# 关联分析：高级概念

电子信息与通信学院 冯 镔
fengbin@hust.edu.cn

# 分类属性

- 如何多种形式的变量进行关联分析？

| Session Id | Country | Session Length (sec) | Number of Web Pages viewed | Gender | Browser Type | Buy |
|---|---|---|---|---|---|---|
| 1 | USA | 982 | 8 | Male | IE | No |
| 2 | China | 811 | 10 | Female | Netscape | No |
| 3 | USA | 2125 | 45 | Female | Mozilla | Yes |
| 4 | Germany | 596 | 4 | Male | IE | Yes |
| 5 | Australia | 123 | 9 | Male | Mozilla | No |
| … | … | … | … | … | … | … |

关联规则例子:

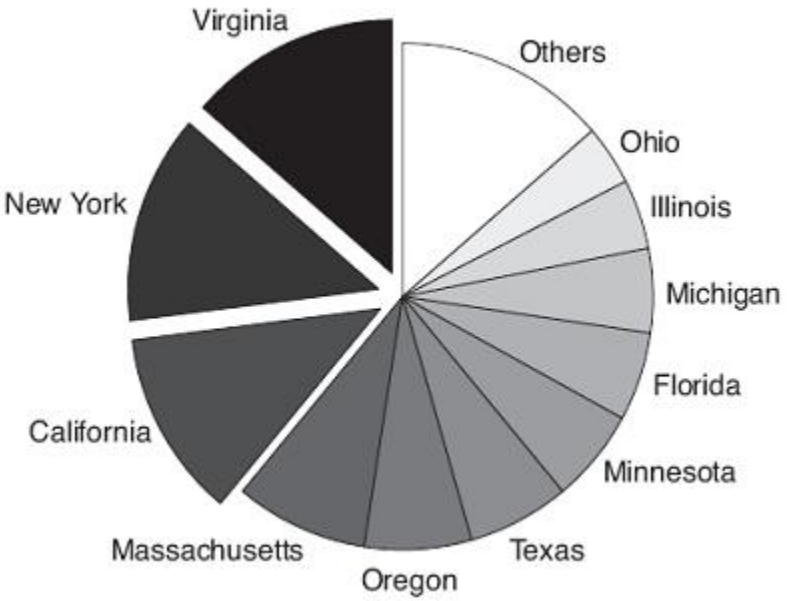
$$\{\text{Number of Pages} \in [5,10) \wedge (\text{Browser=Mozilla})\} \rightarrow \{\text{Buy = No}\}$$

# 分类属性

- 将分类属性变换成非对称二进制变量
- 通过为每个不同的属性-值对创建一个新的项来实现
  - 将**Browser Type**属性替换为
    - **Browser Type = Internet Explorer**
    - **Browser Type = Mozilla**
    - **Browser Type = Netscape**

# 分类属性

- 潜在问题：
- 1. 有些属性值可能不够频繁，不能成为频繁模式的一部分
  - 将相关的属性值分组，形成少数类别



# 分类属性

- **2.** 某些属性值的频率可能比其他属性高很多
  - 如：85%的人都有家庭计算机
    - {家庭计算机=是，网上购物=是}-->{关注隐私=是}
  - 产生许多冗余模式
- **3.** 尽管每个事务的宽度与原始数据中属性的个数相同，但计算时间可能增加
  - 避免产生包含多个来自同一属性的项的候选项集

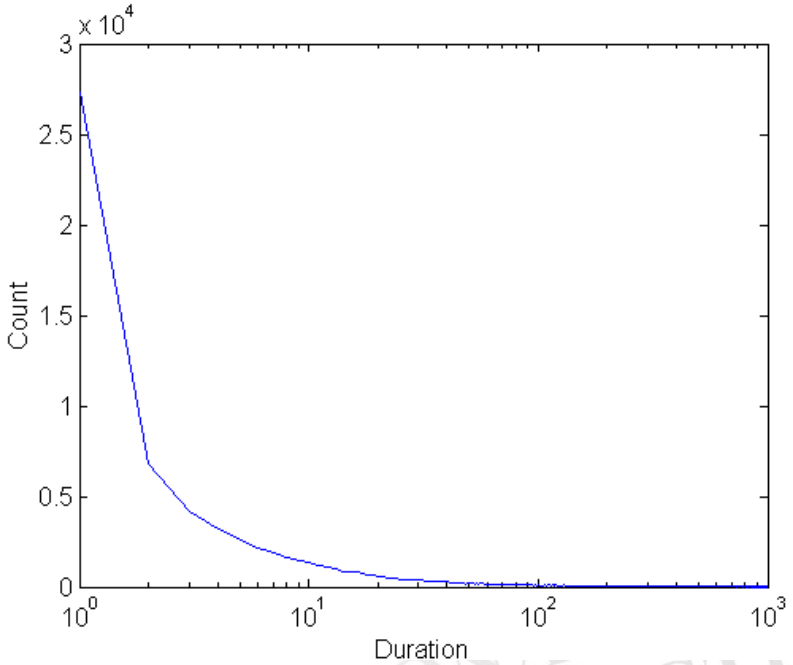
# 连续属性

- 不同类型的规则
  - Age∈[21,35) ∧ Salary∈[70k,120k) → Buy
  - Salary∈[70k,120k) ∧ Buy → Age: μ=28, σ=4
- 包含连续属性的规则通常称作量化关联规则
- 常用处理方法
  - 基于离散化方法
  - 基于统计学方法

# 连续属性

- 基于离散化的方法
  - 等区间宽度
  - 等频率
  - 基于熵或聚类

年龄∈[12, 16),年龄∈[16, 20),年龄∈[20, 24), …,年龄∈[56, 60)




| 男 | 女 | … | 年龄<13 | 年龄∈[13, 21） | 年龄∈[21, 30） | … | 关注隐私=是 | 关注隐私=否 |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | … | 0 | 0 | 1 | … | 1 | 0 |
| 1 | 0 | … | 0 | 0 | 0 | … | 0 | 1 |
| 1 | 0 | … | 0 | 0 | 1 | … | 1 | 0 |
| 0 | 1 | … | 0 | 0 | 0 | … | 1 | $0^7$ |

# 连续属性

- 属性离散化的一个关键参数是用于划分每个属性的区间个数

**R1:**年龄∈[16,24) -> 网上聊天=是（s=8.8%, c=81.5%）

**R2:**年龄∈[44,60) -> 网上聊天=否（s=16.8%, c=70%）

当s > 5%, c > 65%时，认为规则是有趣的

| 年龄组 | 网上聊天=是 | 网上聊天=否 |
|---|---|---|
| [12, 16) | 12 | 13 |
| [16, 20) | 11 | 2 |
| [20, 24) | 11 | 3 |
| [24, 28) | 12 | 13 |
| [28, 32) | 14 | 12 |
| [32, 36) | 15 | 12 |
| [36, 40) | 16 | 14 |
| [40, 44) | 16 | 14 |
| [44, 48) | 4 | 10 |
| [48, 52) | 5 | 11 |
| [52, 56) | 5 | 10 |
| [56, 60) | 2 | 11 |

# 连续属性

- 如果区间太宽，则可能因为缺乏置信度而丢失某些模式
  - **R1:**年龄∈**[12,36) -> **网上聊天=是 **s=30%, c=57.7%**
  - **R2:**年龄∈**[36,60) -> **网上聊天=否 **s=28%, c=58.3%**
- 如果区间太窄，则可能因为缺乏支持度而丢失某些模式
  - $R_{11}$:年龄∈**[16,20) -> **网上聊天=是 **s=4.4%, c=84.6%**
  - $R_{12}$:年龄∈**[20,24) -> **网上聊天=否 **s=4.4%, c=78.6%**

# 连续属性

- 当区间宽度为8岁时，规则R2被分裂成：
  - $R_{21}$:年龄∈[44,52) -> 网上聊天=否 s=8.4%, c=70%
  - $R_{22}$:年龄∈[52,60) -> 网上聊天=否 s=8.4%, c=70%
  - 两个子规则都有足够的支持度和置信度，因此R2可以通过聚合两个子规则而恢复
- 规则R1被分裂成：
  - $R_{11}$:年龄∈[12,20) -> 网上聊天=是 s=9.2%, c=60.5%
  - $R_{12}$:年龄∈[20,28) -> 网上聊天=是 s=9.2%, c=60.0%
  - 不能通过合并来恢复R1

# 连续属性

- 考虑邻近区间的每种可能的分组
  - 从较小的区间开始，逐步合并成较宽的区间
- 计算开销非常大
  - 区间的组合数量很大
  - 如果对应于区间[a,b)的项是频繁的，则包含[a,b)的区间对应的所有项也必然是频繁的
  - 可以使用最大支持度阈值，防止创建对应于非常宽的区间的项，并减少项集的数量
- 产生许多冗余规则
  - $R_3$:{年龄∈[16,20), 性别=男} -> {网上聊天=是}
  - $R_4$:{年龄∈[16,24), 性别=男} -> {网上聊天=是}

# 连续属性

- 基于统计学的方法

  {年收入>$100K， 网上购物=是} -> 年龄：均值 = 38

- 量化关联规则可以用来推断总体的统计性质
- 1. 规则产生
  - 指定并保留目标属性，对其余分类属性和连续属性二元化
  - 利用已有算法提取频繁项集(Apriori算法或FP增长)
  - 使用统计量对目标属性在每个频繁项集内的分布进行汇总

# 连续属性

- **2. 规则确认**
  - 一个量化关联规则是有趣的，仅当由规则覆盖的事务计算的统计量不同于由未被规则覆盖的事务计算的统计量
  - 统计假设检验，μ－ μ’ ＞ △？
  - 给定两个相反的假设分别是原假设和备择假设

$$H_0 : \mu' = \mu + \Delta \qquad H_1 : \mu' > \mu + \Delta$$

$$Z = \frac{\mu' - \mu - \Delta}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}$$

# 连续属性

{年收入>$100K， 网上购物=是} -> 年龄：均值 = 38

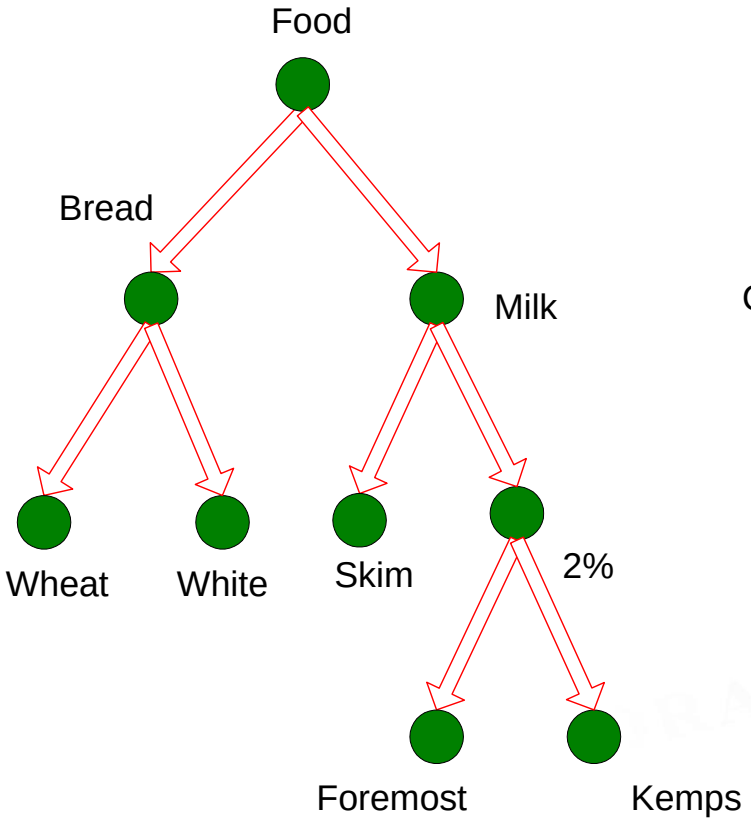
- 假设有50个用户支持前件，年龄标准差为3.5
- 不支持前件的用户200个，平均年龄30岁，标准差6.5
- 仅当 μ 与 μ’ 之间的差大于5时，认为量化关联规则是有趣的

