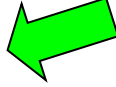

Data Mining:

Concepts and Techniques

Mining Frequent Patterns, Association and Correlations: Basic Concepts and Methods

- Basic Concepts 
- Frequent Itemset Mining Methods
- Which Patterns Are Interesting?—Pattern Evaluation Methods
- Summary

What Is Frequent Pattern Analysis?

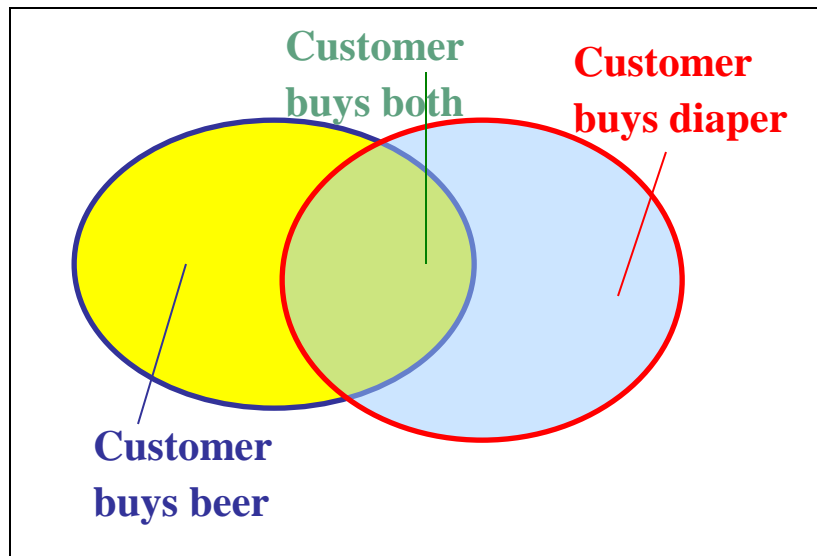
- **Frequent pattern**: a pattern (a set of items, subsequences, substructures, etc.) that occurs frequently in a data set
- First proposed by Agrawal, Imielinski, and Swami [AIS93] in the context of **frequent itemsets** and **association rule mining**
- Motivation: Finding inherent regularities in 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?

- Freq. pattern: An intrinsic and important property of datasets
- 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: discriminative, frequent pattern analysis
 - Cluster analysis: frequent pattern-based clustering
 - Data warehousing: iceberg cube and cube-gradient
 - Semantic data compression: fascicles
 - Broad applications

Basic Concepts: Frequent Patterns

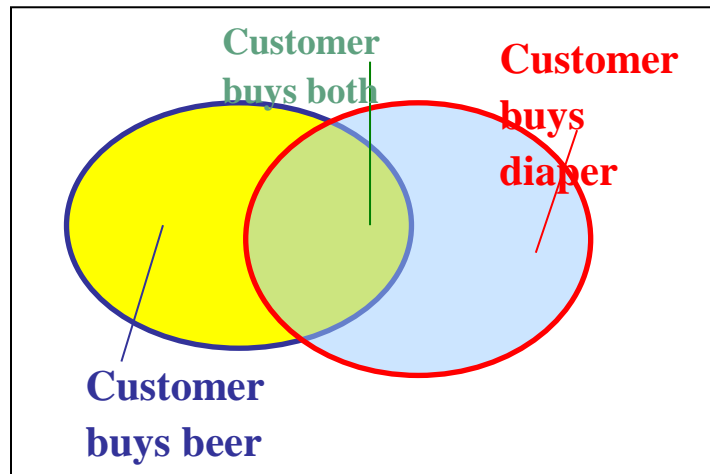
| Tid | Items bought |
|-----|----------------------------------|
| 10 | Beer, Nuts, Diaper |
| 20 | Beer, Coffee, Diaper |
| 30 | Beer, Diaper, Eggs |
| 40 | Nuts, Eggs, Milk |
| 50 | Nuts, Coffee, Diaper, Eggs, Milk |



- **itemset**: A set of one or more items
- **k-itemset** $X = \{x_1, \dots, x_k\}$
- **(absolute) support**, or, **support count** of X : Frequency or occurrence of an itemset X
- **(relative) support**, s , is the fraction of transactions that contains X (i.e., the **probability** that a transaction contains X)
- An itemset X is **frequent** if X 's support is no less than a *minsup* threshold

