



武汉光电国家实验室(筹)
WUHAN NATIONAL LABORATORY FOR OPTOELECTRONICS

关联分析： 基本概念和算法

电子信息与通信学院 冯 斌

fengbin@hust.edu.cn



关联规则挖掘

- 给定一系列记录，找到其中隐含的令人感兴趣的联系，用以根据记录中某些项的出现来预测其他项的产生

商场购物篮事务

<i>TID</i>	<i>Items</i>
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

关联规则的例子

$\{\text{Diaper}\} \rightarrow \{\text{Beer}\},$
 $\{\text{Milk, Bread}\} \rightarrow \{\text{Eggs, Coke}\},$
 $\{\text{Beer, Bread}\} \rightarrow \{\text{Milk}\},$

关联意味着同时出现，而并非因果关系

关联规则挖掘

- 一、从大型事务数据集中发现模式可能在计算上要付出很高的代价
 - 有效挖掘模式的算法
- 二、所发现的某些模式可能是虚假的，因为它们可能是偶然发生的
 - 模式的评估问题

关联规则挖掘

➤ 二元表示

➤ 非对称二元变量

➤ 项集

➤ 包含0个或多个项的集合

➤ K-项集

➤ 支持度计数

➤ 包含特定项集的事务个数

$$\sigma(X) = |\{t_i \mid X \subseteq t_i, t_i \in T\}|$$

➤ $\sigma(\{\text{牛奶, 面包, 尿布}\}) = 2$

TID	面包	牛奶	尿布	啤酒	鸡蛋	可乐
1	1	1	0	0	0	0
2	1	0	1	1	1	0
3	0	1	1	1	0	1
4	1	1	1	1	0	0
5	1	1	1	0	0	1

TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

关联规则挖掘

➤ 关联规则

➤ 形如 $X \rightarrow Y$ 的蕴含表达式, 其中 X 和 Y 是不相交的项集

➤ 如: $\{\text{牛奶}, \text{尿布}\} \rightarrow \{\text{啤酒}\}$

TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

➤ 规则评估度量

➤ 支持度(s): 规则可以用于给定数据集的频繁程度

$$s(X \rightarrow Y) = \frac{\sigma(X \cup Y)}{N}$$

$s(\text{牛奶}, \text{尿布} \rightarrow \text{啤酒}) = 0.4$

➤ 置信度(c): Y 在包含 X 的事务中出现的频繁程度

$$c(X \rightarrow Y) = \frac{\sigma(X \cup Y)}{\sigma(X)}$$

$c(\text{牛奶}, \text{尿布} \rightarrow \text{啤酒}) = 0.67$

关联规则挖掘

- 习题3
- (a)规则 空集 $\rightarrow A$ 和 $A \rightarrow$ 空集的置信度是多少
- (b)令 c_1 , c_2 和 c_3 分别是规则 $\{p\} \rightarrow \{q\}$, $\{p\} \rightarrow \{q, r\}$ 和 $\{p, r\} \rightarrow \{q\}$ 的置信度, 它们之间可能存在什么关系
- (c)假定(b)中的规则具有相同的支持度, 重复(b)的分析
- (d)传递性: 假定规则 $A \rightarrow B$ 和 $B \rightarrow C$ 的置信度都大于阈值 minconf , 规则 $A \rightarrow C$ 可能具有小于 minconf 的置信度吗



关联规则挖掘

- 问题描述：给定事务的集合T，关联规则发现是指找出支持度大于等于**minsup**并且置信度大于等于**minconf**的所有规则
 - **minsup**是支持度阈值
 - **minconf**是置信度阈值
- 最直接的方法：计算每个可能规则的支持度和置信度，排除不满足条件的规则
 - 可能的规则总数为 $R=3^d-2^{d+1}+1$

关联规则挖掘

Example of Rules:

TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

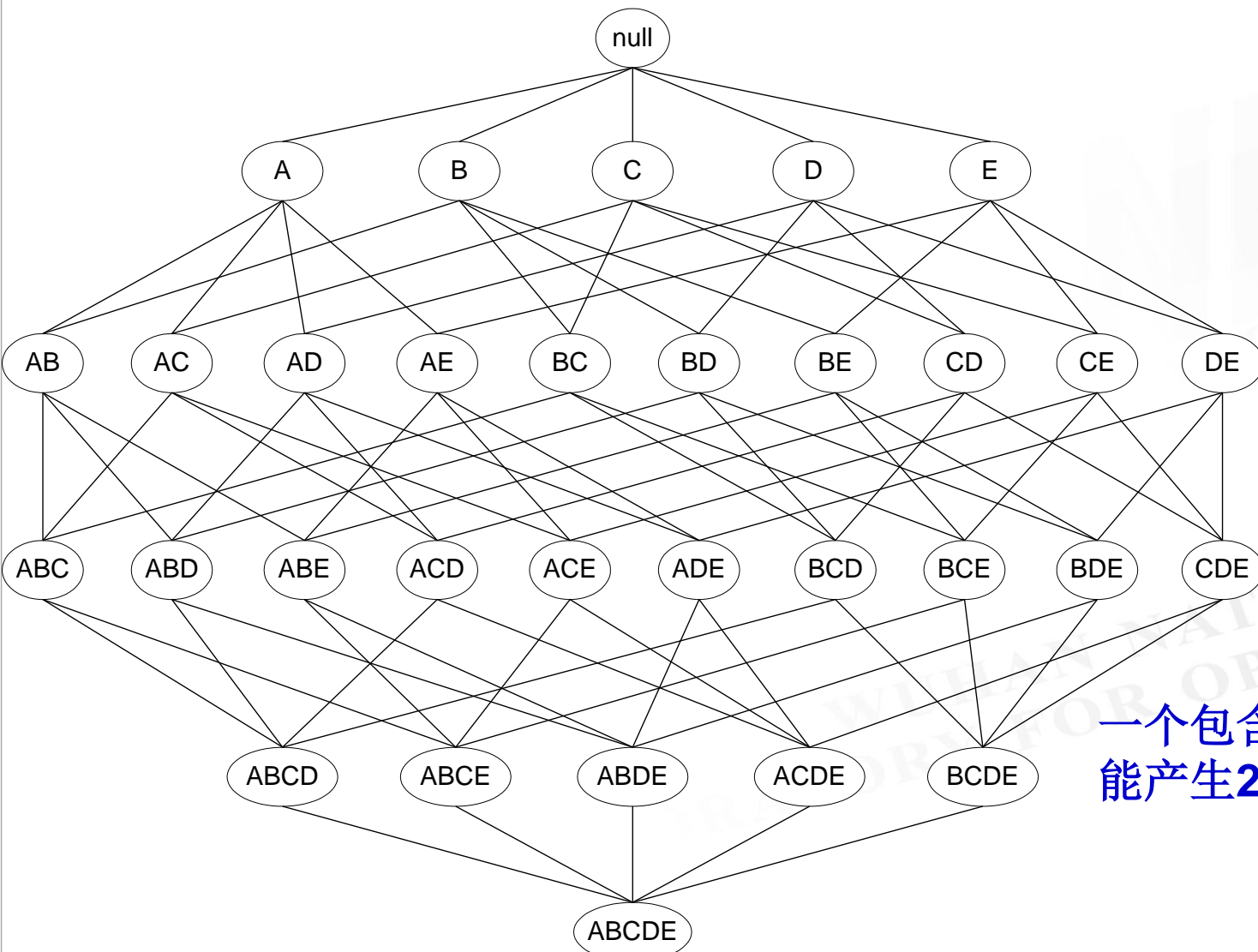
$\{\text{Milk, Diaper}\} \rightarrow \{\text{Beer}\} (s=0.4, c=0.67)$
 $\{\text{Milk, Beer}\} \rightarrow \{\text{Diaper}\} (s=0.4, c=1.0)$
 $\{\text{Diaper, Beer}\} \rightarrow \{\text{Milk}\} (s=0.4, c=0.67)$
 $\{\text{Beer}\} \rightarrow \{\text{Milk, Diaper}\} (s=0.4, c=0.67)$
 $\{\text{Diaper}\} \rightarrow \{\text{Milk, Beer}\} (s=0.4, c=0.5)$
 $\{\text{Milk}\} \rightarrow \{\text{Diaper, Beer}\} (s=0.4, c=0.5)$

- 上述规则涉及的项来自同一个项集 {牛奶, 尿布, 啤酒}
- 从同一个项集中推导出的规则必然具有相同的支持度, 置信度可能不同
- 可将支持度和置信度两个任务分解

关联规则挖掘

- 两步法:
- 1. 频繁项集产生: 目标是发现满足最小支持度阈值的所有项集, 称作频繁项集
- 2. 规则的产生: 从发现的频繁项集中提取所有高置信度的规则, 称为强规则
- 频繁项集产生所需的计算开销远大于规则的产生所需的计算开销

频繁项集产生



一个包含 k 个项的数据集可能产生 2^k 个候选项集

频繁项集产生

➤ Brute-force法:

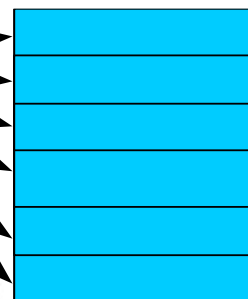
- 将每个格中的项集作为候选频繁项集
- 计算每个候选项集的支持度

Transactions

TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

↑
N
↓

List of
Candidates



↑
M
↓

- 将每个候选项集与每个事务进行比较
- 复杂度 $O(NMw)$

频繁项集产生

➤ 减少候选项集的数目(M)

➤ 全搜索: $M=2^d$

➤ 利用剪枝技术减小M

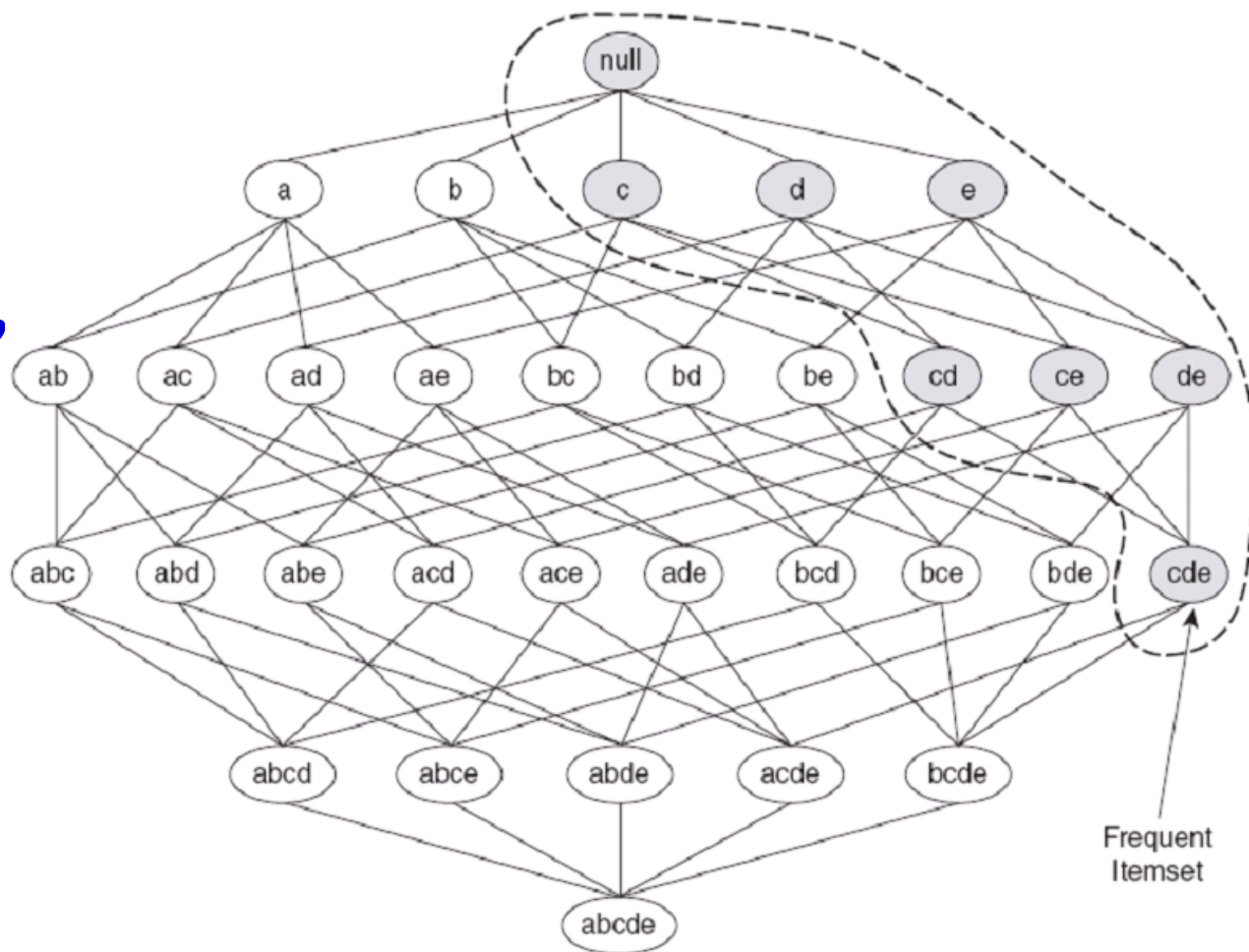
➤ 减少比较次数(NM)

➤ 使用更高级的数据结构存储候选项集或事务集

➤ 不需要在每个候选集和事务之间匹配

频繁项集产生

➤ 先验原理：
如果一个项集是频繁的，
则它的所有子集一定也是频繁的



频繁项集产生

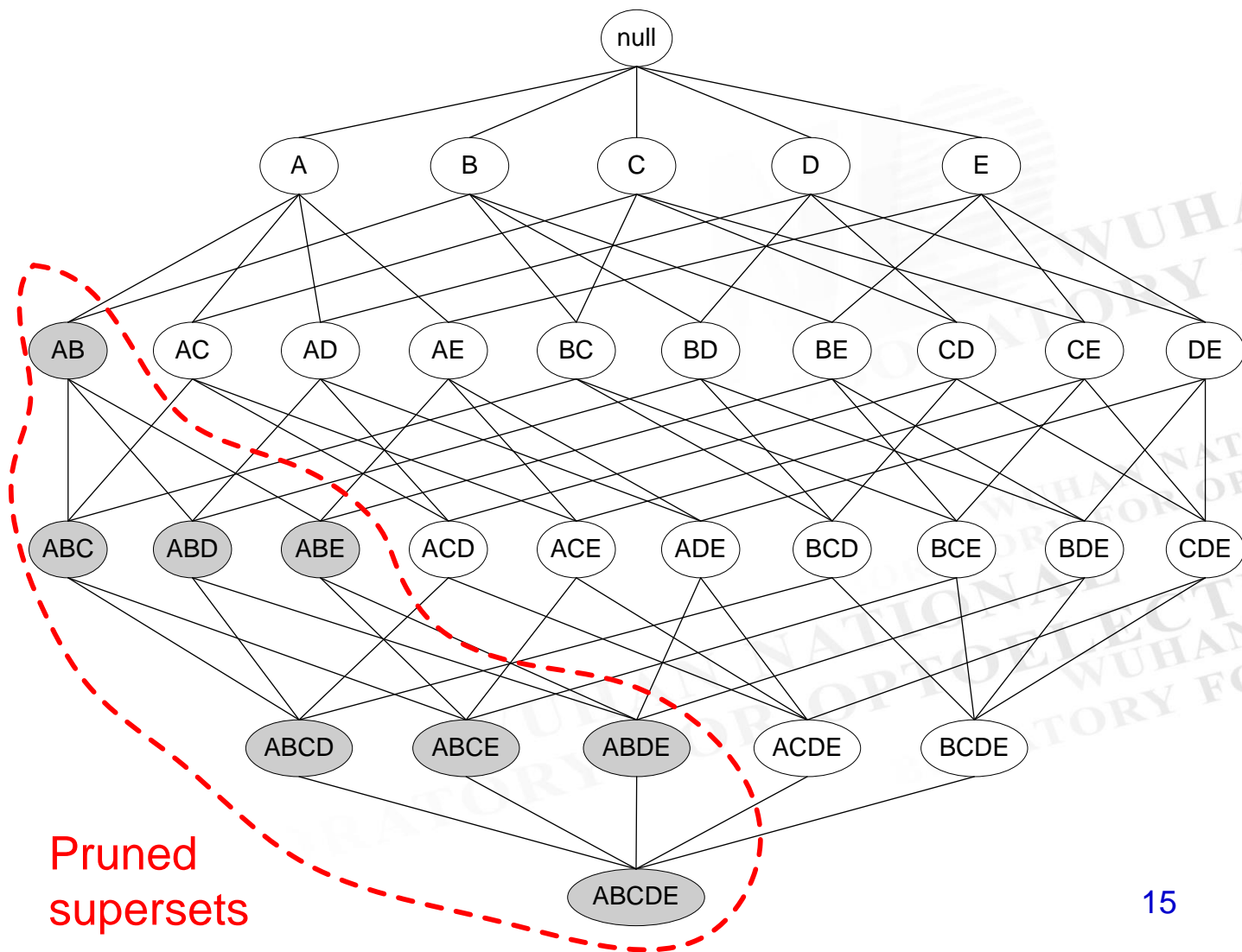
- 相反，如果一个项集是非频繁的，则它的所有超集也一定是非频繁的
- 基于支持度度量的一个关键性质：一个项集的支持度不会超过它的子集的支持度

$$\forall X, Y : (X \subseteq Y) \Rightarrow s(X) \geq s(Y)$$

- 反单调性

频繁项集产生

➤ 基于支持度的剪枝



Items (1-itemsets)

Item	Count
Bread	4
Coke	2
Milk	4
Beer	3
Diaper	4
Eggs	1



Itemset	Count
{Bread,Milk}	3
{Bread,Beer}	2
{Bread,Diaper}	3
{Milk,Beer}	2
{Milk,Diaper}	3
{Beer,Diaper}	3

Pairs (2-itemsets)

(No need to generate candidates involving Coke or Eggs)

Minimum Support = 3

If every subset is considered,
 $C_1^6 + C_2^6 + C_3^6 = 41$
 With support-based pruning,
 $6 + 6 + 1 = 13$

TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke



Triplets (3-itemsets)

