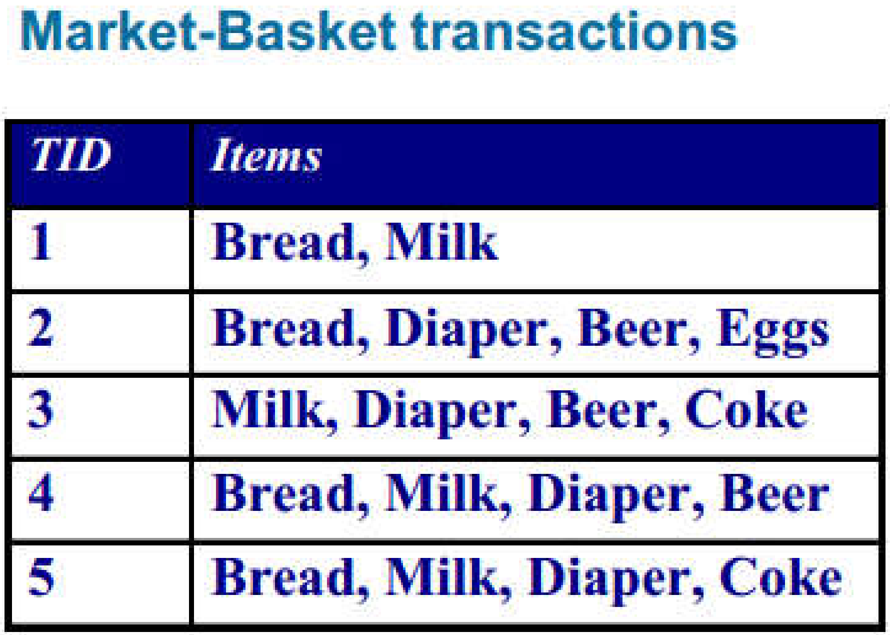
大数据分析 | 关联分析

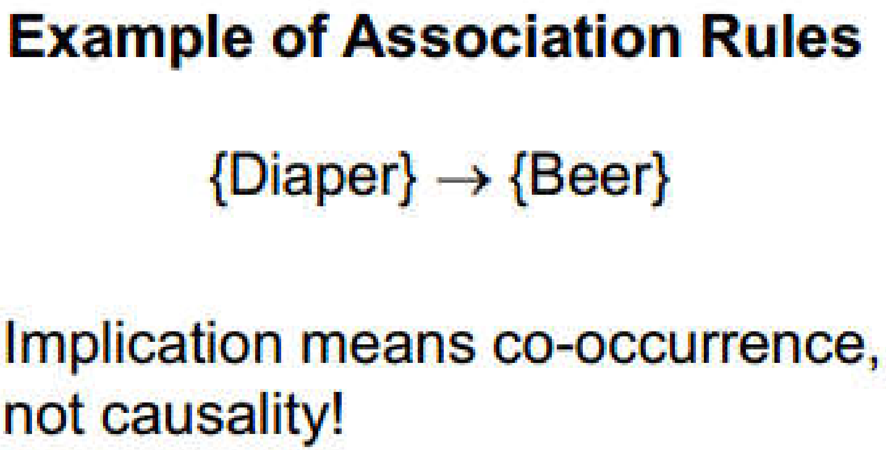
关联规则分析

如果两个或多个变量之间存在一定的关联，那么其中一个变量的状态就能通过其他变量进行预测。



Market-Basket 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



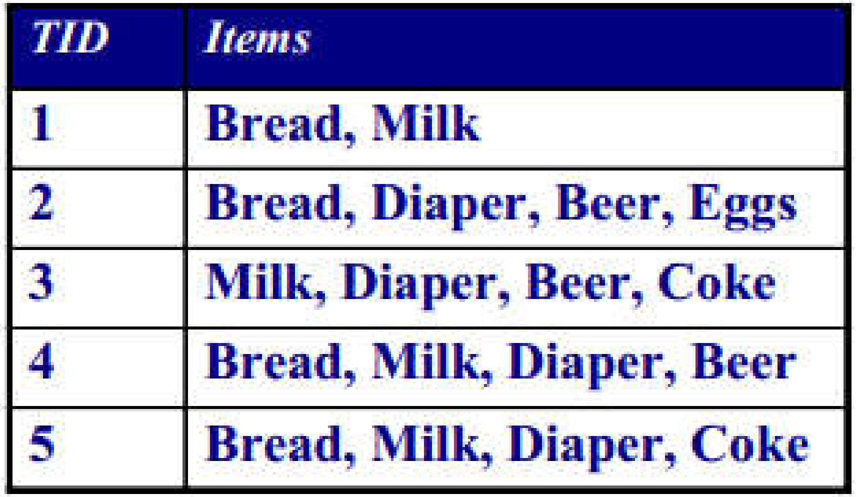
Example of Association Rules

{Diaper} → {Beer}

Implication means co-occurrence, not causality!

频繁项集

- **Itemset**
  - A collection of one or more items
    - Example: {Milk, Bread, Diaper}
  - k-itemset
    - An itemset that contains k items
- **Support count (σ)**
  - Frequency of occurrence of an itemset
  - E.g. σ({Milk, Bread,Diaper}) = 2
- **Support**
  - Fraction of transactions that contain an itemset
  - E.g. s({Milk, Bread, Diaper}) = 2/5



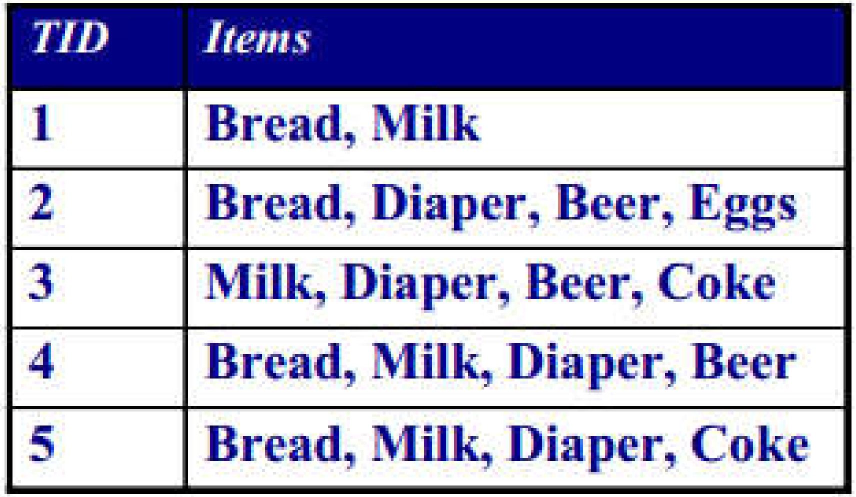
TID	*Items*
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

