1. Sequential Pattern

Sequential Pattern（序列模式）

- Sequential Pattern 是指在时间顺序上经常出现的模式。
- 它描述了事件或事务之间的时间序列关系。
- 例如：
  - 顾客先购买 iPad，然后购买 Apple Pencil，最后购买 Magic Keyboard。
  - 股票价格上涨后，另一只股票可能在两天内上涨。

Sequence Support（序列支持度）

- Sequence Support 是指一个序列在序列数据库中出现的频率。
- 计算方式：
  - 支持度 = 包含该序列的客户数 / 总客户数
  - 注意：支持度是基于客户的，而不是单个事务。

实例

```
Customer ID | Transaction Time | Items Bought
1           | Jun 25           | {30}
1           | Jun 30           | {90}
2           | Jun 10           | {10, 20}
2           | Jun 15           | {30}
2           | Jun 20           | {40, 60, 70}
3           | Jun 25           | {30, 50, 70}
4           | Jun 25           | {30}
4           | Jun 30           | {40, 70}
4           | Jul 25           | {90}
5           | Jun 12           | {90}
```

Sequential Version

将每个客户的交易按时间顺序排列：

```
Customer ID | Sequence
1           | <(30), (90)>
2           | <(10 20), (30), (40 60 70)>
3           | <(30 50 70)>
4           | <(30), (40 70), (90)>
5           | <(90)>
```

计算支持度

- 目标序列：`<(30), (90)>`
  - 包含该序列的客户：C1 和 C4。

- 总客户数：5。
- 支持度 = 2 / 5 = 40%。

解释

- 该序列表示客户先购买商品 30，然后购买商品 90。
- 支持度为 40%，说明有 40% 的客户符合该模式。

Sequential Association Rule Mining（序列关联规则挖掘）步骤

1. **Sort Phase（排序阶段）**：将事务数据库转换为按客户 ID 和时间排序的序列数据库。
2. **Frequent Itemset Phase（频繁项集阶段）**：使用 Apriori 算法挖掘频繁项集。
3. **Transformation Phase（转换阶段）**：将每个客户序列转换为频繁项集的表示形式，例如 `<s1, s2, ...>` 转换为 `<l1, l2, ...>`，其中 `li` 是频繁项集。
4. **Sequence Phase（序列阶段）**：使用类似 Apriori 的算法基于频繁项集挖掘频繁序列。
5. **Maximal Phase（最大序列阶段）**：找到最长的频繁序列（无法进一步扩展的序列）。

实例

假设有以下序列数据库：

```
Customer ID | Sequence
1           | <(30), (90)>
2           | <(10 20), (30), (40 60 70)>
3           | <(30 50 70)>
4           | <(30), (40 70), (90)>
5           | <(90)>
```

- 最小支持度 = 25%。
- 频繁子序列：`<(30), (90)>`，支持度为 40%（C1 和 C4 满足该模式）。

AprioriAll Algorithm（AprioriAll 算法）

1. **Initialization（初始化）**：从所有频繁 1-项集开始。
2. **Candidate Generation（候选生成）**：从频繁序列大小为 `k−1` 的序列生成大小为 `k` 的候选序列。
3. **Support Counting（支持度计数）**：计算每个候选序列在序列数据库中的支持度。
4. **Pruning（剪枝）**：删除不满足最小支持度的候选序列。
5. **Iteration（迭代）**：对更大的序列重复步骤 2-4，直到无法生成更多候选序列。

Multi-level Association Rule（多层次关联规则）

？？看实例，看懂算法

- **定义**: Multi-level Association Rule 是指在不同抽象层次上挖掘的关联规则。
  - 例如，"牛奶"（Milk）和"面包"（Bread）的规则可以进一步细化为"2% 牛奶"（2% Milk）和"全麦面包"（Wheat Bread）。
- **特点**:

1. 项目通常具有层次结构（Hierarchy），例如：
    - Level 0: Electronics
    - Level 1: Laptop, Desktop
    - Level 2: Accessories (e.g., Mouse, Printer)
2. 高层次规则可能是常识性知识，而低层次规则可能更具体但支持度较低。
3. 需要选择适当的支持度阈值（Support Threshold）以平衡规则的数量和质量。

挖掘方法

1. **Top-down Approach（自顶向下方法）**：
    - 从高层次（Level 1）开始，逐步向下挖掘更具体的规则，直到无法找到更多频繁项集。
    - 每一层可以使用 Apriori 或其变体算法。
2. **支持度选择**:
    - **统一支持度**: 所有层次使用相同的支持度阈值。
    - **递减支持度**: 层次越低，支持度阈值越小。

实例

假设有以下层次结构和交易数据：

```
Level 0: Electronics
Level 1: Laptop, Desktop
Level 2: Accessories (e.g., Mouse, Printer)

Transaction ID | Items
T100           | Laptop, Printer
T200           | Desktop, Mouse
T300           | Mouse, Wrist Pad
T400           | Laptop, Digital Camera
T500           | Desktop, Antivirus
```

- 在 Level 1，可能发现规则：`Laptop -> Desktop`。
- 在 Level 2，可能发现更具体的规则：`Mouse -> Wrist Pad`。

冗余问题

- 如果低层次规则的支持度接近其高层次规则的“期望值”，则该规则可能是冗余的。
    - 例如：
        - R1: Milk -> Bread [支持度 = 8%]
        - R2: 2% Milk -> Bread [支持度 = 2%]
        - R2 是 R1 的子规则，可能是冗余的。

应用

- **零售分析**: 发现不同层次的商品购买模式。
- **医疗数据**: 分析不同层次的疾病关联。
- **市场营销**: 针对不同层次的客户需求制定策略。

Multi-dimensional Association Rule（多维关联规则）

- **定义**: Multi-dimensional Association Rule 是指涉及多个维度（Dimensions）或谓词（Predicates）的关联规则。
  - 例如，"年龄在 19-25 岁之间的学生更可能购买笔记本电脑"。
- **特点**:
  1. 规则可以是跨维度的（Inter-dimension）或混合维度的（Hybrid-dimension）。
  2. 跨维度规则涉及不同的维度，例如年龄和职业。
  3. 混合维度规则可能在同一维度中重复使用谓词，例如购买不同商品的关联。

挖掘方法

1. **Inter-dimension Association Rules（跨维度关联规则）**:
   - 不同维度之间的关联，例如：
     - `age(X, "19-25") ∧ occupation(X, "student") -> buys(X, "Laptop")`
2. **Hybrid-dimension Association Rules（混合维度关联规则）**:
   - 同一维度中重复使用谓词，例如：
     - `age(X, "19-25") -> buys(X, "Chips") ∧ buys(X, "Laptop")`
3. **处理方法**:
   - 将多维数据转换为单维数据，通过增加虚拟项（Virtual Items）表示不同维度的属性值。
   - 使用传统的关联规则挖掘算法（如 Apriori）进行挖掘。

实例

假设有以下数据：

```
Transaction ID | Age  | Occupation | Items Bought
T1             | 22   | Student    | Laptop
T2             | 30   | Engineer   | Smartphone
T3             | 19   | Student    | Chips, Laptop
T4             | 25   | Student    | Chips
T5             | 35   | Manager    | Laptop
```

- 跨维度规则：`age(19-25) ∧ occupation(Student) -> buys(Laptop)`。
- 混合维度规则：`age(19-25) -> buys(Chips) ∧ buys(Laptop)`。

应用

- **市场分析**: 分析不同年龄、职业和购买行为之间的关联。
- **医疗数据**: 研究不同年龄段和健康状况之间的关联。
- **教育领域**: 分析学生的学习行为和成绩之间的关系。

Mining Quantitative Association Rule（挖掘数量型关联规则）

- **定义**: Quantitative Association Rule 是指涉及数量型属性（Quantitative Attributes）的关联规则，例如年龄、工资等连续值。
  - 例如，"年龄在 30-40 岁之间且工资在 30K-40K 之间的人更可能购买电视"。