Basic Concepts: Association Rules

| Tid | Items bought |
|-----|----------------------------------|
| 10 | Beer, Nuts, Diaper |
| 20 | Beer, Coffee, Diaper |
| 30 | Beer, Diaper, Eggs |
| 40 | Nuts, Eggs, Milk |
| 50 | Nuts, Coffee, Diaper, Eggs, Milk |



- 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 $minsup = 50\%$, $minconf = 50\%$

Freq. Pat.: Beer:3, Nuts:3, Diaper:4, Eggs:3, {Beer, Diaper}:3

- Association rules: (many more!)
 - $Beer \rightarrow Diaper$ (60%, 100%)
 - $Diaper \rightarrow Beer$ (60%, 75%)

Closed Patterns and Max-Patterns

- A long pattern contains a combinatorial number of sub-patterns, e.g., $\{a_1, \dots, a_{100}\}$ contains $\binom{100}{1} + \binom{100}{2} + \dots + \binom{100}{100} = 2^{100} - 1 = 1.27 \times 10^{30}$ sub-patterns!
- Solution: Mine *closed patterns* and *max-patterns* instead
- An itemset X is **closed** if X is *frequent* and there exists *no* super-pattern $Y \supset X$, with the same support as X (proposed by Pasquier, et al. @ ICDT'99)
- An itemset X is a **max-pattern** if X is frequent and there exists no frequent super-pattern $Y \supset X$ (proposed by Bayardo @ SIGMOD'98)
- Closed pattern is a lossless compression of freq. patterns
 - Reducing the # of patterns and rules

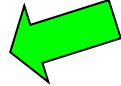
Closed Patterns and Max-Patterns

- Exercise. $DB = \{ \langle a_1, \dots, a_{100} \rangle, \langle a_1, \dots, a_{50} \rangle \}$
 - $Min_sup = 1$.
- What is the set of **closed itemset**?
 - $\langle a_1, \dots, a_{100} \rangle: 1$
 - $\langle a_1, \dots, a_{50} \rangle: 2$
- What is the set of **max-pattern**?
 - $\langle a_1, \dots, a_{100} \rangle: 1$
- What is the set of **all patterns**?
 - !!

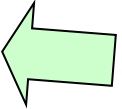
Computational Complexity of Frequent Itemset Mining

- How many itemsets are potentially to be generated in the worst case?
 - The number of frequent itemsets to be generated is sensitive to the minsup threshold
 - When minsup is low, there exist potentially an exponential number of frequent itemsets
 - The worst case: M^N where M : # distinct items, and N : max length of transactions
- The worst case complexity vs. the expected probability
 - Ex. Suppose Walmart has 10^4 kinds of products
 - The chance to pick up one product 10^{-4}
 - The chance to pick up a particular set of 10 products: $\sim 10^{-40}$
 - What is the chance this particular set of 10 products to be frequent 10^3 times in 10^9 transactions?

Mining Frequent Patterns, Association and Correlations: Basic Concepts and Methods

- Basic Concepts
- Frequent Itemset Mining Methods 
- Which Patterns Are Interesting?—Pattern Evaluation Methods
- Summary

Scalable Frequent Itemset Mining Methods

- Apriori: A Candidate Generation-and-Test Approach 
- Improving the Efficiency of Apriori
- FPGrowth: A Frequent Pattern-Growth Approach
- ECLAT: Frequent Pattern Mining with Vertical Data Format

