

Association Rule Mining

Dr.M. Venkatesan

Assistant Professor

Department of Computer Science and

Engineering

NITK

Mining Frequent Patterns, Association

- Efficient and scalable frequent itemset mining methods
- Mining various kinds of association rules

What Is Frequent Pattern Analysis?

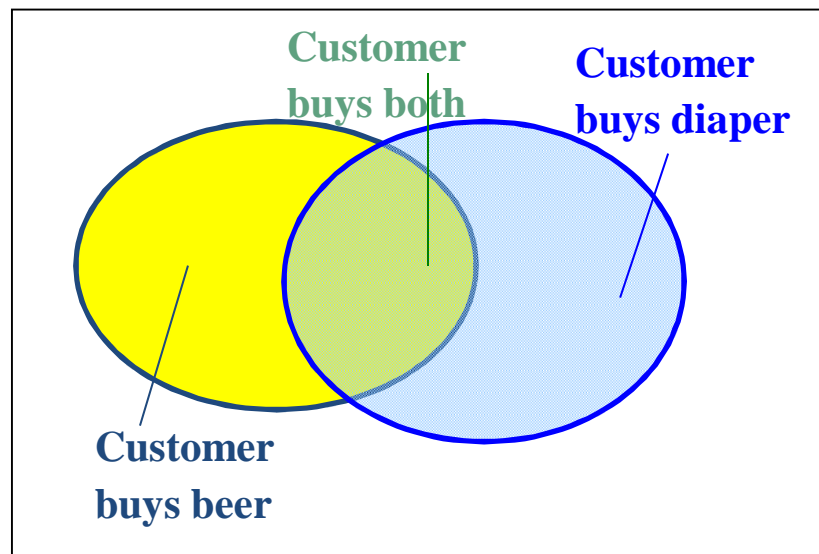
- **Frequent pattern**: a pattern (a set of items, subsequences, substructures, etc.) that occurs frequently in a data set
- First proposed by Agrawal, Imielinski, and Swami [AIS93] in the context of **frequent itemsets** and **association rule mining**
- Motivation: Finding inherent regularities in data
 - What products were often purchased together?— Beer and diapers?!
 - What are the subsequent purchases after buying a PC?
 - What kinds of DNA are sensitive to this new drug?
 - Can we automatically classify web documents?
- Applications
 - Basket data analysis, cross-marketing, catalog design, sale campaign analysis, Web log (click stream) analysis, and DNA sequence analysis.

Why Is Freq. Pattern Mining Important?

- Discloses an intrinsic and important property of data sets
- Forms the foundation for many essential data mining tasks
 - Association, correlation, and causality analysis
 - Sequential, structural (e.g., sub-graph) patterns
 - Pattern analysis in spatiotemporal, multimedia, time-series, and stream data
 - Classification: associative classification
 - Cluster analysis: frequent pattern-based clustering
 - Data warehousing: iceberg cube and cube-gradient
 - Semantic data compression: fascicles
 - Broad applications

Basic Concepts: Frequent Patterns and Association Rules

Transaction-id	Items bought
10	A, B, D
20	A, C, D
30	A, D, E
40	B, E, F
50	B, C, D, E, F



- Itemset $X = \{x_1, \dots, x_k\}$
- Find all the rules $X \rightarrow Y$ with minimum support and confidence
 - **support**, s , **probability** that a transaction contains $X \cup Y$
 - **confidence**, c , **conditional probability** that a transaction having X also contains Y

Let $sup_{min} = 50\%$, $conf_{min} = 50\%$

Freq. Pat.: $\{A:3, B:3, D:4, E:3, AD:3\}$

Association rules:

$A \rightarrow D$ (60%, 100%)

$D \rightarrow A$ (60%, 75%)

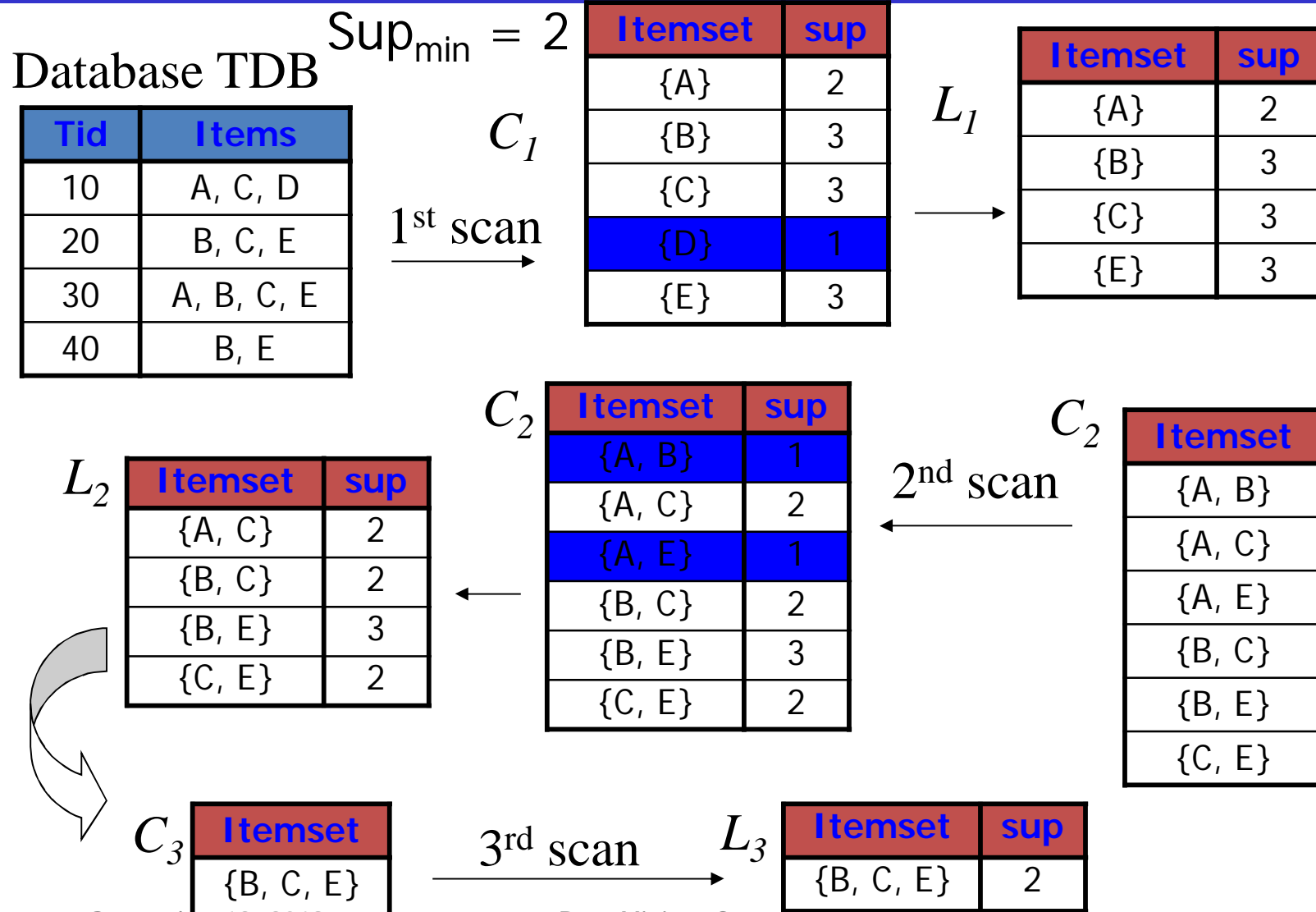
Scalable Methods for Mining Frequent Patterns

- The **downward closure** property of frequent patterns
 - Any subset of a frequent itemset must be frequent
 - If **{beer, diaper, nuts}** is frequent, so is **{beer, diaper}**
 - i.e., every transaction having {beer, diaper, nuts} also contains {beer, diaper}
- Scalable mining methods: Three major approaches
 - Apriori (Agrawal & Srikant@VLDB'94)
 - Freq. pattern growth (FPgrowth—Han, Pei & Yin @SIGMOD'00)
 - Vertical data format approach (Charm—Zaki & Hsiao @SDM'02)