- **特点**:
    1. 数量型属性需要离散化（Discretization）以便进行关联规则挖掘。
    2. 离散化方法可以是静态的（Static）或动态的（Dynamic）。
    3. 规则可以结合多个数量型属性和分类属性。

挖掘方法

1. **Static Discretization（静态离散化）**:
    - 基于预定义的概念层次（Concept Hierarchies）对数量型属性进行分段。
    - 例如，将工资分为“0-20K”、“21-30K”等区间。
2. **Dynamic Discretization（动态离散化）**:
    - 根据数据分布动态生成区间，以满足某些挖掘标准（如最大化规则的置信度或紧凑性）。
3. **Clustering（聚类）**:
    - 使用基于距离的聚类方法对数量型属性进行分组。
    - 例如，"年龄在 30-40 岁且工资在 30K-40K 之间的人更可能购买电视"。

实例

假设有以下交易数据：

```
Transaction ID | Age  | Salary  | Items Bought
T1             | 32   | 35K     | TV
T2             | 45   | 50K     | Laptop
T3             | 28   | 25K     | Smartphone
T4             | 36   | 38K     | TV
T5             | 40   | 42K     | TV
```

- 静态离散化：将年龄分为“20-30”、“31-40”、“41-50”，将工资分为“20K-30K”、“31K-40K”、“41K-50K”。
- 动态离散化：根据数据分布生成区间，例如“28-36 岁”和“35K-42K”。
- 可能发现的规则：`Age(31–40) ∧ Salary(31K–40K) -> Buys(TV)`。

应用

- **零售分析**: 分析不同年龄段和收入水平的客户购买行为。
- **市场营销**: 针对特定年龄和收入群体设计促销活动。
- **医疗数据**: 研究不同年龄和健康指标之间的关联。

Mining Stock Data（股票数据挖掘）

- **定义**: Mining Stock Data 是指通过分析股票价格的时间序列数据，发现股票趋势和关联模式。
- **方法**:
    1. **Intra-stock Mining（单股票挖掘）**:
        - 分析单只股票的价格趋势，例如“上涨、上涨，然后下跌”。
        - 客户：股票（如 IBM, MSFT, INTC）。

        - 项目：价格趋势（如 Go-up, Go-down）。
    2. **Inter-stock Mining（跨股票挖掘）**：
        - 分析不同股票在同一时间窗口内的行为。
        - 每个时间窗口被视为一个“事务”，用于分析股票之间的关联。
        - 示例：开盘价-最高价-收盘价-最低价模式。

- **应用**:

    1. **趋势预测**: 通过挖掘历史数据预测未来的股票价格趋势。
    2. **投资策略**: 发现股票之间的关联规则，优化投资组合。
    3. **风险管理**: 识别可能的市场异常或风险模式。

Contributions of Association Rule Mining（关联规则挖掘的贡献）

- **灵活性**: Association Rule Mining 是一种灵活的技术，可以应用于几乎所有领域和数据类型。
    - 例如：零售、医疗、教育、金融等领域。
- **高效性**: 提供了高效的数据库解决方案，能够处理非常大的数据库。
    - 通过限制内存使用（如候选项集的管理），可以在有限的资源下完成挖掘。
- **统计分析**: 本质上是一种统计技术，能够发现数据中的隐藏模式和关系。
    - 例如：发现商品之间的购买关联、客户行为模式等。
- **易用性**: 算法简单易用，用户可以根据具体需求调整支持度和置信度等参数。
- **应用广泛**:
    1. **零售分析**: 发现商品之间的关联规则（如“啤酒 -> 尿布”）。
    2. **医疗数据**: 分析疾病与症状之间的关系。
    3. **市场营销**: 设计促销活动和推荐系统。
    4. **股票分析**: 研究股票价格的变化模式。
- **支持多种扩展**:
    - 支持多层次、多维度、数量型、稀有模式等扩展，满足不同场景的需求。