Itemset	Count
{Bread,Milk,Diaper}	3



频繁项集产生

Apriori 算法的频繁项集产生↵

1: $k=1$ ↵

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

3: repeat↵

4: $k=k+1$ ↵

5: $C_k = \text{apriori-gen}(F_{k-1})$ {产生候选项集}↵

6: for 每个事务 $t \in T$ do↵

7: $C_t = \text{subset}(C_k, t)$ {识别属于 t 的所有候选}↵

8: for 每个候选项集 $c \in C_t$ do↵

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

10: end for↵

11: end for↵

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

13: until $F_k = \text{空集}$ ↵

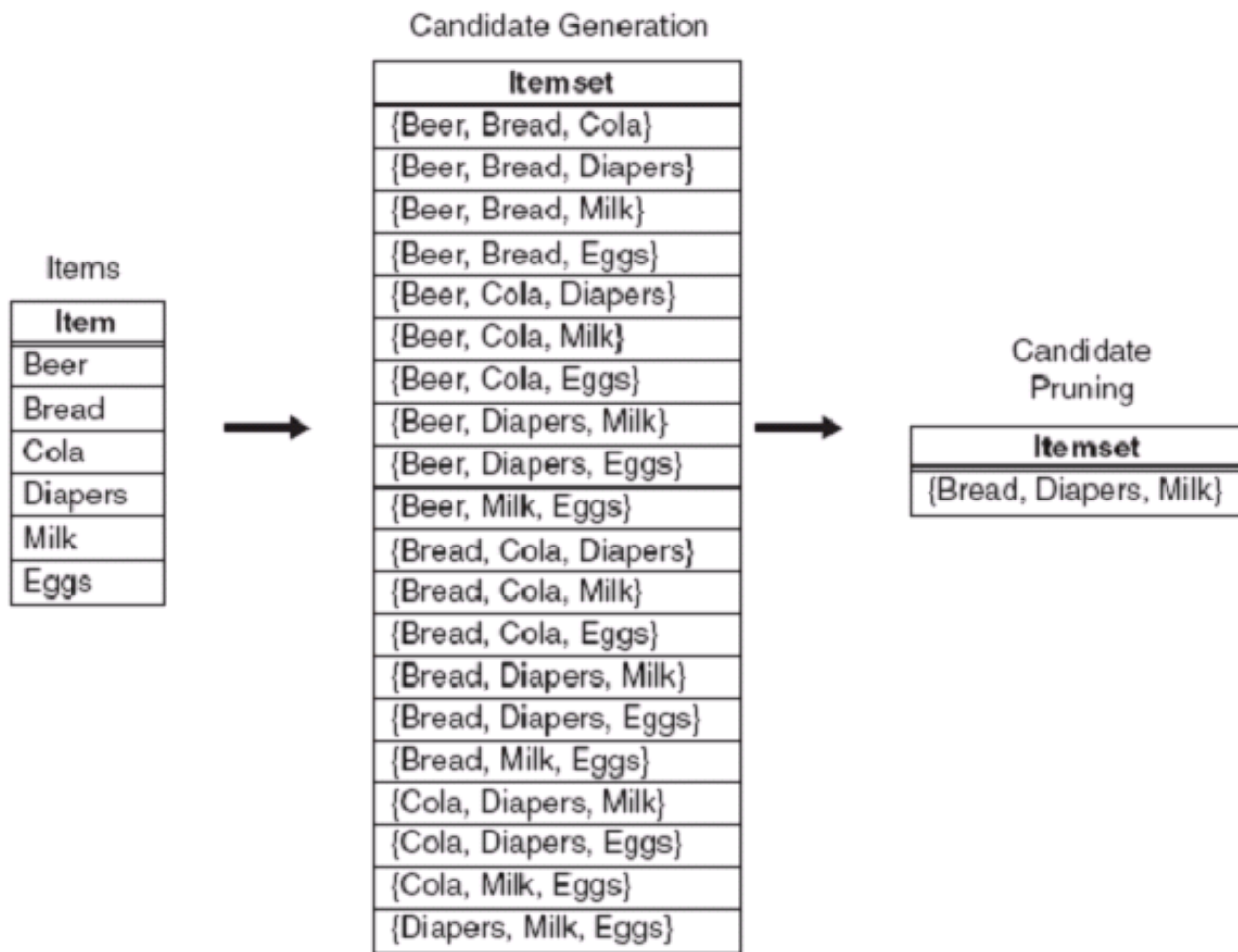
14: Result = $\bigcup F_k$ ↵

频繁项集产生

- **Apriori-gen**函数通过如下两个操作产生候选项集：
 - **1. 候选项集的产生**
 - 由前一次迭代发现的频繁(**k-1**)-项集产生新的候选**k**-项集
 - **2. 候选项集的剪枝**
 - 采用基于支持度的剪枝策略，删除一些候选**k**-项集

频繁项集产生

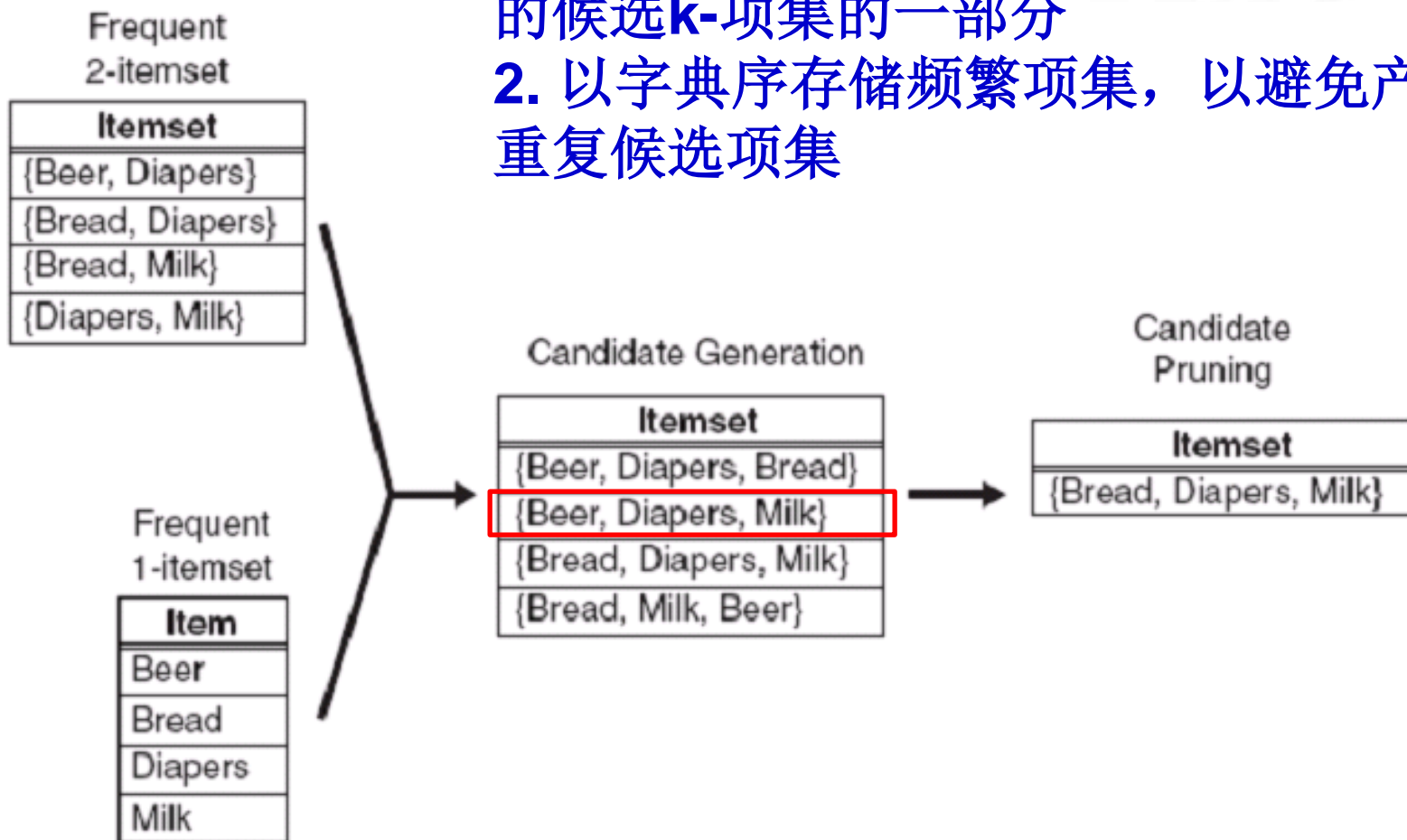
➤ Brute-forth法



频繁项集产生

➤ $F_{k-1} \times F_1$ 法

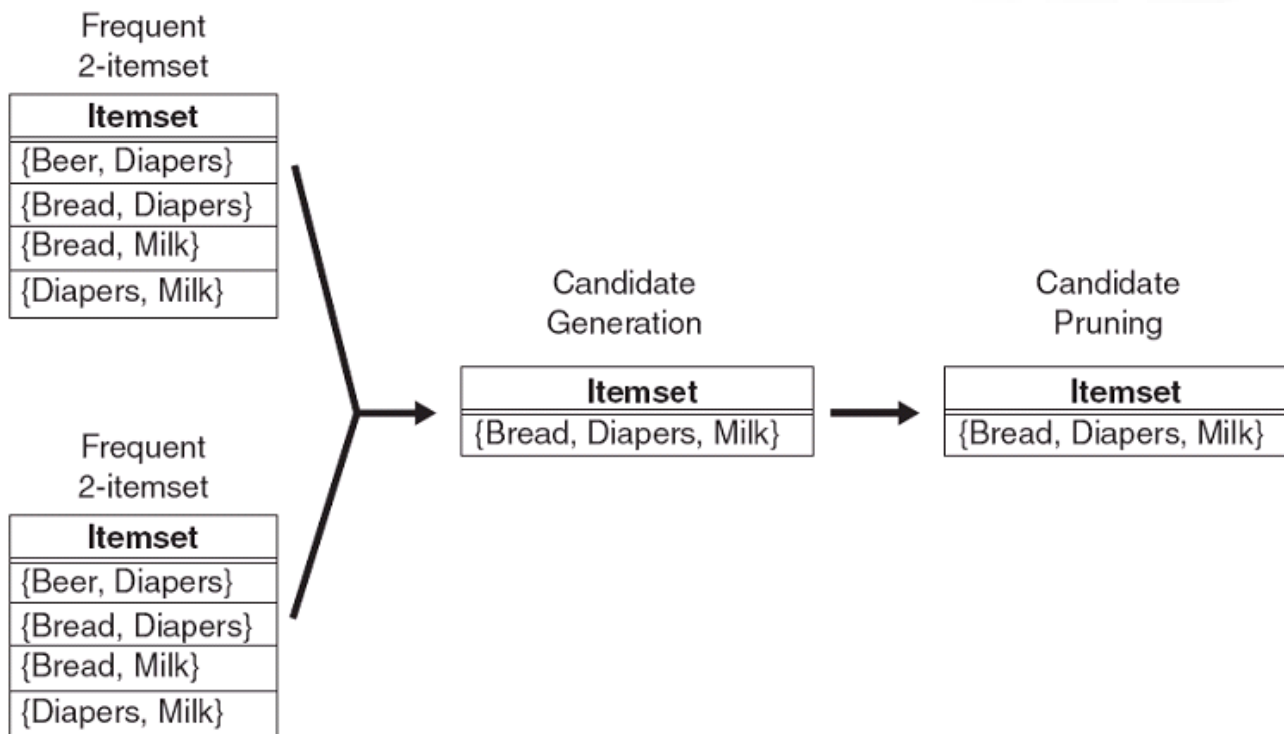
1. 所有的频繁k-项集是这种方法所产生的候选k-项集的一部分
2. 以字典序存储频繁项集，以避免产生重复候选项集



频繁项集产生

➤ $F_{k-1} \times F_{k-1}$ 法

- 合并一对频繁 $k-1$ 项集，仅当它们的前 $k-2$ 个项集都相同



频繁项集产生

➤ 习题7

➤ 考虑下面的频繁3-项集的集合:

$\{1,2,3\}, \{1,2,4\}, \{1,2,5\}, \{1,3,4\}, \{1,3,5\}, \{2,3,4\},$
 $\{2,3,5\}, \{3,4,5\}$, 假定数据集中只有5个项。

(a) 列出采用 $F_{k-1} \times F_1$ 合并策略, 由候选产生过程得到的所有候选4-项集

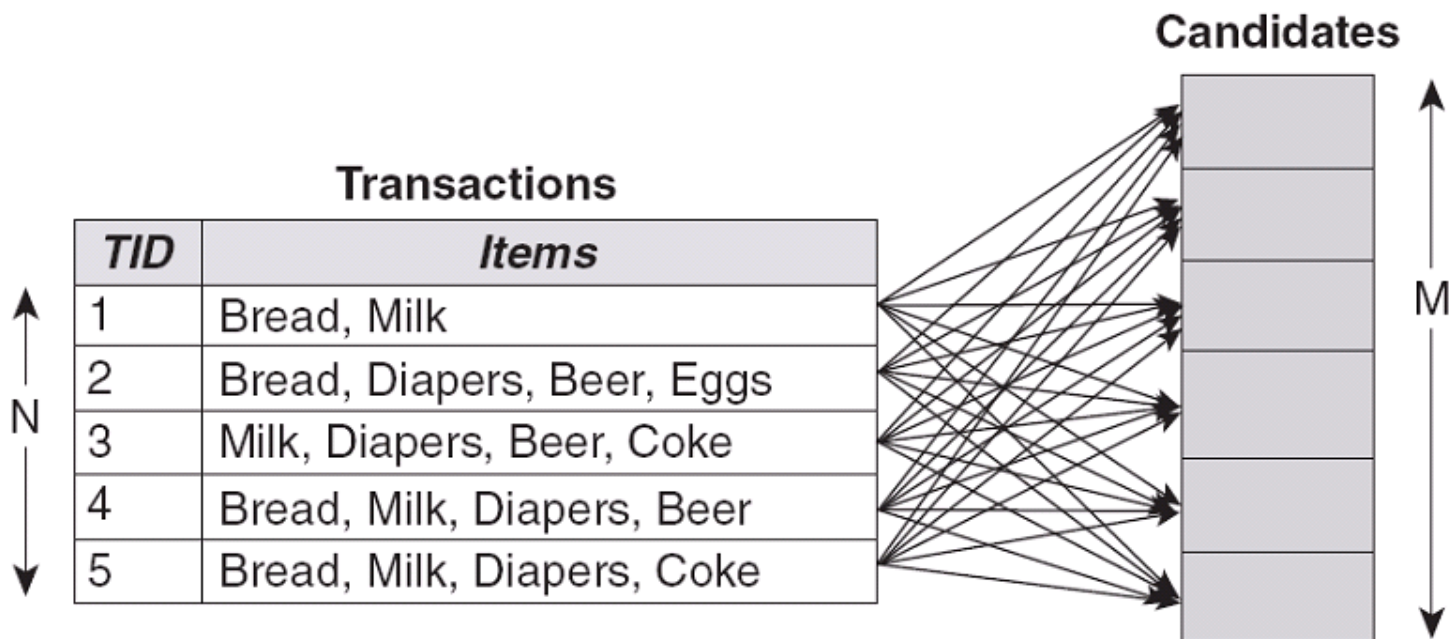
(b) 列出由Apriori算法的候选产生过程得到的所有候选4-项集

(c) 列出Apriori算法候选剪枝步骤后剩下的所有候选4-项集



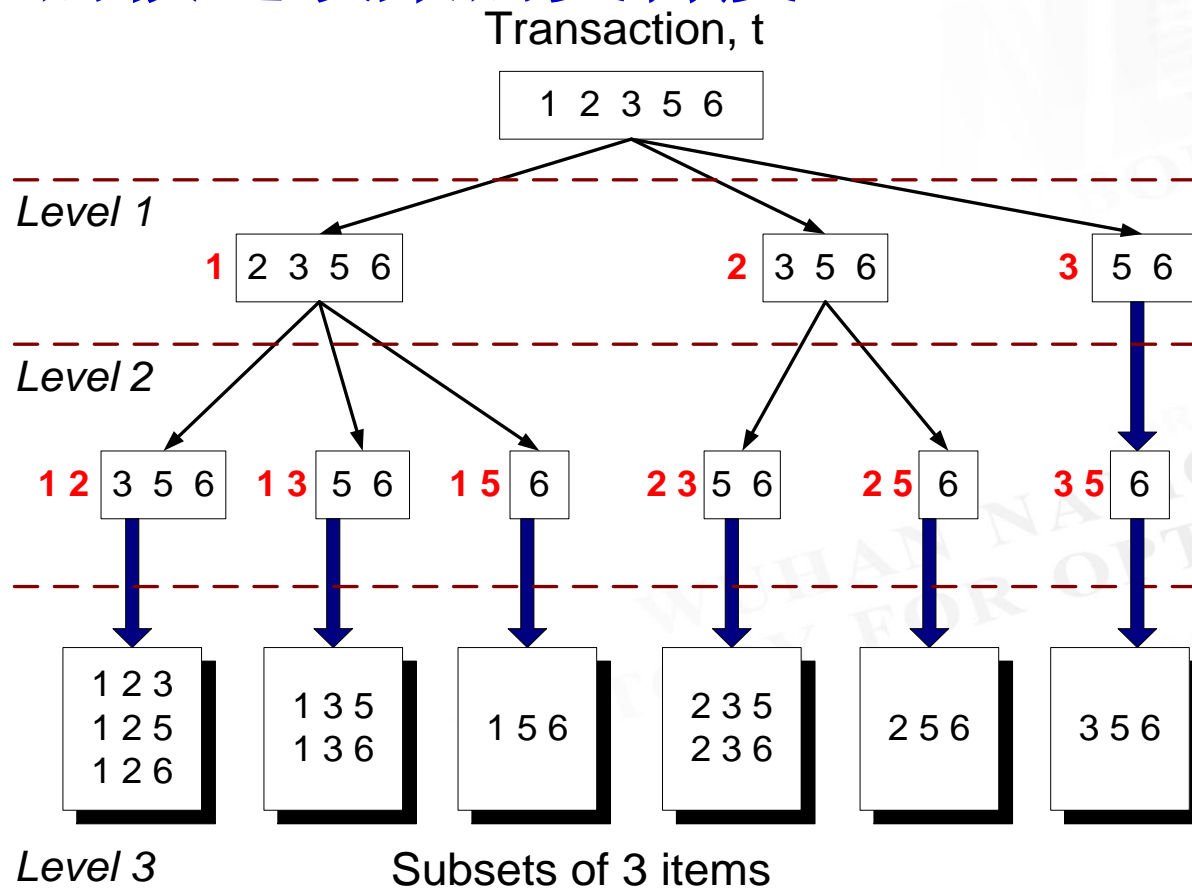
支持度计数

- 确定在候选项剪枝步骤保留下来的每个候选项集出现的频繁程度
- 方法1：将每个事务与所有的候选项集进行比较，并更新包含在事务中的候选项集支持度计数



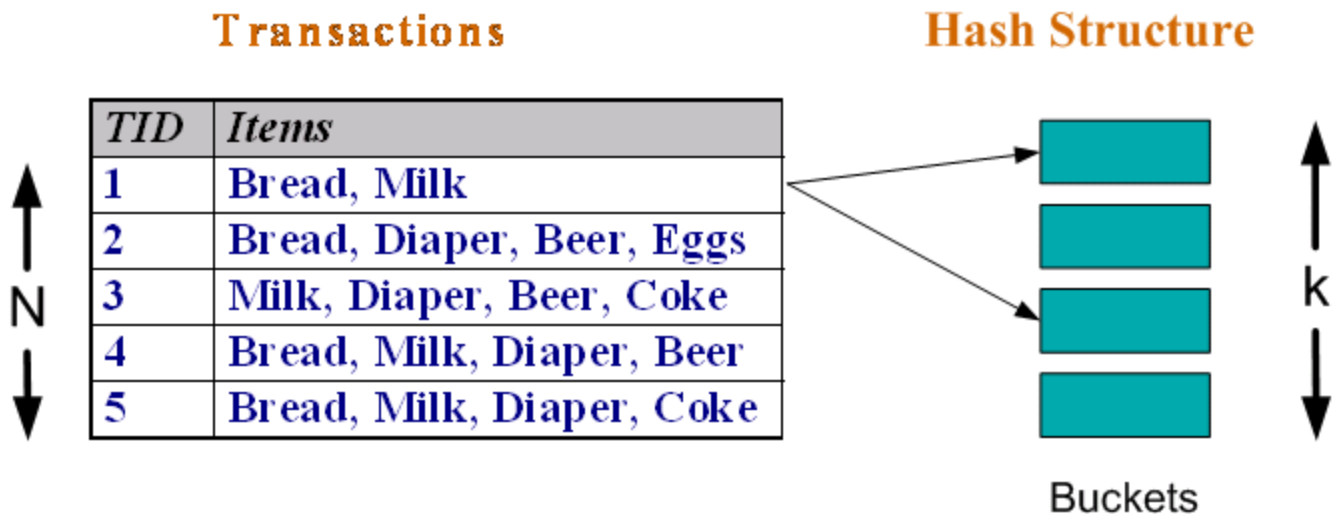
支持度计数

- **方法2：** 枚举每个事务所包含的项集，并更新对应的候选项集的支持度

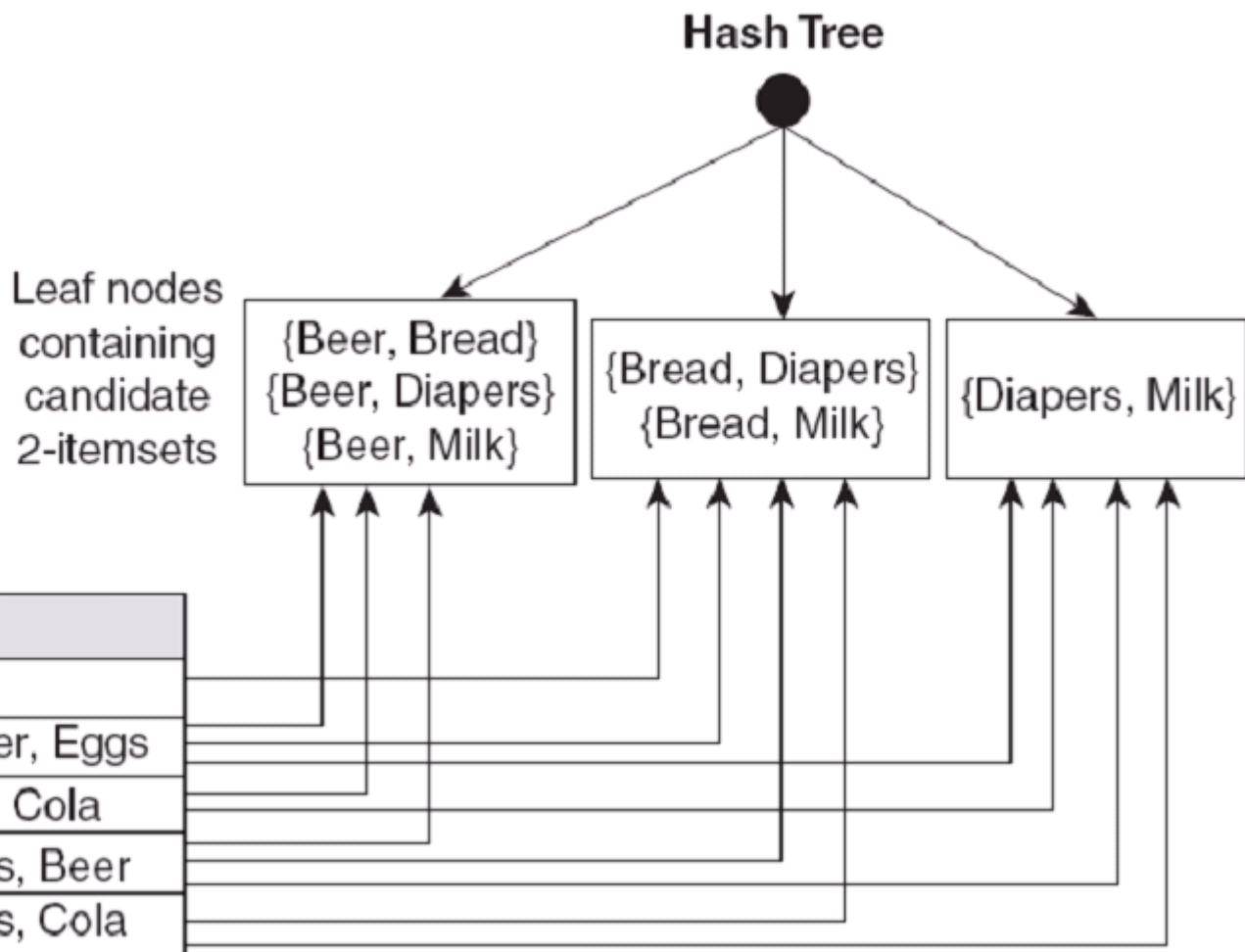


支持度计数

- 将候选项集划分，以hash树结构存储
- 包含在事务中的项集也散列到相应的桶中
- 不是将事务中的每个项集与所有的候选项集进行比较，而是与处于同一桶内的候选项集进行匹配