Apriori: A Candidate Generation-and-Test Approach

- Apriori pruning principle: If there is **any** itemset which is infrequent, its superset should not be generated/tested!
(Agrawal & Srikant @VLDB'94, Mannila, et al. @ KDD' 94)
- Method:
 - Initially, scan DB once to get frequent 1-itemset
 - Generate length $(k+1)$ candidate itemsets from length k frequent itemsets
 - Test the candidates against DB
 - Terminate when no frequent or candidate set can be generated

The Apriori Algorithm—An Example



The Apriori Algorithm

- Pseudo-code:

C_k : Candidate itemset of size k

L_k : frequent itemset of size k

$L_1 = \{\text{frequent items}\};$

for ($k = 1; L_k \neq \emptyset; k++$) **do begin**

C_{k+1} = candidates generated from L_k ;

for each transaction t in database **do**

 increment the count of all candidates in C_{k+1}
 that are contained in t

L_{k+1} = candidates in C_{k+1} with min_support

end

return $\cup_k L_k$;

Important Details of Apriori

- How to generate candidates?
 - Step 1: self-joining L_k
 - Step 2: pruning
- How to count supports of candidates?
- Example of Candidate-generation
 - $L_3 = \{abc, abd, acd, ace, bcd\}$
 - Self-joining: $L_3 * L_3$
 - $abcd$ from abc and abd
 - $acde$ from acd and ace
 - Pruning:
 - $acde$ is removed because ade is not in L_3
 - $C_4 = \{abcd\}$

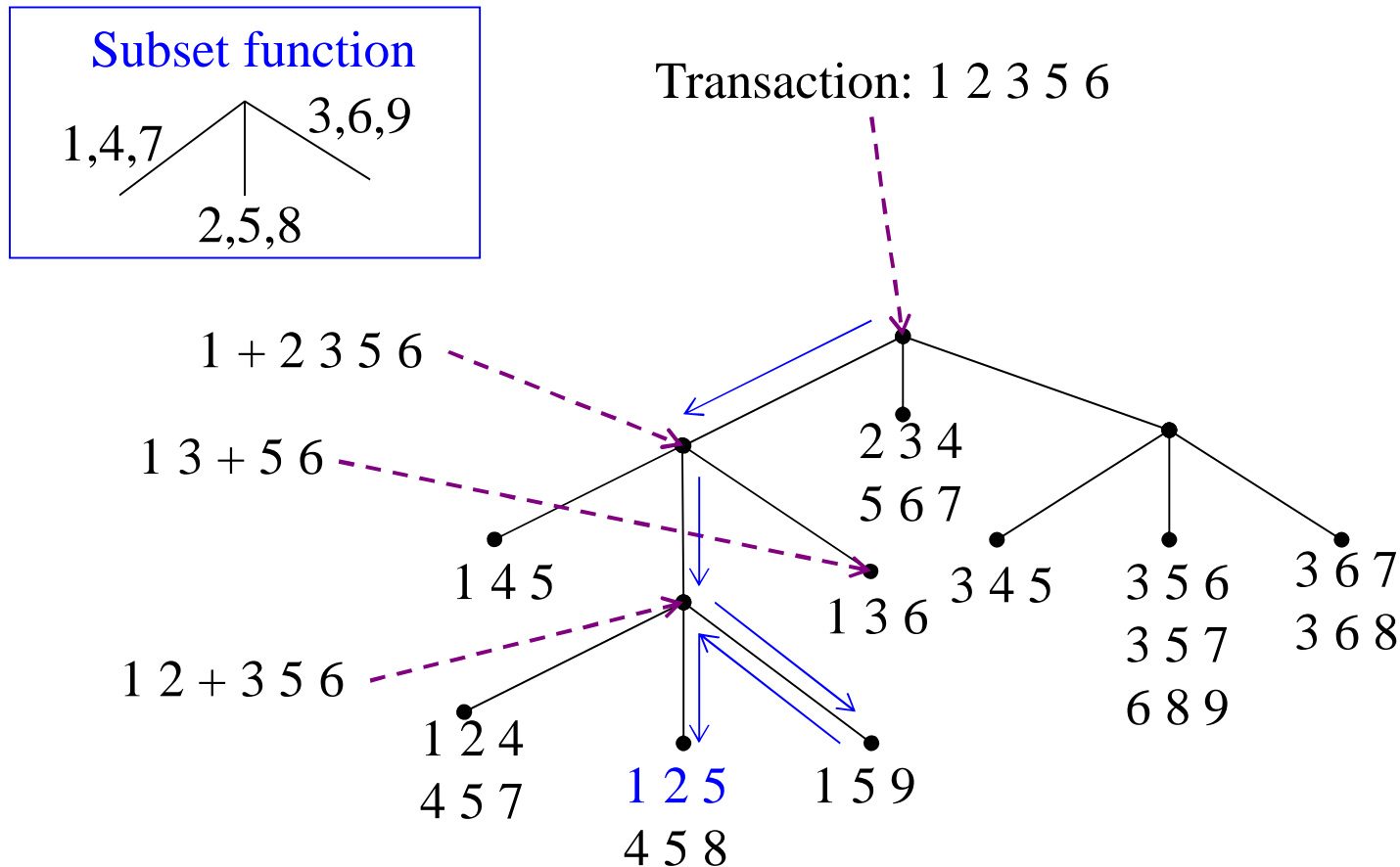
How to Generate Candidates?

- Suppose the items in L_{k-1} are listed in an order
- Step 1: self-joining L_{k-1}
insert into C_k
select **$p.item_1, p.item_2, \dots, p.item_{k-1}, q.item_{k-1}$**
from **$L_{k-1} p, L_{k-1} q$**
where **$p.item_1 = q.item_1, \dots, p.item_{k-2} = q.item_{k-2}, p.item_{k-1} <$**
 $q.item_{k-1}$
- Step 2: pruning
for all ***itemsets* c in C_k** do
 for all ***(k-1)-subsets* s of c** do
 if (s is not in L_{k-1}) then delete c from C_k

How to Count Supports of Candidates?

- Why counting supports of candidates a problem?
 - The total number of candidates can be very huge
 - One transaction may contain many candidates
- Method:
 - Candidate itemsets are stored in a *hash-tree*
 - *Leaf node* of hash-tree contains a list of itemsets and counts
 - *Interior node* contains a hash table
 - *Subset function*: finds all the candidates contained in a transaction

Example: Counting Supports of Candidates



Challenges of Frequent Pattern Mining

- Challenges
 - Multiple scans of transaction database
 - Huge number of candidates
 - Tedious workload of support counting for candidates
- Improving Apriori: general ideas
 - Reduce passes of transaction database scans
 - Shrink number of candidates
 - Facilitate support counting of candidates

Eclat Algorithm

ECLAT Algorithm

- **Equivalence Class Clustering and bottom up Lattice Traversal- ECLAT**
- Method for Frequent Itemset Generation
- Searches in a DFS manner.
- Represent the data in **vertical format**.

Eclat(cont..)

- Both Apriori and FP-growth use **horizontal data format**
- Alternatively data can also be represented in **vertical format**

TID	Items
1	Bread,Butter,Jam
2	Butter,Coke
3	Butter,Milk
4	Bread,Butter,Coke
5	Bread,Milk
6	Butter,Milk
7	Bread,Milk
8	Bread,Butter,Milk,Jam
9	Bread,Butter,Milk

Item Set	TID set
Bread	1,4,5,7,8,9
Butter	1,2,3,4,6,8,9
Milk	3,5,6,7,8,9
Coke	2,4
Jam	1,8

Eclat(cont..)