- **Frequent Itemset**
  - An itemset whose support is greater than or equal to a *minsup* threshold

关联规则

- **Association Rule**
  - An implication expression of the form X → Y, where X and Y are itemsets
  - Example:
    {Milk, Diaper} → {Beer}

- **Rule Evaluation Metrics**
  - Support (s)
    - Fraction of transactions that contain both X and Y
  - Confidence (c)
    - Measures how often items in Y appear in transactions that contain X



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**

Example:

$$\{\text{Milk}, \text{Diaper}\} \Rightarrow \{\text{Beer}\}$$

$$s = \frac{\sigma(\text{Milk}, \text{Diaper}, \text{Beer})}{|\text{T}|} = \frac{2}{5} = 0.4$$

$$c = \frac{\sigma(\text{Milk}, \text{Diaper}, \text{Beer})}{\sigma(\text{Milk}, \text{Diaper})} = \frac{2}{3} = 0.67$$

关联规则发现

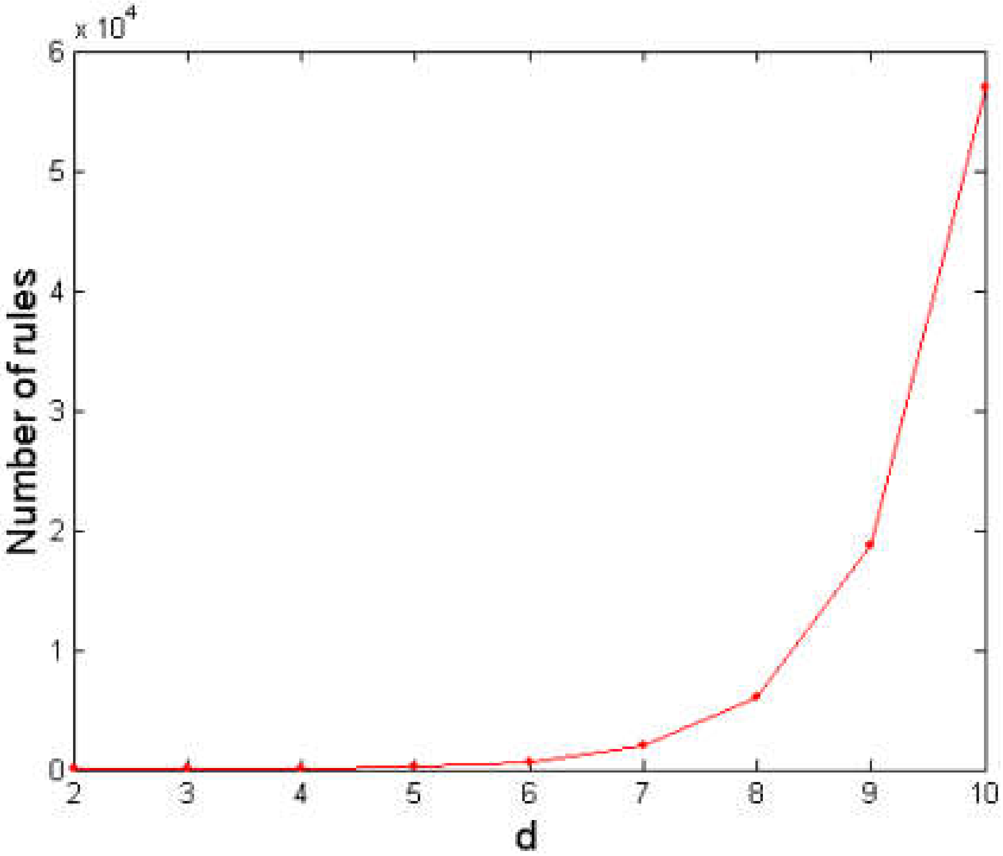
- 给定事务的集合T，找出 **支持度**$\geq minsup$ 并且 **置信度**$\geq minconf$ 的所有规则
- Minsup：支持度阈值
- Minconf：置信度阈值
- 计算每个可能规则的支持度和置信度，代价很高，令人望而却步

