# 第十二章 关联规则

## 一、基本理念

关联规则在数据分析中非常有用。

- 在超市中，数据通常是通过条码扫描器收集的。 这种数据库包含大量交易记录，每条记录都列出顾客一次购物中购买的所有商品。 因此，管理者可以识别哪些商品经常一起被购买，并基于统计结果优化超市布局、交叉销售策略、以及促销活动。
- 电信用户购买的可选增值服务（如电话等待、快速拨号、ISDN 等），可以帮助运营商确定如何将这些服务组合销售，以实现利润最大化。
- 银行业务中，分析客户使用的服务类型（如货币市场账户、存款证、汽车贷款等），可以用于识别可能对其他业务感兴趣或潜在需要新服务的客户。
- 在保险领域，出现异常组合的理赔项目可能意味着欺诈行为。
- 医疗患者的历史记录中，不同治疗方式组合出现的规律，可能用于预测潜在并发症，为诊断与治疗提供依据。

---

## 二、概念

1. 关联规则的体现
   - 关联规则可以通过数据挖掘自动生成：
     - 通过筛选数据找出频繁模式（频繁项集）
     - 规则通常表现为 **“If A then B（若A则B）”** 的形式
   - 在规则 *If A then B* 中：
     - 条件 A 称为 **前件（antecedent）**
     - 结果 B 称为 **后件（consequent）**
     - 前件与后件必须 **互斥**（即两者的商品项不能重叠）
   - 示例规则：
     - **面包 和 牛奶 → 汽水 和 花生酱　[s=50%, c=90%]**
     - 表示： 在全部交易中，有 50% 同时包含面包和牛奶，且其中 90% 的交易也包含汽水和花生酱。
2. 关联规则概念
   - 事务：一组同时出现的商品或项目的集合，例如一次购物中顾客购买的所有商品。
   - 项集：由若干商品组成的集合，可以作为规则的 **前件（antecedent）** 或 **后件（consequent）**。
   - K项集：包含 **K 个商品** 的项集。例如 `[egg, milk]` 是一个 **2-项集**。
   - 频繁项集：如果某个项集的出现次数（支持度计数）大于设定的最小支持度阈值，则该项集称为 **频繁项集**。

# 三、方法

1. 度量指标
   - 支持度
     **某项集在所有事务中出现的次数与总事务数的比例**

$$support(A \Rightarrow B) = P(A \cap B) = \frac{包含A和B的事务数}{总事务数}$$

   - 置信度
     **前件中的项与后件中的项出现在同一笔交易中的百分比，其实就是条件概率，包含前件的事务中包含后件的比例**

$$Confidence(A \Rightarrow B) = P(B|A) = \frac{support(A \cup B)}{support(A)} = \frac{support(A \Rightarrow B)}{support(A)}$$

| 指标 | 支持度（Support） | 置信度（Confidence / 条件概率） |
| --- | --- | --- |
| **定义** | 衡量某个项集在数据集中出现的频率 | 衡量在另一个项集出现的前提下，该项集出现的可能性 |
| **作用** | 用于识别数据集中出现频繁的项集 | 用于评估规则的强度 |
| **解释** | 项集出现的事务占总事务的百分比 | 在包含左侧项集的事务中，右侧项集出现的百分比 |

# 四、Apriori算法

## 1. 核心原理：Apriori 属性（反单调性）

这是算法高效运行的基石，用于减少搜索空间。

- **定义：**
  - 一个**频繁项集**的所有**子集**必须也是频繁的。
  - 如果一个项集**不是**频繁的，那么它的所有**超集**也一定不是频繁的。
- **具体说明：**
  - 如果一个项集 $G$ 不满足最小支持度阈值（$P(G) < min\text{-}sup$），那么它不是频繁项集。
  - 如果在 $G$ 中加入一个物品 $A$，新项集 $G \cup \{A\}$ 的出现频率不可能比 $G$ 高。因此，新项集支持度一定 $< min\text{-}sup$，可以直接排除。

## 2. 步骤一：频繁项集生成 (Frequent Itemset Generation)

**目标：** 找出所有满足最小支持度的项集。
**基本流程：** **先统计单个物品（1-项集），保留满足最小支持度的；然后迭代生成 2-项集、3-项集……直到无法生成**。

这一步主要包含两个具体操作循环：**连接 (Join)** 和 **剪枝 (Prune)**。

### 2.1 连接步骤 (Join Step)

- **目的：** 把已有的频繁 $k-1$ 项集组合，生成候选 $k$ 项集（$C_k$）。
- **前提：** 每个项集（Itemset）内的元素必须是**排序**的。
- **生成规则：**
  - 将 $L_{k-1}$ 中的每一项与其自身进行连接。
  - **连接条件：** 要连接 $L_1[k-1]$ 和 $L_2[k-1]$，它们必须有 $(k-2)$ 个共同元素（即前缀相同），且 $l_1[k-1] < l_2[k-1]$（最后一个元素前者小于后者）。
  - *注：对于 $L_3$（生成 3-项集），前两个元素应该匹配。*
  - **操作：** $L_1[k-1] \bowtie L_2[k-1]$。
- **作用：** 该条件仅仅是为了确保**不生成重复项**。