Eclat: algorithm

1. Get tidlist for each item (DB scan)
2. Tidlist of $\{a\}$ is exactly the list of transactions containing $\{a\}$
3. Intersect tidlist of $\{a\}$ with the tidlists of all other items, resulting in tidlists of $\{a,b\}$, $\{a,c\}$, $\{a,d\}$, ...
= $\{a\}$ -conditional database (if $\{a\}$ removed)
4. Repeat from 1 on $\{a\}$ -conditional database
5. Repeat for all other items

Eclat(Cont..)

Frequent 1-itemsets

min_sup=2

Item Set	TID Set
Bread	1,4,5,7,8,9
Butter	1,2,3,4,6,8,9
Milk	3,5,6,7,8,9
Coke	2,4
Jam	1,8

Frequent 2-itemsets

Item Set	TID set
{Bread,Butter}	1,4,8,9
{Bread,Milk}	5,7,8,9
{Bread,Coke}	4
{Bread,Jam}	1,8
{Butter,Milk}	3,6,8,9
{Butter,Coke}	2,4
{Butter,Jam}	1,8
{Milk,Jam}	8

Cont..

Frequent 3-itemsets

Item Set	TID Set
{Bread,Butter,Milk}	8,9
{Bread,Butter,Jam}	1,8

- This process repeats, with k incremented by 1 each time, until no frequent items or no candidate itemsets can be found.

Association Rule Mining

- Given a set of transactions, find rules that will predict the occurrence of an item based on the occurrences of other items in the transaction

Market-Basket transactions

<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

Example of Association Rules

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

Implication means co-occurrence,
not causality!

Definition: Frequent Itemset

- **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. $\sigma(\{\text{Milk, Bread, Diaper}\}) = 2$
- **Support**
 - Fraction of transactions that contain an itemset
 - E.g. $s(\{\text{Milk, Bread, Diaper}\}) = 2/5$
- **Frequent Itemset**
 - An itemset whose support is greater than or equal to a *minsup* threshold

<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

Definition: Association Rule

- Association Rule
 - An implication expression of the form $X \rightarrow Y$, where X and Y are itemsets
 - Example:
 $\{\text{Milk, Diaper}\} \rightarrow \{\text{Beer}\}$

<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

- 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

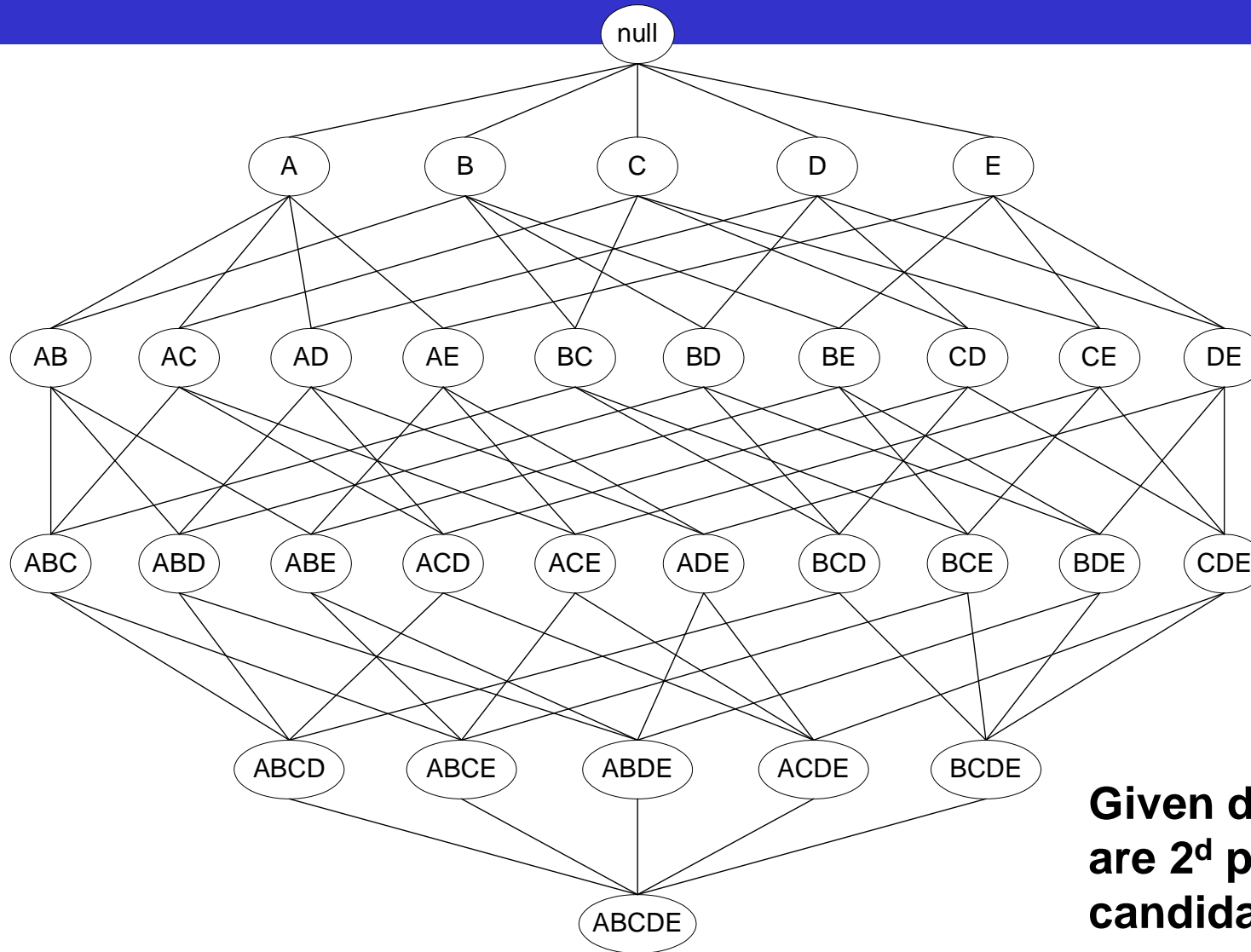
Example:

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

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

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

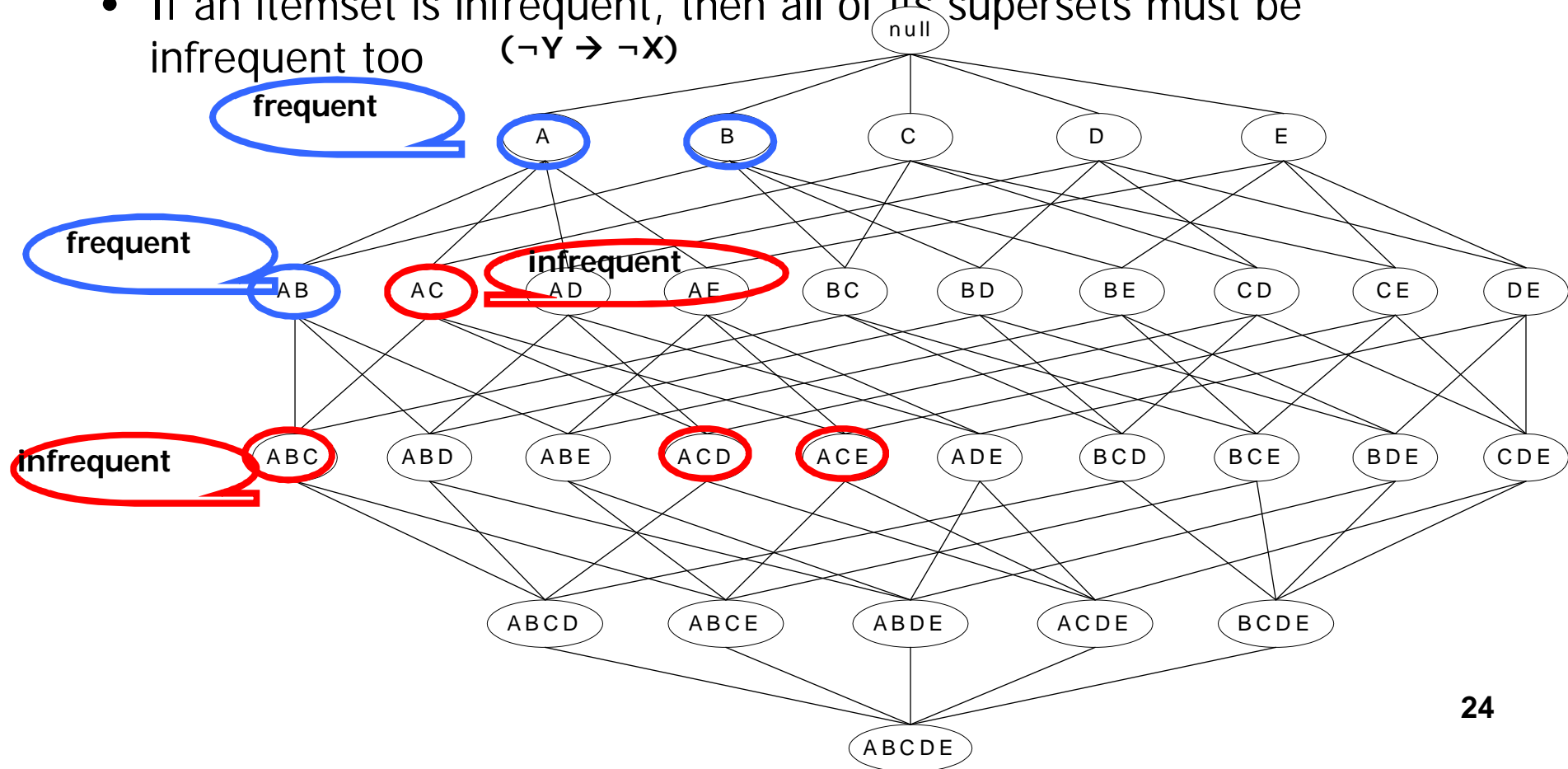
Frequent Itemset Generation



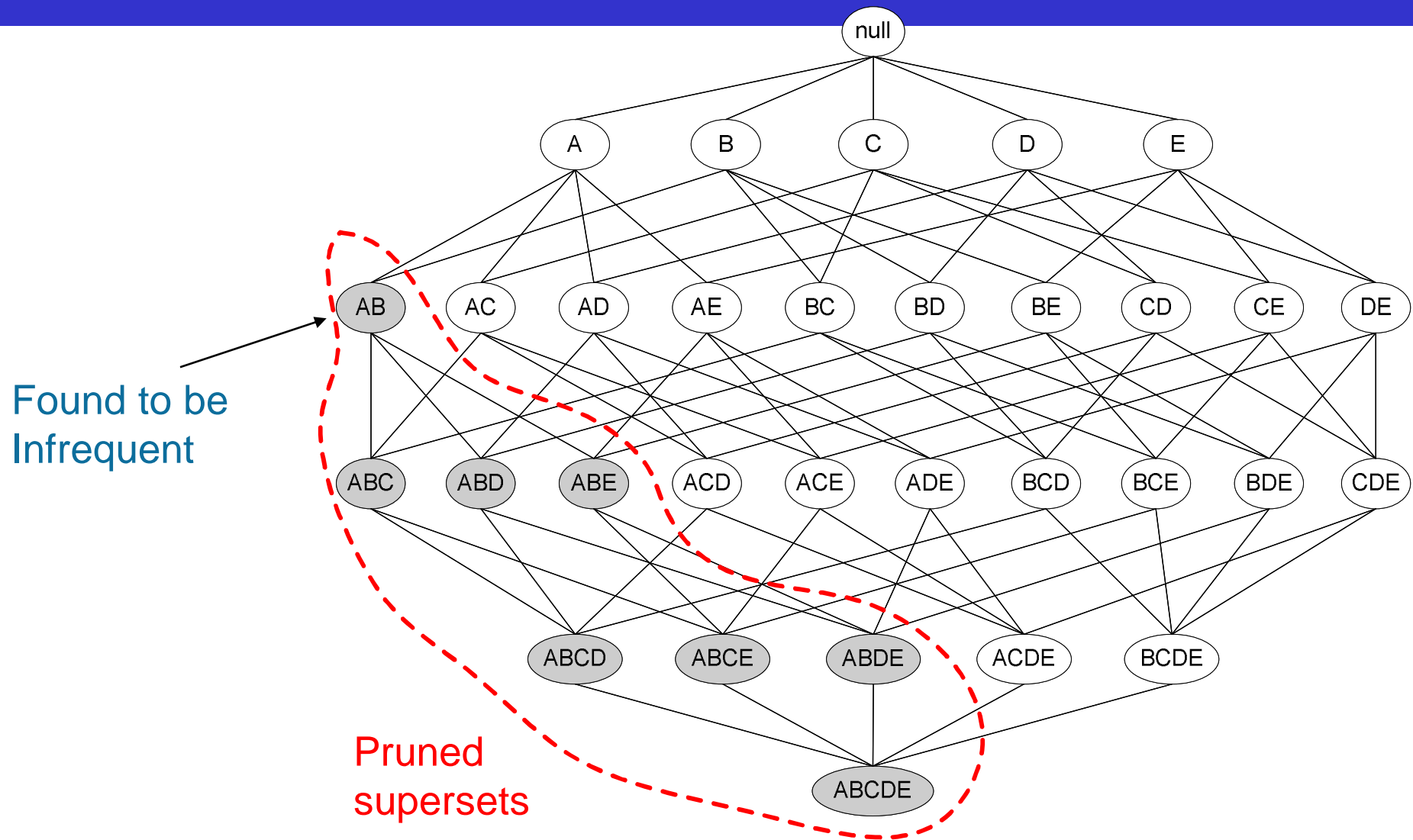
Given d items, there are 2^d possible candidate itemsets

Apriori Principle

- If an itemset is frequent, then all of its subsets must also be frequent $(X \rightarrow Y)$
- If an itemset is infrequent, then all of its supersets must be infrequent too $(\neg Y \rightarrow \neg X)$

