

名企课 lesson 2 关联规则

推荐系统的几种算法：

Content base: 内容特征表示，特征学习，推荐列表 静态

协同过滤：群体智能，用户历史行为 动态

关联规则：Transaction, 频繁项集，关联规则挖掘

基于效用：效用函数的定义

基于知识：知识图谱创建 → 直接给结果

组合推荐：实际工作常用，综合几种算法

关联规则：association rules / basket analysis

ex: customer buy A, then possibility of buy B?

支持度：某商品组合出现的次数和总次数的比例

支持度 \uparrow 出现频率 \uparrow

(support)

置信度（条件概率）：购买商品A，会有多少概率买商品B

confidence

↓ 条件 $P(B|A)$ 与 $P(A|B)$ 是不一定相同的

提升度：商品A的出现对商品B出现概率提升程度

成因 A 对 B 的影响 solo

提升度 $(A \rightarrow B) = \text{置信度}(A \rightarrow B) / \text{支持度}(B)$ 大小作比较

3种情况：提升度 $(A \rightarrow B) > 1$: 有提升

$= 1$: 没有提升也没下降

< 1 : 有下降

最重要的指标是提升度

Apriori 算法：查找 frequent item set 的过程

frequent set: 支持度大于等于最小支持度 (Min Support)阀值的项集

non-frequent set: 支持度小于 Min support 的 set

K 代表商品个数, ex: $K=1$ 一个商品 不同 dataset 的 mini support 不同
数量少的情况下, mini support 可以取的大一些
商品组合 $K=2$ 当商品 4, 6 没有在 $K=1$ 的 frequent item set 里时,
那么, 在 $K=2$ 的情况下也不会出现在 frequent item set 中
因为他们组合会更少
mini support 是固定的筛子

$K=2$

1, 2

1, 3

2, 3

3, 5

$K=3$

1, 2, 3

2, 3, 5

1, 3, 5

1, 2, 5

从 $K=2$ 的选项中挑选

1, 2, 3, 5 组合

SKU 越大, Mini support 会设的越小 ex: 10 万 SKU, mini support 设为 0.1 不好, 0.01 有道理

SKU: 商品的编号 ex: iPhone 12 Mini support 越大, frequent item set 的数 目越多。

比商品的平均值 → 频率的定义

Apriori 流程: ① $K=1$, 计算 K 项集支持度 ② 剔掉 $< \text{minisupport}$ 物项

③ 如果 $K-1$ 项集为空, 则对应 $K-1$ 项集的结果为最终结果。否则 $K+1$,
repeat Step 1-3

工具包:

efficient, apriori 率比较高 比较快, 结论简单
maxextend, 慢一些, 但结论多一些

集合: 无顺序, 不可重复

efficient - apriori: itemsets 频繁项集 rules 频繁规则: min confidence

Min confidence = 1: 买A的情况下必买B

min support \rightarrow itemsets \rightarrow min-confidence

support solo confidence 条件

mlxtend 使用 apriori 的时候需要提前做一个 one-hot 编码

min support 可以先设个比较小的值再排序

lift 提升度 离散变量用 one-hot, 连续变量就不用了

爬虫: requests+bs4 八爪鱼 selenium

selenium 需要 chrome 浏览器以及对应版本的 chrome driver

电影演员案例: lift 值比较高 商品案例: lift 比较小

关联规则: 基于购物清单(个体) 当前的需求: 当下的一次购买
协同过滤: 个性化(某一群体) 长期偏好: 用户历史的行驶轨迹
把数据都看作 transaction \rightarrow 购物篮分析
最小支持度及 min置信度是实验得的。

min support: 可以从高到低输出前 20 个项集支持度作参考
(0.01 - 0.5) 跨度很大

min confidence: 0.5 ~

lift > 1

Apriori 的不足：①可能产生大量候选集，原因是排列组合把可能的项集都组合出来
②每次计算都重新扫描 dataset, Compute 每个项集 support, waste 空间, 时间

MySQL 索引 | B+树 }
Oracle B树 } = 叉树

FP Growth: ① FP 树存储 frequent set, 减少存储空间
② 只遍历 dataset 2 次, 减少计算量

创建 item header table

① 先扫描数据集, 对满足最小支持度 K=1 按 support 从高到低排序
删除不满足 min support 的项

header table 包括 项 支持度 链表

② 对每条购买记录, 按项头表顺序进行排序并过滤
构造 FP 树, 根节点记为 Null 节点

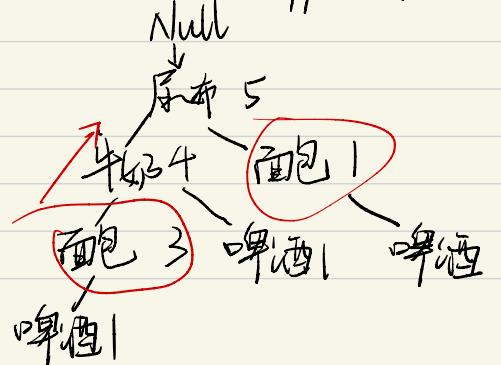
③ 再次扫描数据集, 把 step 2 的记录插入 FP 树中, 节点如果存在
就计数 Count + 1, 不存在就创建, 并更新链表。

④ 从“啤酒”(频数表最后一次)开始, 条件模式基: 去掉“啤酒”

然后在剩下的选项中通过 Min support 来筛选频繁项集

项	support
尿布	5
牛奶	4
面包	4
啤酒	3

✓



找出“面包”节点分支：
尿布3, 牛奶: 3, 面包: 3

→ TID
T1 尿布, 牛奶? 3
T2 尿布? 1

尿布: 1, 面包: 1

然后“牛奶”的频繁项集，最后“尿布”

树 | 借存
|
搜索, 检查

开源工具: pypi.org

import fptools as fp

Spark.mllib 并行

实战 spark → FP Growth

Python → apriori, mlxtend

spark 运行比较占内存

相关性分析：如果 $x \& y$ 没有相关性，回归意义不大

pearson系数: $P_{x,y} = \frac{\text{cov}(X,Y)}{\sigma_x \sigma_y}$

回归分析(Regression):
MSE, $\text{loss} = \frac{1}{m} \sum_{i=1}^m (\hat{y}_i - y_i)^2$
MIE, $\text{loss} = \frac{1}{m} \sum_{i=1}^m |\hat{y}_i - y_i|$

MSE 用的比较多，原因是更容易收敛。计算效率更好，梯度下降斜率更大
MIE 的可解释性更强

clf = linear_model.LinearRegression()

coef - 存放系数

Score(X, y) R方确定系数

$$SS_{res} = \sum (y_i - f(x_i))^2$$

$$SS_{tot} = \sum_{i=1}^n (y_i - \bar{y})^2$$

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

R-squared → 模型对现实数据拟合程度，评估预测效果

logistic regression 可以做多分类任务 $R^2 = 0.8$ 回归关系可以解释因变量 80% 的变异

R^2 越高越好，可解释性强 特征越重要，系数越高

时间序列效果一般都不错，但容易出现过拟合