支持度计数

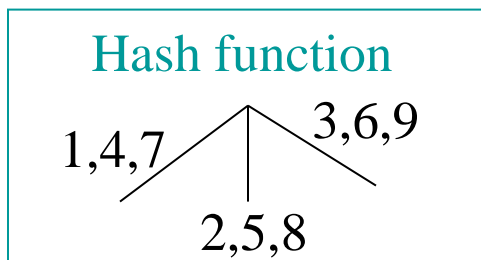


生成hash树

➤ 假设有15个候选3-项集

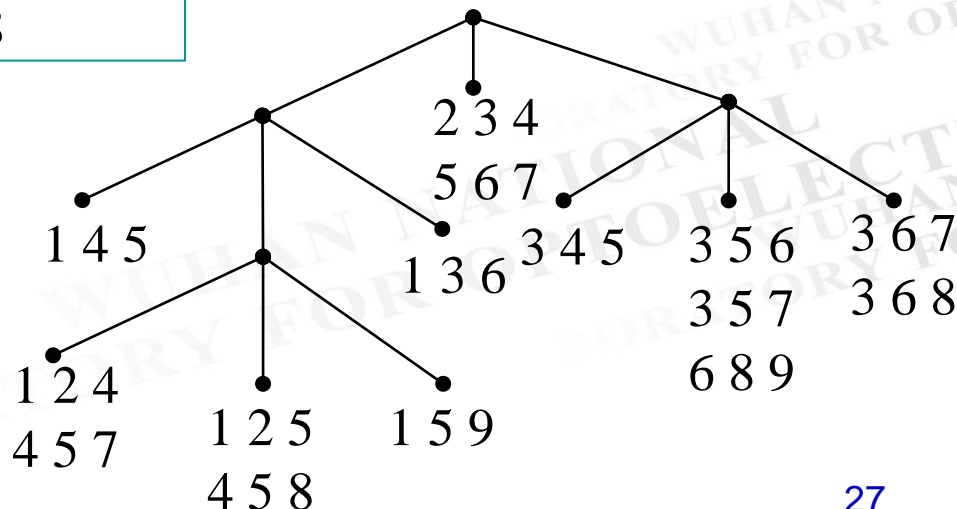
{1 4 5}, {1 2 4}, {4 5 7}, {1 2 5}, {4 5 8}, {1 5 9}, {1 3 6}, {2 3 4},
{5 6 7}, {3 4 5}, {3 5 6}, {3 5 7}, {6 8 9}, {3 6 7}, {3 6 8}