### 2.2 扩展并剪枝 (Expand & Prune)

- **扩展：** 基于连接步骤的结果，形成候选 $(k+1)$-项集。
- **剪枝（核心）：**
  - 利用 Apriori 原则，检查候选项集的所有**真子集**。
  - **判断逻辑：** 频繁项集的所有非空子集也必须是频繁的。
  - **操作：** 只要有一个子集不在上一轮的频繁项集列表中，该候选项集就作废（直接删除），不再进行数据库计数。
  - *例子：* 如果想要 $[A, B, C, D]$ 成为频繁项集，那么它的所有 3-项子集（如 $[A, B, C]$ 等）都必须在 $L_3$ 中。

## 3. 步骤二：生成关联规则 (Rule Generation)

**目标：** 在找到的频繁项集基础上，挖掘出满足置信度的规则。

- **基本操作：**
  - 对于每一个频繁项集 $G$，生成 $G$ 的所有非空子集（忽略空集和全集）。
  - 对于 $G$ 的每一个非空子集 $s$，输出规则：$s \Rightarrow (G-s)$。
  - *例子：* 频繁项集 $\{A, B, C\}$，可以生成规则 $A, B \to C$ 或 $A \to B, C$。
- **筛选条件：**
  - 计算每一条规则的**置信度**。
  - 剪枝掉那些不满足**最小置信度 (min-conf)** 阈值的规则。
- **置信度的反单调性（重要性质）：**
  - 如果我们把更多的项从规则的左边（条件）移动到右边（结果），规则的置信度只会**降低**（或者持平），决不会增高。
  - *注：利用此性质也可以进行规则生成的剪枝。*

**总结：** 最终选择的是那些同时具有**良好支持度**（步骤一筛选）和**良好置信度**（步骤二筛选）的规则。

---

## 五、误导性强关联规则

高支持度和高置信度并不一定代表规则是有用的

1. 规则无效
   比如说在总体1000人中有400个男人600个女人，所有人都买了牛奶，根据生成规则和指标计算出$Milk \Rightarrow male$的支持度是0.4，置信度是0.4
   在设定最小支持度的水平下可以判定这个是一条强规则，但是实际上他是误导性的，因为男性顾客在所有人群中的比例本身就是40%
2. 负相关

| | 吃早饭 | 不吃早饭 |
|---|---|---|
| 优秀 | 60 | 40 |
| 不优秀 | 66 | 14 |

根据上述表格计算$优秀 \rightarrow 吃早饭$的关联关系
通过计算得到支持度为$\frac{1}{3}$，随后计算置信度，优秀且吃早饭的人在优秀的人中的比例为$0.6$
吃早饭人的比例为0.7，这么进行比较发现，优秀的人反而更少吃早饭，这就对我们的关联分析产生了负向的影响

---

## 六、解决办法——提升度Lift

针对上述问题，引入了第三个指标提升度

$$Lift = \frac{confidence(A \Rightarrow B)}{support(A)support(B)} = \frac{p(AB)}{P(A)P(B)}$$

该公式计算的是该关联关系的置信度除以后件发生概率，得到的是关联关系针对后件的一个大小情况，其回答了这么一个问题，前件的出现会不会使得后件更加容易发生，即前件对后件的的出现概率是否有影响
总结一下

- **Lift = 1**：代表两者独立，无关联（就是本图的情况，规则无效）。
- **Lift > 1**：代表正相关（买了A确实更有可能买B，这才是我们要找的规则）。
- **Lift < 1**：代表负相关（买了A反而更不可能买B）。
  使用该方法我们可以剔除掉$Lift <= 1$的情况

---

# 七、总结

1. 算法缺点
   虽然Apriori算法逻辑清晰，但在处理大数据集或长模式的时候效率特别低
   - 候选项集数量巨大——核心痛点
     这个是该算法最大的缺陷，通常被称为组合爆炸
     对于处理海量候选项集的时候非常低效，通常被称为组合爆炸
     假设数据中有长度为100的频繁项集，对于这个长度为100的项集，算法必须先找出它所有的子集，计算得到一个很大的数字
   - 多次扫描事务数据库
     Apriori算法是迭代的，每次寻找k-项集，都要扫描数据库一次，如果数据库很大，就需要重复读取消耗非常多的时间，导致速度极其缓慢
2. 算法优点
   - 易于理解
     关联规则挖掘的结果非常容易理解和解释，这使得我们能够清楚地说明在数据中发现的模式
   - 应用范围广
     关联规则挖掘可以用于零售，金融和医疗保健等广泛领域，有助于改进决策并增加收入