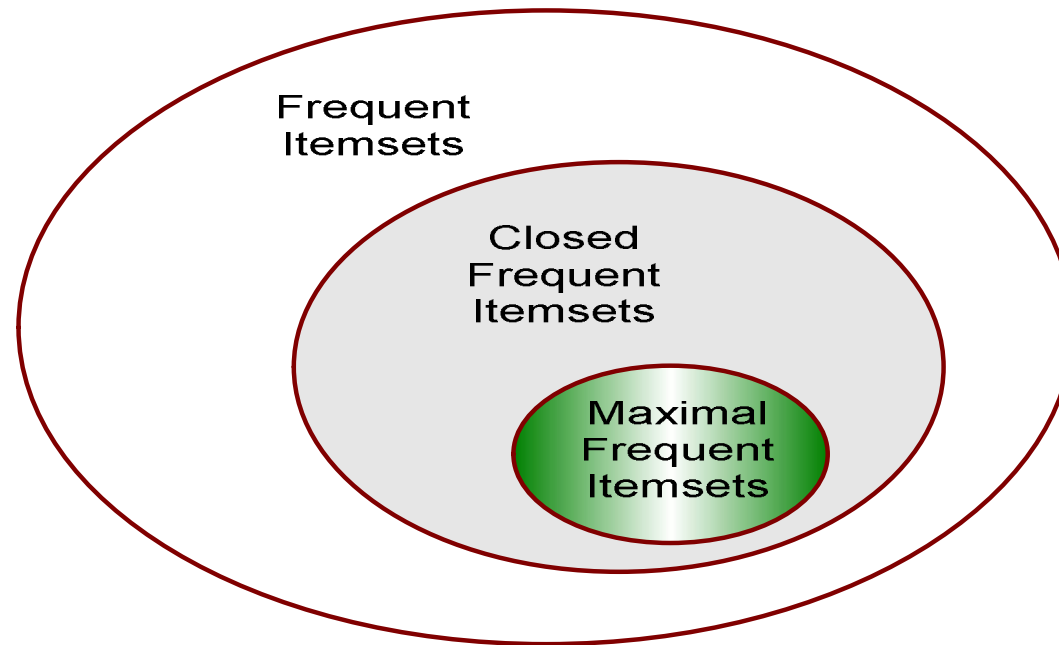


Illustrating Apriori Principle

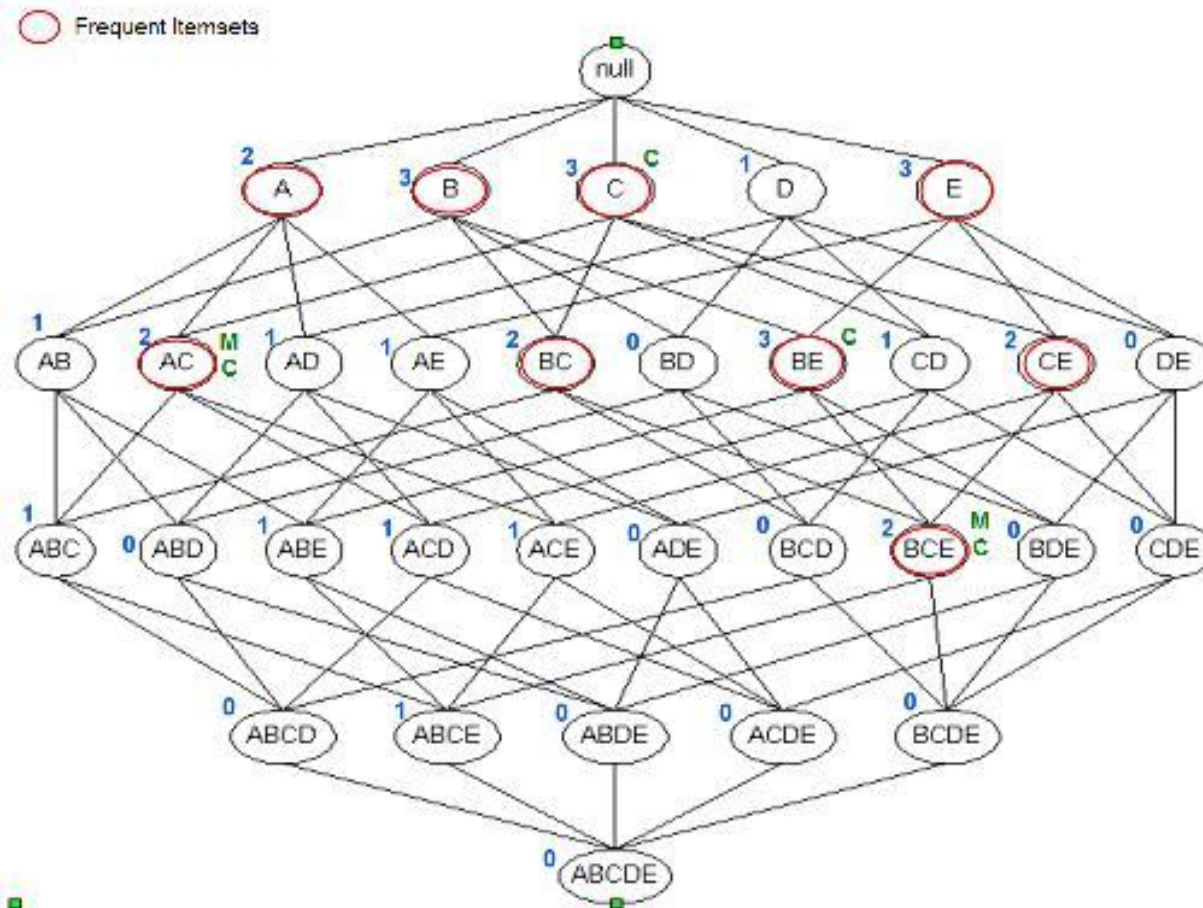


Closed Itemset: support of all parents are not equal to the support of the itemset.

Maximal Itemset: all parents of that itemset must be infrequent.



Itemset {c} is closed as support of parents (supersets) {A C}:2, {B C}:2, {C D}:1, {C E}:2 not equal support of {c}:3. And the same for {A C}, {B E} & {B C E}.
 Itemset {A C} is maximal as all parents (supersets) {A B C}, {A C D}, {A C E} are infrequent. And the same for {B C E}.



Mining Frequent Patterns Without Candidate Generation

- Grow long patterns from short ones using local frequent items
 - “abc” is a frequent pattern
 - Get all transactions having “abc”: DB|abc
 - “d” is a local frequent item in DB|abc → abcd is a frequent pattern

Construct FP-tree from a Transaction Database

<i>TID</i>	<i>Items bought</i>	<i>(ordered) frequent items</i>
100	{f, a, c, d, g, i, m, p}	{f, c, a, m, p}
200	{a, b, c, f, l, m, o}	{f, c, a, b, m}
300	{b, f, h, j, o, w}	{f, b}
400	{b, c, k, s, p}	{c, b, p}
500	{a, f, c, e, l, p, m, n}	{f, c, a, m, p}

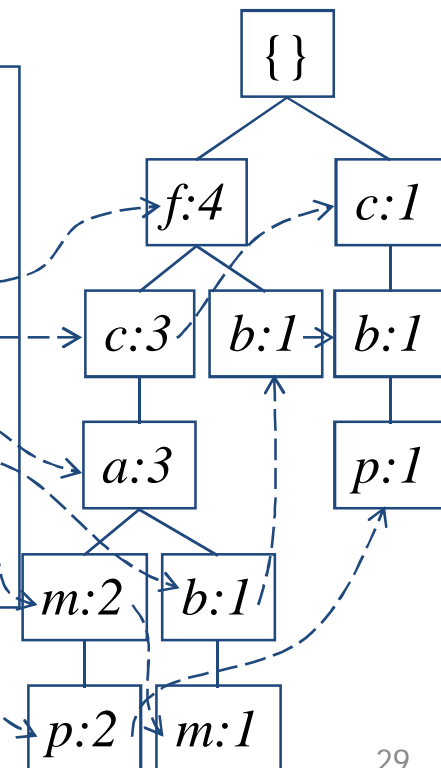
$min_support = 3$

1. Scan DB once, find frequent 1-itemset (single item pattern)
2. Sort frequent items in frequency descending order, f-list
3. Scan DB again, construct FP-tree