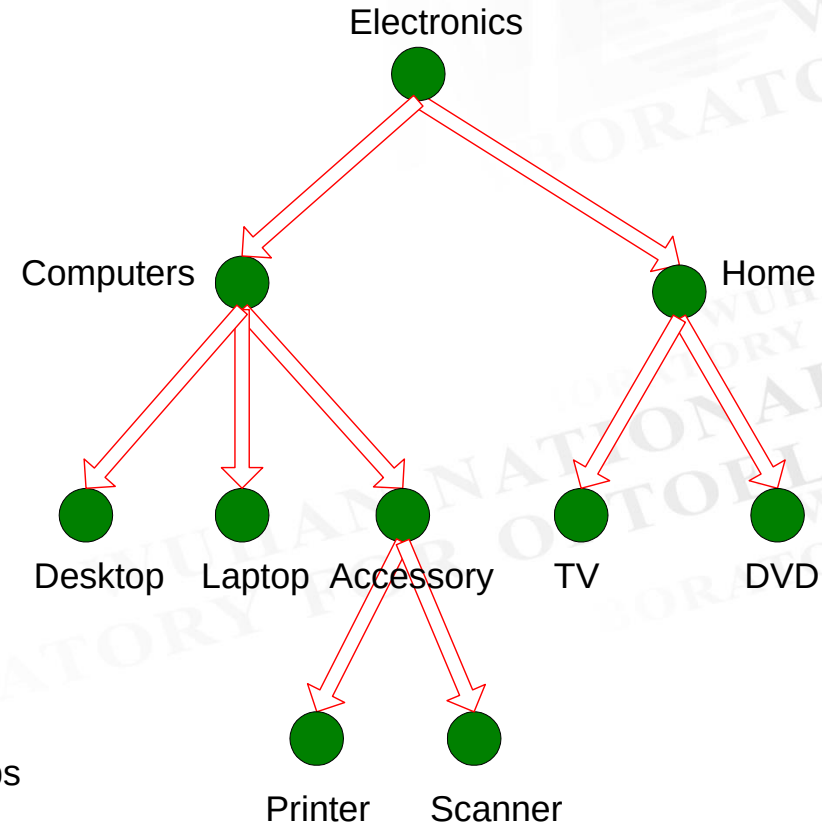
$$Z = \frac{38-30-5}{\sqrt{\frac{3.5^2}{50}+\frac{6.5^2}{200}}} = 4.4414 \qquad Z > Z_{0.95} = 1.64$$

原假设被拒绝，该量化规则是有趣的

# 处理概念分层

- 概念分层是定义在一个特定的域中的各种实体或概念的多层组织



# 处理概念分层

- 为什么要将概念分层引入关联分析中？
- 1. 位于层次结构较下层的项可能没有足够的支持度，从而不在任何频繁项集中出现
  - 可能丢失涉及较下层的项的有趣模式
- 2. 在概念分层的较低层发现的规则倾向于过于特殊，可能不如较高层的规则感兴趣
  - 脱脂牛奶-->普通面包，2%牛奶-->白面包
  - 牛奶-->面包

# 处理概念分层

- 方法1：通过在每个事务t中扩展所有项的对应祖先，得到扩展事务
  - {DVD, 普通面包}→{DVD, 普通面包, 家电, 电子产品, 面包, 食品}
- 局限：
- 1. 处于较高层的项比处于较低层的项趋向于具有较高的支持度计数
- 2. 分层的引入增加了关联分析的计算时间
- 3. 概念分层的引入可能产生冗余规则

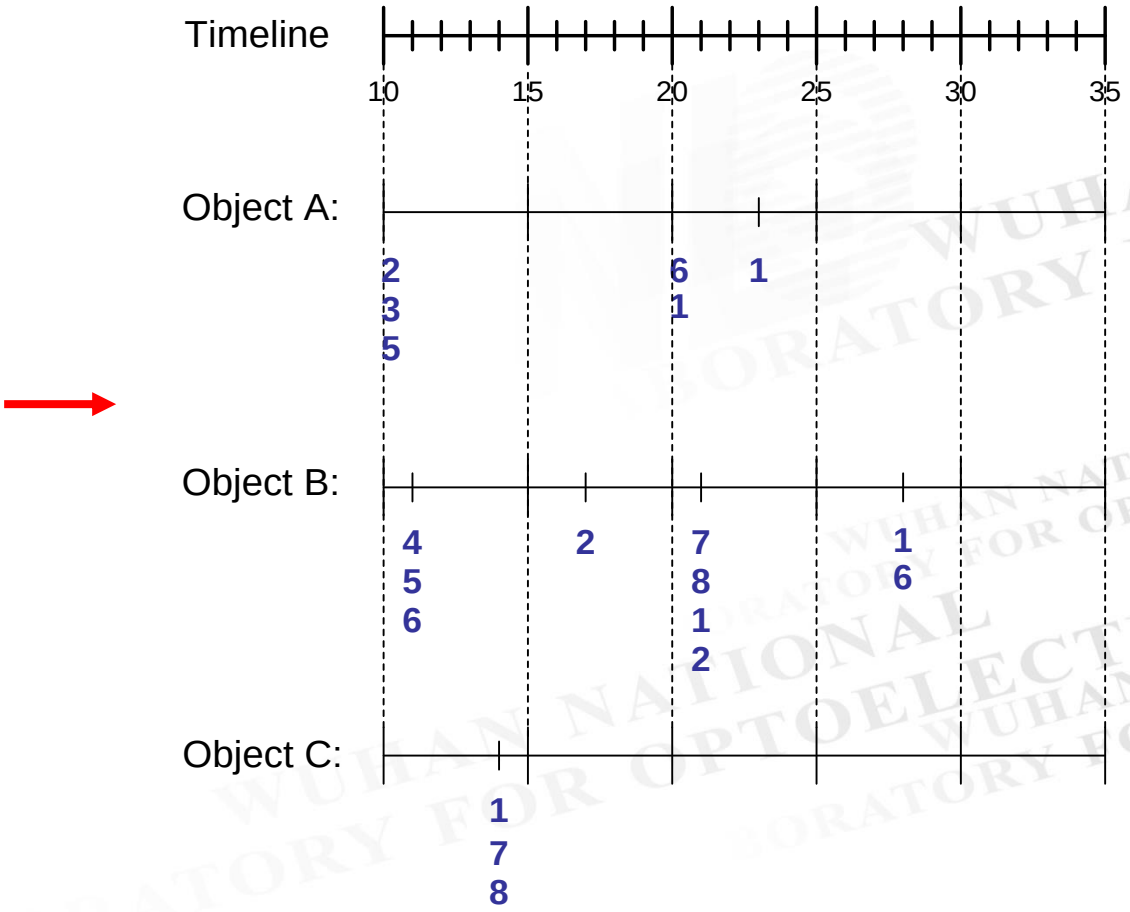
# 处理概念分层

- 方法2：首先在最高层次上产生频繁项集，然后逐层往下依次产生频繁项集

- 局限：
- 1. 复杂度太高
  - 需要多次扫描数据
- 2. 可能会丢失感兴趣的跨层关联模式

# 序列模式

| Object | Timestamp | Events |
|---|---|---|
| A | 10 | 2, 3, 5 |
| A | 20 | 6, 1 |
| A | 23 | 1 |
| B | 11 | 4, 5, 6 |
| B | 17 | 2 |
| B | 21 | 7, 8, 1, 2 |
| B | 28 | 1, 6 |
| C | 14 | 1, 8, 7 |



# 序列模式

- 序列是元素的有序列表
  - $s = < e_1\ e_2\ e_3\ \ldots >$
- 每个元素$e_i$是一个或多个事件的集族
  - $e_i = \{i_1, i_2, \ldots, i_k\}$
- 每个事件具有其特有的时间或空间属性
- 一个序列可以用它的长度和出现的事件个数来刻画
  - 序列的长度对应于出现在序列中的元素个数

# 序列模式

- web站点访问者访问的web页面序列
  - {主页} {电子产品} {照相机和摄像机} {数码相机} {购物车} {订购确认} {返回购物}
  - 7个元素和7个事件
- 计算机科学主修课程序列
  - {算法与数据结构，操作系统引论} {数据库系统，计算机体系结构} {计算机网络，软件工程} {计算机图形学，并行程序设计}
  - 4个元素和8个事件

# 序列模式

- 如果t中每个有序元素都是s中一个有序元素的子集，则序列t是序列s的子序列
  - 如果存在整数$1 \le j_1 < j_2 < \ldots < j_m < n$，使得$t_1 \subseteq s_{j1}$, $t_2 \subseteq s_{j2}$, …, $t_m \subseteq s_{jm}$, 则序列$t=<t_1 t_2 \ldots t_m>$是序列$s=<s_1 s_2 \ldots s_n>$的子序列
  - $n \ge m$

| Data sequence | Subsequence | Contain? |
|---|---|---|
| < {2,4} {3,5,6} {8} > | < {2} {3,5} > | Yes |
| < {1,2} {3,4} > | < {1} {2} > | No |
| < {2,4} {2,4} {2,5} > | < {2} {4} > | Yes |

# 序列模式

- 序列s的支持度是包含s的所有数据序列所占的比例
- 如果序列s的支持度大于或等于用户指定的阈值minsup，则称s是一个序列模式(频繁序列)
- 序列模式发现
  - 给定：序列数据集D和用户指定的最小支持度阈值minsup
  - 任务：找出支持度≥ minsup的所有子序列

# 序列模式

| Object | Timestamp | Events |
|---|---|---|
| A | 1 | 1,2,4 |
| A | 2 | 2,3 |
| A | 3 | 5 |
| B | 1 | 1,2 |
| B | 2 | 2,3,4 |
| C | 1 | 1, 2 |
| C | 2 | 2,3,4 |
| C | 3 | 2,4,5 |
| D | 1 | 2 |
| D | 2 | 3, 4 |
| D | 3 | 4, 5 |
| E | 1 | 1, 3 |
| E | 2 | 2, 4, 5 |

*Minsup* = 50%

**Examples of Frequent Subsequences:**

| | | |
|---|---|---|
| < {1,2} > | s=60% | A B C |
| < {2,3} > | s=60% | A B C |
| < {2,4}> | s=80% | A B C E |
| < {3} {5}> | s=80% | A C D E |
| < {1} {2} > | s=80% | A B C E |
| < {2} {2} > | s=60% | A B C |
| < {1} {2,3} > | s=60% | A B C |
| < {2} {2,3} > | s=60% | A B C |
| < {1,2} {2,3} > | s=60% | A B C |