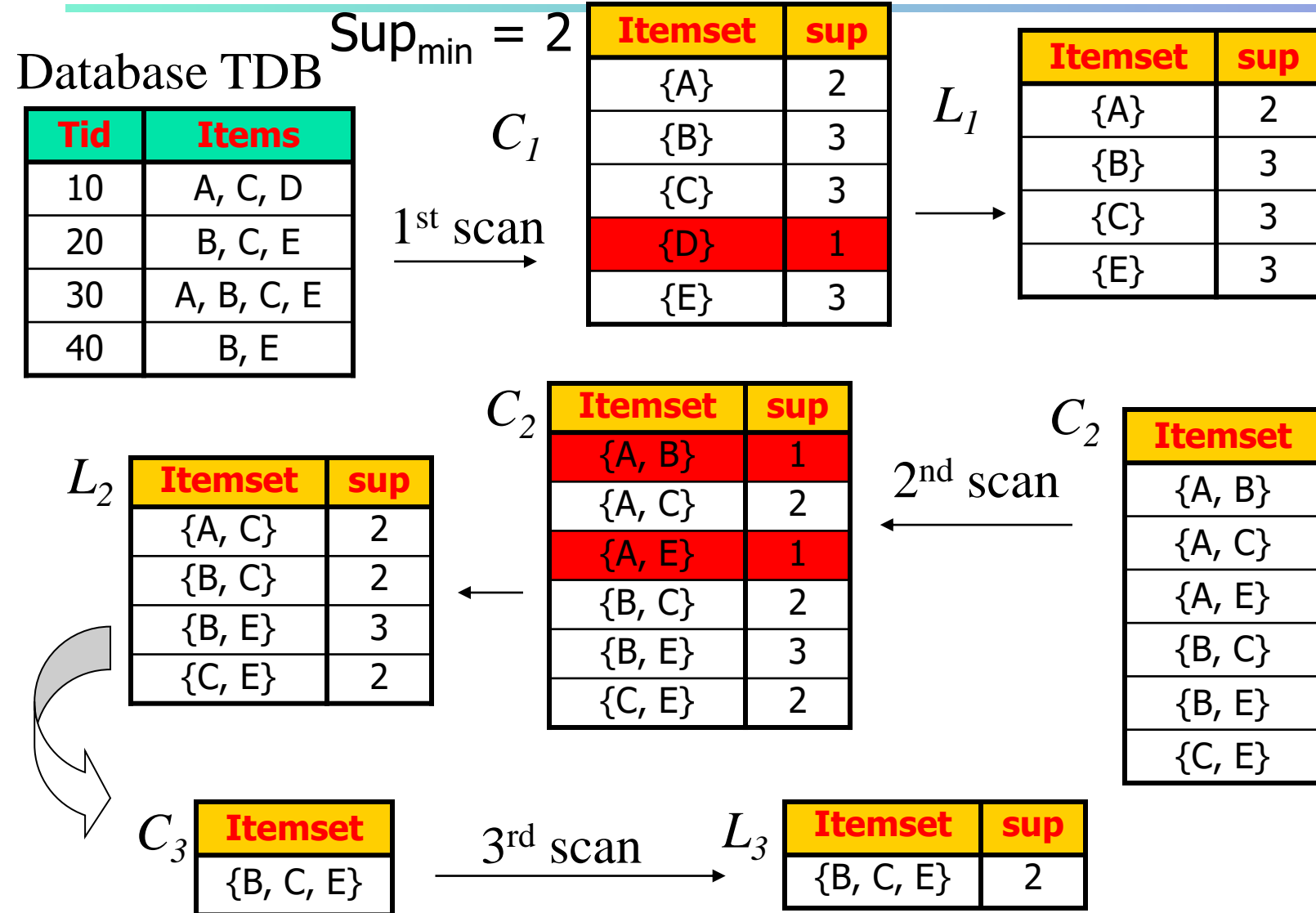
The Downward Closure Property and Scalable Mining Methods

- 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 & 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 $\cup_k L_k$;