➤ Hash函数



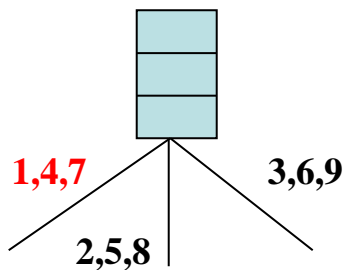
➤ 最大叶大小

➤ 叶结点存储项的最大个数

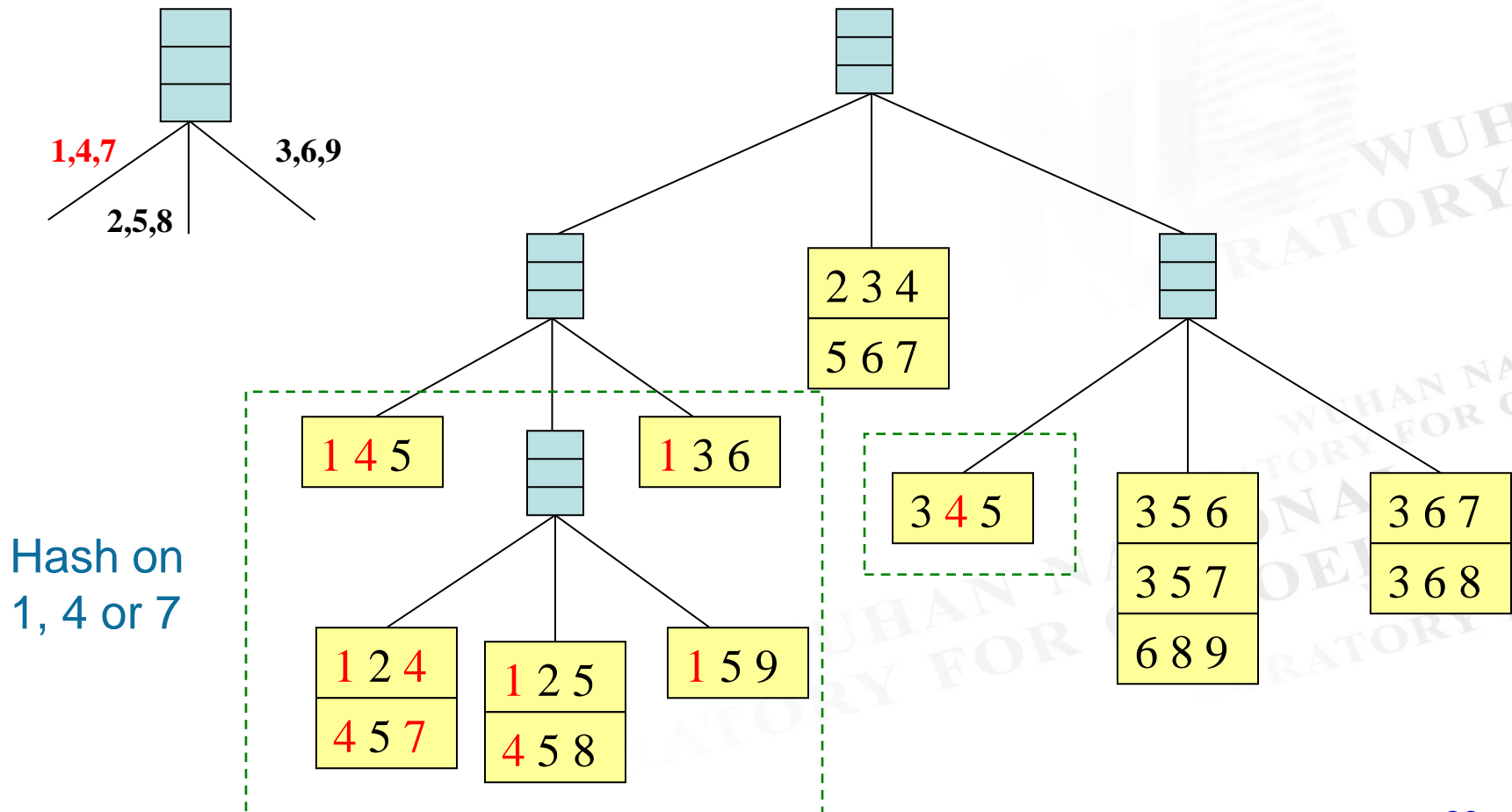


Hash树

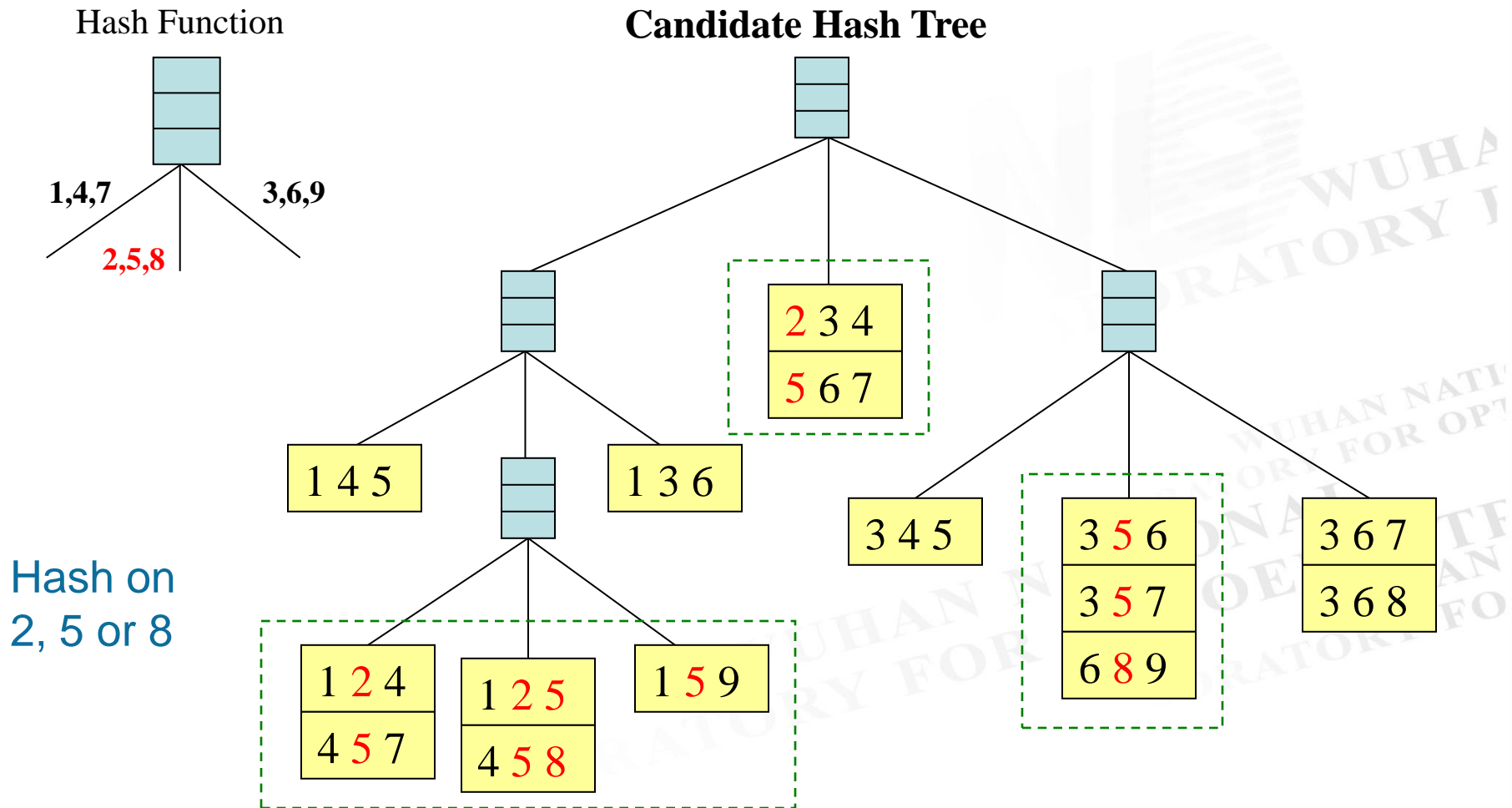
Hash Function



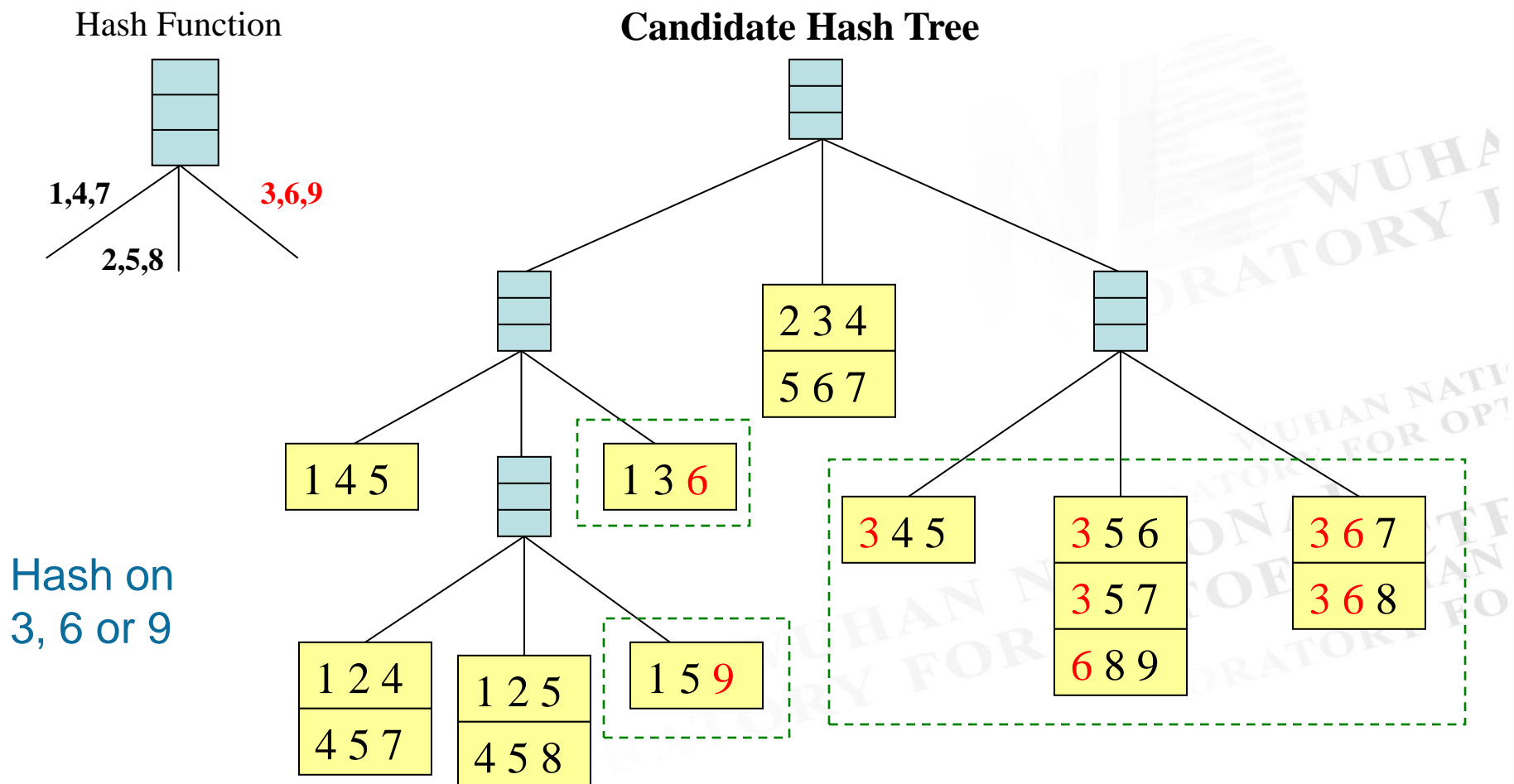
Candidate Hash Tree



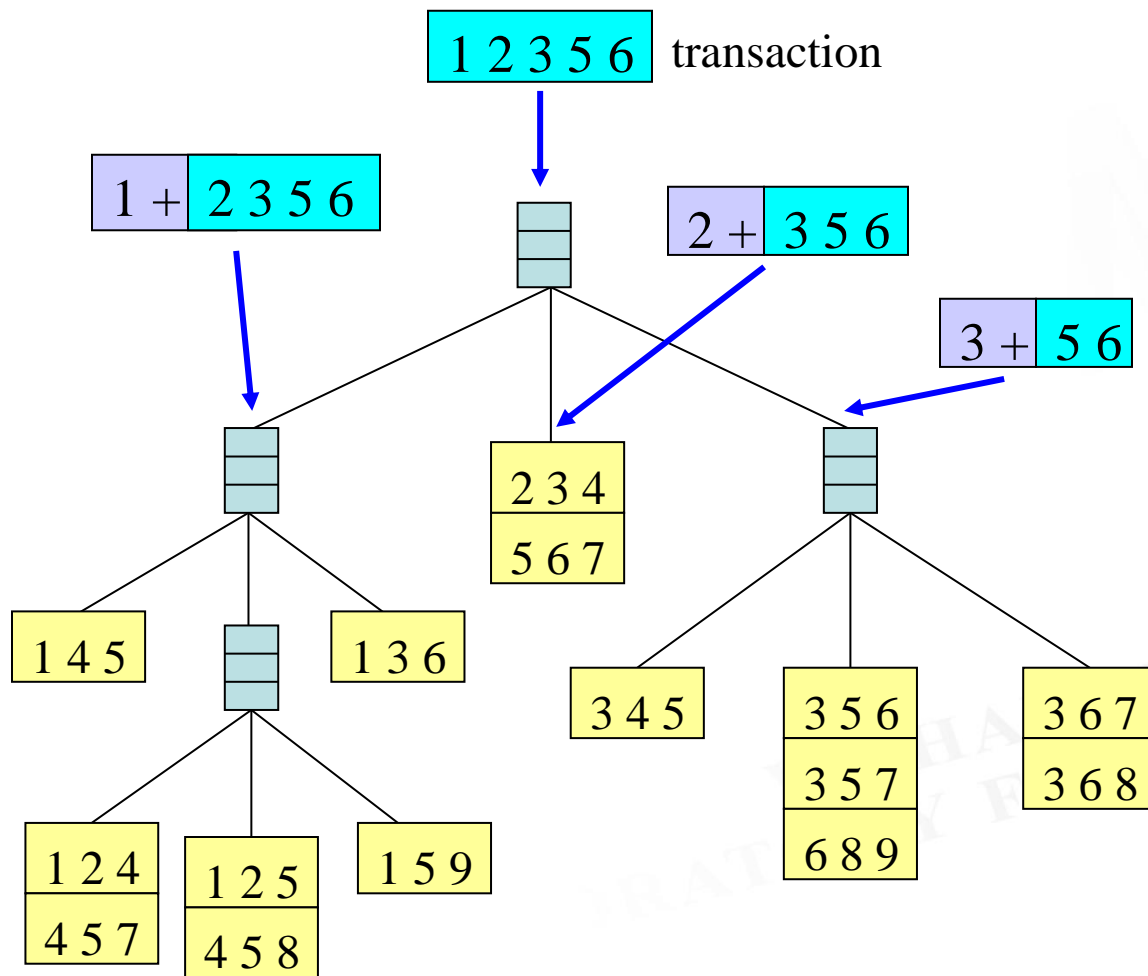
Hash树



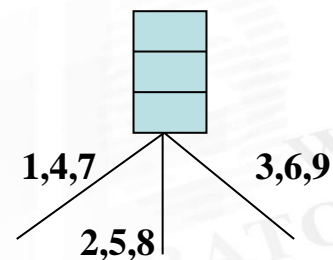
Hash树



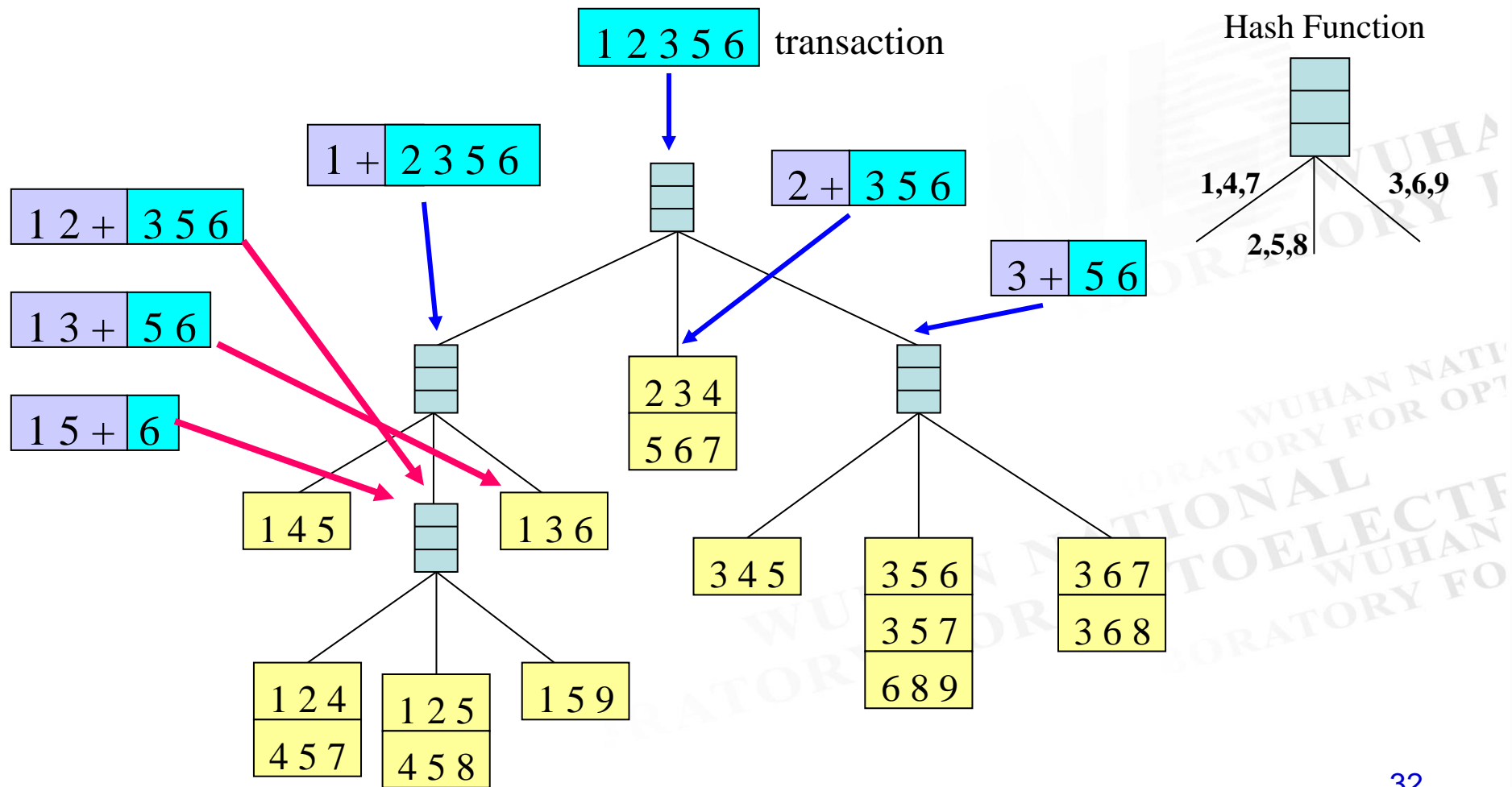
Hash树的子集操作



Hash Function

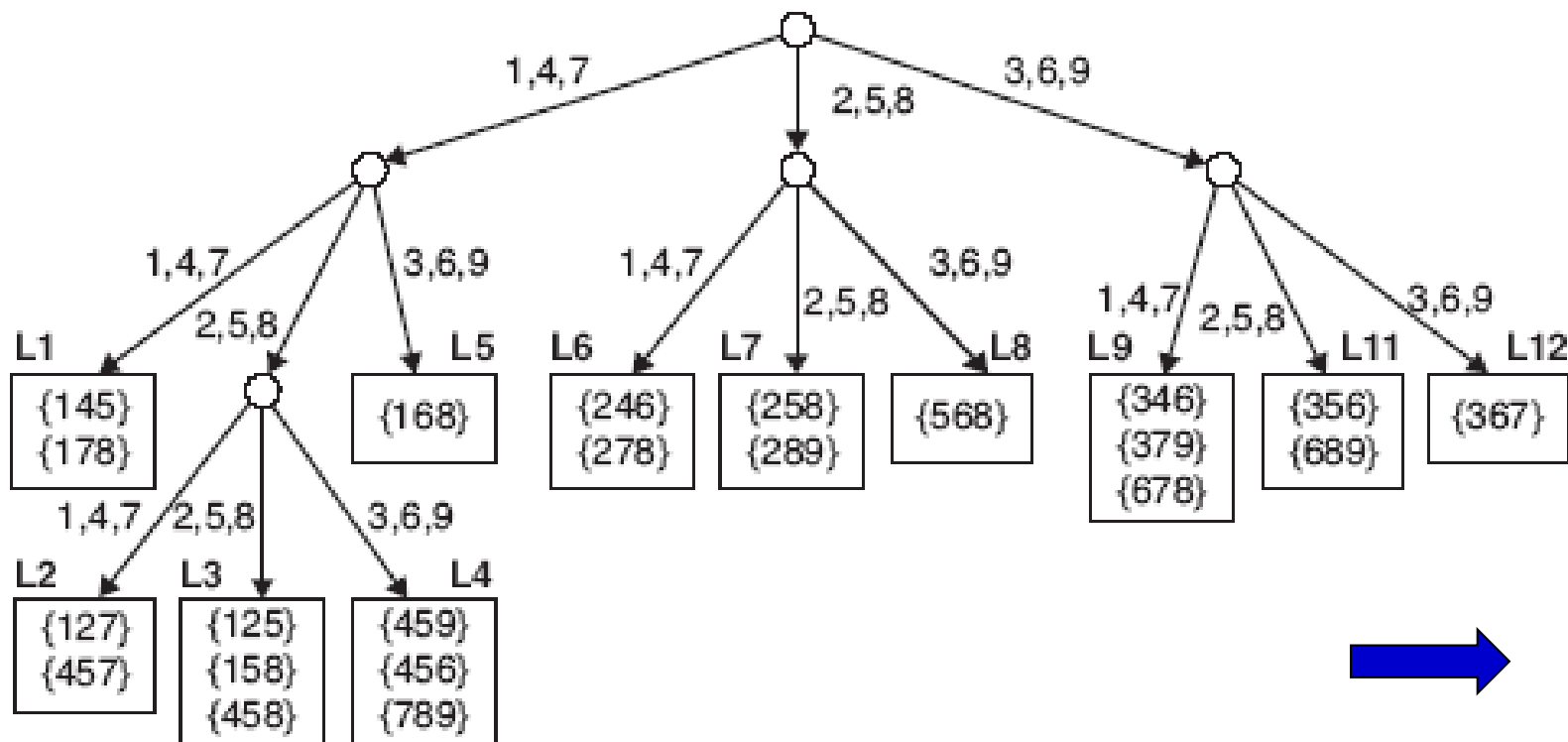


Hash树的子集操作



Hash树的子集操作

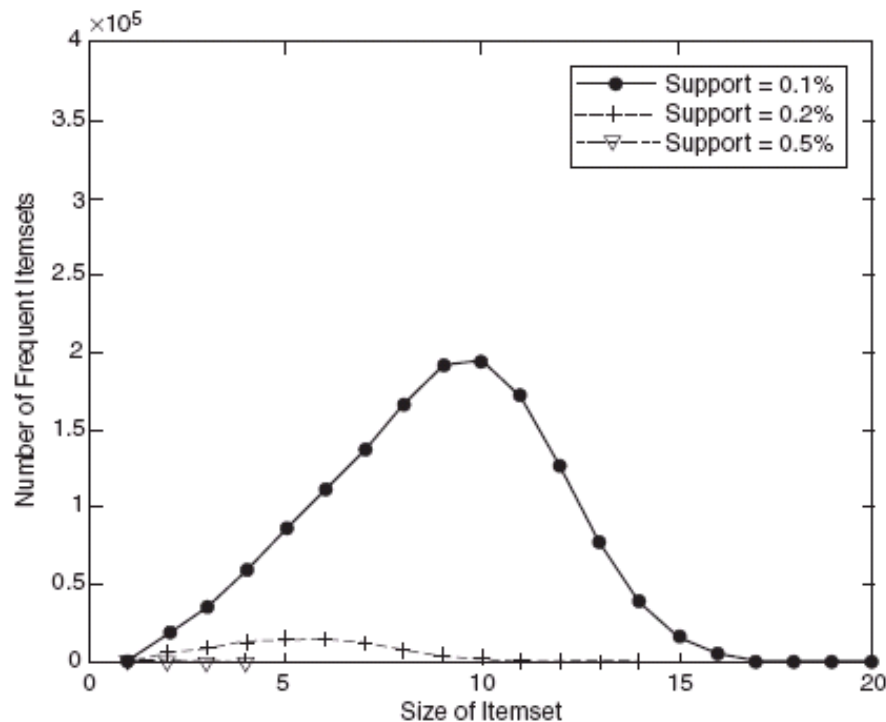
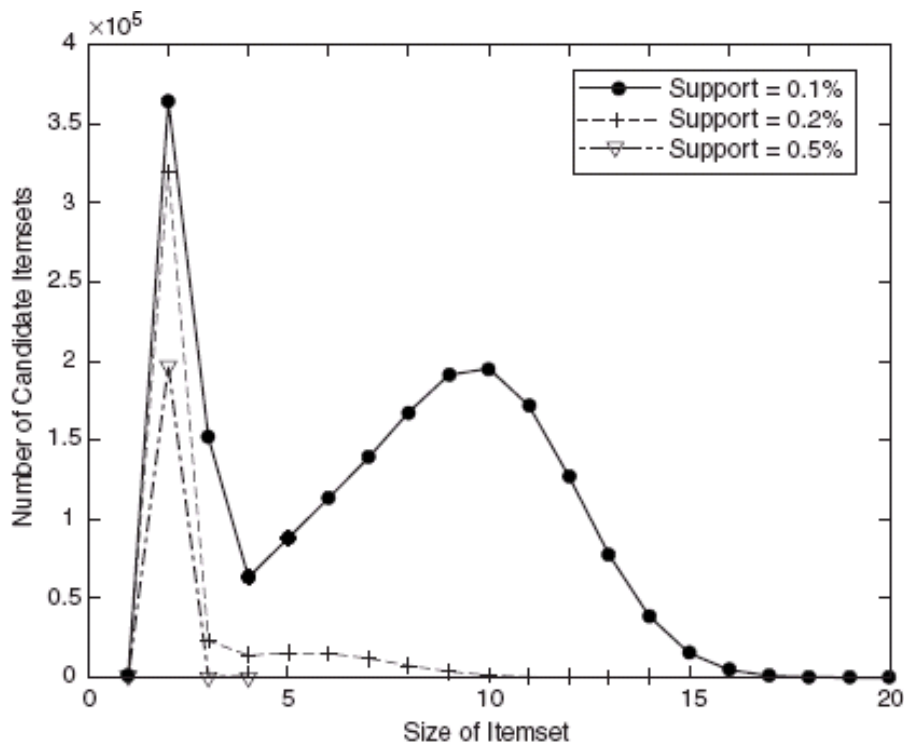
- 习题9
- (a) 给定一个包含项{1,3,4,5,8}的事务，在寻找该事务的候选项集时，访问了Hash树的哪些叶结点？
- (b) 使用访问的叶结点确定包含的候选项集



Apriori计算复杂度

➤ 支持度阈值

- 降低支持度阈值将导致更多的项集是频繁的
- 会增加频繁项集的最大长度



Apriori计算复杂度

➤ 数据集的维度

- 需要更多的空间来存储项的支持度计数
- 如果频繁项集的数目也增长，计算量和IO开销将增加

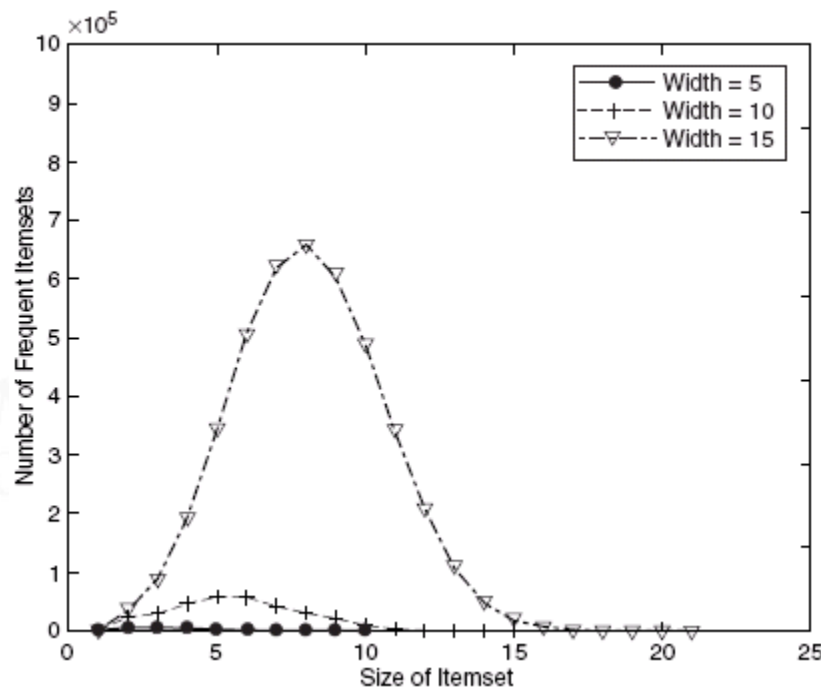
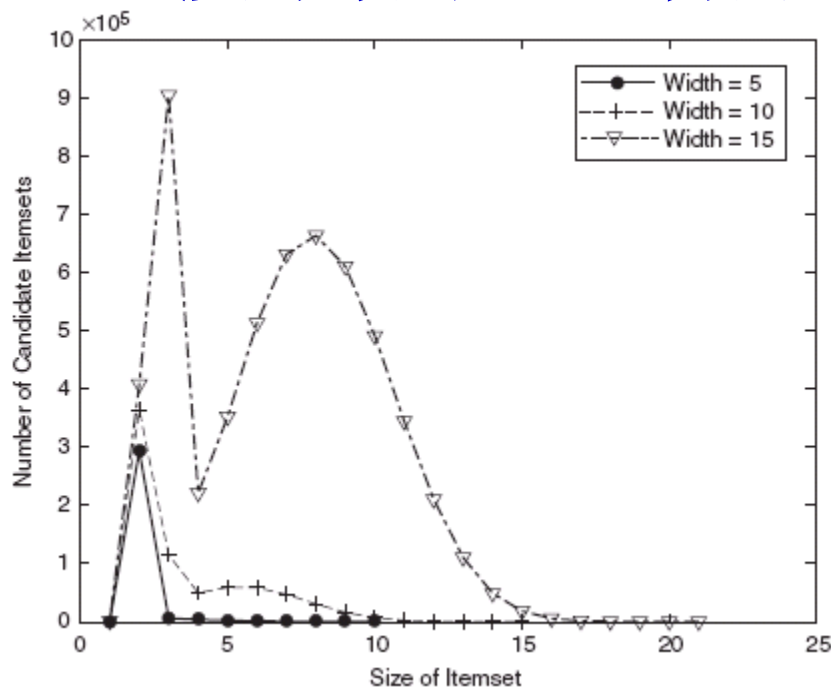
➤ 事务数

- 由于需要反复扫描数据集，因此其运行时间随着事务数增加而增加

Apriori计算复杂度

事务的平均宽度

- 频繁项集的最大长度随事务平均宽度增加而增加
- 事务中包含的项集也随之增加，同时将增加支持度计数时hash树的遍历次数



关联规则挖掘

- 两步法:
- 1. 频繁项集产生: 目标是发现满足最小支持度阈值的所有项集, 称作频繁项集
- 2. 规则的产生: 从发现的频繁项集中提取所有高置信度的规则, 称为强规则

规则产生

- 每个频繁**k**-项集能产生最多 **2^k-2** 个关联规则
 - 将项集**Y**划分成两个非空的子集**X**和**Y-X**，使得**X \rightarrow Y-X**满足置信度阈值

$$\sum_{i=1}^{K-1} C_i^K = 2^K - 1 - 1 = 2^K - 2$$

- 从频繁项集提取的规则必然已经满足支持度阈值

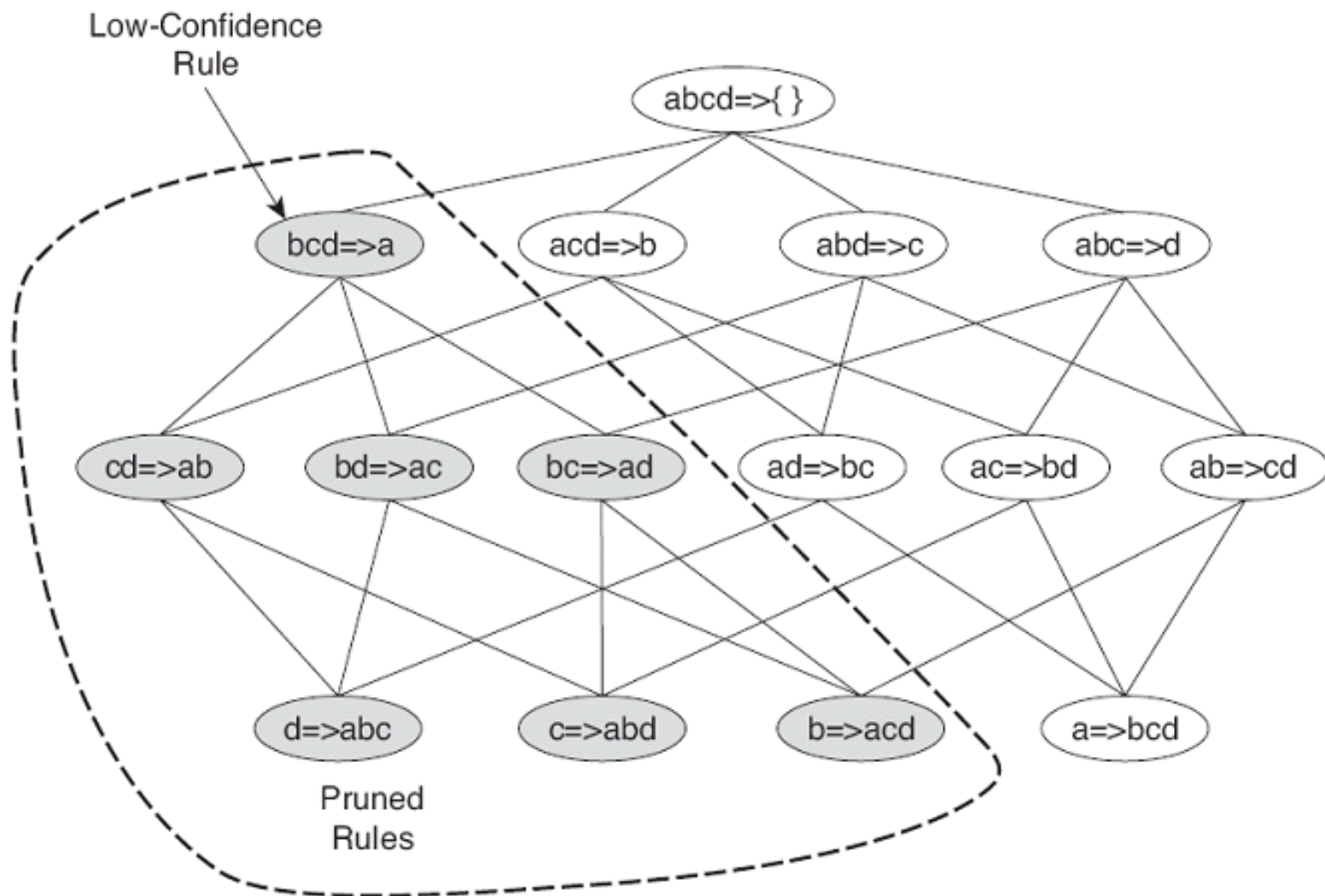
规则产生

- 置信度不具有任何单调性
 - $X \rightarrow Y$ 的置信度可能大于, 小于或等于规则 $X^* \rightarrow Y^*$ 的置信度, 其中 $X^* \in X$ 且 $Y^* \in Y$
- 定理: 如果规则 $X \rightarrow Y$ 不满足置信度阈值, 则 $X' \rightarrow Y$ 的规则一定也不满足置信度阈值, 其中 X' 是 X 的子集
 - $\sigma(Y)/\sigma(X) \geq \sigma(Y)/\sigma(X')$