关联规则发现（续）

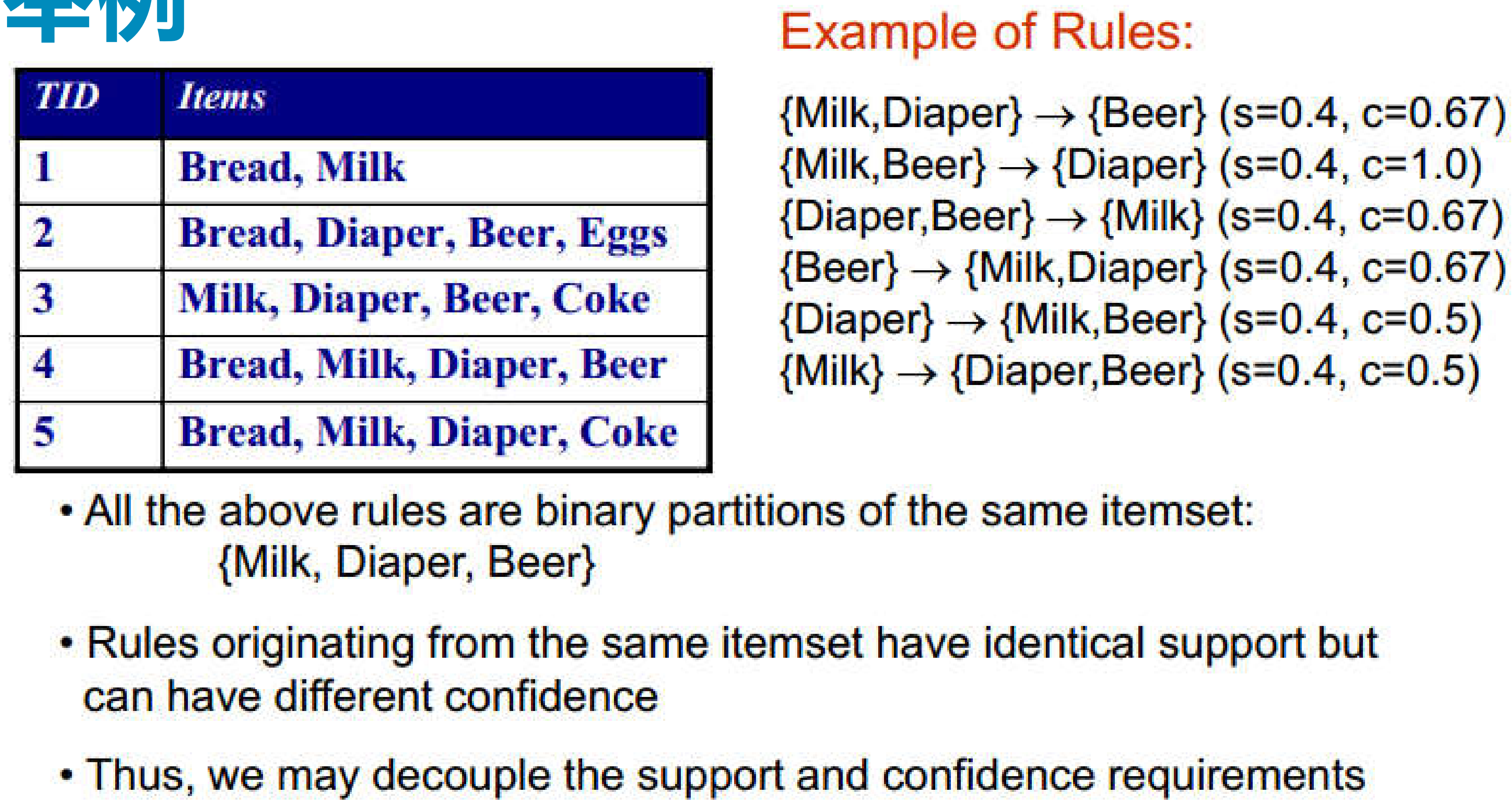
- 从包含d个项的数据集提取的可能规则的总数为

$$3^d - 2^{d+1} + 1$$

- d=6，602条规则



举例



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**

Example of Rules:

{Milk,Diaper} → {Beer} (s=0.4, c=0.67)
{Milk,Beer} → {Diaper} (s=0.4, c=1.0)
{Diaper,Beer} → {Milk} (s=0.4, c=0.67)
{Beer} → {Milk,Diaper} (s=0.4, c=0.67)
{Diaper} → {Milk,Beer} (s=0.4, c=0.5)
{Milk} → {Diaper,Beer} (s=0.4, c=0.5)

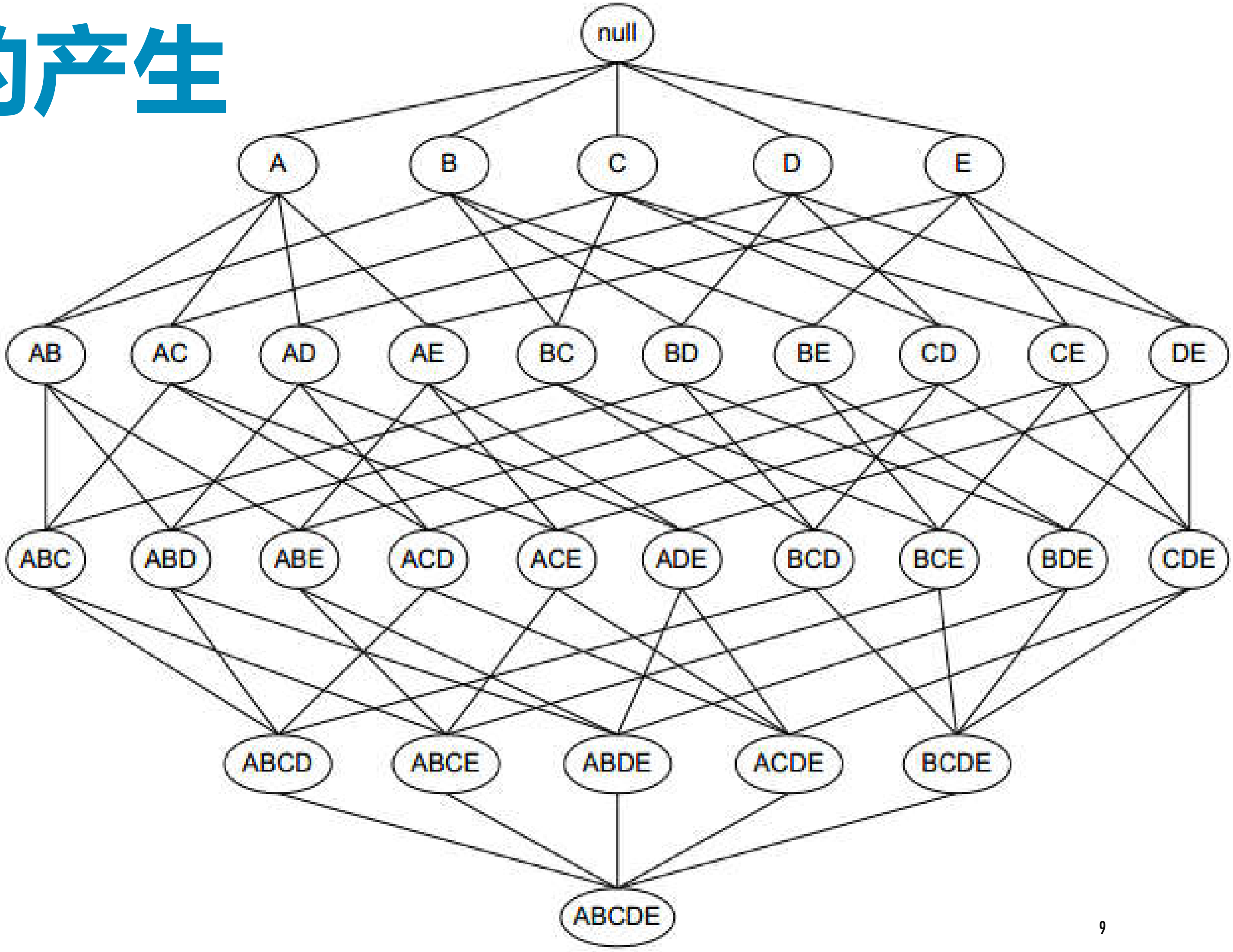
- All the above rules are binary partitions of the same itemset:
  {Milk, Diaper, Beer}
- Rules originating from the same itemset have identical support but can have different confidence
- Thus, we may decouple the support and confidence requirements

关联规则挖掘步骤

- step 1：频繁项集的产生
  - 发现满足最小支持度阈值的所有频繁项集
- step 2：规则的产生
  - 从上一步发现的频繁项集中提取所有高置信度的规则，这些规则称为强规则