# 序列模式

- 给定一个序列：**<{a b} {c d e} {f} {g h i}>**
  - 可能的子序列如
    - **<{a} {c d} {f} {g} >, < {c d e} >, < {b} {g} >**
- 出现在具有**n**个事件的数据序列中的**k-**序列总数为 $C_n^k$
- 具有**n**个事件的数据序列中包含的序列总数为

$$C_n^1 + C_n^2 + \cdots + C_n^n = 2^n - 1$$

# 序列模式

- 蛮力法
- 给定n个事件: $i_1, i_2, i_3, \ldots, i_n$
- 候选1-序列:
  - $\langle\{i_1\}\rangle, \langle\{i_2\}\rangle, \langle\{i_3\}\rangle, \ldots, \langle\{i_n\}\rangle$
- 候选2-序列:
  - $\langle\{i_1, i_2\}\rangle, \langle\{i_1, i_3\}\rangle, \ldots, \langle\{i_1\}\{i_1\}\rangle, \langle\{i_1\}\{i_2\}\rangle, \ldots, \langle\{i_{n-1}\}\{i_n\}\rangle$
- 候选3-序列:
  - $\langle\{i_1, i_2, i_3\}\rangle, \langle\{i_1, i_2, i_4\}\rangle, \ldots, \langle\{i_1, i_2\}\{i_1\}\rangle, \langle\{i_1, i_2\}\{i_2\}\rangle, \ldots,$
  - $\langle\{i_1\}\{i_1, i_2\}\rangle, \langle\{i_1\}\{i_1, i_3\}\rangle, \ldots, \langle\{i_1\}\{i_1\}\{i_1\}\rangle, \langle\{i_1\}\{i_1\}\{i_2\}\rangle, \ldots$

# 序列模式

- 候选序列的个数比候选项集的个数大的多
- 1. 一个项在项集中最多出现一次，但一个事件在序列中可以出现许多次
  - $\{i_1, i_2\}$
  - $<\{i_1, i_2\}>,<\{i_1\}\{i_2\}>,<\{i_2\}, \{i_2\}>,<\{i_1\}, \{i_1\}>$
- 2. 次序在项集中不重要，在序列中重要
  - {1,2}和{2,1}表示同一个项集
  - $<\{i_1\}\{i_2\}>$和$<\{i_2\}\{i_1\}>$对应于不同的序列

# 序列模式

- 先验原理：包含特定的k-序列的任何数据序列必然包含该k序列的所有(k-1)-子序列

序列模式发现的类 Apriori 算法

1: k=1

2: $F_k = \{ i \mid i \in I \wedge \sigma(\{i\}) \geq N \times minsup \}$ {发现所有的频繁 1-序列}

3: repeat

4: k=k+1

5: $C_k$ = apriori - gen($F_{k-1}$) {产生候选 k-序列}

6: for 每个数据序列 t∈T do

7: $C_t$ = subsequence( $C_k$, t ) {识别包含在 t 中的所有候选}

8: for 每个候选 k-序列 c∈$C_t$ do

9: $\sigma(c) = \sigma(c) + 1$ {支持度计数增值}

10: end for

11: end for

12: $F_k = \{ c \mid c \in C_k \wedge \sigma(c) \geq N \times minsup \}$ {提取频繁 k-序列}

13: until $F_k$ = 空集

14: Result = U $F_k$

# 序列模式

- 候选产生
  - 仅当从$s^{(1)}$中去掉第一个事件得到的子序列与从$s^{(2)}$中去掉最后一个事件得到的子序列相同时，序列$s^{(1)}$与另一个序列$s^{(2)}$合并
  - 候选结果是序列$s^{(1)}$与$s^{(2)}$的最后一个事件的连接

# 序列模式

- 合并$w_1$=<{1} {2 3} {4}>和$w_2$ =<{2 3} {4 5}>可以产生候选序列< {1} {2 3} {4 5}>
- 合并$w_1$=<{1} {2 3} {4}>和$w_2$ =<{2 3} {4} {5}>可以产生候选序列< {1} {2 3} {4} {5}>
- 不能合并序列$w_1$ =<{1} {2 6} {4}>和$w_2$ =<{1} {2} {4 5}>

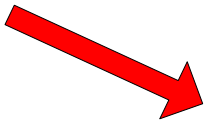
# 序列模式

- $s^{(2)}$的最后一个事件可以作为最后一个事件合并到$s^{(1)}$的最后一个元素中，也可以作为一个不同的元素，取决于：
  - (1)如果$s^{(2)}$的最后两个事件属于相同的元素，则$s^{(2)}$的最后一个事件在合并后的序列中是$s^{(1)}$的最后一个元素的一部分
  - (2)如果$s^{(2)}$的最后两个事件属于不同的元素，则$s^{(2)}$的最后一个事件在合并后的序列中成为连接到$s^{(1)}$的尾部的单独元素

# 序列模式

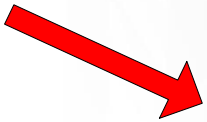
Frequent 3-sequences

< {1} {2} {3} >
< {1} {2 5} >
< {1} {5} {3} >
< {2} {3} {4} >
< {2 5} {3} >
< {3} {4} {5} >
< {5} {3 4} >



Candidate Generation

< {1} {2} {3} {4} >
< {1} {2 5} {3} >
< {1} {5} {3 4} >
< {2} {3} {4} {5} >
< {2 5} {3 4} >



Candidate Pruning

< {1} {2 5} {3} >

- 候选剪枝
  - 一个候选**k-**序列被剪掉，如果它的**(k-1)-**序列至少有一个是非频繁的

# 序列模式

- 时限约束
  - 学生A：<{统计学} {数据库系统} {数据挖掘}>
  - 学生B：< {数据库系统} {统计学} {数据挖掘}>
- 序列模式的每个元素都与一个时间窗口[l, u]相关联
  - l是该时间窗口内事件的最早发生时间
  - u是该时间窗口内事件的最晚发生时间

# 序列模式

- 最大跨度约束：
  - 整个序列中所允许的事件的最晚和最早发生时间的最大时间差
  - **maxspan越长，检测到模式的可能性就越大**
  - 可能捕获不真实的模式
  - 在支持度计数时，必须忽略给定模式中事件的第一次和最后一次发生的时间间隔大于**maxspan**的情况
- 假定**maxspan=3**

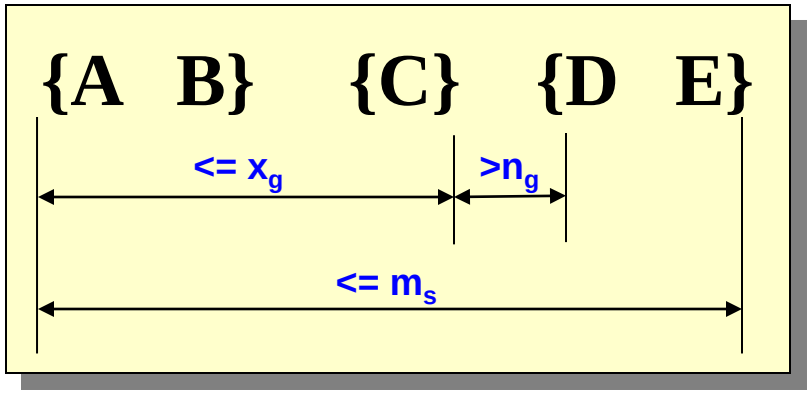
| 数据序列 s | 序列模式 t | 包含？ |
|---|---|---|
| <{1,3}{3,4}{4}{5}{6,7}{8}> | <{3} {4}> | 是 |
| <{1,3}{3,4}{4}{5}{6,7}{8}> | <{3} {6}> | 是 |
| <{1,3}{3,4}{4}{5}{6,7}{8}> | <{1, 3} {6}> | 否 |

# 序列模式

- 最小间隔和最大间隔约束
  - 限制两个相邻元素之间的时间差
- 假定maxgap=3，mingap=1

| 数据序列 s | 序列模式 t | maxgap | mingap |
|---|---|---|---|
| <{1,3}{3,4}{4}{5}{6,7}{8}> | <{3} {6}> | 通过 | 通过 |
| <{1,3}{3,4}{4}{5}{6,7}{8}> | <{6} {8}> | 通过 | 未通过 |
| <{1,3}{3,4}{4}{5}{6,7}{8}> | <{1, 3} {6}> | 未通过 | 通过 |
| <{1,3}{3,4}{4}{5}{6,7}{8}> | <{1} {3} {8}> | 未通过 | 未通过 |

# 序列模式




$x_g$ = 2, $n_g$ = 0, $m_s$= 4

$x_g$: 最大间隔

$n_g$: 最小间隔

$m_s$: 最大跨度约束

| 数据序列 | 序列模式 | 包含? |
|---|---|---|
| < {2,4} {3,5,6} {4,7} {4,5} {8} > | < {6} {5} > | Yes |
| < {1} {2} {3} {4} {5}> | < {1} {4} > | No |
| < {1} {2,3} {3,4} {4,5}> | < {2} {3} {5} > | Yes |
| < {1,2} {3} {2,3} {3,4} {2,4} {4,5}> | < {1,2} {5} > | No |

# 序列模式

- 无约束时
  - **<{2}{5}>和<{2}{3}{5}>的支持度都是60%**
  - **A C D**
- **maxgap为1时**
  - **<{2}{5}>支持度是40%**
  - **A C**
  - **<{2}{3}{5}>支持度是60%**
- 违反先验原理
  - 当序列中的事件个数增加时，支持度增加了

| Object | Timestamp | Events |
|---|---|---|
| A | 1 | 1,2,4 |
| A | 2 | 2,3 |
| A | 3 | 5 |
| B | 1 | 1,2 |
| B | 2 | 2,3,4 |
| C | 1 | 1, 2 |
| C | 2 | 2,3,4 |
| C | 3 | 2,4,5 |
| D | 1 | 2 |
| D | 2 | 3, 4 |
| D | 3 | 4, 5 |
| E | 1 | 1, 3 |
| E | 2 | 2, 4, 5 |

# 序列模式

- 方法1：
  - 在无时限约束条件下挖掘时序模式
  - 对发现的模式后处理
- 方法2：
  - 修改挖掘算法，自动剪枝违背时限约束的候选

# 序列模式

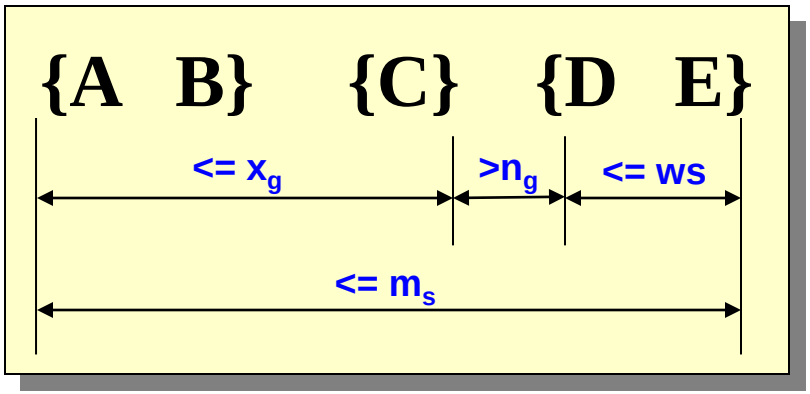
- 邻接子序列
- 序列s是序列$w=<e_1e_2\ldots e_k>$的邻接子序列，如果下列条件之一成立：
- s是从$e_1$或$e_k$中删除一个事件后由w得到；
- s是从至少包含两个事件的任意$e_i \in w$中删除一个事件后由w得到；
- s是t的邻接子序列，而t是w的邻接子序列