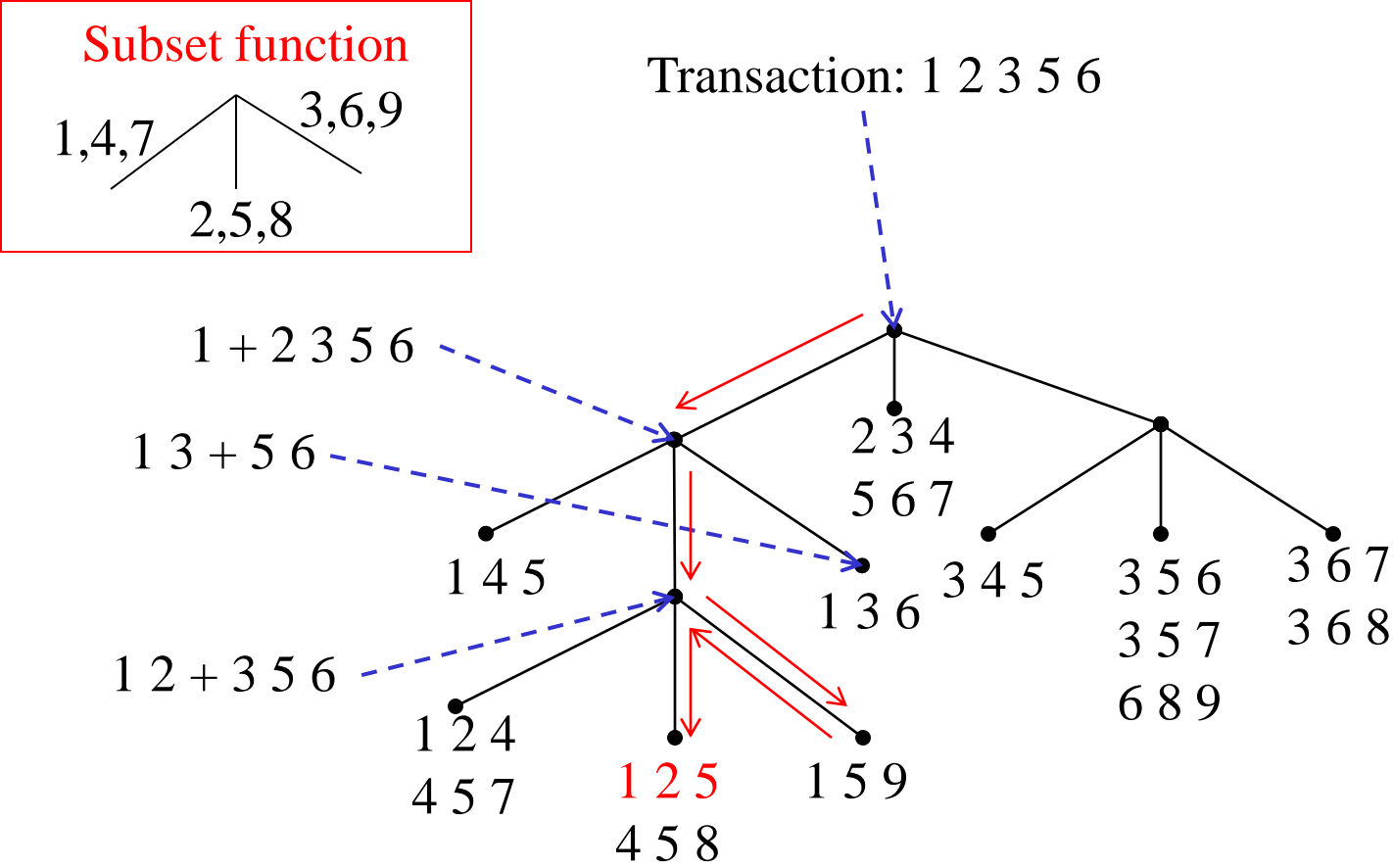
Implementation of Apriori

- How to generate candidates?
 - Step 1: self-joining L_k
 - Step 2: pruning
- 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 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

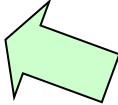
Counting Supports of Candidates Using Hash Tree



Candidate Generation: An SQL Implementation

- SQL Implementation of candidate generation
 - 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
forall ***itemsets c in C_k*** do
 forall ***($k-1$)-subsets s of c*** do
 if (s is not in L_{k-1}) then delete c from C_k
- Use object-relational extensions like UDFs, BLOBs, and Table functions for efficient implementation [See: S. Sarawagi, S. Thomas, and R. Agrawal. Integrating association rule mining with relational database systems: Alternatives and implications. SIGMOD'98]