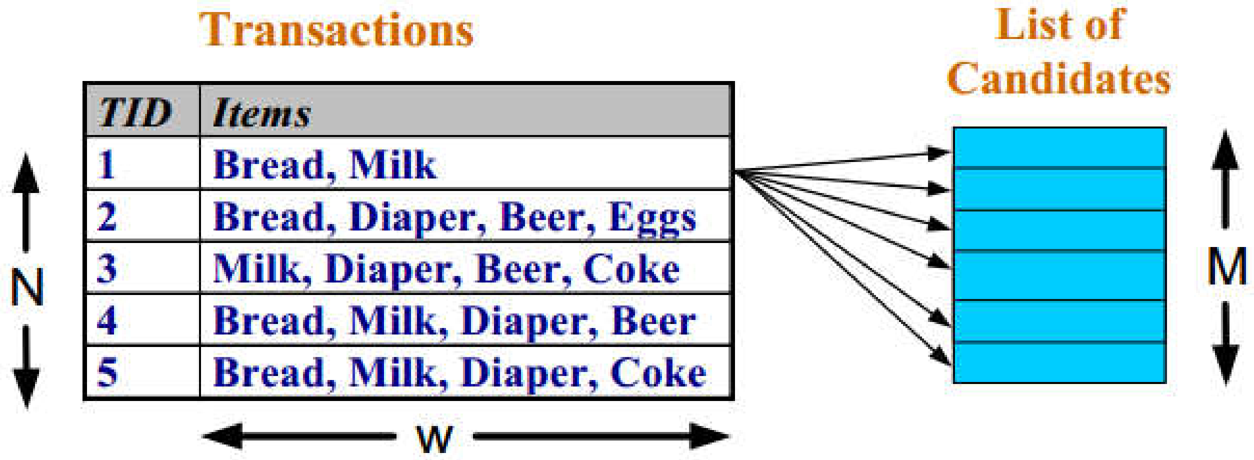
频繁项集的产生

一个包含k个项的数据集可能产生2^k-1个频繁项集



频繁项集

- Each itemset in the lattice is a candidate frequent itemset
- Count the support of each candidate by scanning the database



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

List of Candidates

- Match each transaction against every candidate
- Complexity ~ O(NMw) => Expensive since M = 2^d !!!

APRIORI原理

- Apriori principle:
  - If an itemset is frequent, then all of its subsets must also be frequent

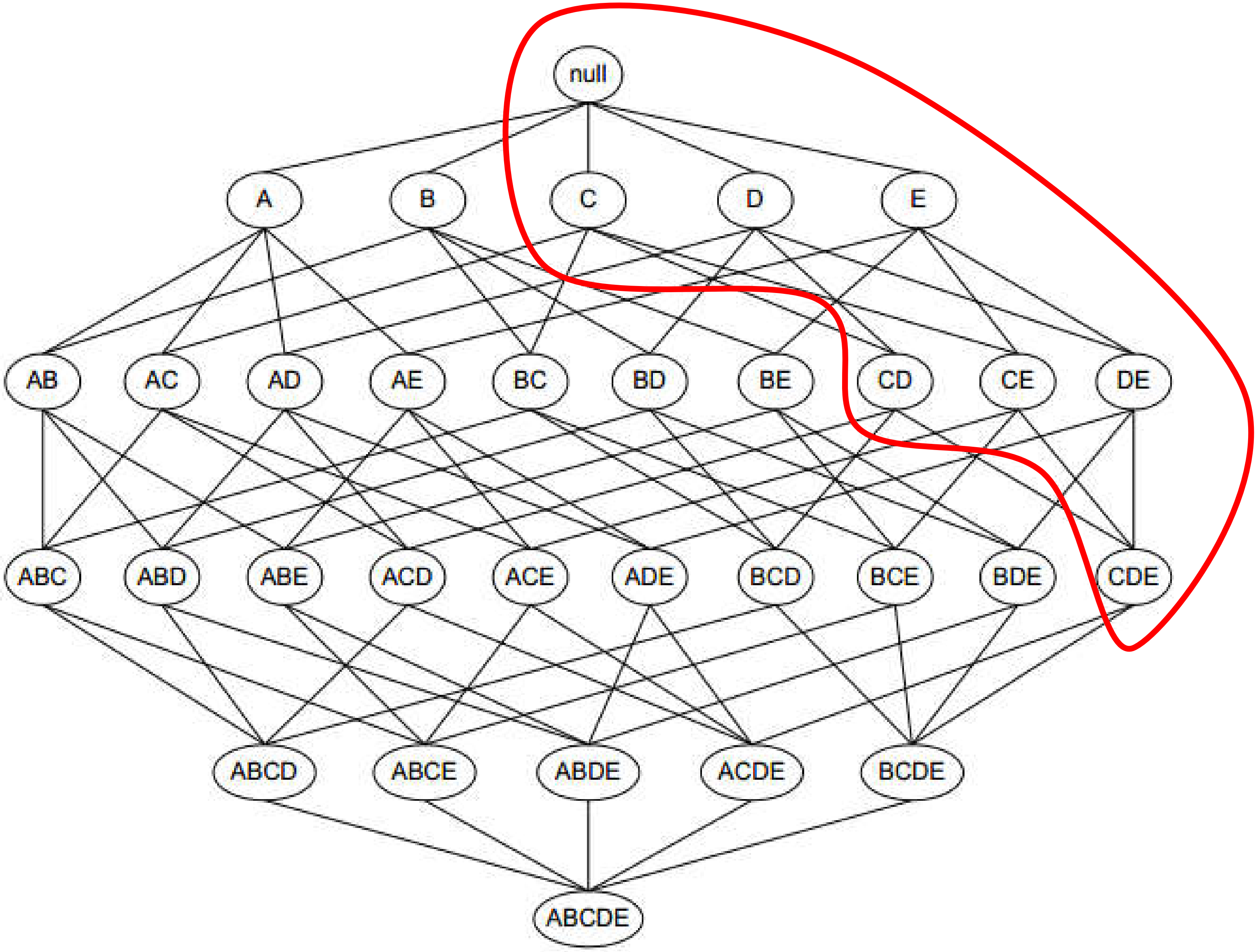
- Apriori principle holds due to the following property of the support measure:

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

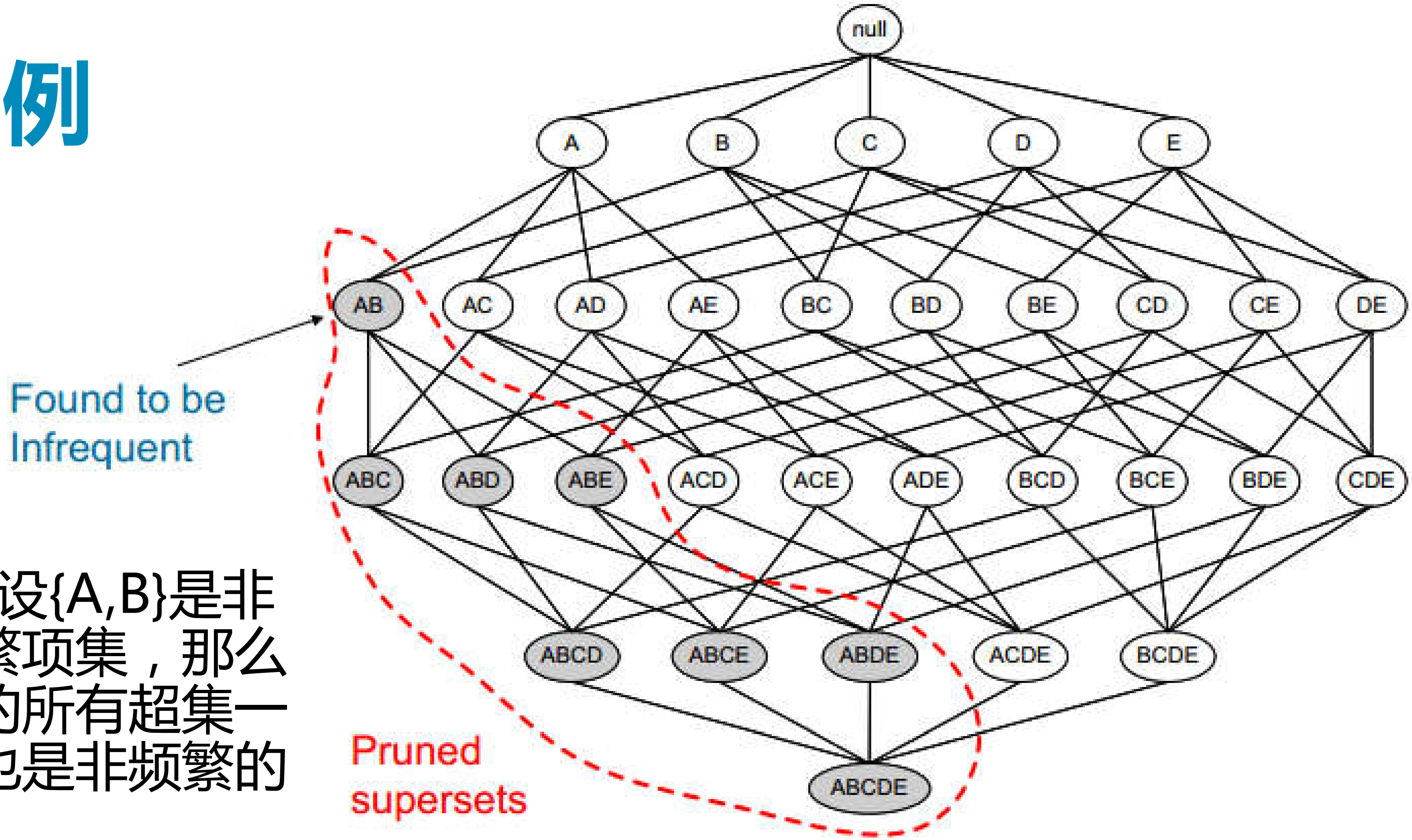
  - Support of an itemset never exceeds the support of its subsets
  - This is known as the anti-monotone property of support

举例

- 假设{C,D,E}是频繁项集，那么它的所有子集一定也是频繁的



举例



- 假设{A,B}是非频繁项集，那么它的所有超集一定也是非频繁的

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

Items (1-itemsets)

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

Pairs (2-itemsets)

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

(No need to generate candidates involving Coke or Eggs)

Minimum Support = 3

Triplets (3-itemsets)

Itemset	Count
{Bread,Milk,Diaper}	**3**

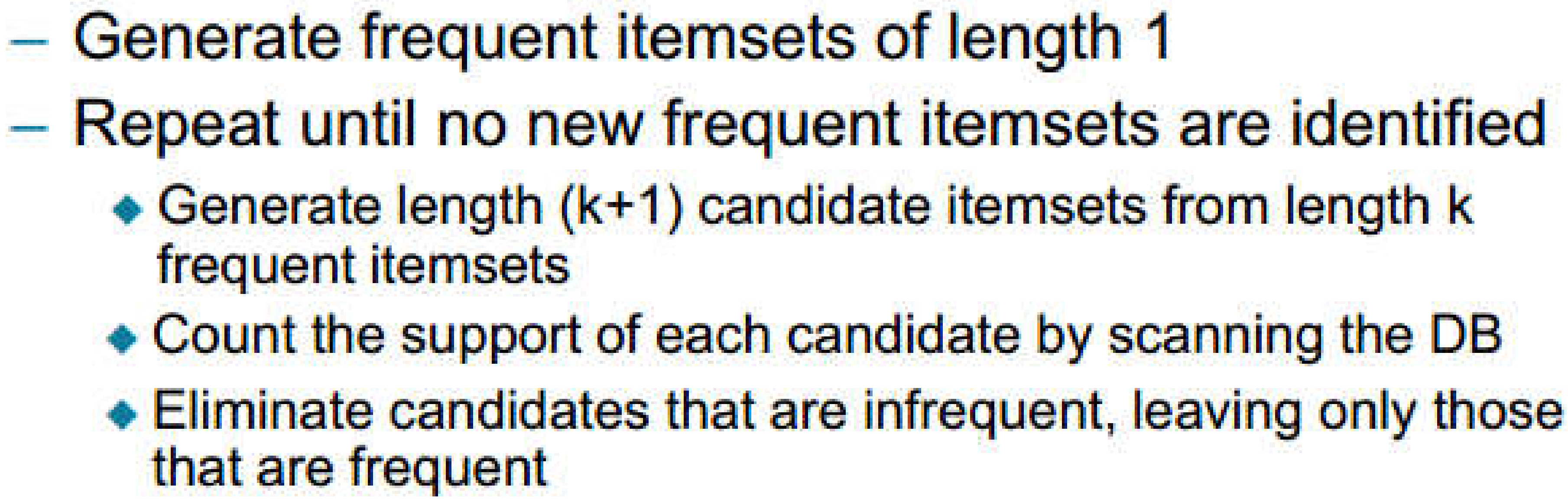
If every subset is considered,

$^6C_1 + {}^6C_2 + {}^6C_3 = 41$

With support-based pruning,

$6 + 6 + 1 = 13$

APRIORI算法



- Generate frequent itemsets of length 1
- Repeat until no new frequent itemsets are identified
  - Generate length (k+1) candidate itemsets from length k frequent itemsets
  - Count the support of each candidate by scanning the DB
  - Eliminate candidates that are infrequent, leaving only those that are frequent

