lec16.1 Association Pattern Mining Problem

关联模式挖掘 (Association Pattern Mining)

应用:

- 超市数据
  - 识别哪些商品经常一起购买
  - 提供关于目标市场营销和商品陈列的有用见解
- 文本挖掘
  - 识别共现的术语和关键词
- 泛化到依赖导向的数据类型
  - 网络日志分析
  - 软件错误检测
  - 时空事件检测

术语 (Terminology):
借用超市类比:

- 数据集中的对象称为transactions
- 输出: large itemsets（频繁项集或频繁模式）

用途:
频繁项集可用于生成关联规则 $X \Rightarrow Y$，其中 X 和 Y 是项目集（例如，$\{\text{Eggs}, \text{Milk}\} \Rightarrow \{\text{Yourgt}\}$）

- 向经常购买鸡蛋和牛奶的顾客推销酸奶
- 将酸奶放置在靠近鸡蛋和牛奶的货架上

The Frequent Pattern Mining Model

1. 项目的全集 (U):
   - 项目的全集包含 d 个项目，记作 U
2. 项目集:
   - 项目集是多个项目的集合
3. 数据集 ($\mathscr{D}$):
   - 数据集包含 n 个交易 $T_1, \ldots, T_n$，每个交易都是一个项目集
   - 每个交易可以表示为一个 d 维的二进制向量
   - 交易中的每个二进制属性表示来自 U 的一个特定项目
4. 支持度（support）:
   - 项目集 I 的支持度是数据集 $\mathscr{D}$ 中包含 I 作为子集的交易的比例，记作 $\sup(I)$

频繁项目集挖掘问题:

- 定义:
  - 给定一个包含交易的数据集 $\mathscr{D}$ 和一个频率阈值 f，确定所有在 $\mathscr{D}$ 中至少出现在 f 比例交易中的项目集

- 注意:
  - 较低的频率阈值会产生更多的频繁项目集
  - 过高的频率阈值可能导致没有频繁项目集

Example (let $f = 0.65$)

Transaction	Milk	Butter	Bread	Mushrooms	Onion	Carrot
1234	1	1	1	0	1	0
324	0	0	0	1	1	1
234	1	1	1	0	1	0
2125	1	1	1	1	0	1
113	1	0	0	1	1	0
5653	1	1	1	1	1	0

{Milk, Butter, Bread} is a large itemset

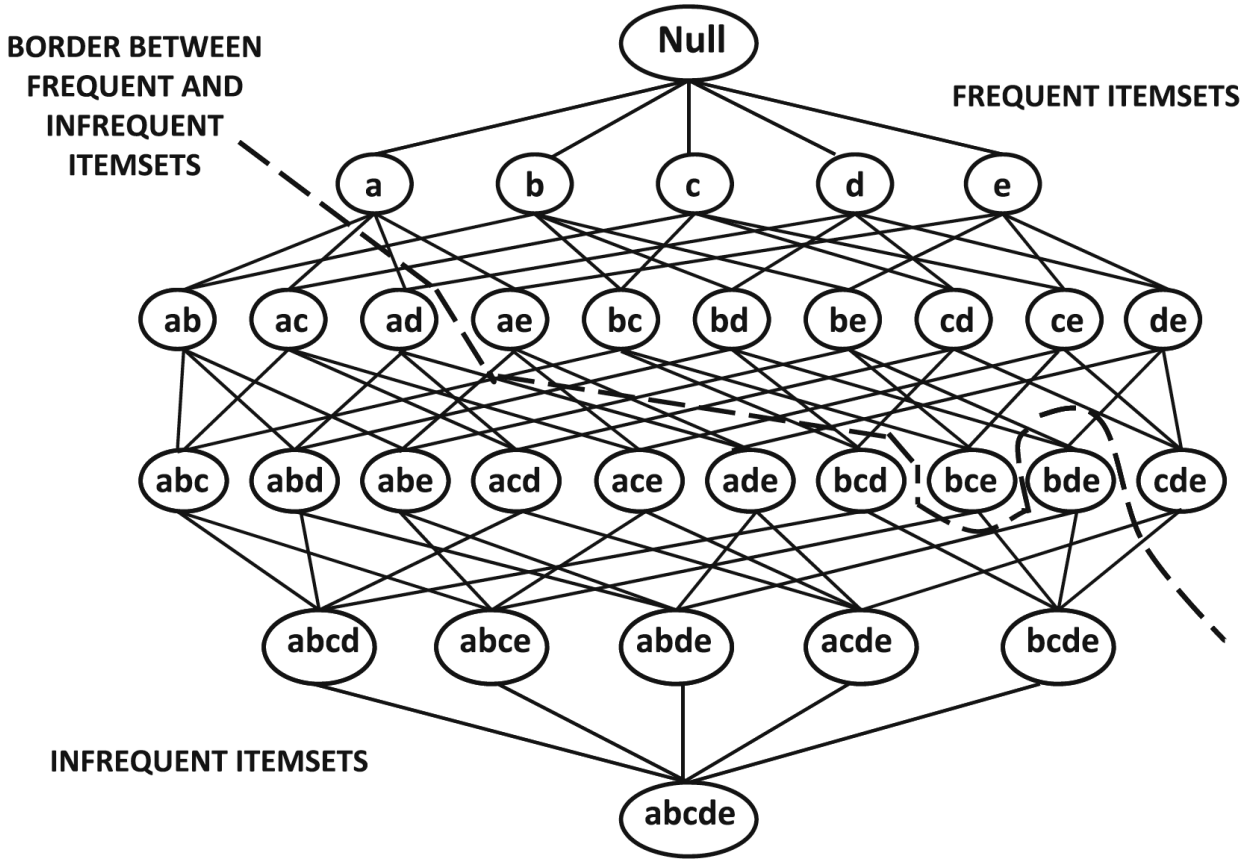
{Mushrooms, Onion, Carrot} is **not** is a large itemset