# 序列模式

| 数据序列 s | 序列模式 t | t 是 s 的邻接子序列？ |
|---|---|---|
| <{1}{2,3}> | <{1} {2}> | 是 |
| <{1,2}{2}{3}> | <{1} {2}> | 是 |
| <{3,4}{1,2}{2,3}{4}> | <{1} {2}> | 是 |
| <{1}{3}{2}> | <{1} {2}> | 否 |
| <{1,2}{1}{3}{2}> | <{1} {2}> | 否 |

- 修订的先验原理
- 如果一个k序列是频繁的，则它的所有邻接(k-1)子序列也一定是频繁的

# 序列模式




$x_g$: 最大间隔

$n_g$: 最小间隔

ws: 窗口大小

$m_s$: 最大跨度约束

- ws用来指定序列模式的任意元素中事件最晚和最早出现之间的最大允许时间差
  - 窗口大小为0表示模式同一元素中的所有事件必须同时出现

# 序列模式

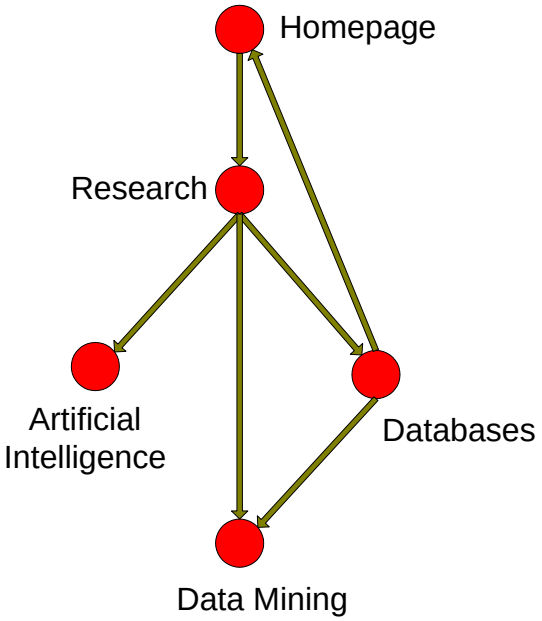
- ws=2, mingap=0, maxgap=3

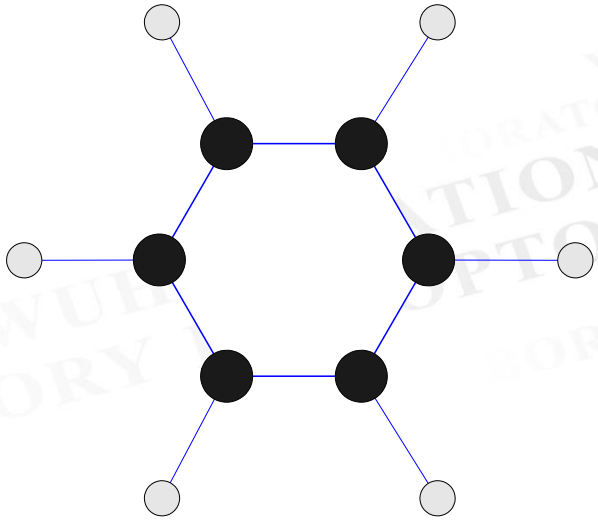
| 数据序列 s | 序列模式 t | s 支持 t? |
|---|---|---|
| <{1,3}{3,4}{4}{5}{6,7}{8}> | <{3, 4} {5}> | 是 |
| <{1,3}{3,4}{4}{5}{6,7}{8}> | <{4, 6} {8}> | 是 |
| <{1,3}{3,4}{4}{5}{6,7}{8}> | <{3, 4, 6} {8}> | 否 |
| <{1,3}{3,4}{4}{5}{6,7}{8}> | <{1, 3, 4} {6, 7, 8}> | 否 |

- 窗口大小约束会影响序列模式发现算法的支持度计数

# 子图模式

- 利用图形表示来建模一些比项集和序列更加复杂的实体
- 频繁子图挖掘
  - 在图的集合中发现一组公共子结构

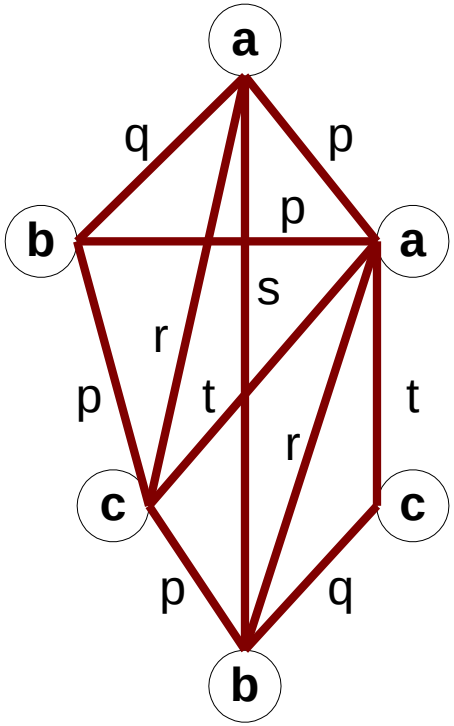


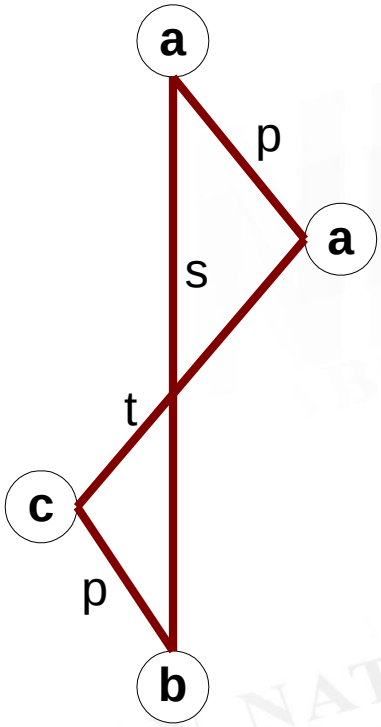
# 子图模式

- 图是一种可以用来表示实体集之间联系的数据结构
  - 顶点集V，连接顶点对的边集E
  - 每条边用顶点对表示
- 子图
  - 图G‘=(V’, E’)是另一个图G=(V, E)的子图，如果它的顶点集V’是V的子集，并且它的边集E‘是E的子集
  - $G' \subseteq_s G$

# 子图模式




(a) Labeled Graph

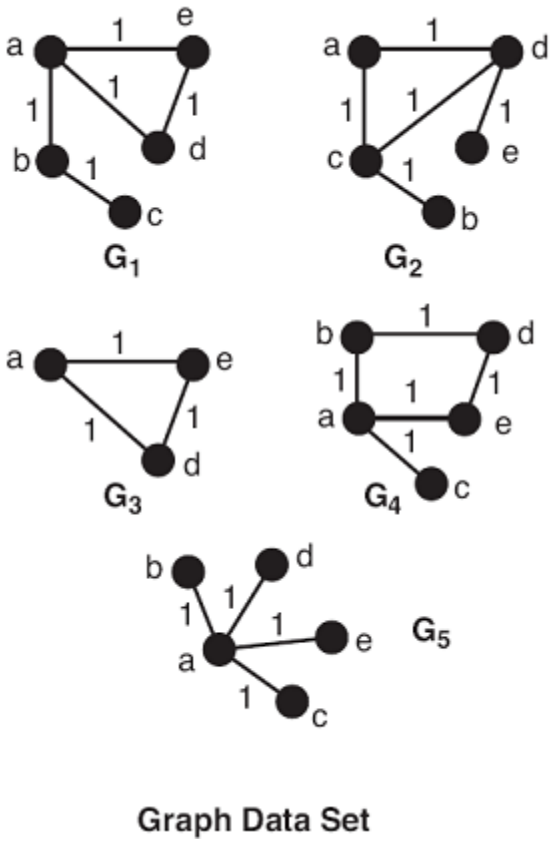



(b) Subgraph

# 子图模式

- 支持度：给定一个图的集族ς，子图g的支持度定义为包含它的所有图所占的百分比，即

$$s(g)=\frac{\left|\{G_i \mid g \subseteq_s G_i, G_i \in \varsigma\}\right|}{|\varsigma|}$$



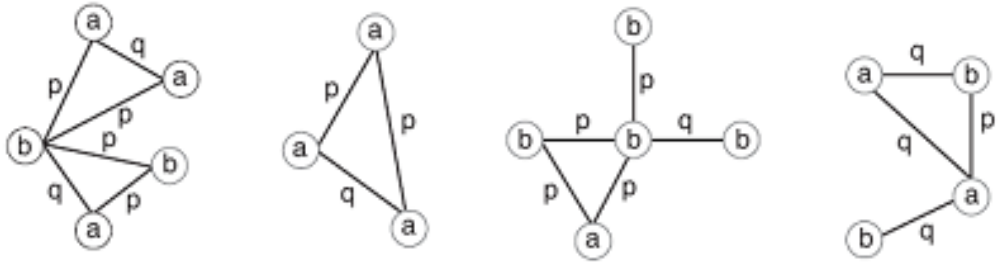
# 子图模式

- 频繁子图挖掘
  - 给定图的集合 $\zeta$ 和支持度阈值minsup
  - 目标是找出所有使得**s(g)≥minsup**的子图**g**
- 无向连通图
  - 一个图是连通的，如果图中每对顶点之间都存在一条路径
  - 一个图是无向的，如果它只包含无向边
    - $(v_i, v_j)$和$(v_j, v_i)$无区别

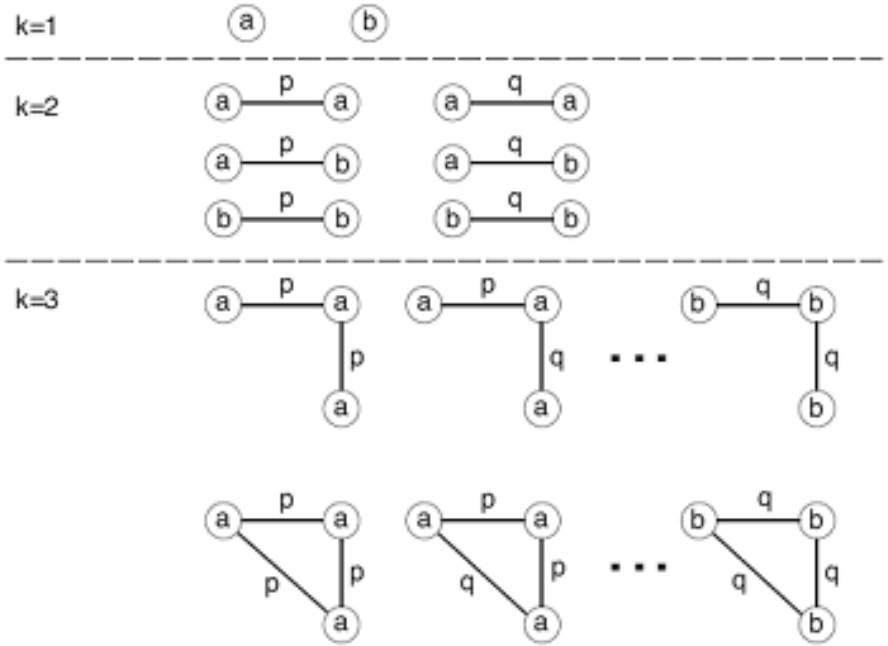
# 子图模式

- 挖掘频繁子图是一项计算量很大的任务
  - 有很多非连通子图
  - 一个项在项集中至多出现一次，而顶点标号可能在一个图中出现多次
  - 相同的顶点标号对可以有多种边标号选择
- 蛮力方法：产生所有的连通子图作为候选，并计算他们各自的支持度

# 子图模式



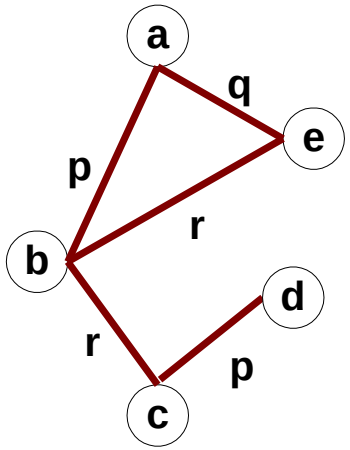

(a) Example of a graph data set.