候选项集的产生与剪枝-1

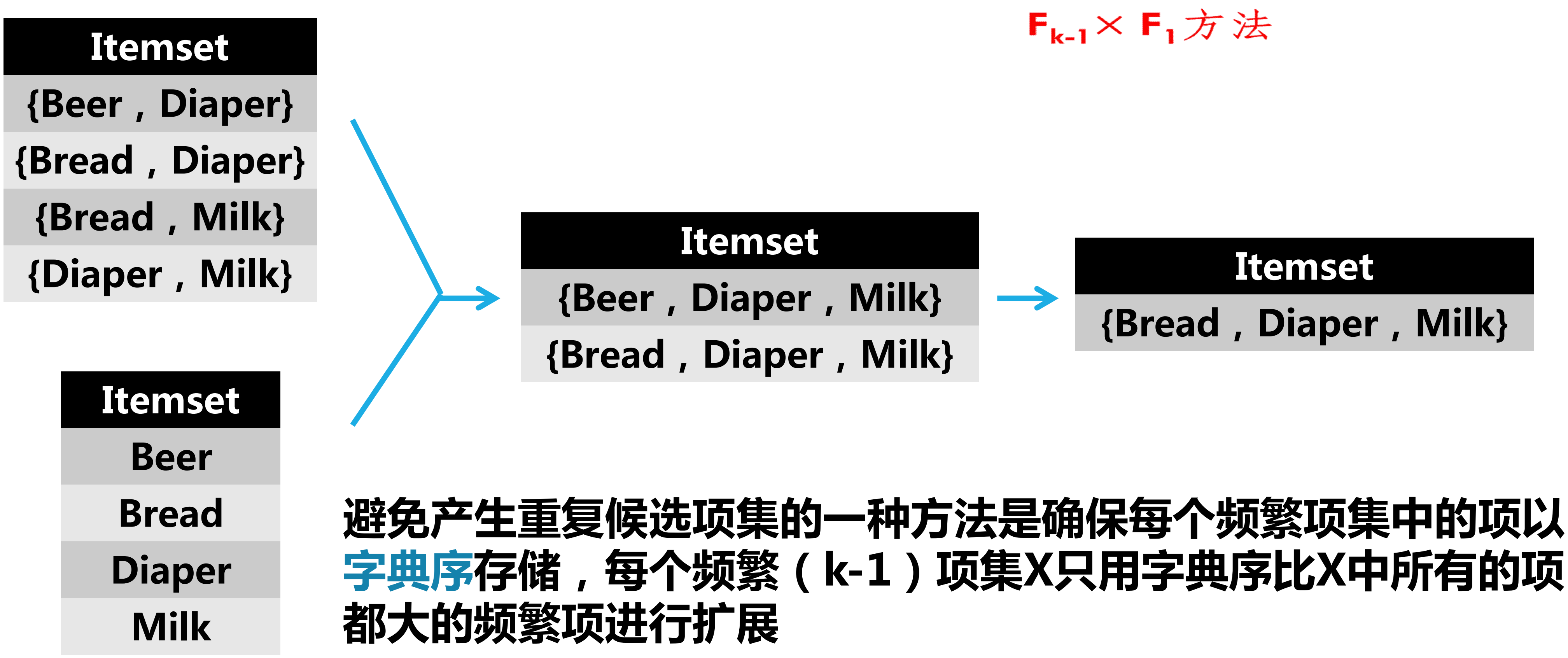
F_{k-1}× F_1方法

Itemset
{Beer , Diaper}
{Bread , Diaper}
{Bread , Milk}
{Diaper , Milk}

Itemset
Beer
Bread
Diaper
Milk

Itemset
{Beer , Diaper , Bread}
{Beer , Diaper , Milk}
{Bread , Diaper , Beer}
{Bread , Diaper , Milk}
{Bread , Milk , Beer}
{Bread , Milk , Diaper}
{Diaper , Milk, Beer}
{Diaper , Milk, Bread}

