lec16.2 Brute Force Algorithms

关联模式挖掘的暴力算法

关联规则生成框架

1. 阶段1：为给定的频率阈值 f 生成所有频繁项目集
   - 暴力算法
   - `Apriori` 算法
2. 阶段2：从频繁项目集中，生成在给定置信度阈值 c 下的关联规则
   - 对于每个频繁项目集 I：
     - 将 I 分割成所有可能的子集对 (X, Y)，使得 $Y = I - X$ 且 $X \cup Y = I$
     - 计算规则 $X \Rightarrow Y$ 的置信度。如果至少为 c，则存储规则 $X \Rightarrow Y$

暴力算法（Brute Force Algorithm）

过程说明：

- 假设：
  - 设 U 为项目的全集，且 $d = |U|$
- 项目集的数量：
  - U 的非空子集共有 $2^d - 1$ 个
- 候选项目集：
  - 每个子集都是一个可能的频繁项目集（即候选项目集）

暴力算法步骤：

1. 输入：
   - 项目的全集 U
   - 数据集 $\mathscr{D}$
   - 频率阈值 f
2. 操作：
   - 对于 U 的每个非空子集 I：
     - 计算 I 的支持度 $\sup(I)$
     - 如果 $\sup(I) \geq f$，则将 I 添加到频繁项目集家族中

主要问题：

- 时间复杂度：
  - 暴力算法的主要问题是时间复杂度呈指数增长。如果 $|U| = 1000$，则总共有 $2^{1000} > 10^{300}$ 个候选项目集需要计算

修剪搜索空间（Pruning the Search Space）

向下闭包性质（Downward Closure Property）

- **性质**: 每个频繁项目集的子集也是频繁的
- **含义**: 如果一个项目集是频繁的，那么它的所有子集也必须是频繁的

项目集的定义

- **定义**: 一个 k-项目集是包含 k 个元素的项目集
- **含义**: 例如，包含 3 个项目的项目集被称为 3-项目集（或3项集）

关键结论

- **结论**: 如果没有 k-项目集是频繁的，那么也不会有 $(k+1)$-项目集是频繁的
- **应用**: 通过使用向下闭包性质，可以显著减少搜索空间。具体来说，如果我们发现一个 k-项目集不是频繁的，我们就不需要考虑它的超集（$(k+1)$-项目集）

示例

假设我们有如下项目集：

- $\{A, B\}$
- $\{A, C\}$
- $\{B, C\}$
- $\{A, B, C\}$

如果我们发现 $\{A, B\}$ 不是频繁的，那么我们可以推断出 $\{A, B, C\}$ 也不会是频繁的，因为 $\{A, B\}$ 是它的子集

改进的暴力算法（Improved Brute Force Algorithm）

基本概念

- **频繁项目集性质**: 如果没有 k-项目集是频繁的，那么 $(k+1)$-项目集也不会是频繁的
- **目的**: 利用上述性质来减少计算量，避免检查不可能频繁的项目集

改进的暴力算法步骤

1. **输入**:
   - 项目的全集 U
   - 数据集 $\mathscr{D}$
   - 频率阈值 f
2. **过程**:
   - 对于 k 从 1 到 $|U|$:
     - 对于每个 k-项目集 I:
       - 计算 I 的支持度 $\sup(I)$
       - 如果 $\sup(I) \geq f$，则将 I 添加到频繁项目集家族中
     - 如果没有 k-项目集是频繁的，则停止

改进的暴力算法优势

- **更高效**: 相比于原始暴力算法，在稀疏数据集（每个交易中项目数量较少的数据集）上表现更好
- **减少计算量**: 通过提前停止（如果没有 k-项目集是频繁的），避免了不必要的计算

具体细节

- **假设**: 设数据集中一个交易中最多有 l 个项目
- **候选项目集数量**: 至多有 $\sum_{i=1}^{l}\binom{|U|}{i}$ 个候选项目集，比 $2^{|U|}$ 小得多。

例子

设 $|U|=1000$ 且 $l=10$
候选项目集数量为 $\sum_{i=1}^{10}\binom{1000}{i}$，数量级为 10^{23}