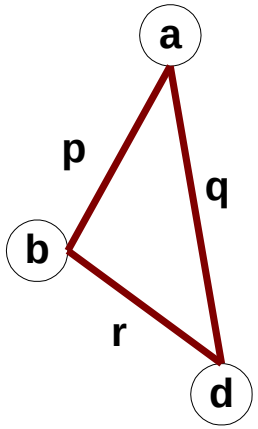
# 子图模式

- 开发一种类Apriori算法来找出频繁子图
- 数据变换
  - 将图变换成类似于事务的形式，使得可以使用已有的算法
  - 边标号与对应的顶点标号组合被映射到一个“项”
  - “事务”的宽度由图的边数决定
  - 仅当图中每一条边都具有唯一的顶点和边标号组合时，该方法才可行

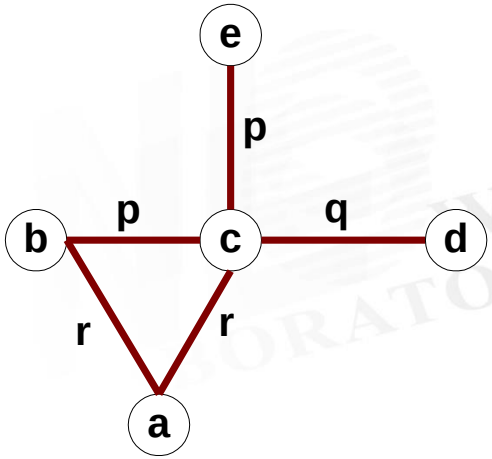
# 子图模式




G1




G2




G3

| | (a,b,p) | (a,b,q) | (a,b,r) | (b,c,p) | (b,c,q) | (b,c,r) | … | (d,e,r) |
|---|---|---|---|---|---|---|---|---|
| G1 | 1 | 0 | 0 | 0 | 0 | 1 | … | 0 |
| G2 | 1 | 0 | 0 | 0 | 0 | 0 | … | 0 |
| G3 | 0 | 0 | 1 | 1 | 0 | 0 | … | 0 |
| G3 | … | … | … | … | … | … | … | … |

# 子图模式

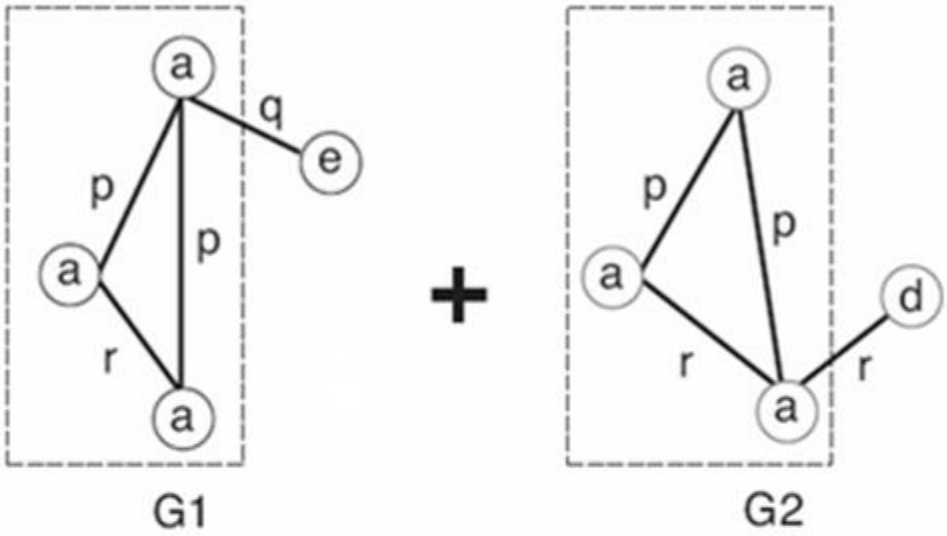
- 频繁子图挖掘算法一般结构
- 1. 候选产生：合并包含频繁(k-1)-子图对，得到候选k-子图
- 2. 候选剪枝：丢弃包含非频繁的(k-1)-子图的所有候选k-子图
- 3. 支持度计数：统计 $\zeta$ 中包含每个候选的图的个数
- 4. 候选删除：丢弃支持度小于阈值的子图

# 子图模式

- 如何定义子图的大小k？
- 顶点增长(vertex growing)
  - k是图中的顶点个数
- 边增长(edge growing)
  - k是图中的边的个数
- 为了避免产生重复的候选，两个(k-1)-子图必须共享一个共同的(k-2)-子图
  - 共同的子图称为核

# 子图模式

- 顶点增长



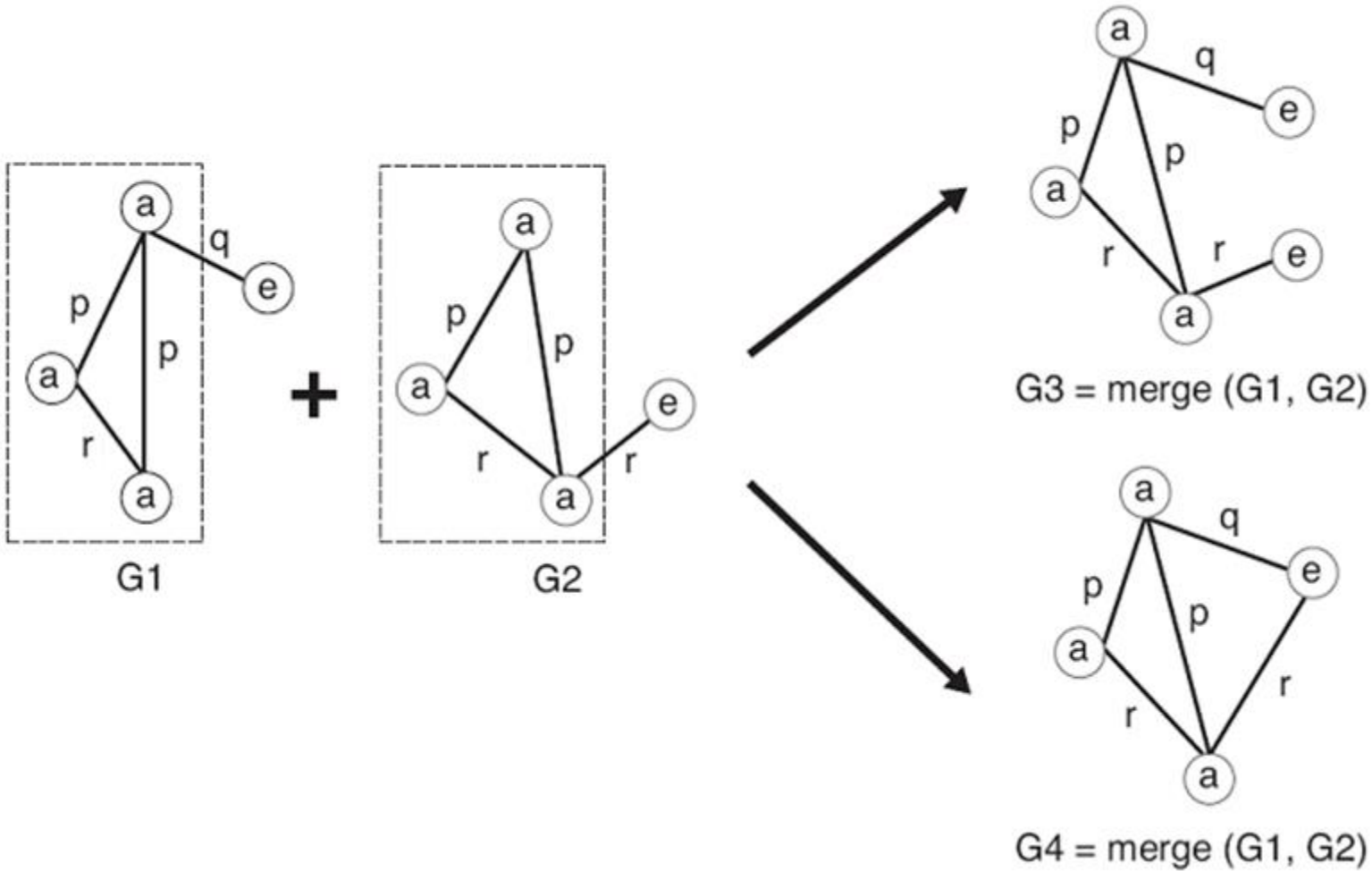

$$M_{G1}=\begin{pmatrix} 0 & p & p & q \\ p & 0 & r & 0 \\ p & r & 0 & 0 \\ q & 0 & 0 & 0 \end{pmatrix} \quad M_{G2}=\begin{pmatrix} 0 & p & p & 0 \\ p & 0 & r & 0 \\ p & r & 0 & r \\ 0 & 0 & r & 0 \end{pmatrix} \quad M_{G3}=\begin{pmatrix} 0 & p & p & q & 0 \\ p & 0 & r & 0 & 0 \\ p & r & 0 & 0 & r \\ q & 0 & 0 & 0 & ? \\ 0 & 0 & r & ? & 0 \end{pmatrix}$$

# 子图模式

- 邻接矩阵$M^{(1)}$与另一个邻接矩阵$M^{(2)}$合并，如果删除$M^{(1)}$和$M^{(2)}$的最后一行和最后一列得到的子矩阵相同
- 结果矩阵是$M^{(1)}$，添加上$M^{(2)}$的最后一行和最后一列
- 新矩阵的其余项或者用0，或者用连接顶点对的合法的边标号替换

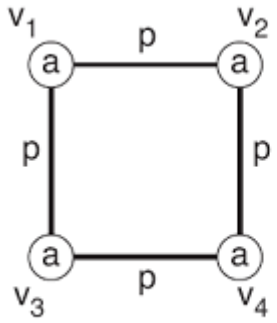
# 子图模式

- 边增长

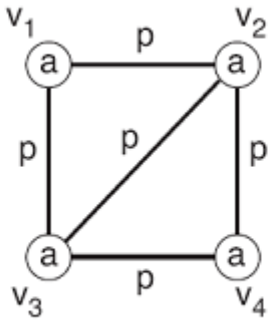


# 子图模式

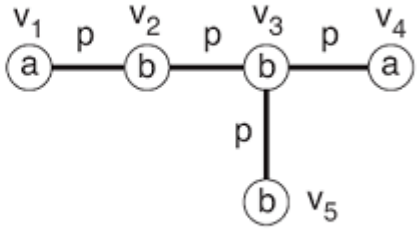
- 一个频繁子图$g^{(1)}$与另一个频繁子图$g^{(2)}$合并，仅当$g^{(1)}$删除一条边得到的子图与从$g^{(2)}$删除一条边得到的子图拓扑等价。
- 合并后，结果子图是$g^{(1)}$，添加$g^{(2)}$的那条额外的边
- 顶点拓扑等价