Header Table

<i>Item</i>	<i>frequency</i>	<i>head</i>
f	4	
c	4	
a	3	
b	3	
m	3	
p	3	

F-list = f-c-a-b-m-p



Benefits of the FP-tree Structure

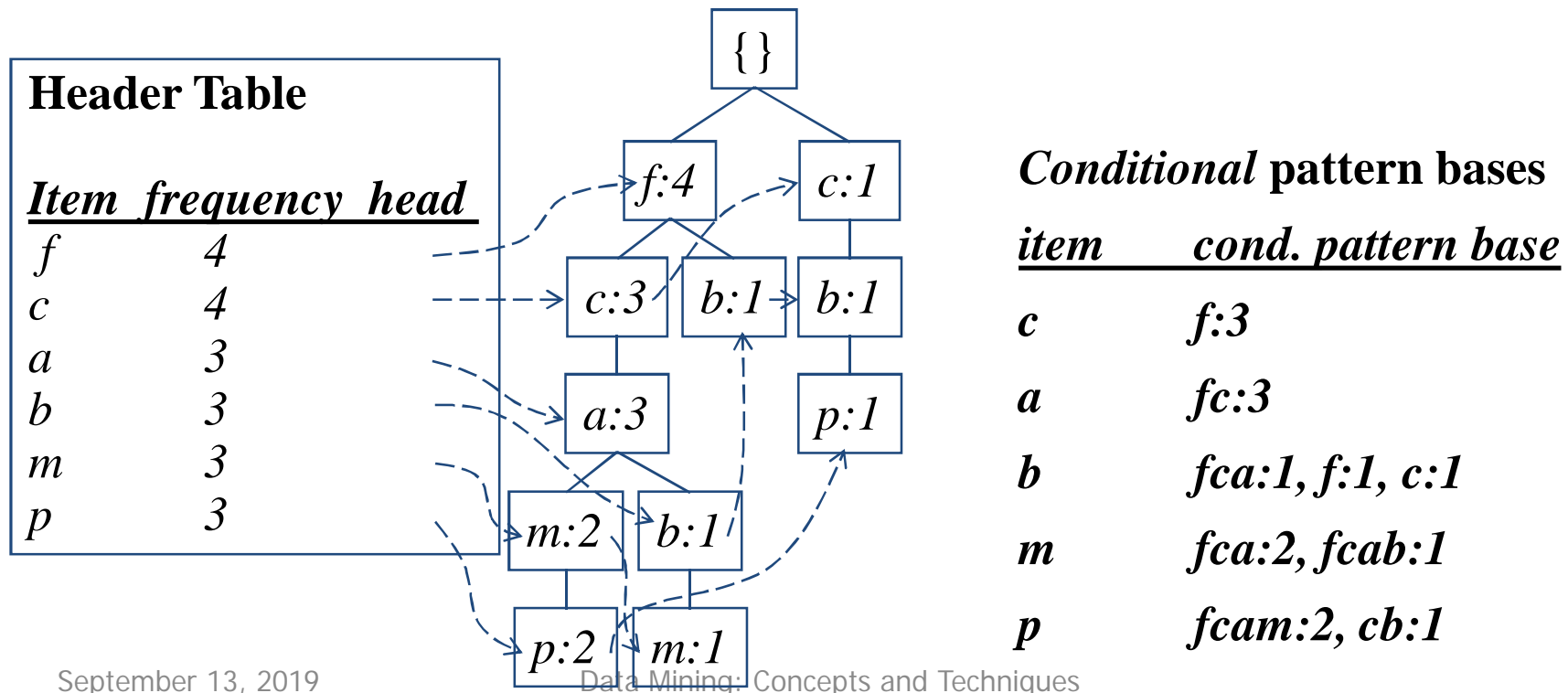
- Completeness
 - Preserve complete information for frequent pattern mining
 - Never break a long pattern of any transaction
- Compactness
 - Reduce irrelevant info—infrequent items are gone
 - Items in frequency descending order: the more frequently occurring, the more likely to be shared
 - Never be larger than the original database (not count node-links and the *count* field)
 - For Connect-4 DB, compression ratio could be over 100

Partition Patterns and Databases

- Frequent patterns can be partitioned into subsets according to f-list
 - F-list=f-c-a-b-m-p
 - Patterns containing p
 - Patterns having m but no p
 - ...
 - Patterns having c but no a nor b, m, p
 - Pattern f
- Completeness and non-redundancy

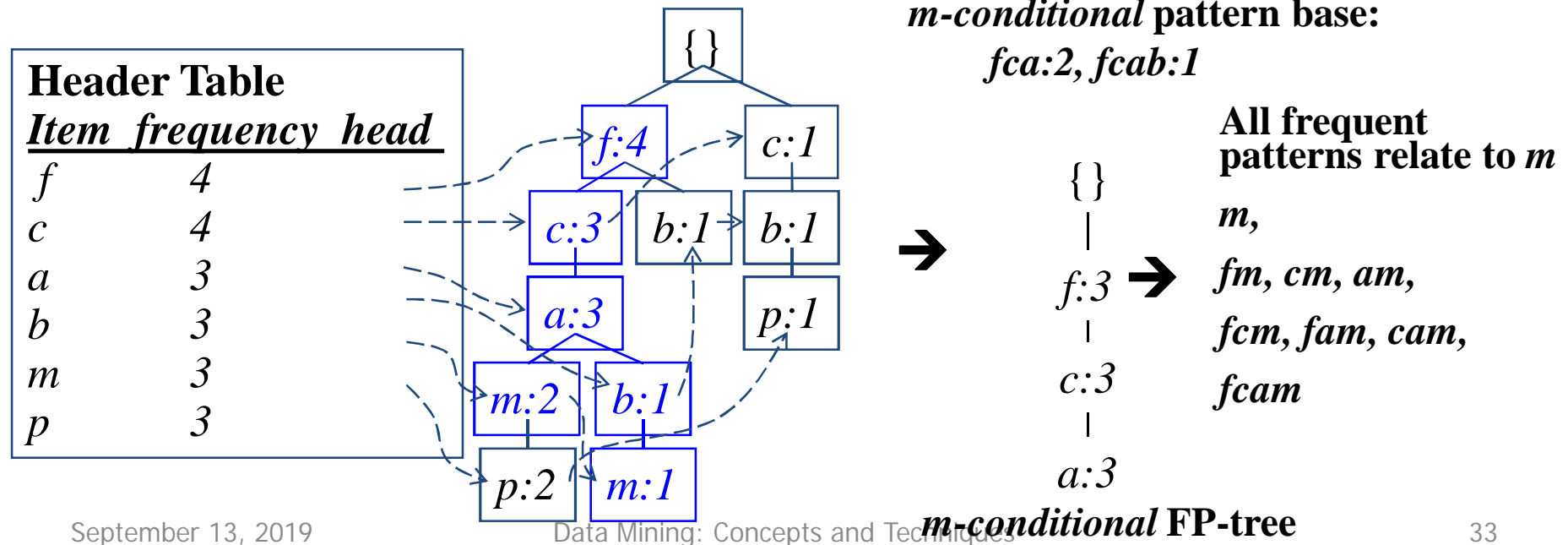
Find Patterns Having P From P-conditional Database

- Starting at the frequent item header table in the FP-tree
- Traverse the FP-tree by following the link of each frequent item p
- Accumulate all of *transformed prefix paths* of item p to form p 's conditional pattern base



From Conditional Pattern-bases to Conditional FP-trees

- For each pattern-base
 - Accumulate the count for each item in the base
 - Construct the FP-tree for the frequent items of the pattern base



Mining Various Kinds of Association Rules

- Mining multilevel association
- Mining multidimensional association
- Mining quantitative association
- Mining interesting correlation patterns

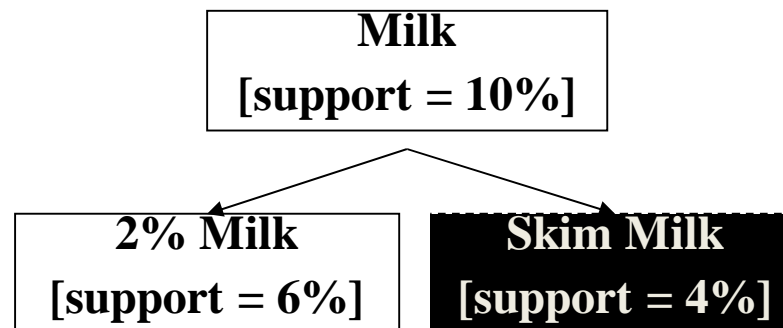
Mining Multiple-Level Association Rules

- Items often form hierarchies
- Flexible support settings
 - Items at the lower level are expected to have lower support
- Exploration of *shared* multi-level mining (Agrawal & Srikant@VLB'95, Han & Fu@VLDB'95)

uniform support

Level 1
min_sup = 5%

Level 2
min_sup = 5%



reduced support

Level 1
min_sup = 5%

Level 2
min_sup = 3%

Mining Multi-Dimensional Association

- Single-dimensional rules:

$\text{buys}(X, \text{"milk"}) \Rightarrow \text{buys}(X, \text{"bread"})$

- Multi-dimensional rules: ≥ 2 dimensions or predicates
 - Inter-dimension assoc. rules (*no repeated predicates*)