规则产生

- **Apriori**算法采用逐层算法来产生关联规则，初始提取规则后件只含一个项的所有高置信度规则；然后使用这些规则来产生新的候选规则
 - 每层对应规则后件中的项数
 - 由 $\{acd\} \rightarrow \{b\}$ 和 $\{abd\} \rightarrow \{c\}$ 合并成 $\{ad\} \rightarrow \{bc\}$
- 如果任意结点具有低置信度，则可以立即剪掉该结点生成的整个子图

规则产生



规则产生

Apriori 算法的规则产生

```
1: for 每一个频繁 k-项集  $f_k$ ,  $k \geq 2$  do
2:    $H_1 = \{i \mid i \in f_k\}$  {规则的 1-项后件}
3:   call ap-genrules( $f_k$ ,  $H_1$ )
4: end for
```

过程 ap-genrules(f_k , H_m)

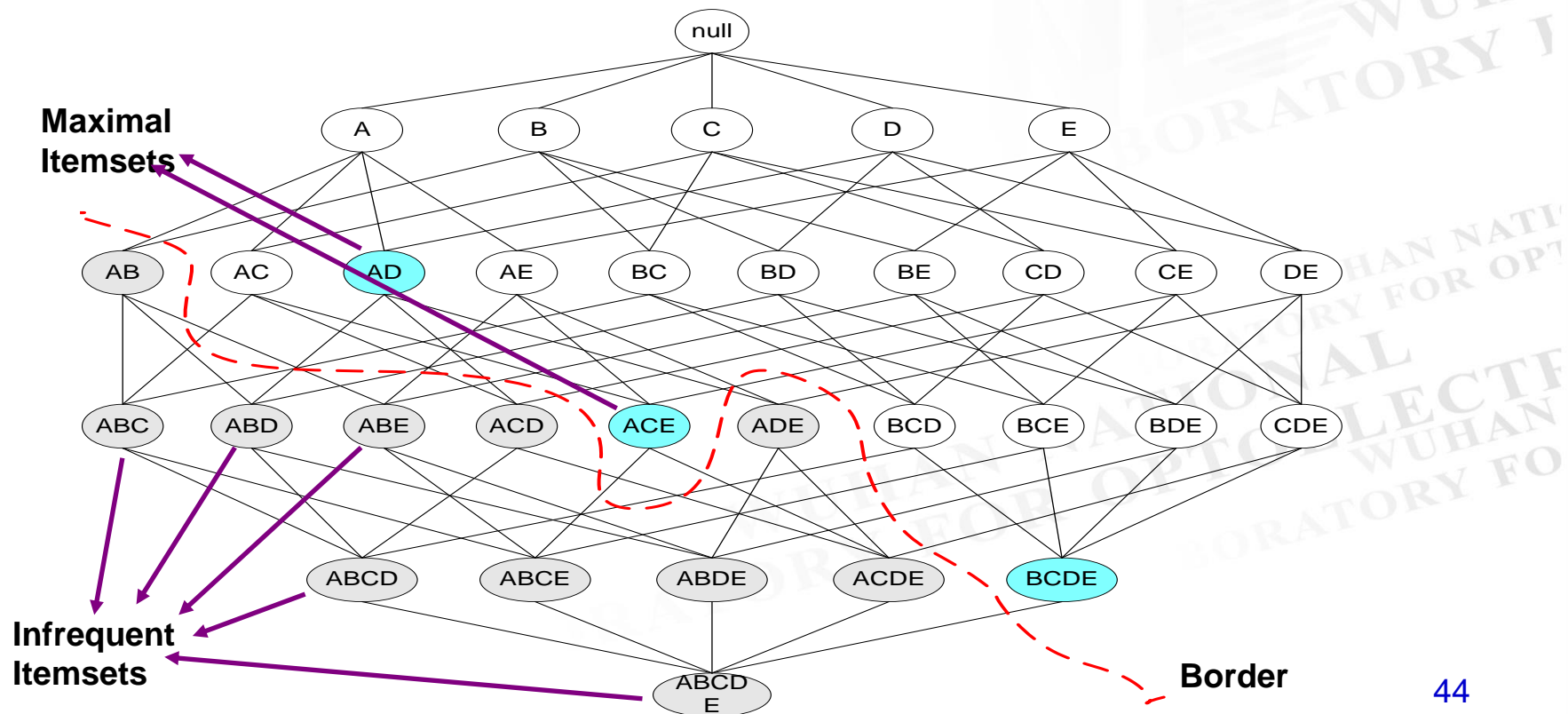
```
1:  $k = |f_k|$  {频繁项集的大小}
2:  $m = |H_m|$  {规则后件的大小}
3: if  $k > m + 1$  then
4:    $H_{m+1} = \text{apriori-gen}(H_m)$ 
5:   for 每个  $h_{m+1} \in H_{m+1}$  do
6:      $\text{conf} = \sigma(f_k) / \sigma(f_k - h_{m+1})$ 
7:     if  $\text{conf} \geq \text{minconf}$  then
8:       output: 规则  $(f_k - h_{m+1}) \rightarrow h_{m+1}$ 
9:     else
10:      从  $H_{m+1}$  delete  $h_{m+1}$ 
11:    end if
12:  end for
13:  call ap-genrules( $f_k$ ,  $H_{m+1}$ )
14: end if
```

频繁项集的紧凑表示

- 由事务数据集产生的频繁项集的数量可能非常大，需要从中识别出可推导出其他所有的频繁项集的，较小的，具有代表性的项集
- 最大频繁项集和频繁闭项集

频繁项集的紧凑表示

- 最大频繁项集(Maximal Frequent Itemset)
- 其直接超集都不是频繁的频繁项集



频繁项集的紧凑表示

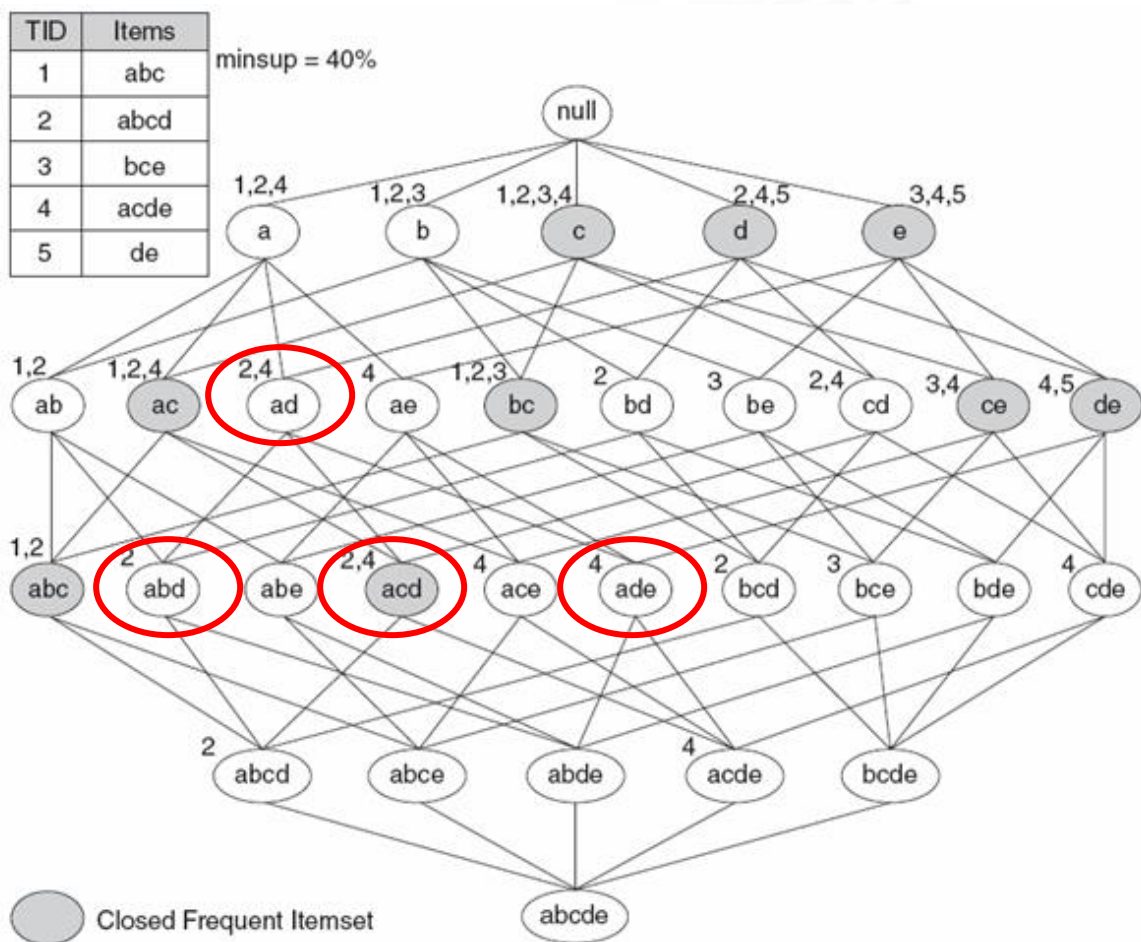
➤ 闭项集(Closed Itemset)

- 项集 X 是闭的，如果它的直接超集都不具有和它相同的支持度计数
- 项集的一种最小表示，同时不丢失支持度信息
- 如果至少存在一个 X 的直接超集，其支持度计数与 X 相同， X 就不是闭的

频繁项集的紧凑表示

➤ 频繁闭项集(Closed Frequent Itemset)

➤ 一个项集是频繁闭项集，如果它是闭的，且它的支持度大于或等于最小支持度阈值



频繁项集的紧凑表示

- 使用频繁闭项集的支持度确定那些非闭的频繁项集的支持度
- 非闭项集的支持度一定与它的某个直接超集中的最大支持度相同

使用频繁闭项集进行支持度计数↵

1: 设 C 是频繁闭项集的集合↵

2: 设 k_{\max} 是频繁闭项集的最大长度↵

3: $F_{k_{\max}} = \{f | f \in C, |f| = k_{\max}\}$ (找出长度为 k_{\max} 的所有频繁项集)↵

4: for $k = k_{\max} - 1$ downto 1 do↵

5: $F_k = \{f | f \subset F_{k+1}, |f| = k\}$ (找出长度为 k 的所有频繁项集)↵

6: for 每个 $f \in F_k$ do↵

7: if $f \notin C$ then↵

8: $f.\text{support} = \max\{f'.\text{support} | f' \in F_{k+1}, f \subset f'\}$ ↵

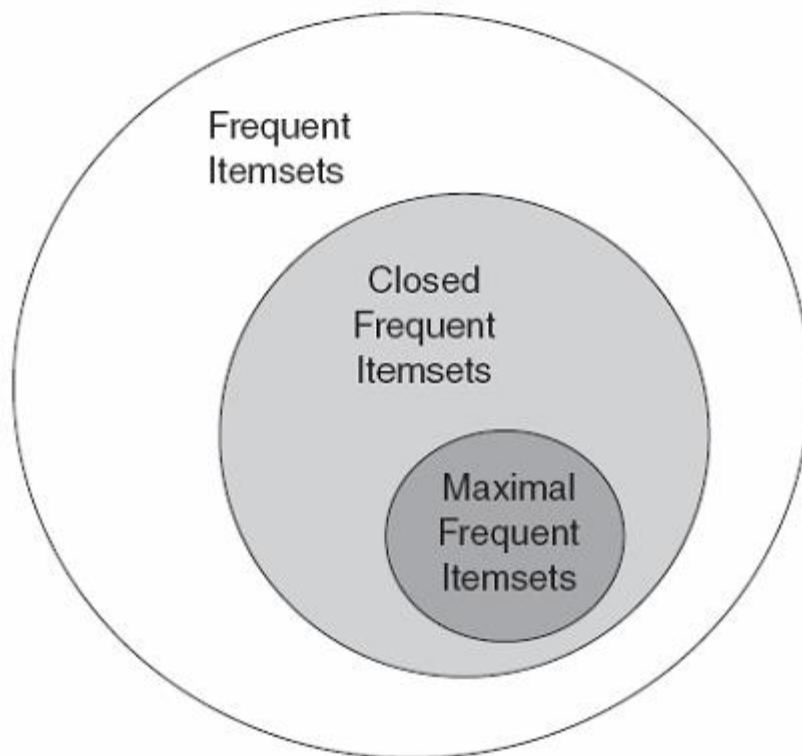
9: end if↵

10: end for↵

11: end for↵

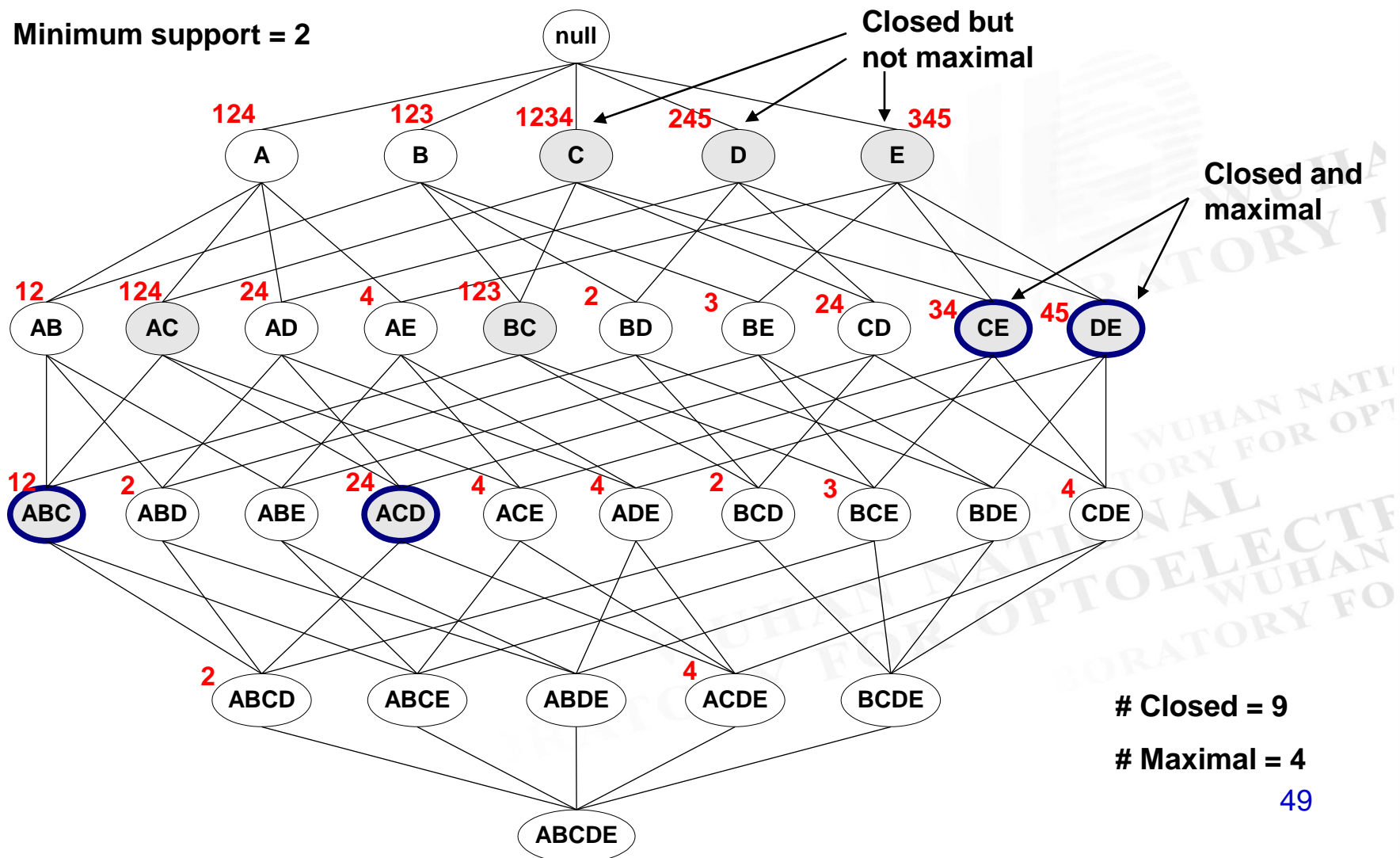
频繁项集的紧凑表示

- 最大频繁项集都是闭的，因为任何最大频繁项集都不可能与其直接超集具有相同的支持度计数



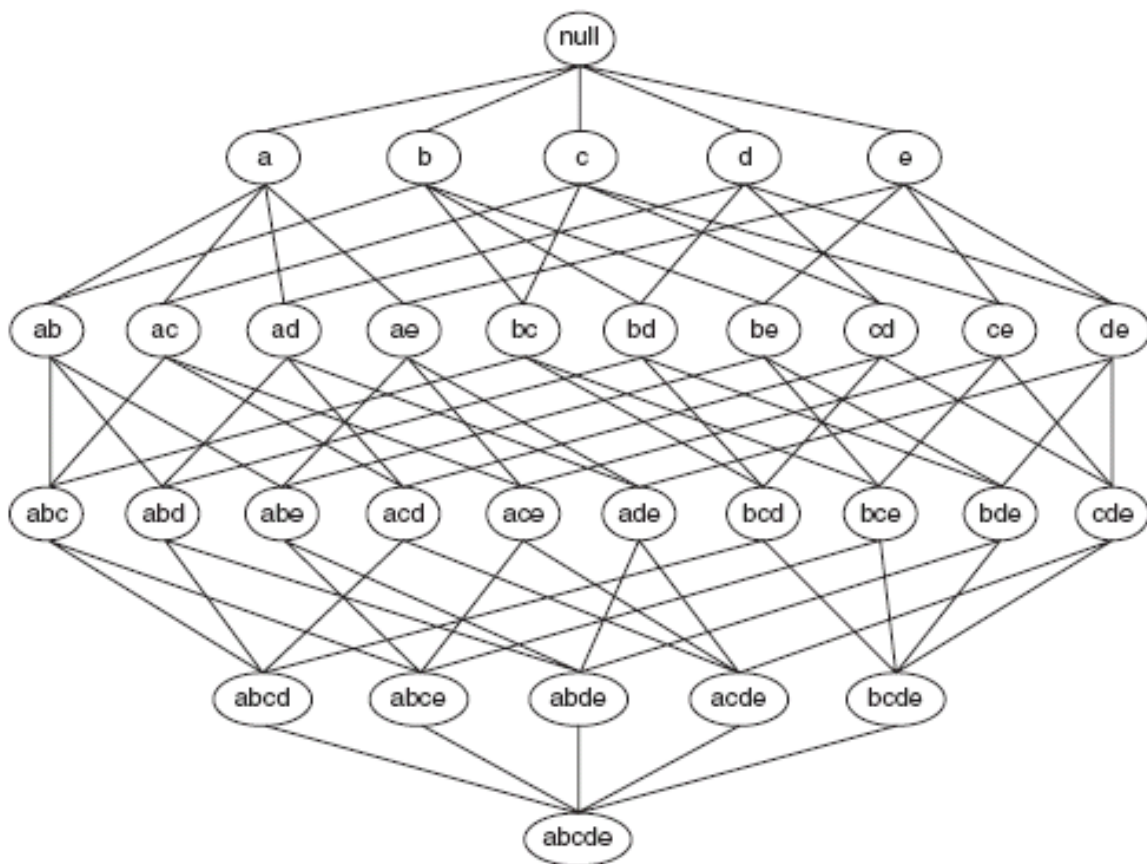
频繁项集的紧凑表示

Minimum support = 2



习题11

➤ 给定如下格结构和事务，标记其中每个结点



Transaction ID	Items Bought
1	{a, b, d, e}
2	{b, c, d}
3	{a, b, d, e}
4	{a, c, d, e}
5	{b, c, d, e}
6	{b, d, e}
7	{c, d}
8	{a, b, c}
9	{a, d, e}
10	{b, d}