候选项集的产生与剪枝-1（续）



F_{k-1}× F_1方法

Itemset
{Beer，Diaper}
{Bread，Diaper}
{Bread，Milk}
{Diaper，Milk}

Itemset
Beer
Bread
Diaper
Milk

Itemset
{Beer，Diaper，Milk}
{Bread，Diaper，Milk}

Itemset
{Bread，Diaper，Milk}

避免产生重复候选项集的一种方法是确保每个频繁项集中的项以**字典序**存储，每个频繁（k-1）项集X只用字典序比X中所有的项都大的频繁项进行扩展

候选项集的产生与剪枝-2

$F_{k-1} \times F_{k-1}$ 方法

频繁（k-1）项集：$A=\{a_1,a_2,\ldots,a_{k-1}\}$，$B=\{b_1,b_2,\ldots,b_{k-1}\}$

Itemset
{Beer，Diaper}
{Bread，Diaper}
{Bread，Milk}
{Diaper，Milk}

Itemset
{Beer，Diaper}
{Bread，Diaper}
{Bread，Milk}
{Diaper，Milk}

→

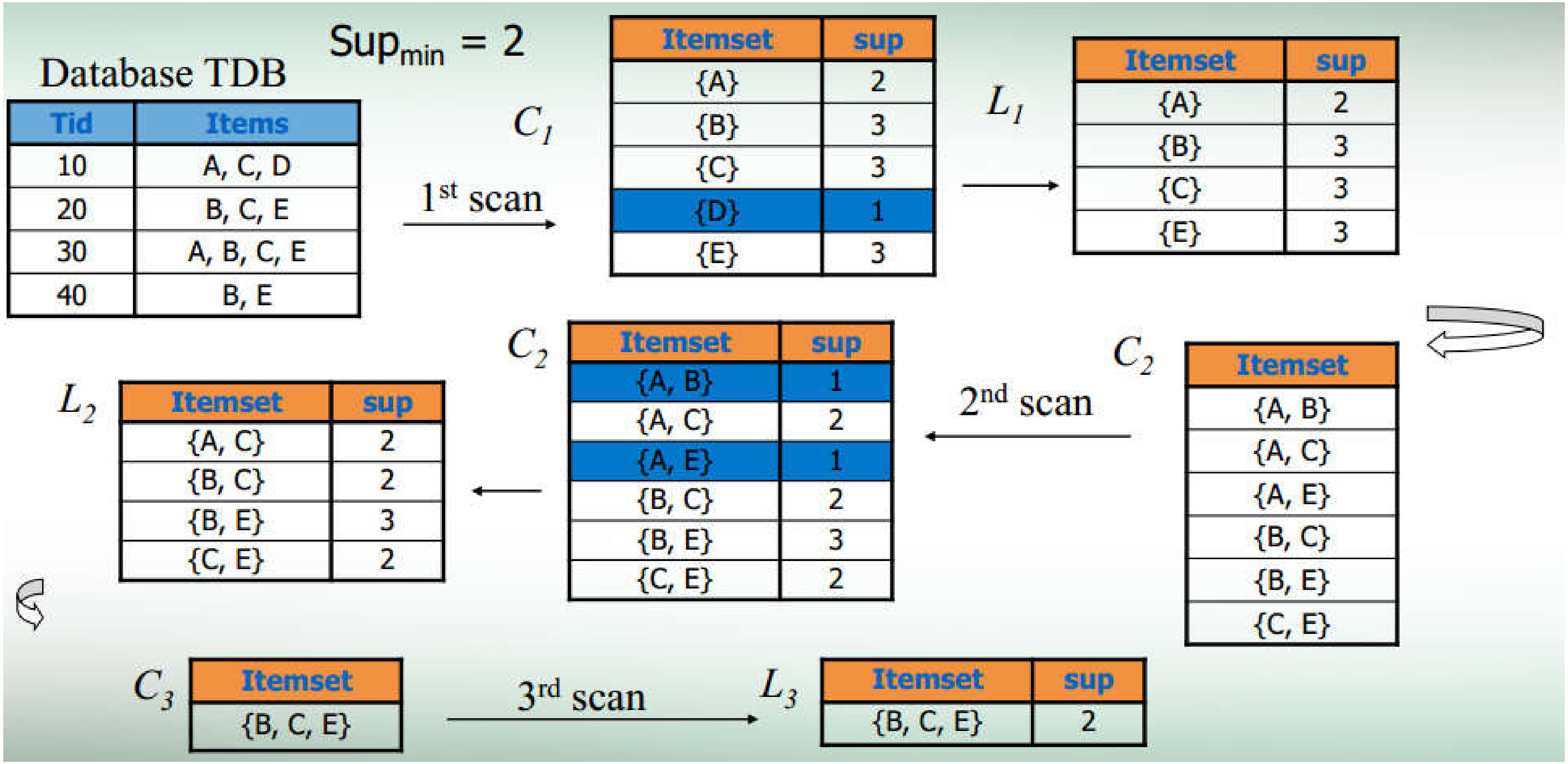
Itemset
{Bread，Diaper，Milk}

→

Itemset
{Bread，Diaper，Milk}

合并A和B，如果它们满足：

$a_i=b_i (i=1,2,\ldots,k-2)$并且$a_{k-1} \neq b_{k-1}$



$Sup_{min} = 2$

Database TDB

Tid	Items
10	A, C, D
20	B, C, E
30	A, B, C, E
40	B, E

1st scan

C_1

Itemset	sup
{A}	2
{B}	3
{C}	3
{D}	1
{E}	3

L_1

Itemset	sup
{A}	2
{B}	3
{C}	3
{E}	3

C_2

Itemset
{A, B}
{A, C}
{A, E}
{B, C}
{B, E}
{C, E}

2nd scan

C_2

Itemset	sup
{A, B}	1
{A, C}	2
{A, E}	1
{B, C}	2
{B, E}	3
{C, E}	2

L_2

Itemset	sup
{A, C}	2
{B, C}	2
{B, E}	3
{C, E}	2

C_3

Itemset
{B, C, E}

3rd scan

L_3

Itemset	sup
{B, C, E}	2

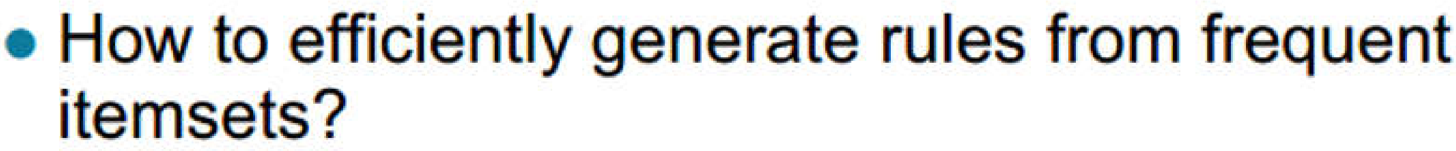
规则产生

- Given a frequent itemset L, find all non-empty subsets $f \subset L$ such that $f \rightarrow L - f$ satisfies the minimum confidence requirement
  - If {A,B,C,D} is a frequent itemset, candidate rules:

    ABC →D, ABD →C, ACD →B, BCD →A,
    A →BCD, B →ACD, C →ABD, D →ABC
    AB →CD, AC → BD, AD → BC, BC →AD,
    BD →AC, CD →AB,

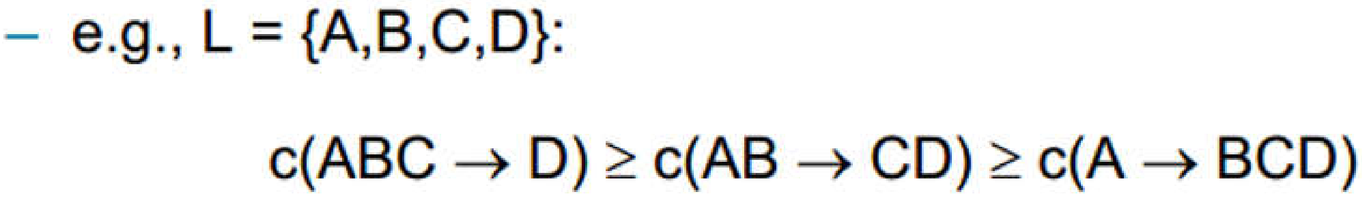
- If |L| = k, then there are $2^k - 2$ candidate association rules (ignoring $L \rightarrow \varnothing$ and $\varnothing \rightarrow L$)

置信度的剪枝



- How to efficiently generate rules from frequent itemsets?