G1



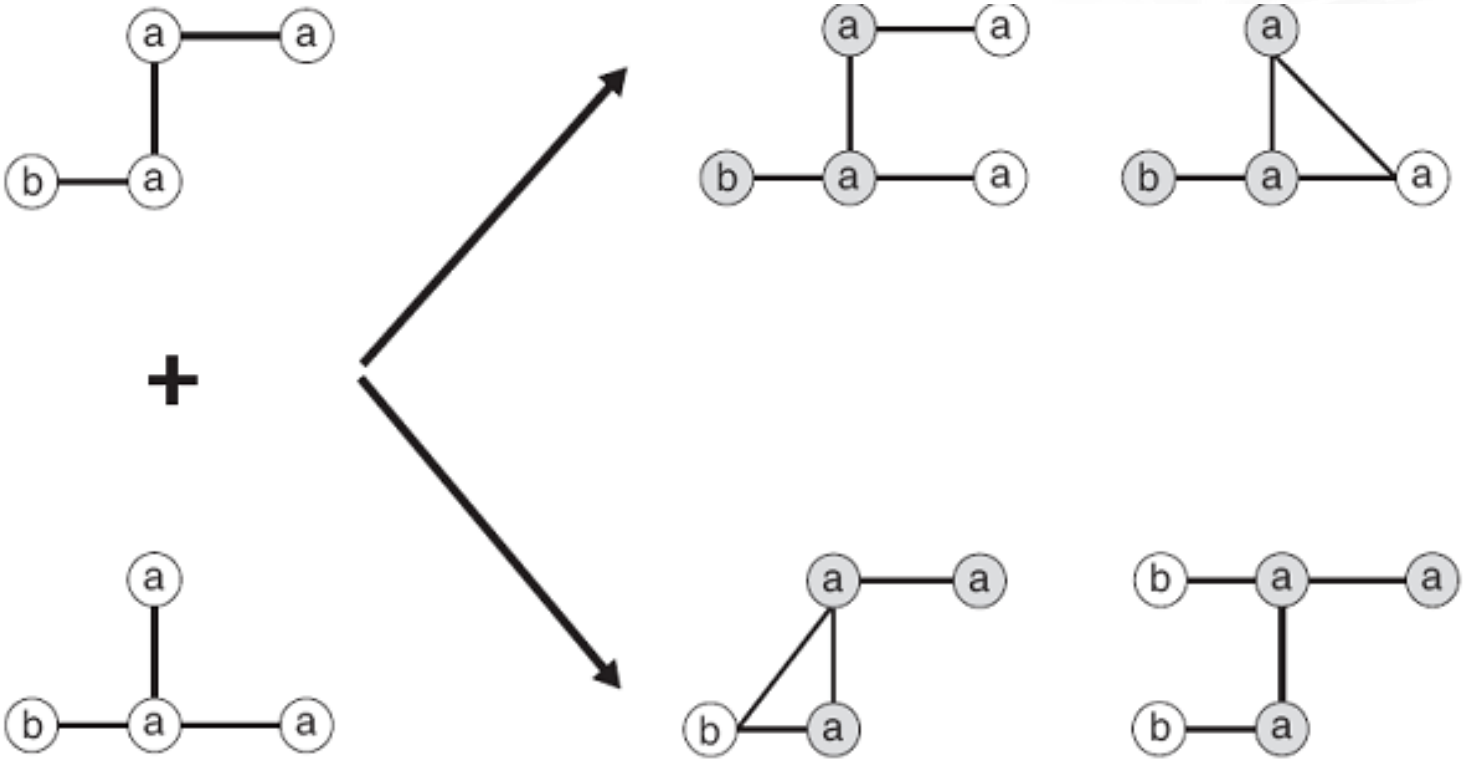
G2



G3

# 子图模式

- 当与一对(k-1)子图相关联的核有多个时，还可能产生多个候选子图

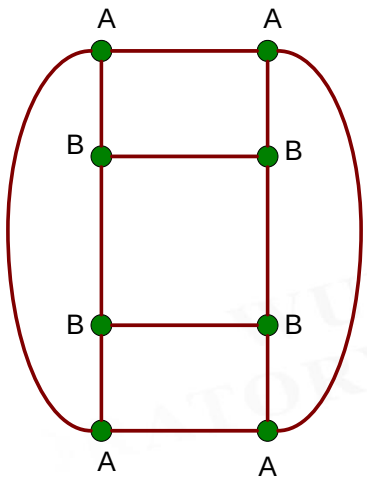


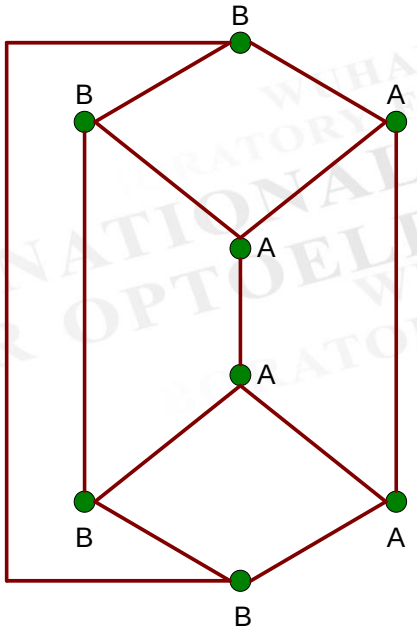
# 子图模式

- 候选剪枝
  - 产生候选k-子图后，需要剪去(k-1)-子图非频繁的候选
- 相继从k-子图删除一条边，并检查对应的(k-1)-子图是否连通且频繁
  - 如果不是，该候选k-子图丢弃

# 子图模式

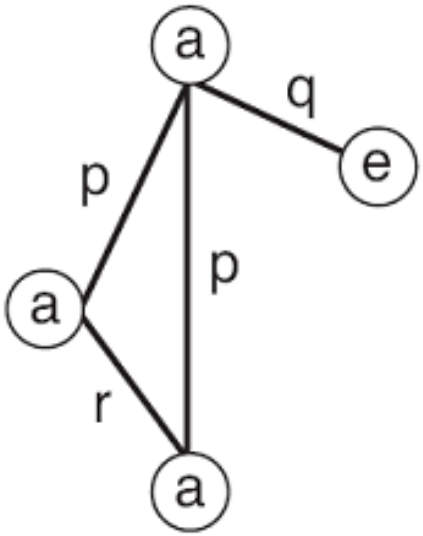
- 为了检查k-1子图是否频繁，需要将它与其他频繁k-1子图匹配
- 判断两个图是否拓扑等价称为图同构问题
- 如果两个图的顶点之间存在一个1-1映射，则他们是同构的



# 子图模式

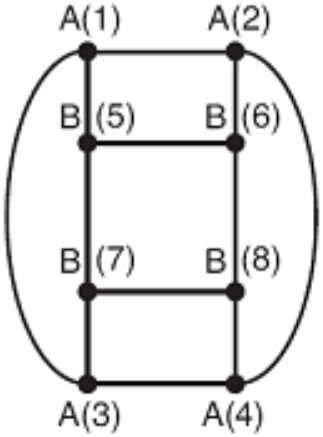
- 处理同构问题
  - 将每一个图都映射到一个唯一的串表达式，称作代码或规范标号
  - 如果两个图是同构的，则它们的代码一定相同
- 第一步：邻接矩阵表示




$$M=\begin{pmatrix} 0 & p & p & q \\ p & 0 & r & 0 \\ p & r & 0 & 0 \\ q & 0 & 0 & 0 \end{pmatrix}$$

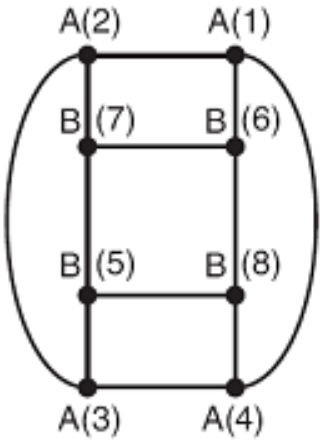
# 子图模式

- 第二步：确定每个邻接矩阵的串表示




| | A(1) | A(2) | A(3) | A(4) | B(5) | B(6) | B(7) | B(8) |
|---|---|---|---|---|---|---|---|---|
| A(1) | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| A(2) | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| A(3) | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| A(4) | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| B(5) | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| B(6) | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| B(7) | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 |
| B(8) | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |

Code = 11001110000100100101000010 11




| | A(1) | A(2) | A(3) | A(4) | B(5) | B(6) | B(7) | B(8) |
|---|---|---|---|---|---|---|---|---|
| A(1) | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| A(2) | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| A(3) | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| A(4) | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| B(5) | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 |
| B(6) | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| B(7) | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| B(8) | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |

Code = 10110100101000001001100011 10

# 非频繁模式

- 非频繁模式
  - 非频繁模式是一个项集或规则，其支持度小于阈值**minsup**
- 有些非频繁模式对于分析是有用的
  - 负相关
    - **DVD vs. VCR**
  - 罕见事件或不合理的例外情况
    - **{火灾=Yes, 警报=on}**

# 非频繁模式

- 为了检测非频繁模式，必须确定模式的期望支持度
  - 如果一个模式的支持度明显低于期望支持度，则认为是一个有趣的非频繁模式
- 挑战：
  - 如何识别有趣的非频繁模式
  - 如何在大型数据集中有效的发现非频繁模式

# 非频繁模式

- 负模式
- 负项 $\bar{i}_k$ 表示项$i_k$不在给定的事务中出现
- 负项集X是具有如下性质的项集：
  - $X=A\cup\overline{B}$，其中A是正项的集合，而 $\overline{B}$是负项的集合，$|\overline{B}|\geq 1$；
  - $s(X)\geq minsup$

# 非频繁模式

- 负关联规则
  - 规则是从一个负项集提取的
  - 规则的支持度大于或等于**minsup**
  - 规则的置信度大于或等于**minconf**
- 负项集和负关联规则统称为负模式
  - 茶→/咖啡