M: 最大频繁项集

C: 频繁闭项集

N: 频繁的，非最大或闭项集

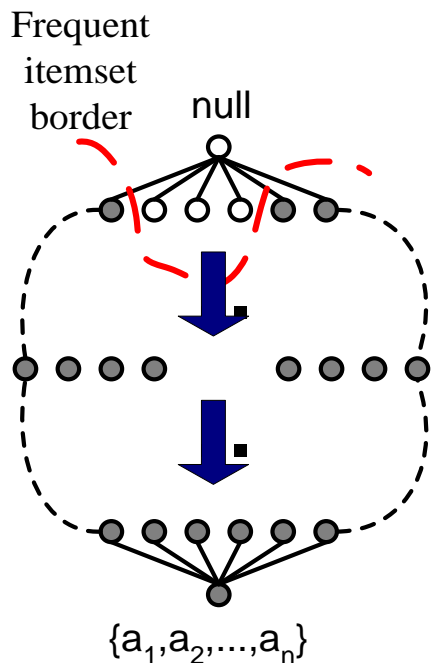
I: 非频繁的

Minsup=30%

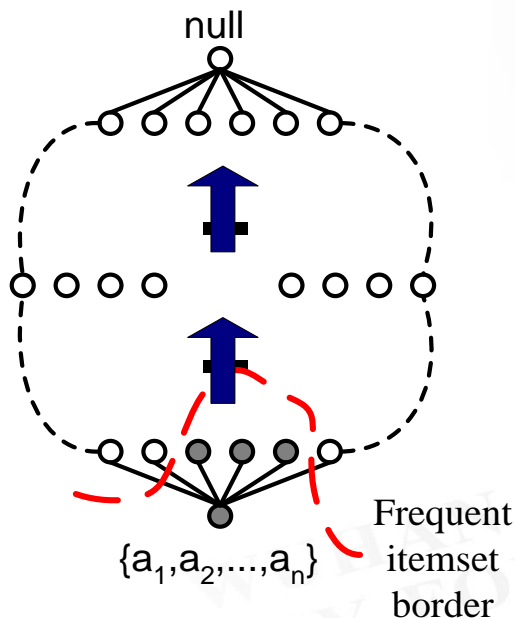


产生频繁项集的其他方法

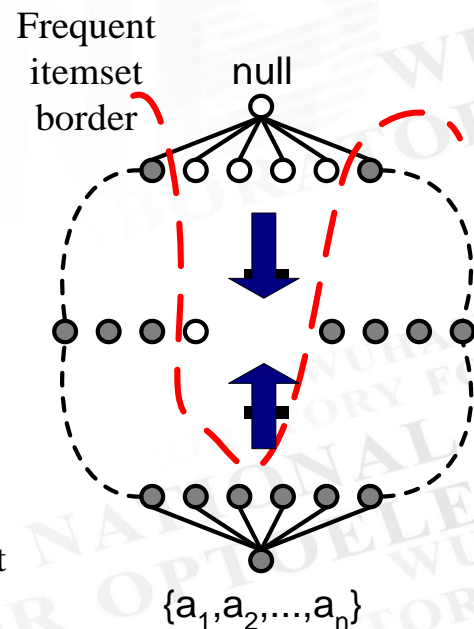
➤ 项集格遍历



(a) General-to-specific



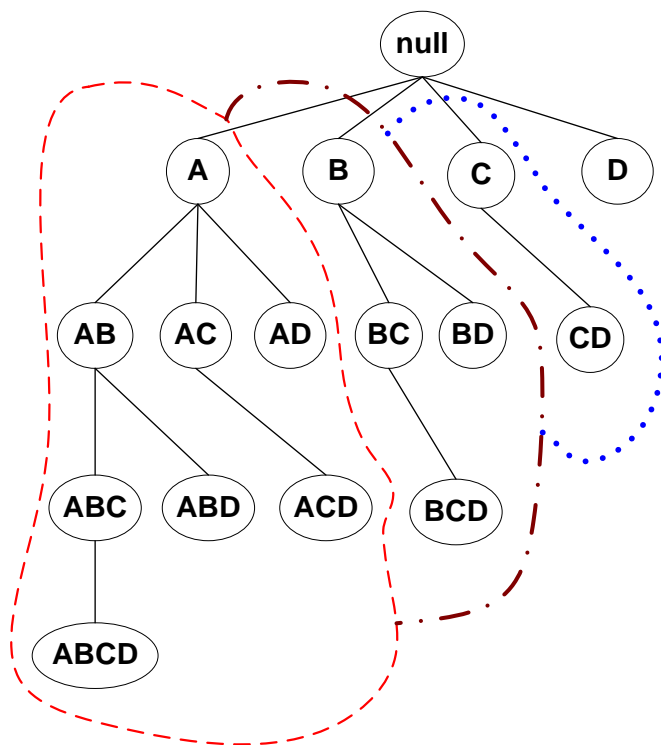
(b) Specific-to-general



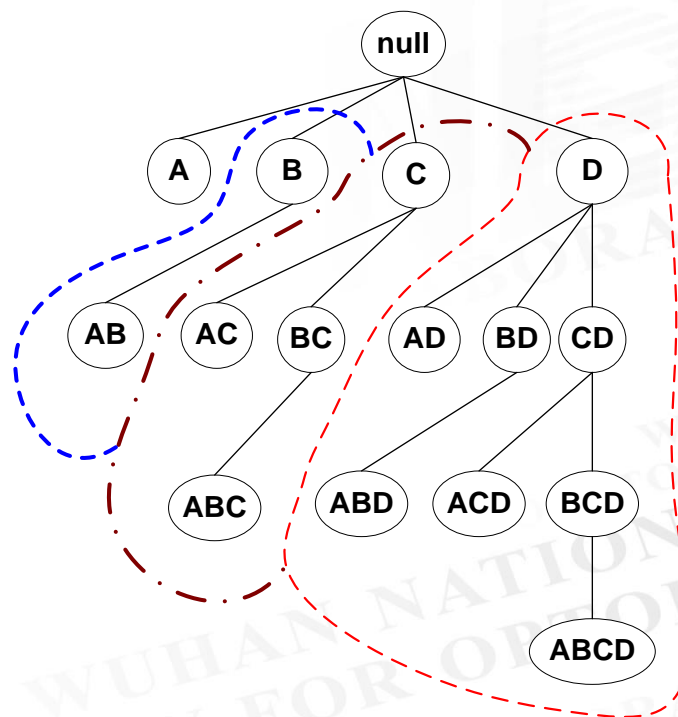
(c) Bidirectional

产生频繁项集的其他方法

➤ 等价类



(a) Prefix tree



(b) Suffix tree

➤ 宽度优先与深度优先



产生频繁项集的其他方法

➤ 事务数据集的表示

➤ 水平和垂直数据形式

Horizontal
Data Layout

TID	Items
1	A,B,E
2	B,C,D
3	C,E
4	A,C,D
5	A,B,C,D
6	A,E
7	A,B
8	A,B,C
9	A,C,D
10	B

Vertical Data Layout

A	B	C	D	E
1	1	2	2	1
4	2	3	4	3
5	5	4	5	6
6	7	8	9	
7	8	9		
8	10			
9				

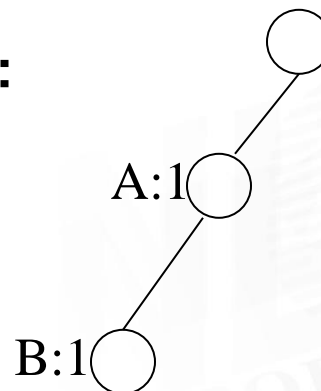
FP增长算法

- **FP树**是一种输入数据的压缩表示方法，通过逐个读入事务，并把每个事务映射到**FP树**中的一条路径来构造
 - 事务中相同的项意味着路径的重叠
- 一旦构建了**FP树**，可直接从中提取频繁项集，而不必重复扫描数据

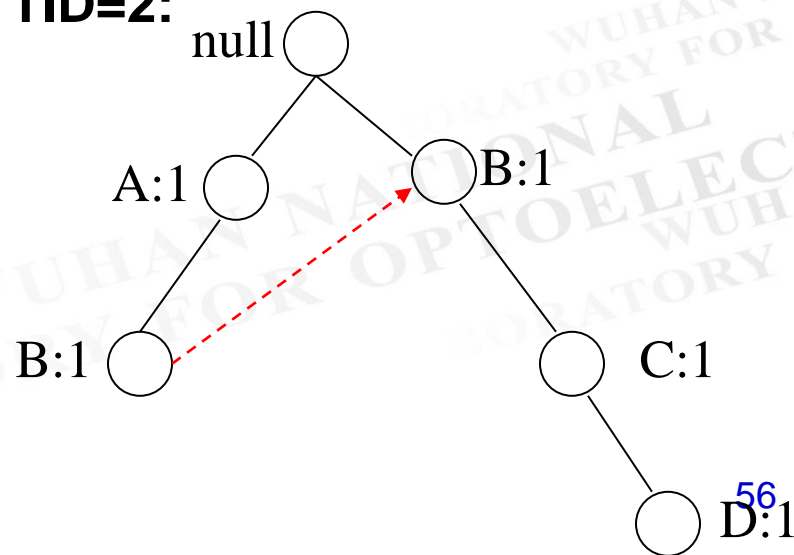
FP增长算法

After reading TID=1:

TID	Items
1	{A,B}
2	{B,C,D}
3	{A,C,D,E}
4	{A,D,E}
5	{A,B,C}
6	{A,B,C,D}
7	{B,C}
8	{A,B,C}
9	{A,B,D}
10	{B,C,E}



After reading TID=2:



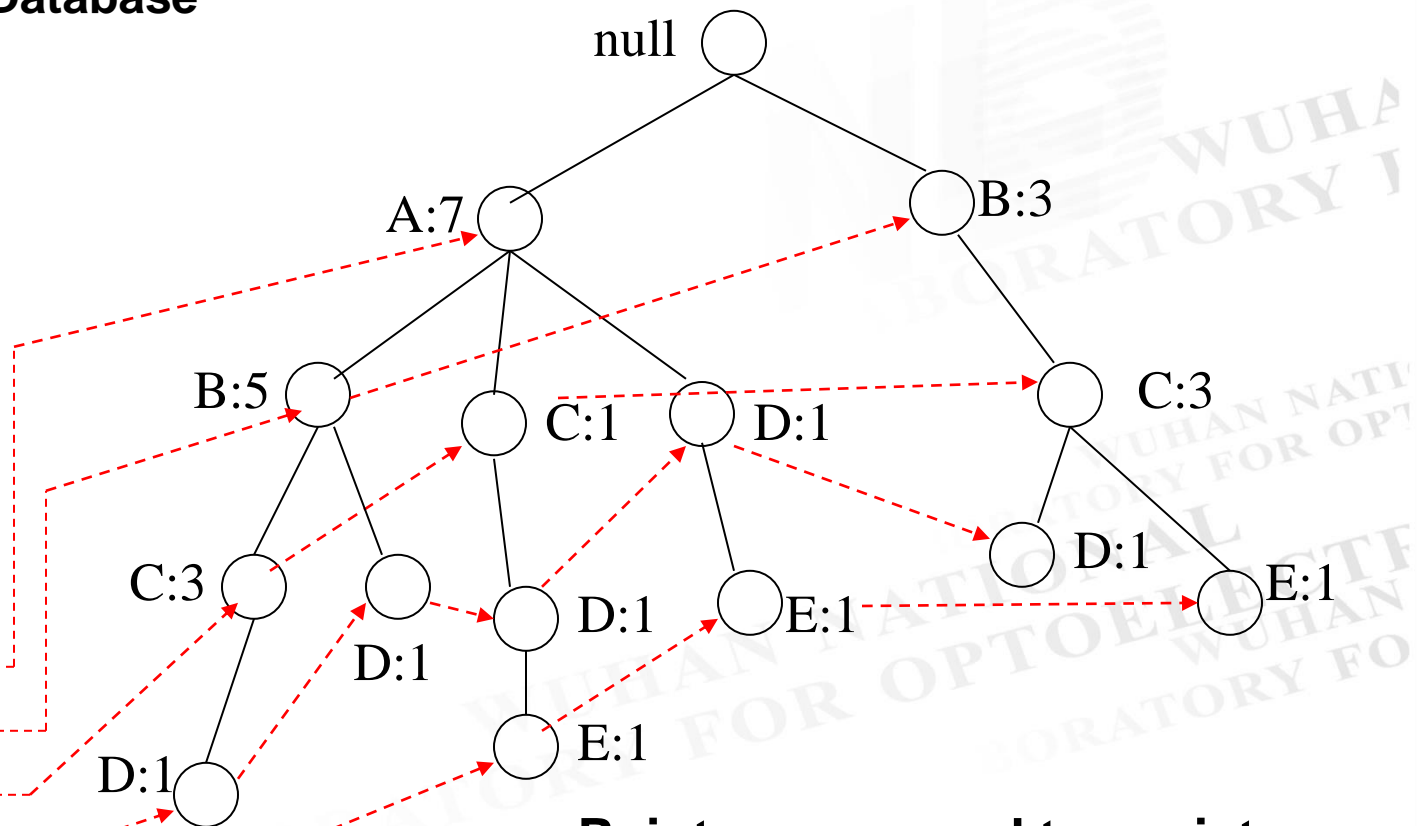
FP增长算法

TID	Items
1	{A,B}
2	{B,C,D}
3	{A,C,D,E}
4	{A,D,E}
5	{A,B,C}
6	{A,B,C,D}
7	{B,C}
8	{A,B,C}
9	{A,B,D}
10	{B,C,E}

Transaction Database

Header table

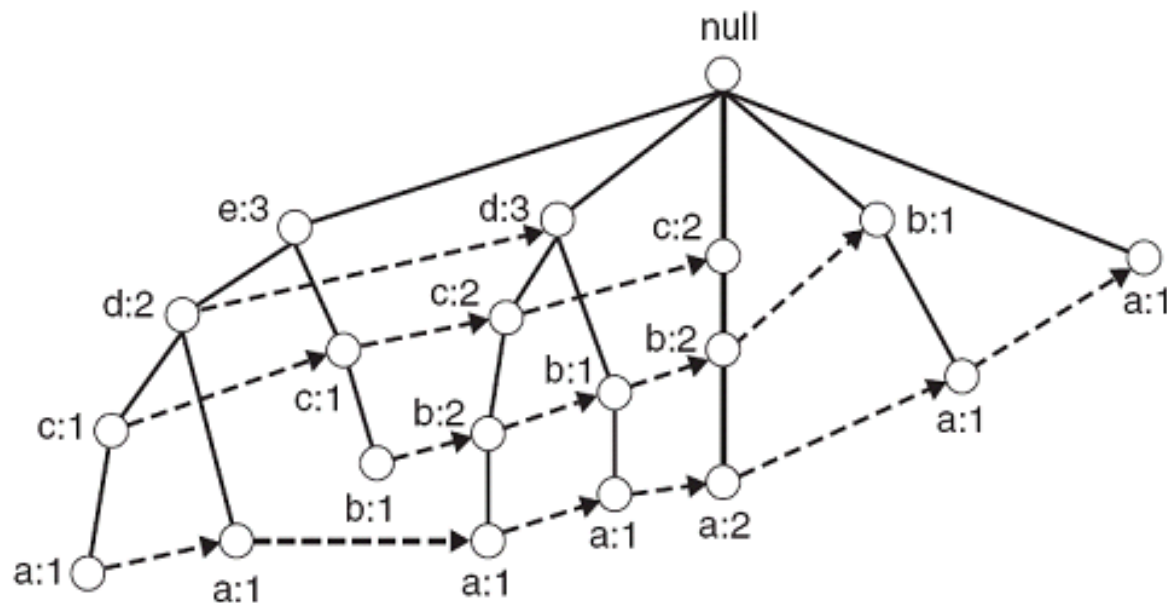
Item	Pointer
A	
B	
C	
D	
E	



Pointers are used to assist frequent itemset generation

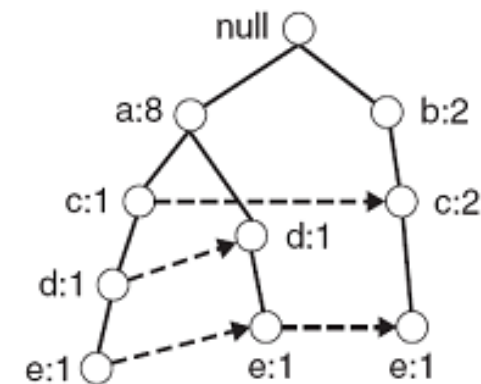
FP增长算法

- **FP树的大小通常比未压缩的数据小**
- **最坏情况下，需要附加的空间为每个项存放结点间的指针和计数，存储空间更大**
- **FP树的大小也依赖于如何对项排序**

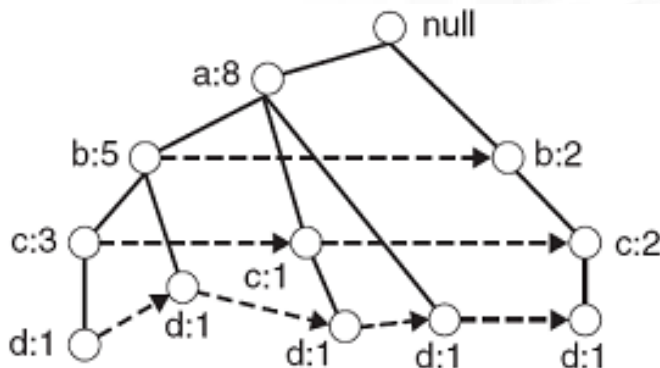


FP增长算法

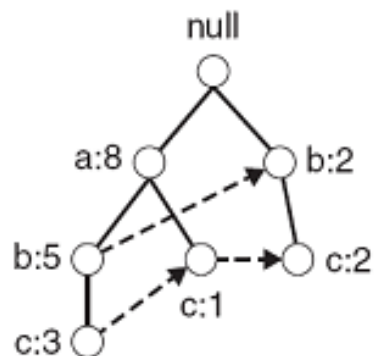
➤ **FP增长**是一种自底向上方式探索树，产生频繁项集的算法



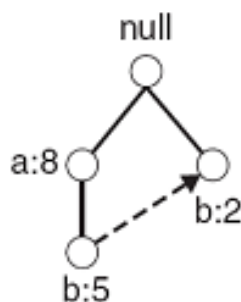
(a) Paths containing node e



(b) Paths containing node d



(c) Paths containing node c



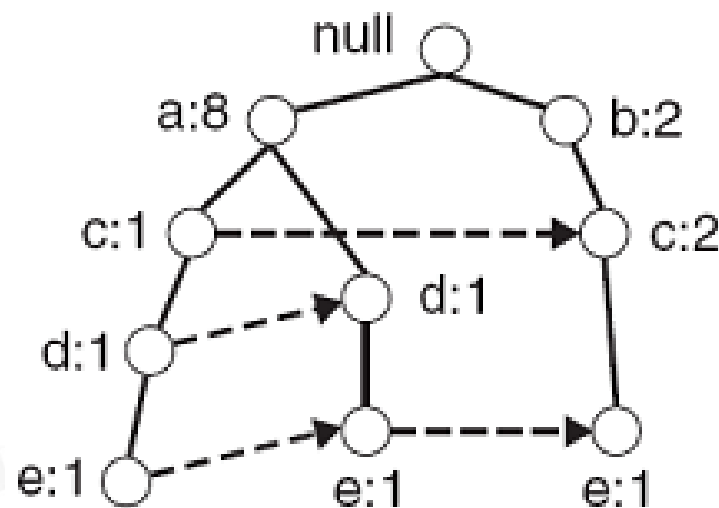
(d) Paths containing node b



(e) Paths containing node a

FP增长算法

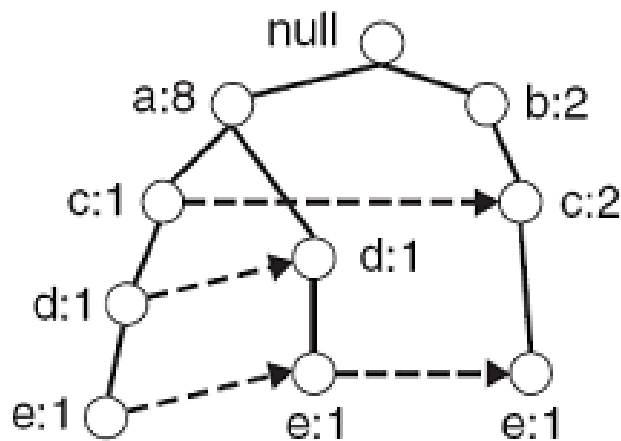
- 考虑发现所有以**e**结尾的频繁项集任务
- 1. 收集包含**e**结点的所有路径，称为前缀路径
- 2. 若最小支持度为2，则通过统计与结点**e**相关联的支持度计数，可知{**e**}是频繁项集