Monotonicity of Support 支持度的单调性

支持度单调性 Support Monotonicity Property:
项目集 I 的每个子集 J 的支持度至少和 I 的一样大，即对于每个 $j \subseteq I$，都有 $\sup(J) \geq \sup(I)$

向下闭包性 Downward Closure Property:
频繁项目集的每个子集也是频繁的

极大频繁项目集:
在给定频率阈值下，一个频繁项目集 I 是极大的：如果它是频繁的且没有它的超集是频繁的



The Frequent Pattern Mining Model

Example (let $f = 0.65$)

Transaction	Milk	Butter	Bread	Mushrooms	Onion	Carrot
1234	1	1	1	0	1	0
324	0	0	0	1	1	1
234	1	1	1	0	1	0
2125	1	1	1	1	0	1
113	1	0	0	1	1	0
5653	1	1	1	1	1	0

There are 3 maximal frequent itemsets: {**Milk, Butter, Bread**} and {**Milk, Onion**} and {**Mushrooms**}, but there 10 frequent itemsets in the dataset:
{**Milk**}, {**Butter**}, {**Bread**}, {**Onion**}, {**Mushrooms**}
{**Milk, Butter**}, {**Milk, Bread**}, {**Butter, Bread**}, {**Milk, Onion**}
{**Milk, Butter, Bread**}

Representation of Frequent Itemsets 频繁项目集的表示

1. 所有频繁项目集都可以从极大频繁项目集中推导出来:
   - 极大频繁项目集包含了所有频繁项目集的信息
2. 极大频繁项目集可以被视为频繁项目集的紧凑表示:
   - 它们提供了一个简洁的方式来表示频繁项目集
3. 然而，这种表示不包含项目集的支持度信息:
   - 尽管极大频繁项目集提供了频繁项目集的紧凑表示，但它们并不存储各个项目集的支持度值

Association Rules 关联规则

关联规则的形式:
我们希望生成形式为 $X \Rightarrow Y$ 的关联规则，这意味着如果一个交易包含项目集 X，那么它“很可能”包含项目集 Y

置信度:
为了衡量关联规则的可能性，我们使用规则的置信度，即在包含项目集 X 的前提下，交易包含项目集 Y 的条件概率

$$\text{conf}(X \Rightarrow Y) = \frac{\sup(X \cup Y)}{\sup(X)}$$

支持度:
根据定义，规则 $X \Rightarrow Y$ 的支持度记作 $\sup(X \Rightarrow Y)$，等于 $\sup(X \cup Y)$

Example: conf({Milk} ⇒ {Butter, Bread})

Transaction	Milk	Butter	Bread	Mushrooms	Onion	Carrot
1234	1	1	1	0	1	0
324	0	0	0	1	1	1
234	1	1	1	0	1	0
2125	1	1	1	1	0	1
113	1	0	0	1	1	0
5653	1	1	1	1	1	0

$\sup(\{\text{Butter, Bread, Milk}\}) = \frac{2}{3}$

$\text{conf}(\{\text{Milk}\} \Rightarrow \{\text{Butter, Bread}\}) = \frac{2}{3} \cdot \frac{6}{5} = \frac{4}{5}$

$\sup(\{\text{Milk}\}) = \frac{5}{6}$

定义

设 X 和 Y 为两个项目集，则规则 $X \Rightarrow Y$ 在频率阈值 f 和置信度阈值 c 下是一个关联规则，当且仅当：

- 规则 $X \Rightarrow Y$ 的支持度（项目集 $X \cup Y$ 的支持度）至少为 f
- 规则 $X \Rightarrow Y$ 的置信度至少为 c

解释:

- 第一个条件确保有足够多的交易与该规则相关
- 第二个条件确保该规则在条件概率方面有足够的强度

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 $X \Rightarrow Y$

置信度单调性性质:
设 X_1、X_2 和 I 是项目集，使得 $X_1 \subseteq X_2 \subseteq I$，则有

$$\text{conf}(X_2 \Rightarrow I - X_2) \geq \text{conf}(X_1 \Rightarrow I - X_1)$$

例如，我们有关联规则 $\{$黄油$\} \Rightarrow \{$牛奶, 面包$\}$ 和 $\{$黄油, 面包$\} \Rightarrow \{$牛奶$\}$，则第二个规则是多余的，因为它与第一个规则具有相同的支持度，但置信度不低于第一个规则？？？？ 什么b解释