$\text{age}(X, \text{"19-25"}) \wedge \text{occupation}(X, \text{"student"}) \Rightarrow \text{buys}(X, \text{"coke"})$

- hybrid-dimension assoc. rules (*repeated predicates*)

$\text{age}(X, \text{"19-25"}) \wedge \text{buys}(X, \text{"popcorn"}) \Rightarrow \text{buys}(X, \text{"coke"})$

- Categorical Attributes: finite number of possible values, no ordering among values—data cube approach
- Quantitative Attributes: numeric, implicit ordering among values—discretization, clustering, and gradient approaches

Mining Other Interesting Patterns

- Flexible support constraints (Wang et al. @ VLDB'02)
 - Some items (e.g., diamond) may occur rarely but are valuable
 - Customized sup_{\min} specification and application
- Top-K closed frequent patterns (Han, et al. @ ICDM'02)
 - Hard to specify sup_{\min} , but top-k with length_{\min} is more desirable
 - Dynamically raise sup_{\min} in FP-tree construction and mining, and select most promising path to mine

Association Mining to Correlation Analysis

- how even strong association rules can be uninteresting and misleading.
- how the support-confidence framework can be supplemented with additional interestingness measures based on statistical significance and correlation analysis.
- correlation measure can be used to augment the support-confidence framework for association rules. This leads to *correlation rules* of the form

$$A \Rightarrow B [\text{support}, \text{confidence}, \text{correlation}].$$

Case Study

- A misleading “strong” association rule. Suppose we are interested in analyzing transactions at *AllElectronics* with respect to the purchase of computer games and videos.
- Let *game* refer to the transactions containing computer games, and *video* refer to those containing videos. Of the 10,000 transactions analyzed, the data show that 6,000 of the customer transactions included computer games, while 7,500 included videos, and 4,000 included both computer games and videos.
- Suppose that a data mining program for discovering association rules is run on the data, using a minimum support of, say, 30% and a minimum confidence of 60%. The following association rule is discovered:

$buys(X, \text{“computer games”}) \Rightarrow buys(X, \text{“videos”})$ [support = 40%, confidence = 66%]

Case study (cont..)

- is a strong association rule?
- Support value of $4,000/10,000 = 40\%$ and confidence value of $4,000/6,000 = 66\%$ satisfy the minimum support and minimum confidence
- Yes. But?
- It is misleading, because the probability of purchasing videos is 75%, which is even larger than 66%.
- computer games and videos are negatively associated because the purchase of one of these items actually decreases the likelihood of purchasing the other
- Confidence only *estimate* of the conditional probability of itemset B given itemset A .
- It does not measure the real strength (or lack of strength) of the correlation and implication between A and B .

Lift

- Lift is a simple correlation measure .
- The occurrence of itemset A is independent of the occurrence of itemset B if $P(A \cup B) = P(A)P(B)$; otherwise, itemsets A and B are dependent and correlated as events.
- The lift between the occurrence of A and B can be measured by computing

$$\text{lift}(A, B) = \frac{P(A \cup B)}{P(A)P(B)}.$$

Lift

- Lift < 1 , then the occurrence of A is *negatively correlated with* the occurrence of B .
- Lift > 1 , then A and B are *positively correlated*, meaning that the occurrence of one implies the occurrence of the other.
- If the resulting value is equal to 1, then A and B are *independent* and there is no correlation between them.

Lift

A 2×2 contingency table summarizing the transactions with respect to game and video purchases.

	<i>game</i>	\overline{game}	Σ_{row}
<i>video</i>	4,000	3,500	7,500
\overline{video}	2,000	500	2,500
Σ_{col}	6,000	4,000	10,000

probability of purchasing a computer game is $P(game) = 0.60$,

probability of purchasing a video is $P(video) = 0.75$

probability of purchasing both is $P(game; video) = 0.40$.

$Lift = P(game, video) / (P(game)P(video)) = 0.40 / (0.60)(0.75) = 0.89 < 1$

negative correlation between the occurrence of *game* and *video*

negative correlation cannot be identified by a support confidence framework.

Interestingness Measure: Correlations (Lift)

- *play basketball* \Rightarrow *eat cereal* [40%, 66.7%] is misleading
 - The overall % of students eating cereal is 75% > 66.7%.
- *play basketball* \Rightarrow *not eat cereal* [20%, 33.3%] is more accurate, although with lower support and confidence
- Measure of dependent/correlated events: **lift**

$$lift = \frac{P(A \cup B)}{P(A)P(B)}$$

	Basketball	Not basketball	Sum (row)
Cereal	2000	1750	3750
Not cereal	1000	250	1250
Sum(col.)	3000	2000	5000

$$lift(B, C) = \frac{2000/5000}{3000/5000 * 3750/5000} = 0.89 \quad lift(B, \neg C) = \frac{1000/5000}{3000/5000 * 1250/5000} = 1.33$$

Chi square(χ^2) measure

The above contingency table, now shown with the expected values.

	<i>game</i>	\overline{game}	Σ_{row}
<i>video</i>	4,000 (4,500)	3,500 (3,000)	7,500
\overline{video}	2,000 (1,500)	500 (1,000)	2,500
Σ_{col}	6,000	4,000	10,000

$$\chi^2 = \Sigma \frac{(observed - expected)^2}{expected} = \frac{(4,000 - 4,500)^2}{4,500} + \frac{(3,500 - 3,000)^2}{3,000} + \frac{(2,000 - 1,500)^2}{1,500} + \frac{(500 - 1,000)^2}{1,000} = 555.6.$$

χ^2 value is greater than one, and the observed value of the slot (*game, video*) = 4,000, which is less than the expected value 4,500, *buying game* and *buying video* are *negatively correlated*.

All_conf and Cosine measures

Given an itemset $X = \{i_1, i_2, \dots, i_k\}$, the *all_confidence* of X is defined as

$$all_conf(X) = \frac{sup(X)}{max_item_sup(X)} = \frac{sup(X)}{max\{sup(i_j) | \forall i_j \in X\}}, \quad (5.24)$$

where $max\{sup(i_j) | \forall i_j \in X\}$ is the maximum (single) item support of all the items in X , and hence is called the *max_item_sup* of the itemset X . The *all_confidence* of X is the minimal confidence among the set of rules $i_j \rightarrow X - i_j$, where $i_j \in X$.

Given two itemsets A and B , the *cosine* measure of A and B is defined as

$$cosine(A, B) = \frac{P(A \cup B)}{\sqrt{P(A) \times P(B)}} = \frac{sup(A \cup B)}{\sqrt{sup(A) \times sup(B)}}. \quad (5.25)$$

Are *lift* and χ^2 Good Measures of Correlation?

- “Buy walnuts \Rightarrow buy milk [1%, 80%]” is misleading
 - if 85% of customers buy milk
- Support and confidence are not good to represent correlations
- So many interestingness measures? (Tan, Kumar, Sritastava @KDD'02)

$$lift = \frac{P(A \cup B)}{P(A)P(B)}$$

$$all_conf = \frac{sup(X)}{max_item_sup(X)}$$

$$coh = \frac{sup(X)}{|universe(X)|}$$

	Milk	No Milk	Sum (row)
Coffee	m, c	~m, c	c
No Coffee	m, ~c	~m, ~c	~c
Sum(col.)	m	~m	Σ

DB	m, c	~m, c	m ~c	~m ~c	lift	all-conf	coh	χ^2
A1	1000	100	100	10,000	9.26	0.91	0.83	9055
A2	100	1000	1000	100,000	8.44	0.09	0.05	670
A3	1000	100	10000	100,000	9.18	0.09	0.09	8172
A4	1000	1000	1000	1000	1	0.5	0.33	0.47

- A measure is null-invariant if its value is free from the influence of null-transactions.
- Null-invariance is an important property for measuring correlations in large transaction databases. Among the four above measures, *all confidence* and *cosine* are null-invariant measures.
- A null-transaction is a transaction that does not contain any of the itemsets being examined. In our example, *mc* represents the number of null-transactions

Comparison of Association Rule Measures

Comparison of the four correlation measures for game-and-video data sets.