Scalable Frequent Itemset Mining Methods

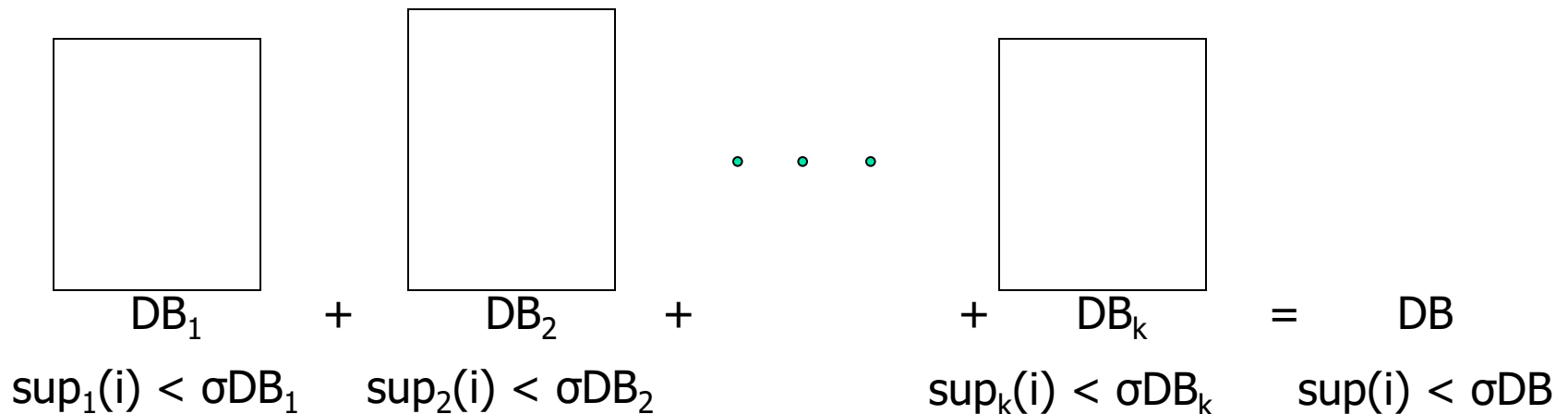
- Apriori: A Candidate Generation-and-Test Approach
- Improving the Efficiency of Apriori 
- FPGrowth: A Frequent Pattern-Growth Approach
- ECLAT: Frequent Pattern Mining with Vertical Data Format
- Mining Close Frequent Patterns and Maxpatterns

Further Improvement of the Apriori Method

- Major computational 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

Partition: Scan Database Only Twice

- Any itemset that is potentially frequent in DB must be frequent in at least one of the partitions of DB
 - Scan 1: partition database and find local frequent patterns
 - Scan 2: consolidate global frequent patterns
- A. Savasere, E. Omiecinski and S. Navathe, *VLDB'95*



DHP: Reduce the Number of Candidates

- A k -itemset whose corresponding hashing bucket count is below the threshold cannot be frequent
 - Candidates: a, b, c, d, e
 - Hash entries
 - {ab, ad, ae}
 - {bd, be, de}
 - ...
 - Frequent 1-itemset: a, b, d, e
 - ab is not a candidate 2-itemset if the sum of count of {ab, ad, ae} is below support threshold
- J. Park, M. Chen, and P. Yu. *An effective hash-based algorithm for mining association rules. SIGMOD'95*

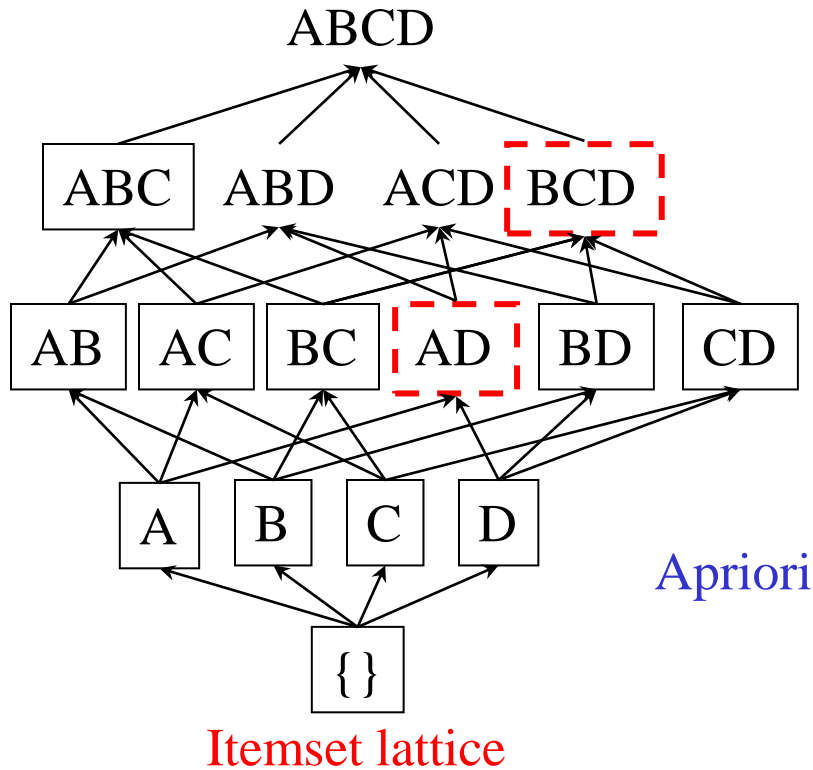
| count | itemsets |
|-------|--------------|
| 35 | {ab, ad, ae} |
| 88 | {bd, be, de} |
| . | . |
| . | . |
| . | . |
| 102 | {yz, qs, wt} |