如果规则f->L-f不满足置信度阈值，则形如f’ ->L-f’ 的规则一定也不满足置信度阈值，其中f’ 是f的子集

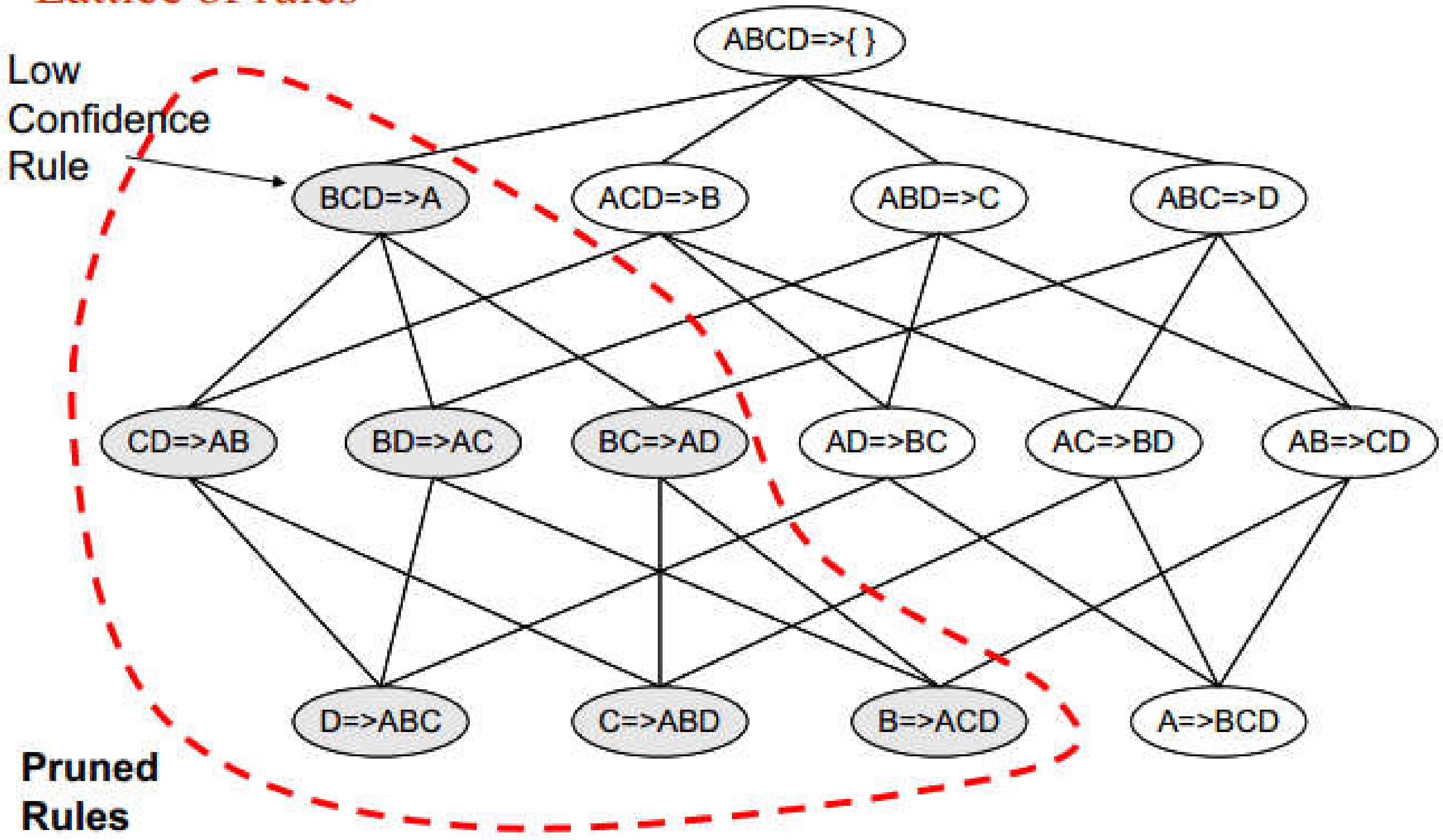


- e.g., L = {A,B,C,D}:

$$c(ABC \rightarrow D) \geq c(AB \rightarrow CD) \geq c(A \rightarrow BCD)$$

APRIORI算法

Lattice of rules



餐饮实例

- 下面通过餐饮企业中的例子演示Apriori在Python中的实现。

- 数据库中部分点餐数据如下表：

序列	时间	订单号	菜品id	菜品名称
1	2014/8/21	101	18491	健康麦香包
2	2014/8/21	101	8693	香煎葱油饼
3	2014/8/21	101	8705	翡翠蒸香茜饺
4	2014/8/21	102	8842	菜心粒咸骨粥
5	2014/8/21	102	7794	养颜红枣糕
6	2014/8/21	103	8842	金丝燕麦包
7	2014/8/21	103	8693	三丝炒河粉
…	…	…	…	…

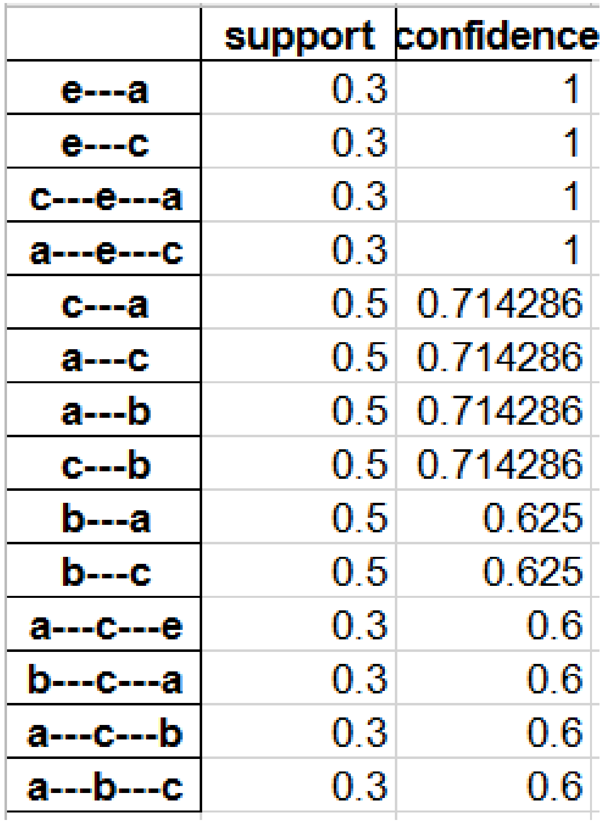
关联规则的实现

■某餐厅的业务数据如下：

订单号	菜品id
1	18491, 8693，8705
2	8842,7794
3	8842，8693
4	18491，8842，8693，7794
5	18491，8842
6	8842，8693
7	18491，8842
8	18491，8842,8693,8705
9	18491，8842,8693
10	18491，8693

订单号	菜品id
1	a，c，e
2	b，d
3	b，c
4	a，b，c，d
5	a，b
6	b，c
7	a，b
8	a，b，c，e
9	a，b，c
10	a，c，e

关联规则运行结果



	support	confidence
e---a	0.3	1
e---c	0.3	1
c---e---a	0.3	1
a---e---c	0.3	1
c---a	0.5	0.714286
a---c	0.5	0.714286
a---b	0.5	0.714286
c---b	0.5	0.714286
b---a	0.5	0.625
b---c	0.5	0.625
a---c---e	0.3	0.6
b---c---a	0.3	0.6
a---c---b	0.3	0.6
a---b---c	0.3	0.6