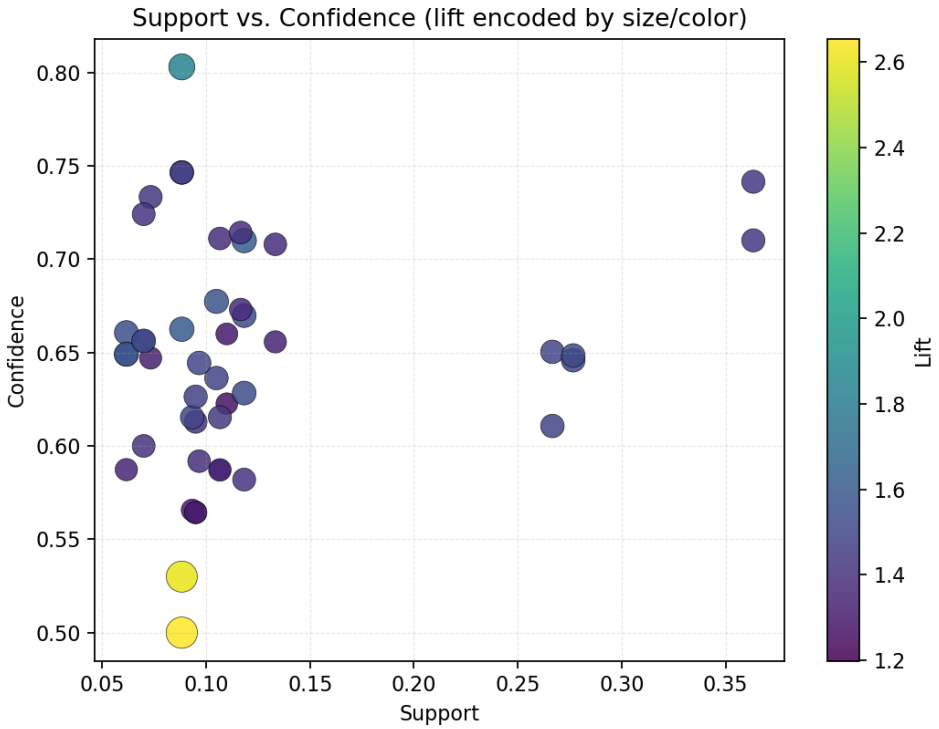

图 1: FP-Growth 规则的支持度-置信度散点图，点大小对应提升度

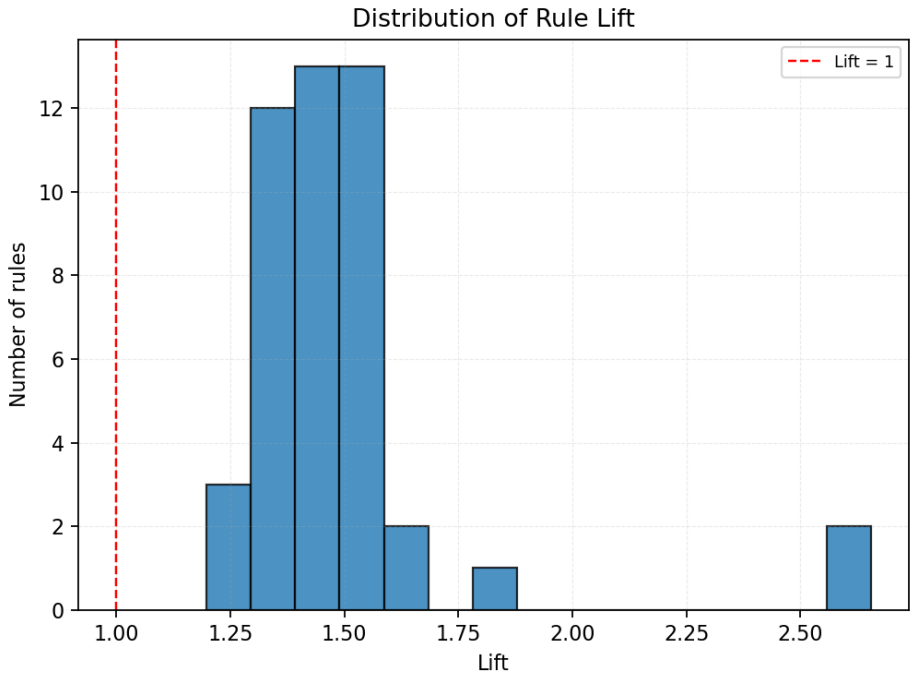

图 2: 提升度分布，突出高关联度规则

6 总结

FP-Growth 通过 FP-tree 与条件模式基避免暴力枚举频繁项集。只要合理设置支持度阈值、排序策略及评价指标，就能在大规模数据上高效挖掘可解释的关联规则。示例展示了如何利用可视化评估模式质量并调整参数。