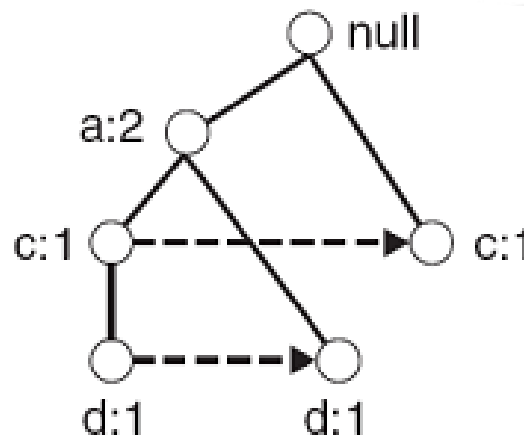
(a) Prefix paths ending in e

FP增长算法

- 3. 以de, ce, be, ae结尾的项集是否是频繁项集？
 - 将前缀路径转化成条件FP树
 - 用于发现以某特定后缀结尾的频繁项集

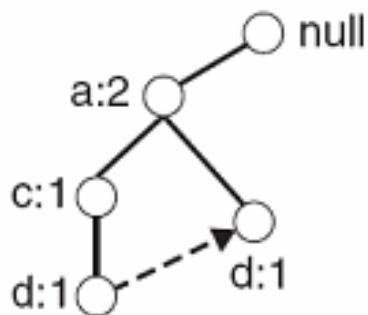


(a) Prefix paths ending in e

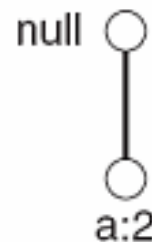


(b) Conditional FP-tree for e

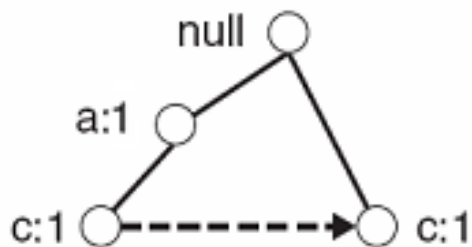
FP增长算法



(c) Prefix paths ending in de



(d) Conditional FP-tree for de



(e) Prefix paths ending in ce



(f) Prefix paths ending in ae

关联模式的评估

- 关联分析算法具有产生大量模式的潜在能力
 - 其中许多是不感兴趣的和冗余的
 - 如果 $\{A, B, C\} \rightarrow \{D\}$ 和 $\{A, B\} \rightarrow \{D\}$ 具有同样的支持度和置信度，则产生了冗余
- 在原先的模式评估中，仅采用支持度和置信度作为评价准则
- 感兴趣度量可以用来删除和排序模式

关联模式的评估

➤ 客观兴趣度度量

➤ 通过统计论据建立

➤ 主观兴趣度度量

➤ 需要领域专家的大量先验知识

➤ {黄油}->{面包}不是有趣的，即使它具有很高的支持度和置信度

关联模式的评估

➤ 兴趣度的客观度量

- 是一种评估关联模式质量的数据驱动的方法，不依赖于领域知识，只需最小限度用户的输入信息

➤ 相依表

Contingency table for $X \rightarrow Y$

	Y	/Y	
X	f_{11}	f_{10}	f_{1+}
/X	f_{01}	f_{00}	f_{0+}
	f_{+1}	f_{+0}	$ T $

f_{11} : support of X and Y

f_{10} : support of \underline{X} and \overline{Y}

f_{01} : support of \overline{X} and \underline{Y}

f_{00} : support of \overline{X} and \overline{Y}

用来定义多种度量

u support, confidence, lift, Gini, J-measure, etc.

关联模式的评估

➤ 置信度的局限性

	Coffee	/Coffee	
Tea	150	50	200
/Tea	650	150	800
	800	200	1000

Association Rule: Tea \rightarrow Coffee

Confidence = $P(\text{Coffee}|\text{Tea}) = 0.75$

but $P(\text{Coffee}) = 0.8$

\Rightarrow 虽然置信度很高，但却是误导

$\Rightarrow P(\text{Coffee}|\text{Tea}) = 0.8125$

置信度的缺陷在于忽略了
规则后件中项集的支持度

关联模式的评估

➤ 提升度(lift)

- 计算规则置信度和规则后件中项集的支持度之间的比率

$$lift(A \rightarrow B) = \frac{c(A \rightarrow B)}{b(B)} \quad Lift = \frac{P(Y | X)}{P(Y)}$$

- 对于二元变量，提升度等效为兴趣因子

$$I(A, B) = \frac{s(A, B)}{s(A) \times s(B)} = \frac{Nf_{11}}{f_{1+}f_{+1}}$$

关联模式的评估

➤ 兴趣因子比较模式的频率与统计独立假设下计算的基准频率

➤ 基准频率 $\frac{f_{1+}}{N} \times \frac{f_{+1}}{N}$

➤ 模式频率 $\frac{f_{11}}{N}$

	Coffee	/Coffee	
Tea	150	50	200
/Tea	650	150	800
	800	200	1000

$$1000 \times 150 / (200 \times 800) = 0.9375$$

– $I(A,B) = 1 \Rightarrow$ 统计独立

– $I(A,B) > 1 \Rightarrow$ 正相关

– $I(A,B) < 1 \Rightarrow$ 负相关

关联模式的评估

➤ 相关分析

$$\phi = \frac{f_{11}f_{00} - f_{01}f_{10}}{\sqrt{f_{1+}f_{+1}f_{0+}f_{+0}}}$$

➤ 相关度的值从-1到+1，0表示统计独立

➤ 相关分析的局限性

➤ 把项在事务中同时出现和同时不出现视为同等重要

➤ 更适合分析对称的二元变量

关联模式的评估

➤ **IS**是一种用于处理非对称二元变量的度量

$$IS(A, B) = \sqrt{I(A, B) \times s(A, B)} = \frac{s(A, B)}{\sqrt{s(A)s(B)}}$$

- 当模式的兴趣因子和支持度都很大时，**IS**也很大
- 等价于二元变量的余弦度量
- 也可以表示为从一对二元变量中提取出的关联规则的置信度的几何均值

$$IS(A, B) = \sqrt{\frac{s(A, B)}{s(A)} \times \frac{s(A, B)}{s(B)}} = \sqrt{c(A \rightarrow B) \times c(B \rightarrow A)}$$

关联模式的评估

- 客观度量的一致性
- 客观度量的性质
- 好的度量**M**必须满足三点性质:
- 当**A**和**B**统计独立时, $M(A,B) = 0$
- 当**P(A)**和**P(B)**保持不变时, $M(A,B)$ 随**P(A,B)**单调递增
- 当**P(A,B)**和**P(B)**(或**P(A)**)保持不变时, $M(A,B)$ 随**P(A)**(或**P(B)**)单调递减

关联模式的评估

10个相依表

Example	f_{11}	f_{10}	f_{01}	f_{00}
E1	8123	83	424	1370
E2	8330	2	622	1046
E3	9481	94	127	298
E4	3954	3080	5	2961
E5	2886	1363	1320	4431
E6	1500	2000	500	6000
E7	4000	2000	1000	3000
E8	4000	2000	2000	2000
E9	1720	7121	5	1154
E10	61	2483	4	7452

采用不同的度量对相依表排序

#	ϕ	λ	α	Q	Y	κ	M	J	G	s	c	L	V	I	IS	PS	F	AV	S	ζ	K
E1	1	1	3	3	3	1	2	2	1	3	5	5	4	6	2	2	4	6	1	2	5
E2	2	2	1	1	1	2	1	3	2	2	1	1	1	8	3	5	1	8	2	3	6
E3	3	3	4	4	4	3	3	8	7	1	4	4	6	10	1	8	6	10	3	1	10
E4	4	7	2	2	2	5	4	1	3	6	2	2	2	4	4	1	2	3	4	5	1
E5	5	4	8	8	8	4	7	5	4	7	9	9	9	3	6	3	9	4	5	6	3
E6	6	6	7	7	7	7	6	4	6	9	8	8	7	2	8	6	7	2	7	8	2
E7	7	5	9	9	9	6	8	6	5	4	7	7	8	5	5	4	8	5	6	4	4
E8	8	9	10	10	10	8	10	10	8	4	10	10	10	9	7	7	10	9	8	7	9
E9	9	9	5	5	5	9	9	7	9	8	3	3	3	7	9	9	3	7	9	9	8
E10	10	8	6	6	6	10	5	9	10	10	6	6	5	1	10	10	5	1	10	10	7

关联模式的评估

➤ 反演性

A	B
1	0
0	0
0	0
0	0
0	1
0	0
0	0
0	0
0	0
1	0

(a)

C	D
0	1
1	1
1	1
1	1
1	0
1	1
1	1
1	1
1	1
0	1

(b)

E	F
0	0
1	0
1	0
1	0
1	0
1	1
1	0
1	0
1	0
1	0
0	0

(c)

如果度量在反演操作下是不变的，则对(C,D)的度量值和对(A,B)的度量值应当向等

关联模式的评估

- 定义：如果交换频度计数 f_{11} 和 f_{00} ， f_{10} 和 f_{01} 后，它的值不变，则称客观度量 M 在反演操作下是不变的
- 反演操作下保持不变的度量，不适合分析非对称的二元数据

关联模式的评估

- 零加性
- 向数据集中添加不相关数据的过程就是所谓的“零加”操作
- 如果增加 f_{00} 而保持相依表中所有其他的频度不变并不影响 M 的值，则度量 M 在零加操作下是不变的
- 对文档分析或购物篮分析时，期望度量在零加操作下保持不变

关联模式的评估

➤ 缩放不变性

	男	女	
高	30	20	50
低	40	10	50
	70	30	100

	男	女	
高	60	60	120
低	80	30	110
	140	90	230

➤ 客观度量 M 在行/列缩放操作下是不变的，如果 $M(T)=M(T')$ ，其中 T 是频度计数为 $[f_{11};f_{10};f_{01};f_{00}]$ 的相依表， T' 是频度计数为 $[k_1k_3f_{11};k_2k_3f_{10};k_1k_4f_{01};k_2k_4f_{00}]$ 的相依表

➤ k_1, k_2, k_3, k_4 是正常量

关联模式的评估

➤ 多个二元变量的度量

c	b	/b	
a	f_{111}	f_{101}	f_{1+1}
/a	f_{011}	f_{001}	f_{0+1}
	f_{+11}	f_{+01}	f_{++1}

/c	b	/b	
a	f_{110}	f_{100}	f_{1+0}
/a	f_{010}	f_{000}	f_{0+0}
	f_{+10}	f_{+00}	f_{++0}

关联模式的评估

➤ 辛普森悖论

➤ 隐藏的变量可能会导致观察到的一对变量之间的联系消失或逆转方向

➤ $c(\text{HDTV} \rightarrow \text{健身器}) = 99/180 = 55\%$

➤ $c(/ \text{HDTV} \rightarrow \text{健身器}) = 54/120 = 45\%$

➤ 购买了HDTV的顾客
更有可能购买健身器？

买HDTV	买健身器		
	是	否	
是	99	81	180
否	54	66	120
	153	147	300

关联模式的评估

➤ 对于大学生

➤ $c(\text{HDTV} \rightarrow \text{健身器}) = 1/10 = 10\%$

➤ $c(/ \text{HDTV} \rightarrow \text{健身器}) = 4/34 = 11.8\%$

➤ 对于在职人员

➤ $98/170 = 57.7\%$

➤ $50/86 = 58.1\%$

顾客组	买HDTV	买健身器		总数
		是	否	
大学生	是	1	9	10
	否	4	30	34
在职人员	是	98	72	170
	否	50	36	86

关联模式的评估

- 在组合数据下，**HDTV**与健身器呈正相关
- 在分层数据下，**HDTV**与健身器呈负相关
- 需要适当的分层才能避免因辛普森悖论产生虚假的模式
 - $a/b < c/d$ 并且 $p/q < r/s$
 - 当 $(a+p)/(b+q) > (c+r)/(d+s)$ 时，出现悖论

Airbnb实际案例

- **2008年成立，提供短期住宿租赁服务**
- **2017年首次盈利，目前为全球200个国家和地区，6.5万个城市超过**
- **1.6亿房客提供了300万间客房和1400个城堡的租赁服务**
- **Peer-to-Peer服务提供商，利用大数据平台匹配供需双方**

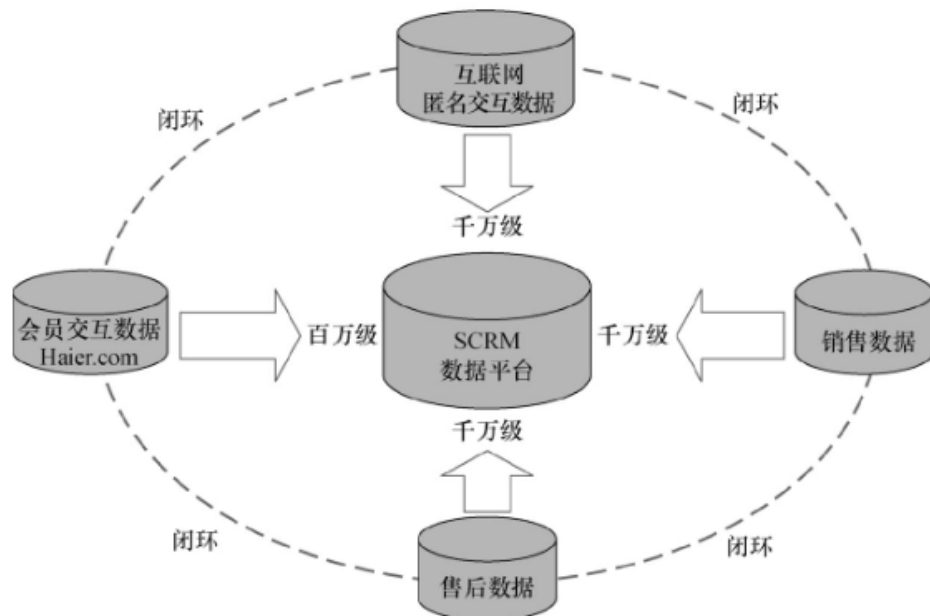
Airbnb实际案例

- 以大数据平台驱动业务
- **Aerosolve:** 帮助房东更好定价的一个机器学习平台
 - 微街区：基于局部特征的动态定价策略，为房东提出针对性的建议



海尔大数据应用

- **SCRM(Social Customer Relationship Management, 社交化用户管理)**
- **与集团12个系统对接**
- **管理1.4亿线下实名用户与19亿线上匿名用户数据**



海尔大数据应用

➤ **SCRM**解决四方面问题:

- 有销售记录，无个性化数据
- 有交易过程，无交互过程
- 有订单数据，无相关数据
- 有营销，缺乏忠诚度

➤ 全方位链接社交行为数据，

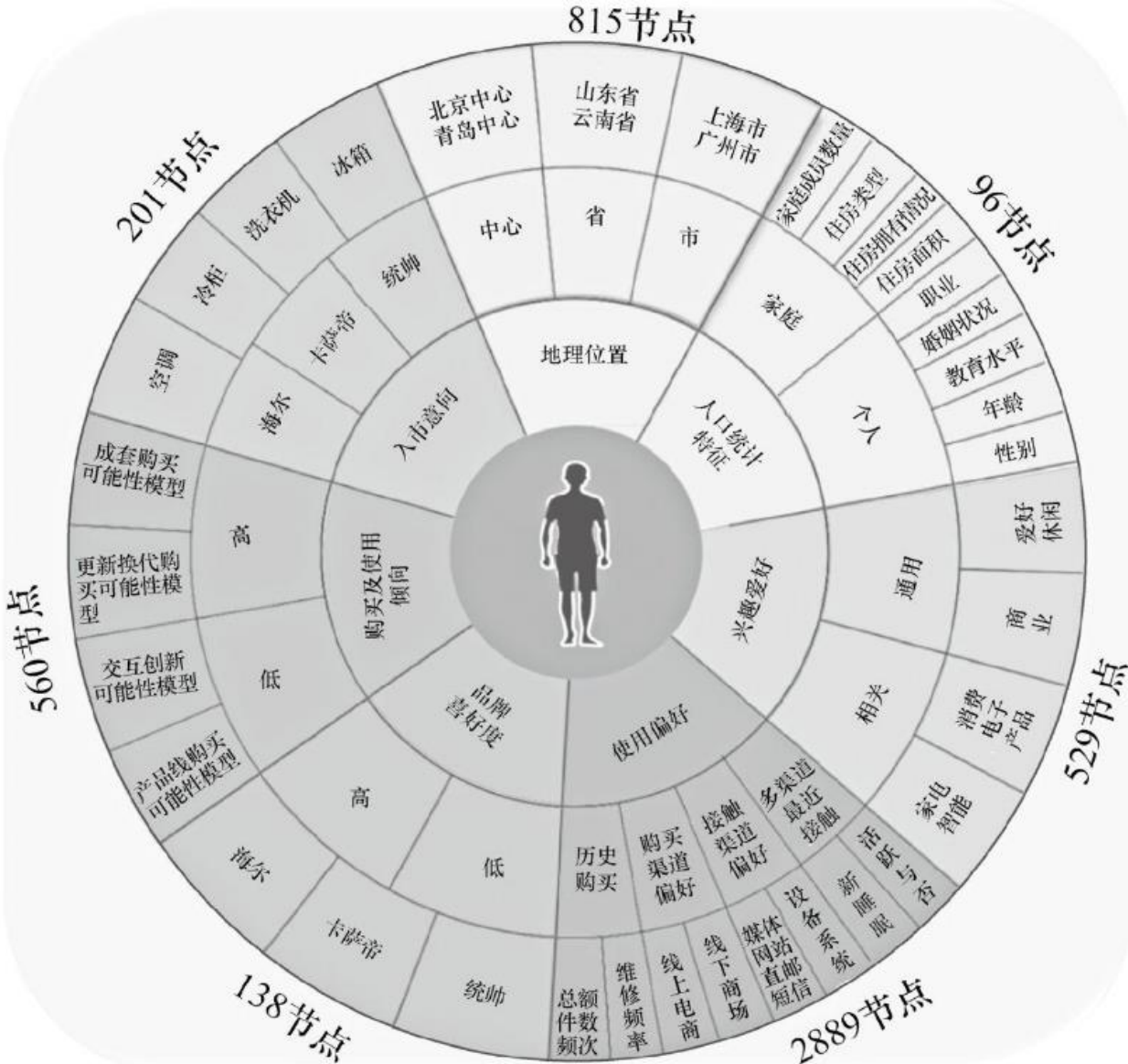
- 挖掘用户画像，满足精准个性需求
- 微博、微信、QQ
- 口碑交互系统，关键词监测

海尔大数据应用

- 挖掘活跃用户
 - 用户画像，用户活跃度模型
- 挖掘用户痛点
 - 对售后数据进行挖掘分析
- 挖掘用户兴趣
 - 对销售数据进行分析，形成爆点
- 用户生活圈挖掘
 - 社交媒体话题挖掘