# 非频繁模式

- 负相关项集：项集X是负相关的，如果

$$s(X) < \prod_{j=1}^{k} s(x_j) = s(x_1) \times s(x_2) \times \cdots s(x_k)$$

  - 一个项集是负相关的，如果它的支持度小于使用统计独立性假设计算出的期望支持度
- 负相关关联规则：X→Y是负相关的，如果

$$s(X \cup Y) < s(X)s(Y)$$

  - 其中X和Y是不相交的项集

# 非频繁模式

- 部分条件 $s(X \cup Y) < s(X)s(Y)$
- 完全条件 $s(X \cup Y) < \prod_i s(x_i) \prod_j s(y_j)$
- 由于X中（Y中）的项通常是正相关的，因此使用部分条件比完全条件来定义负相关规则更实际
  - {眼镜，镜头清洁剂}→{隐形眼镜，盐溶液}

$$s(X) > \prod_{j=1}^{k} s(x_j) = s(x_1) \times s(x_2) \times \cdots s(x_k)$$

# 非频繁模式

- 负相关条件也可以用正项集和负项集的支持度表示

$$
\begin{aligned}
& s(X \cup Y) - s(X)s(Y) \\
&= s(X \cup Y) - [s(X \cup Y) + s(X \cup \bar{Y})][s(X \cup Y) + s(\bar{X} \cup Y)] \\
&= s(X \cup Y)[1 - s(X \cup Y) - s(X \cup \bar{Y}) - s(\bar{X} \cup Y)] - s(X \cup \bar{Y})s(\bar{X} \cup Y) \\
&= s(X \cup Y)s(\bar{X} \cup \bar{Y}) - s(X \cup \bar{Y})s(\bar{X} \cup Y)
\end{aligned}
$$

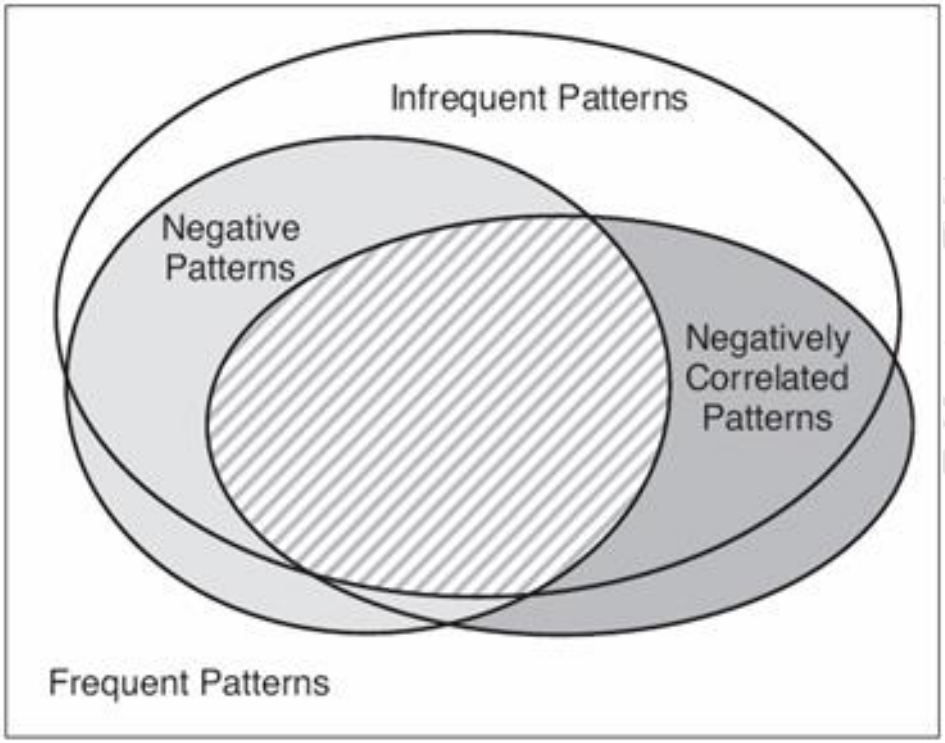
- 负相关条件可表述为

$$s(X \cup Y)s(\bar{X} \cup \bar{Y}) < s(X \cup \bar{Y})s(\bar{X} \cup Y)$$

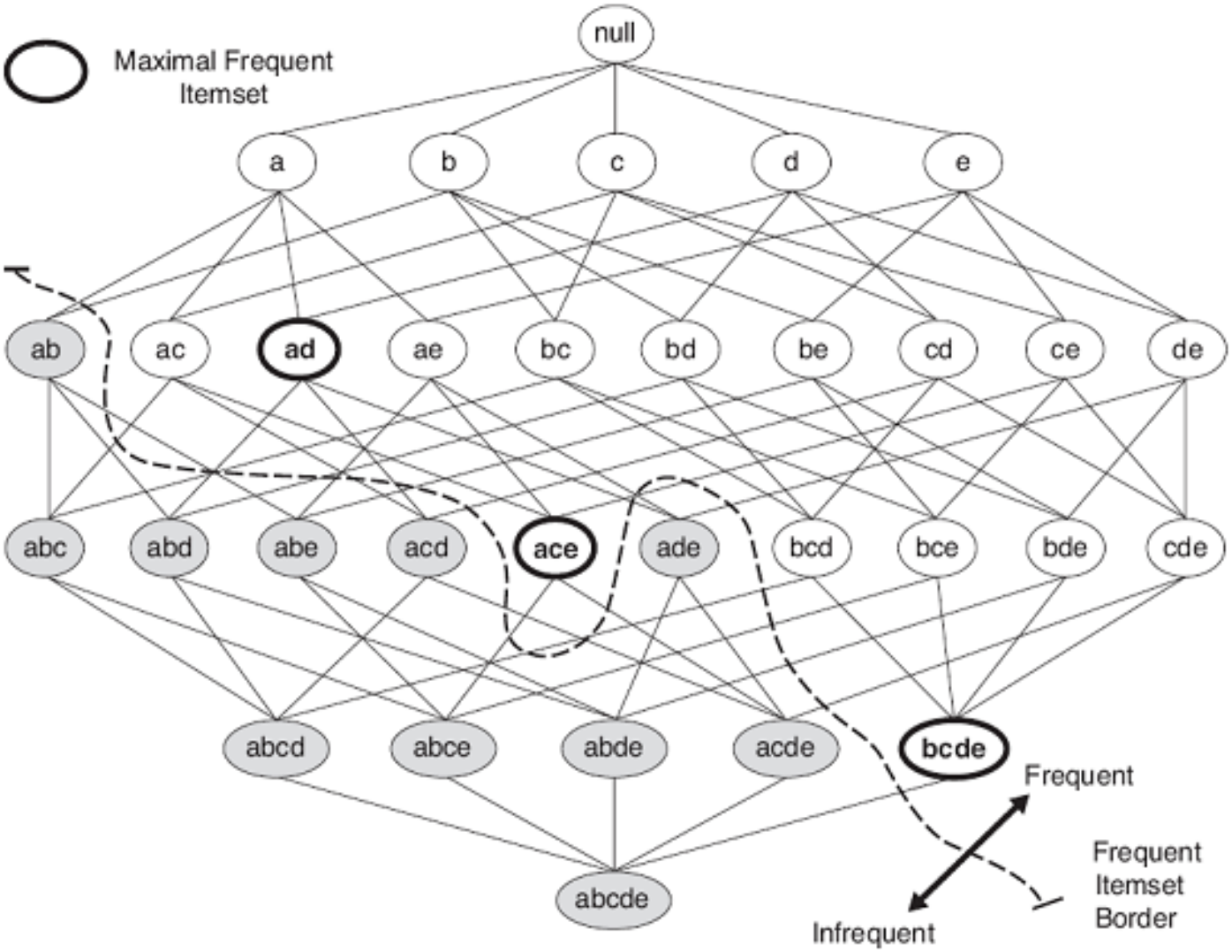
- 负相关项集和负相关关联规则统称为负相关模式

# 非频繁模式

- 许多非频繁模式有对应的负模式
- 许多负相关模式也具有对应的负模式
- 非频繁的负相关模式比频繁的负相关模式更令人感兴趣
  - **XUY**的支持度越低，该模式就越负相关



# 非频繁模式


Maximal Frequent Itemset
null
a
b
c
d
e
ab
ac
ad
ae
bc
bd
be
cd
ce
de
abc
abd
abe
acd
ace
ade
bcd
bce
bde
cde
abcd
abce
abde
acde
bcde
abcde
Frequent
Frequent Itemset Border
Infrequent

# 非频繁模式

- 非频繁模式的数量可能是指数级的
- 通过删除那些不满足负相关条件的非频繁项集得到感兴趣的非频繁模式
- 挖掘负相关项集使用的基于相关性的度量不具有反单调性
- 基于挖掘负模式的方法
- 基于支持度期望的方法

# 非频繁模式

- 基于挖掘负模式的技术
- 将每个项看作对称的二元变量，通过负项增广，将事务数据二元化
- 使用已有的频繁项集产生算法

| TID | Items |
|---|---|
| 1 | {A,B} |
| 2 | {A,B,C} |
| 3 | {C} |
| 4 | {B,C} |
| 5 | {B,D} |

Original Transactions

| TID | A | $\overline{A}$ | B | $\overline{B}$ | C | $\overline{C}$ | D | $\overline{D}$ |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| 2 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 |
| 3 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| 4 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 5 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |

Transactions with Negative Items

# 非频繁模式

- 当每个项都用对应的负项增广时，项的个数会加倍
- 增加负项后，基于支持度的剪枝不再有效
  - x或/x的支持度大于等于50%
- 增加负项后，每个事务的宽度增加
  - $w_{max} \rightarrow d$
- 可以限制被视为对称二元变量的变量数
  - 仅当y频繁时才认为负项/y是有趣的
- 可以限制负模式的类型
  - 仅考虑至少包含一个正项的负模式

# 非频繁模式

- 另一种方法是不用负项增广数据集，而是根据对应的正项集计算负项集的支持度
  - s({p,/q,/r}) = s({p}) – s({p,q}) – s({p,r}) + s({p,q,r})

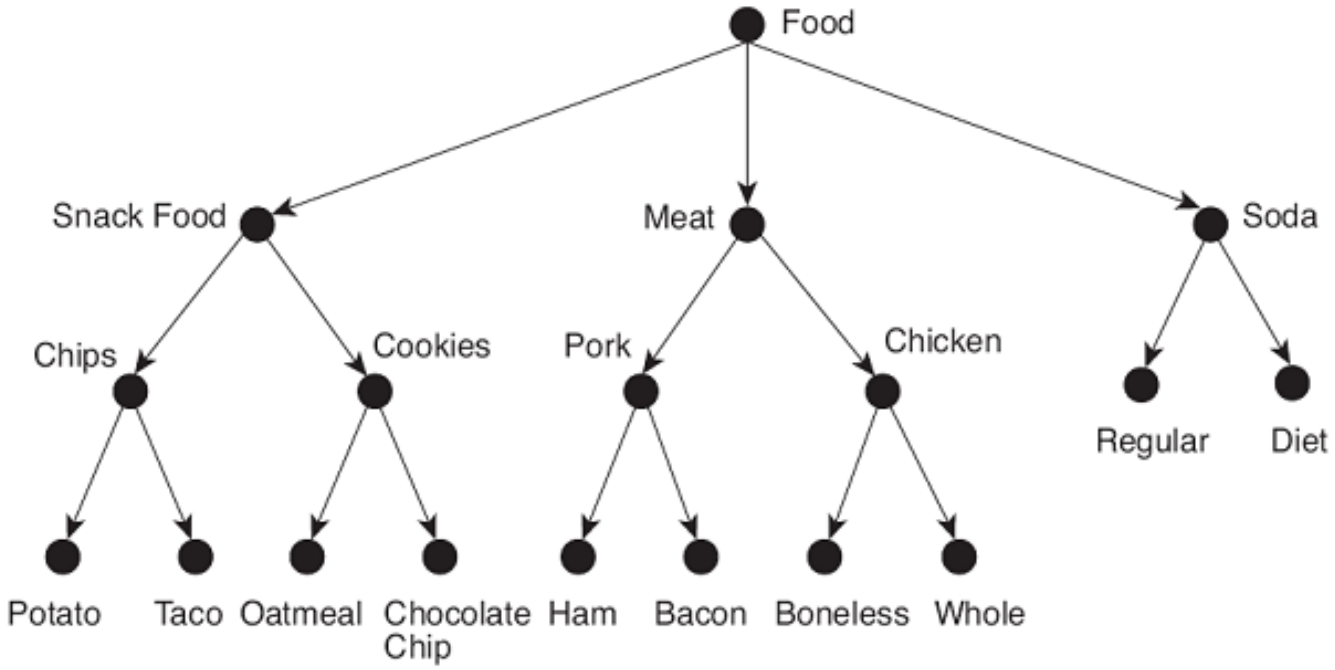
$$s(X \cup \bar{Y}) = s(X) + \sum_{i=1}^{n} \sum_{Z \subset Y, |Z|=i} \{(-1)^i \times s(X \cup Z)\}$$