<i>Data Set</i>	gv	$\bar{g}v$	$g\bar{v}$	$\bar{g}\bar{v}$	<i>all_conf.</i>	<i>cosine</i>	<i>lift</i>	χ^2
D_0	4,000	3,500	2,000	0	0.53	0.60	0.84	1,477.8
D_1	4,000	3,500	2,000	500	0.53	0.60	0.89	555.6
D_2	4,000	3,500	2,000	10,000	0.53	0.60	1.73	2,913.0

Cosine and all confidence value is greater than 0.5, we will claim that g and v are positively correlated in $D1$; however, it has been shown that they are negatively correlated by the lift and chi square analysis. Therefore, a good strategy is to perform the *all confidence* or *cosine* analysis

Which Measures Should Be Used?

- **lift** and χ^2 are not good measures for correlations in large transactional DBs
- **all-conf** or **coherence** could be good measures (Omiecinski@TKDE'03)
- Both **all-conf** and **coherence** have the downward closure property
- Efficient algorithms can be derived for mining (Lee et al. @ICDM'03sub)

symbol	measure	range	formula
ϕ	ϕ -coefficient	-1 ... 1	$\frac{P(A,B) - P(A)P(B)}{\sqrt{P(A)P(B)(1-P(A))(1-P(B))}}$
Q	Yule's Q	-1 ... 1	$\frac{P(A,B)P(\bar{A},\bar{B}) - P(A,\bar{B})P(\bar{A},B)}{P(A,B)P(\bar{A},\bar{B}) + P(A,\bar{B})P(\bar{A},B)}$
Y	Yule's Y	-1 ... 1	$\frac{\sqrt{P(A,B)P(\bar{A},\bar{B})} - \sqrt{P(A,\bar{B})P(\bar{A},B)}}{\sqrt{P(A,B)P(\bar{A},\bar{B})} + \sqrt{P(A,\bar{B})P(\bar{A},B)}}$
k	Cohen's	-1 ... 1	$\frac{P(A,B) + P(\bar{A},\bar{B}) - P(A)P(B) - P(\bar{A})P(\bar{B})}{1 - P(A)P(B) - P(\bar{A})P(\bar{B})}$
PS	Piatetsky-Shapiro's	-0.25 ... 0.25	$P(A,B) - P(A)P(B)$
F	Certainty factor	-1 ... 1	$\max(\frac{P(B A) - P(B)}{1 - P(B)}, \frac{P(A B) - P(A)}{1 - P(A)})$
AV	added value	-0.5 ... 1	$\max(P(B A) - P(B), P(A B) - P(A))$
K	Klosgen's Q	-0.33 ... 0.38	$\sqrt{P(A,B)} \max(P(B A) - P(B), P(A B) - P(A))$
g	Goodman-kruskal's	0 ... 1	$\frac{\sum_j \max_k P(A_j, B_k) + \sum_k \max_j P(A_j, B_k) - \max_j P(A_j) - \max_k P(B_k)}{2 - \max_j P(A_j) - \max_k P(B_k)}$
M	Mutual Information	0 ... 1	$\frac{\sum_i \sum_j P(A_i, B_j) \log \frac{P(A_i, B_j)}{P(A_i)P(B_j)}}{\sum_i \sum_j P(A_i, B_j) \log \frac{P(A_i, B_j)}{P(A_i)P(B_j)}}$
J	J-Measure	0 ... 1	$\min(-\sum_i P(A_i) \log P(A_i) \log P(A_i), -\sum_i P(B_i) \log P(B_i) \log P(B_i))$
G	Gini index	0 ... 1	$\max(P(A, B) \log(\frac{P(A B)}{P(A)}) + P(\bar{A}, B) \log(\frac{P(\bar{A} B)}{P(\bar{A})}),$ $P(B)[P(A B)^2 + P(\bar{A} B)^2] + P(\bar{B})[P(A \bar{B})^2 + P(\bar{A} \bar{B})^2] - P(A)^2 - P(\bar{A})^2$
s	support	0 ... 1	$P(A, B)$
c	confidence	0 ... 1	$\max(P(B A), P(A B))$
L	Laplace	0 ... 1	$\max(\frac{NP(A,B)+1}{NP(A)+2}, \frac{NP(A,B)+1}{NP(B)+2})$
IS	Cosine	0 ... 1	$\frac{P(A,B)}{\sqrt{P(A)P(B)}}$
γ	coherence(Jaccard)	0 ... 1	$\frac{P(A,B)}{P(A)+P(B)-P(A,B)}$
α	all_confidence	0 ... 1	$\frac{\max(P(A), P(B))}{P(A,B)}$
o	odds ratio	0 ... ∞	$\frac{P(A,B)P(\bar{A},\bar{B})}{P(\bar{A},B)P(A,\bar{B})}$
V	Conviction	0.5 ... ∞	$\max(\frac{P(A)P(\bar{B})}{P(A\bar{B})}, \frac{P(B)P(\bar{A})}{P(\bar{B}\bar{A})})$
λ	lift	0 ... ∞	$\frac{P(A,B)}{P(A)P(B)}$
S	Collective strength	0 ... ∞	$\frac{P(A,B) + P(\bar{A}\bar{B})}{P(A)P(B) + P(\bar{A})P(\bar{B})} \times \frac{1 - P(A)P(B) - P(\bar{A})P(\bar{B})}{1 - P(A,B) - P(\bar{A}\bar{B})}$
χ^2	χ^2	0 ... ∞	$\sum_i \frac{(P(A_i) - E_i)^2}{E_i}$

Frequent-Pattern Mining: Summary

- Frequent pattern mining—an important task in data mining
- Scalable frequent pattern mining methods
 - Apriori (Candidate generation & test)
 - Projection-based (FPgrowth, CLOSET+, ...)
 - Vertical format approach (CHARM, ...)
- Mining a variety of rules and interesting patterns
- Constraint-based mining
- Mining sequential and structured patterns
- Extensions and applications

Frequent-Pattern Mining: Research Problems

- Mining fault-tolerant frequent, sequential and structured patterns
 - Patterns allows limited faults (insertion, deletion, mutation)
- Mining truly interesting patterns
 - Surprising, novel, concise, ...
- Application exploration
 - E.g., DNA sequence analysis and bio-pattern classification
 - “Invisible” data mining

Ref: Basic Concepts of Frequent Pattern Mining

- ([Association Rules](#)) R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. SIGMOD'93.
- ([Max-pattern](#)) R. J. Bayardo. Efficiently mining long patterns from databases. SIGMOD'98.
- ([Closed-pattern](#)) N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Discovering frequent closed itemsets for association rules. ICDT'99.
- ([Sequential pattern](#)) R. Agrawal and R. Srikant. Mining sequential patterns. ICDE'95

Ref: Apriori and Its Improvements

- R. Agrawal and R. Srikant. Fast algorithms for mining association rules. VLDB'94.
- H. Mannila, H. Toivonen, and A. I. Verkamo. Efficient algorithms for discovering association rules. KDD'94.
- A. Savasere, E. Omiecinski, and S. Navathe. An efficient algorithm for mining association rules in large databases. VLDB'95.
- J. S. Park, M. S. Chen, and P. S. Yu. An effective hash-based algorithm for mining association rules. SIGMOD'95.
- H. Toivonen. Sampling large databases for association rules. VLDB'96.
- S. Brin, R. Motwani, J. D. Ullman, and S. Tsur. Dynamic itemset counting and implication rules for market basket analysis. SIGMOD'97.
- S. Sarawagi, S. Thomas, and R. Agrawal. Integrating association rule mining with relational database systems: Alternatives and implications. SIGMOD'98.