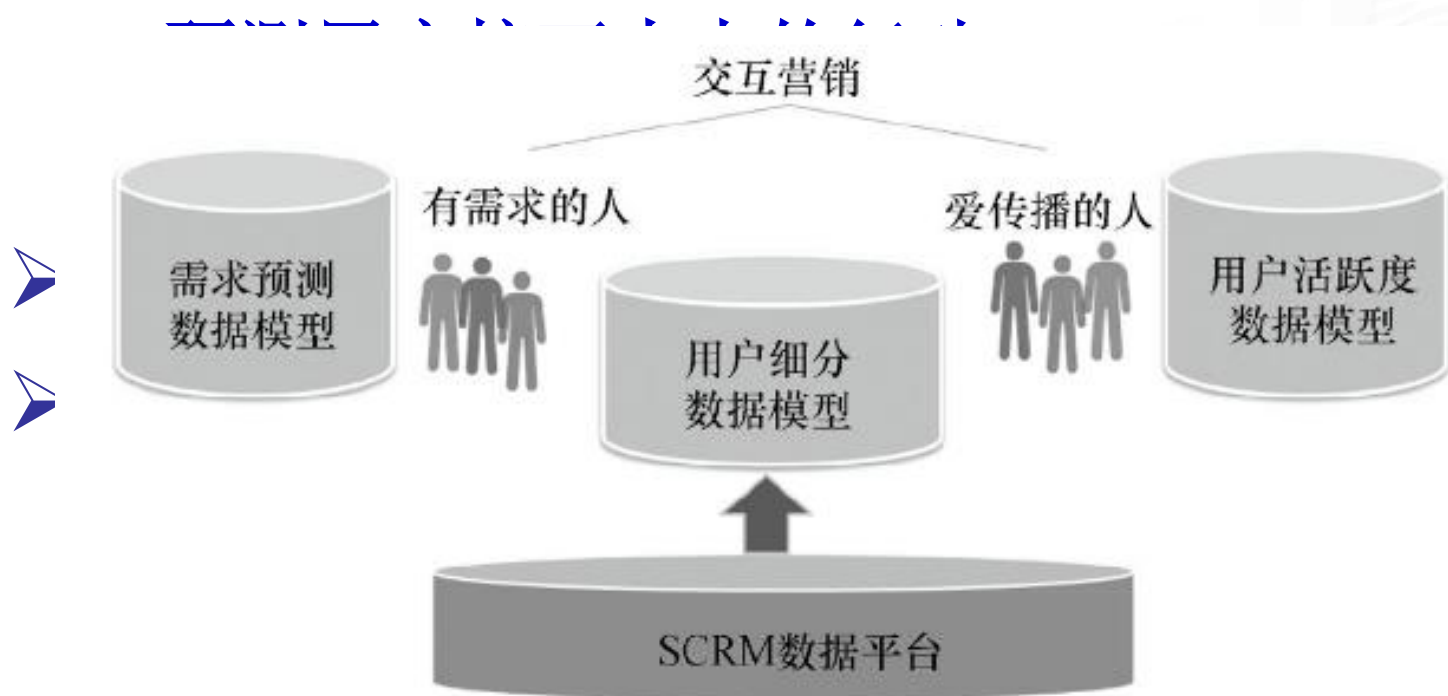
815节点

200



海尔大数据应用

➤ 建立起三大类，十多个数据模型



海尔大数据应用

► 洗衣机需求预测

打分逻辑 判断分类	项目类	分值	判断条件(rules)	时间范围	其他条件	原始数据源
新居/成套	1.大家电购买	3	购买大家电产品(包含洗衣机、冰箱冰柜、热水器、彩电)	≤30日	≥2类	EHUB、HP安装、会员注册产品
新居/成套		2	购买大家电产品(包含洗衣机、冰箱冰柜、热水器、彩电)	≤60日	≥2类	EHUB、HP安装、会员注册产品
新居/成套		1	购买大家电产品(包含洗衣机、冰箱冰柜、热水器、彩电)	≤90日	≥2类	EHUB、HP安装、会员注册产品
新居/成套	2.厨电购买	3	购买厨电类产品	≤30日	子类≥2类	EHUB、HP安装、会员注册产品
新居/成套		1	购买厨电类产品	≤30日	子类=1类	EHUB、HP安装、会员注册产品
新居/成套		2	购买厨电类产品	≤60日	子类≥2类	EHUB、HP安装、会员注册产品
新居/成套		1	购买厨电类产品	≤90日	子类≥2类	EHUB、HP安装、会员注册产品
更新换代	3.已有购买可能性标签	3	逻辑同peter标签:1个月内购买		标签的逻辑判断对象 仅限洗衣机大类产品	HP维修、咨询数据
更新换代		1	逻辑同peter标签:1—6个月内购买		标签的逻辑判断对象 仅限洗衣机大类产品	HP维修、咨询数据/会员数据
更新换代	4.空调维修记录	1	产品有维修记录	≤90日	≥1次洗衣机维修记录	HP维修数据
更新换代		2	产品有维修记录	>90 & ≤365日	1次洗衣机维修记录	HP维修数据
更新换代		3	产品有维修记录	>90 & ≤365日	≥2次洗衣机维修记录	HP维修数据
更新换代	5.空调移机申请	1	空调移机申请,已派单已完成	>30 & ≤365日		HP维修数据
更新换代		2	空调移机申请,已派单已完成	≤30日		HP维修数据
更新换代		3	空调移机申请,已派单已完成			HP维修数据
更新换代	6.小区判断	1	所住小区建成年代在5—13年			节能补贴数据
反馈可能性	7.短信反馈	1	记录回复愿意成为会员			SCRM
反馈可能性	8.官网登录	1	近一个月登录梦享+会员			SCRM
反馈可能性	9.历史记录	-1	之前在破坏性创新活动中已经发送过短信的			通过手动统计

海尔大数据应用

- “大规模生产”向“大规模定制”的转变
- 根据用户开机时间、使用时长等大数据分析自身清洁需求
 - **200**万套自清洁空调为用户节省了**3.2**亿元的清洗费用
- 建立地域差异化的节能模型
 - 为不同地区、不同季节、不同年龄段的用户提供最佳使用温度

海尔大数据应用

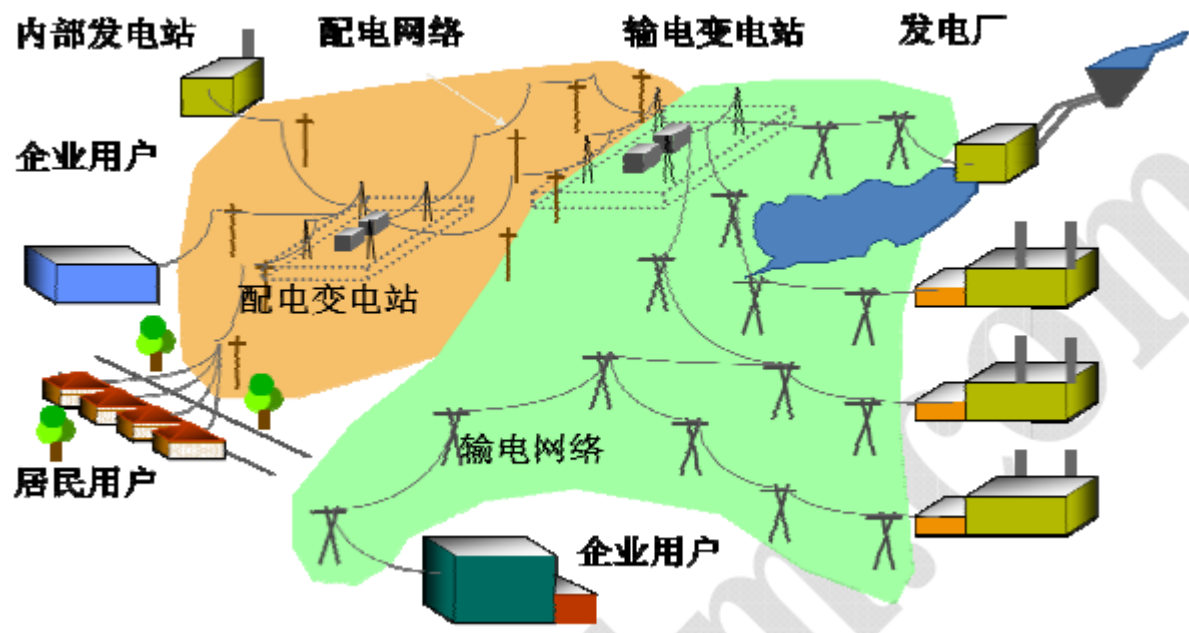
- 将大数据跨界应用到供热、供电领域
 - 与江苏电力公司开展电力削峰合作
 - 降低电网峰谷差近**20%**

- 将空调室内空气大数据用于企业供热小区室温实时监测
 - 辅助适时按需供热

➤ 电力负荷预测

电力负荷预测

- 电力无法大规模存储
- 电力生产和消费必须适配
- 电力负荷预测是电力系统运行调度、生产规划、竞价决策的重要组成部分



电力网结构示意图

电力系统负荷分类

➤ 城市民用负荷

- 具有经常的年增长和明显的季节性波动特点

➤ 商业负荷

- 商业空调、照明、动力，增长迅速，季节性波动

➤ 农村负荷

- 农村居民用电和生产用电，受自然条件影响大
- 大雨可造成北方农业负荷骤降，同时造成南方排涝用电负荷增加

➤ 工业负荷

- 比重最大，与行业特点有关，反映国民经济活跃程度

挖掘目标

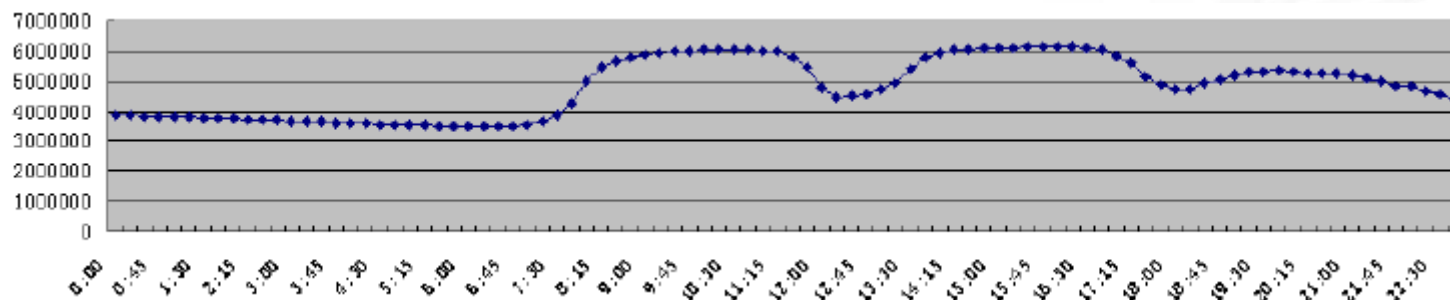
➤ 1. 基于用电特征对用户进行分群

- 电压等级、功率因数、日总用电量、峰用电量、谷用电量、平用电量、**15分钟级别**的用电负荷数据
- **K-means**聚类

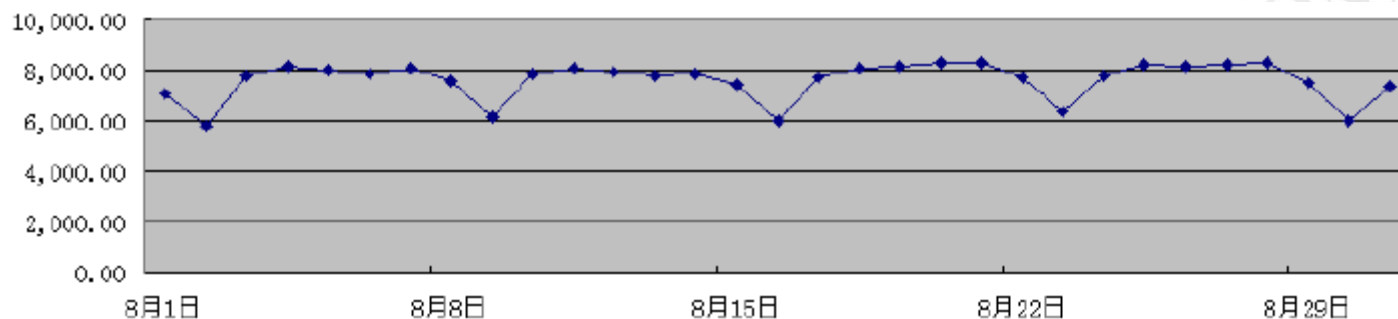
➤ 2. 基于用户分群的结果对用电负荷进行预测

时序预测

- 电力负荷趋势特征
- 1. 实时负荷特征分析

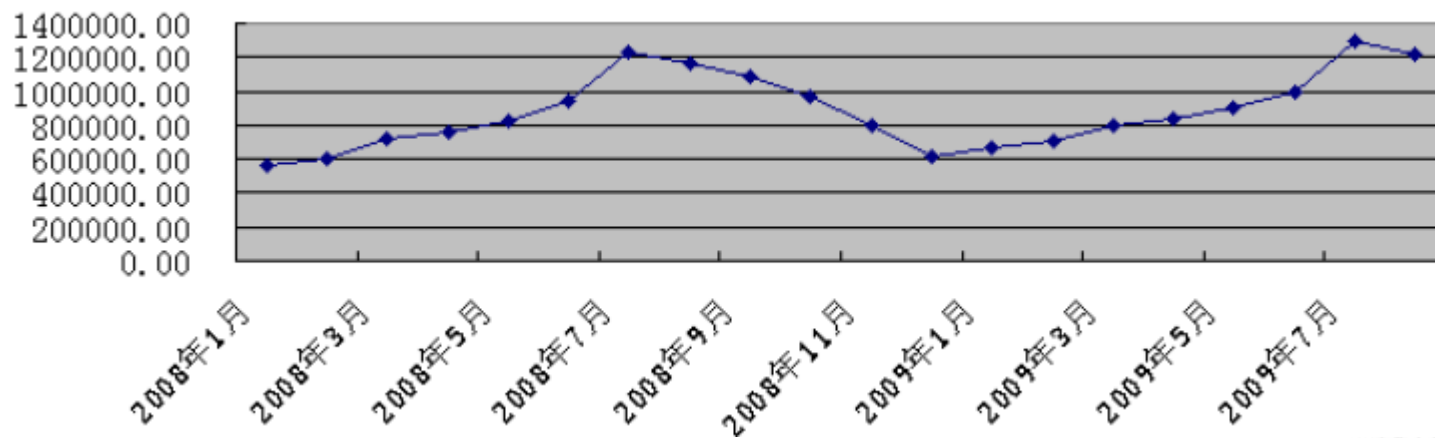


- 2. 日最大负荷特征分析



时序预测

➤ 3. 月最大负荷特征分析



➤ 电力负荷走势具有记忆性

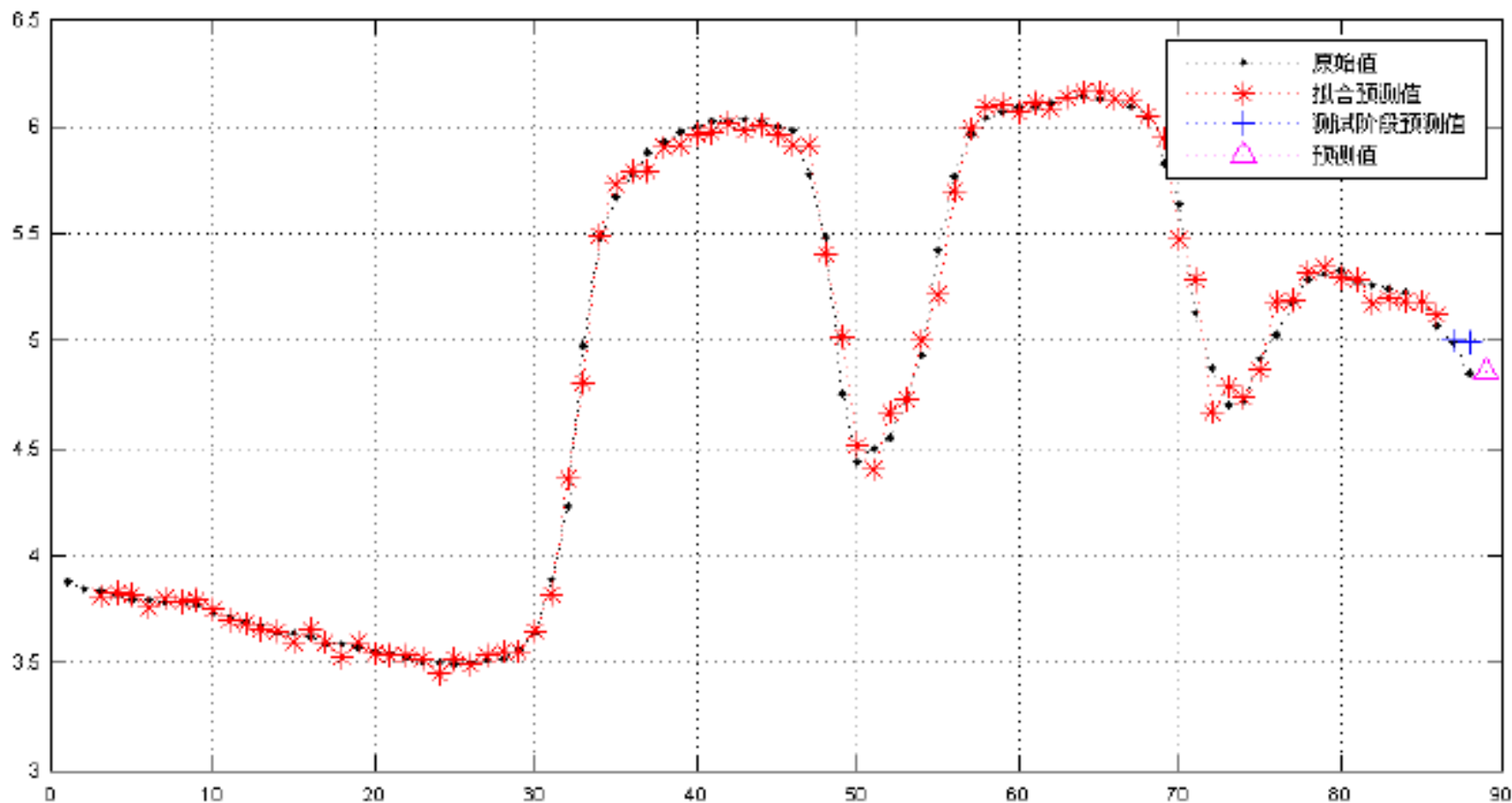
时序预测

- 长期趋势 (long-term trend)
 - 稳定、一般为线性、二次或指数函数
- 季节性波动
- 周期性变动
 - 通常受到经济变动的影响
- 随机影响

用户聚类结果

	样本数量	占比	最大功率	峰总比	平总比	谷总比	负荷率
组 1	372	31.00%	2.2985	0.4131	0.4935	0.0934	0.2509
组 2	1	0.08%	9.6964	0.3040	0.4646	0.2314	0.0037
组 3	3	0.25%	0.1226	0.6491	0.0496	0.3013	0.2792
组 4	86	7.17%	0.1810	0.0000	0.5619	0.0722	0.1602
组 5	20	1.67%	1.2065	0.2289	0.3001	0.4709	0.2167
组 6	8	0.67%	0.6676	0.3332	0.6457	0.0211	0.0633
组 7	6	0.50%	0.5139	0.3558	0.3384	0.3059	0.6090
组 8	23	1.92%	0.4467	0.3160	0.4791	0.2049	0.5352
组 9	18	1.50%	0.6037	0.3826	0.4999	0.7074	0.8237
组 10	4	0.33%	1.5637	0.1110	0.1548	0.7342	0.2102
组 11	38	3.17%	0.5137	0.4223	0.4802	0.0975	0.1051
组 12	7	0.58%	0.2956	0.0000	0.5168	0.1945	0.2440
组 13	52	4.33%	0.5140	0.0000	0.3016	0.4786	0.9634
组 14	68	5.67%	1.0448	0.4415	0.5364	0.0221	0.8076
组 15	35	2.92%	0.4406	0.2965	0.3994	0.3041	0.2258
组 16	50	4.17%	0.7783	0.2999	0.6821	0.0180	0.2534
组 17	23	1.92%	1.5314	0.3497	0.1270	0.5233	0.4867
组 18	301	25.08%	3.3815	0.2143	0.3732	0.4124	0.9002
组 19	51	4.25%	0.0908	0.1900	0.6095	0.2005	0.1122
组 20	34	2.83%	1.9184	0.0000	0.3003	0.5019	0.2340

时序预测



SVR实时负荷预测结果