# 非频繁模式

- 基于支持度期望的技术
  - 仅当非频繁模式的支持度显著小于期望支持度时，才认为它是有趣的
- 期望支持度根据统计独立性假设计算
- 概念分层法
- 间接关联法

# 非频繁模式

- 基于概念分层的支持度期望
  - 预期来自同一族产品的项与其他项具有类似的相互作用

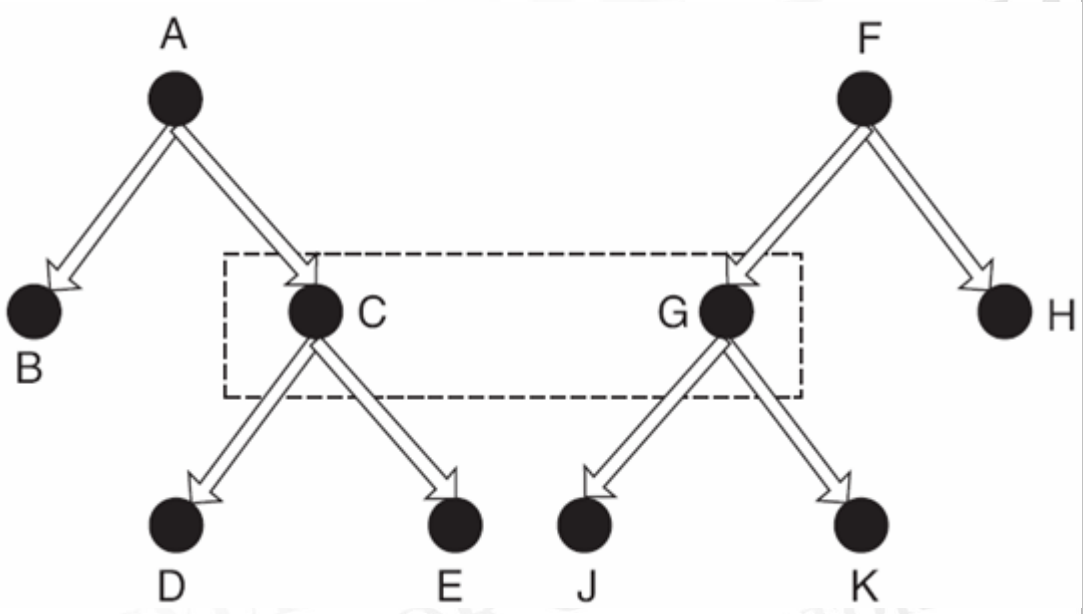


# 非频繁模式

$$\varepsilon(s(E,J)) = s(C,G) \times \frac{s(E)}{s(C)} \times \frac{s(J)}{s(G)}$$

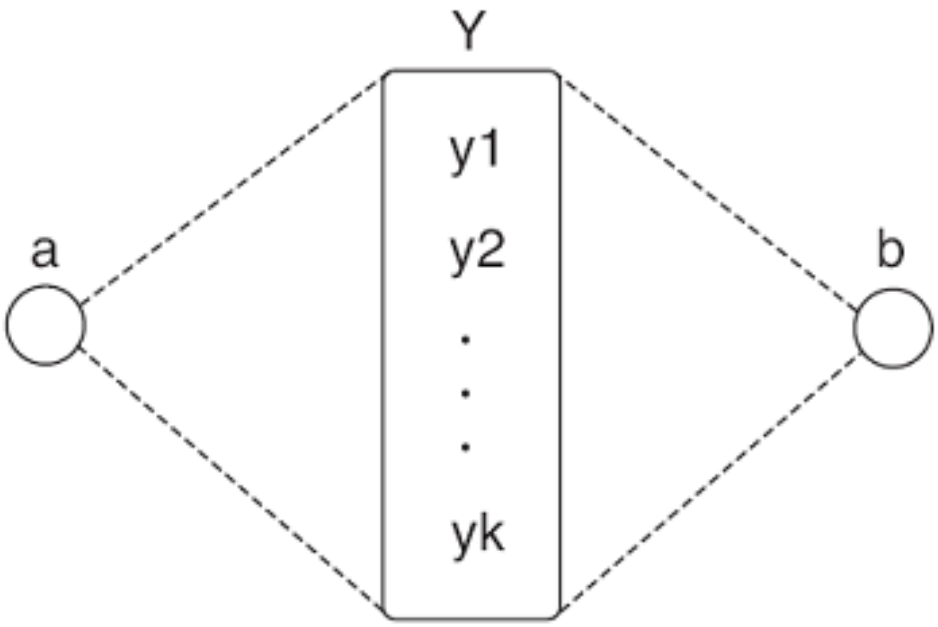
$$\varepsilon(s(C,J)) = s(C,G) \times \frac{s(J)}{s(G)}$$

$$\varepsilon(s(C,H)) = s(C,G) \times \frac{s(H)}{s(G)}$$



# 非频繁模式

- 基于间接关联的支持度期望
  - 通过考虑其他的相关项来确定期望支持度
  - 考察与两个商品一起购买的其他商品
- Y称作中介集



# 非频繁模式

- 间接关联：一对项a，b是通过中介集Y间接关联的，如果下列条件成立：
- 1. $s(\{a, b\}) < t_s$（项对支持度条件）
- 2.

$\exists Y \neq$ 空集，使得：

（a）$s(\{a\} \cup Y) \geq t_f$并且$s(\{b\} \cup Y) \geq t_f$（中介支持度条件）

（b）$d(\{a\}, Y) \geq t_d$，$d(\{b\}, Y) \geq t_d$，其中$d(X, Z)$是$X$和Z之间关联的客观度量（中介依赖条件）

# 非频繁模式

挖掘间接关联的算法

1: 产生频繁项集的集合 $F_k$

2: for k=2 to Kmax do

3: $C_k = \{(a,b,Y) \mid \{a\} \cup Y \in F_k, \{b\} \cup Y \in F_k, a \neq b\}$

4: for 每个候选(a, b, Y)$\in C_k$ do

5: if $s(\{a,b\} < t_s \wedge d(\{a\},Y) \geq t_d \wedge d(\{b\},Y) \geq t_d)$ then

6: $I_k = I_k \cup \{(a,b,Y)\}$

7: end if

8: end for

9: end for

10: Result = U $I_k$

# “今日头条”应用案例

- 基于用户的喜好，推送其最有可能感兴趣的内容
- 对个人行为的系统分析
  - 基本情况、社交行为、阅读喜好、地理位置等
- 自然语言处理和图像自动识别
- 将用户特征、环境特征、与内容特征相匹配，智能化推荐
  - **3**秒完成内容提取和分类，**0.1**秒计算推荐结果
  - **5**秒计算新用户兴趣分配，**10**秒更新用户模型

# 智能推荐引擎


人的特征
兴趣 单反 科技 音乐 网球...
职业 学生 主妇 白领 教授...
年龄 17 24 35 56...
性别 男 女
机型 iphone 三星 小米
用户行为
团队扩张面临的问题
全新宝来发布
李娜的成功是个标杆
北京市民可定制公交车...
环境特征
地理位置 北京东城 上海浦东 长沙雨花区...
时间 早 中 晚 工作日 节假日...
网络 3G 4G WIFI
天气 多云 晴 暴雨 雪 大风...
文章特征
主题词 艺术 科技数码 国际 体育运动...
兴趣标签 瑜伽 旅行 网球 NBA...
热度 762 1788...
时效性 1分钟内 5分钟前 2小时前
质量 1 3 5 6
作者来源 流动科技生活 武汉晨报...
相似文章 来自未来的微波炉
手机还是相机 lumia1020
世界上最大的菜卡相机
李娜雨中憾别温网...
法国学生教你用3D打印技术造出单反相机
流动科技生活 评论 128 07-10 16:55
李娜与杜兰特出席活动粉丝尖叫
武汉晨报 评论 221 07-10 14:51
你关心的
才是头条

# “今日头条”应用案例

## ➢定量计算推荐逻辑

用户：小A　性别：女　年龄：20—25岁
常驻地：合肥　　目前所在地：黄山
兴趣词：考研、旅游、美妆、音乐

| | | |
|---|---|---|
| 1.合肥 | 《考研数学名师来合肥进行见面辅导》 | 20%×0.2=4% |
| 2.合肥 女 年龄20—25岁 | 《合肥大学毕业生落户新政》 | 30%×0.3=9% |
| 3.外地 黄山 | 《黄山旅游奇遇记》 | 35%×0.8=28% |
| 4.音箱 | 《人工智能音箱定制个性化音乐》 | 30%×0.5=15% |

# ➢客户群体细分

# 客户群体细分

- 商业模式从以产品为中心向以客户为中心转变
- 先对客户进行聚类分析，在此基础上提取出规则

# 客户特征分析

- 近度（Recency）
  - 最近一次购买距离分析点的时间
- 频度（Frequency）
  - 一定时期内购买产品的次数
- 值度（Monetary value）
  - 一定时期内购买产品的总金额

# 数据预处理

- 去除冗余、填补遗漏
- 离散化

| 分值 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 时间 | 08 年以下 | 08 年-09 年 | 09 年-10 年 | 10 年-11 年 | 11 年-12 年 |
| 客户 | | | | | |

# 聚类输出

| Cluster Center | $C_1$ | $C_2$ | $C_3$ | $C_4$ | $C_5$ |
|---|---|---|---|---|---|
| $C_1$_R | 1.72 | 4.67 | 1.54 | 4.59 | 1.44 |
| $C_2$_F | 1.18 | 2.92 | 1.23 | 1.17 | 3.22 |
| $C_3$_M | 1.31 | 4.75 | 3.54 | 1.58 | 5.00 |
| $D_i$ | 2.46 | 7.27 | 4.05 | 4.99 | 6.12 |
| 客户价值(输出) | 非常低 | 非常高 | 低 | 中等 | 高 |
| 客户数量 | 93 | 12 | 13 | 59 | 9 |

# 关联规则挖掘

- 事务数据表

| TID | ItemList |
|---|---|
| 0001 | SBW-30,SBW-50... |
| 0002 | SBW-30,SBW-50... |
| 0003 | SGB-50 |
| 0004 | DJA-10,SVC-10 |
| 0005 | SBW-250 |

- 采用Apriori算法挖掘频繁项集，并推导关联规则

# 关联规则挖掘

- 频繁三项集
  - {SBW-30, SBW-50, SBW-100}
  - {SBW-100, SBW-180, SBW-400}
- 强规则
  - SBW-180 ➔ SBW-400，52.63%