Hash Table

Sampling for Frequent Patterns

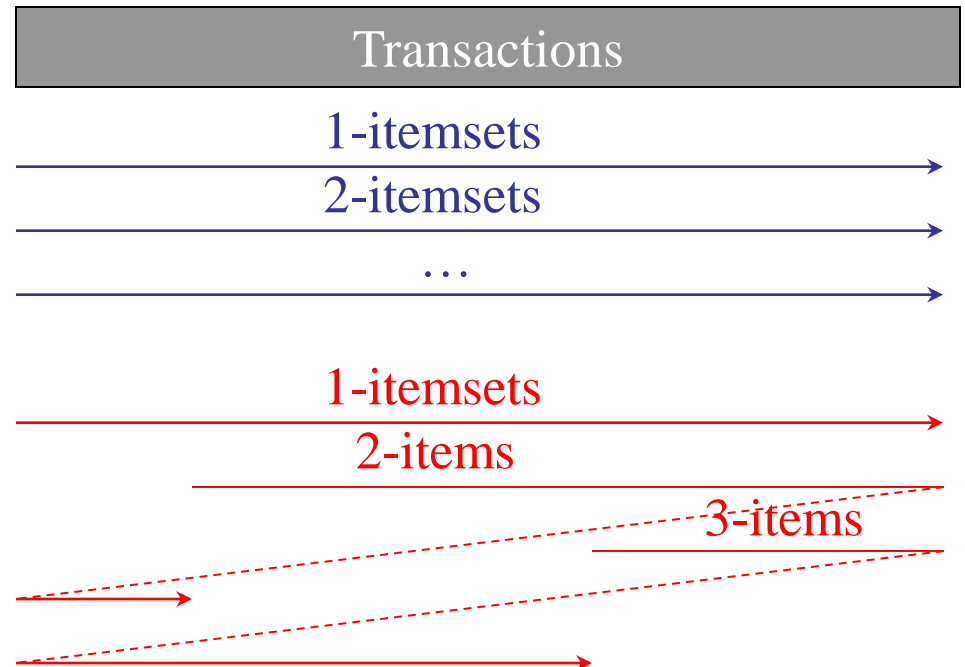
- Select a sample of original database, mine frequent patterns within sample using Apriori
- Scan database once to verify frequent itemsets found in sample, only *borders* of closure of frequent patterns are checked
 - Example: check *abcd* instead of *ab, ac, ..., etc.*
- Scan database again to find missed frequent patterns
- H. Toivonen. Sampling large databases for association rules. In *VLDB'96*

DIC: Reduce Number of Scans



Apriori

- Once both A and D are determined frequent, the counting of AD begins
- Once all length-2 subsets of BCD are determined frequent, the counting of BCD begins

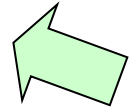


S. Brin R. Motwani, J. Ullman,
and S. Tsur. **Dynamic itemset
counting and implication rules for
market basket data.** *SIGMOD'97*

DIC

Scalable Frequent Itemset Mining Methods

- Apriori: A Candidate Generation-and-Test Approach
- Improving the Efficiency of Apriori
- FPGrowth: A Frequent Pattern-Growth Approach
- ECLAT: Frequent Pattern Mining with Vertical Data Format
- Mining Close Frequent Patterns and Maxpatterns



Pattern-Growth Approach: Mining Frequent Patterns Without Candidate Generation

- Bottlenecks of the Apriori approach
 - Breadth-first (i.e., level-wise) search
 - Candidate generation and test
 - Often generates a huge number of candidates
- The FPGrowth Approach (J. Han, J. Pei, and Y. Yin, SIGMOD' 00)
 - Depth-first search
 - Avoid explicit candidate generation
- Major philosophy: Grow long patterns from short ones using local frequent items only
 - "abc" is a frequent pattern
 - Get all transactions having "abc", i.e., project DB on 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} |

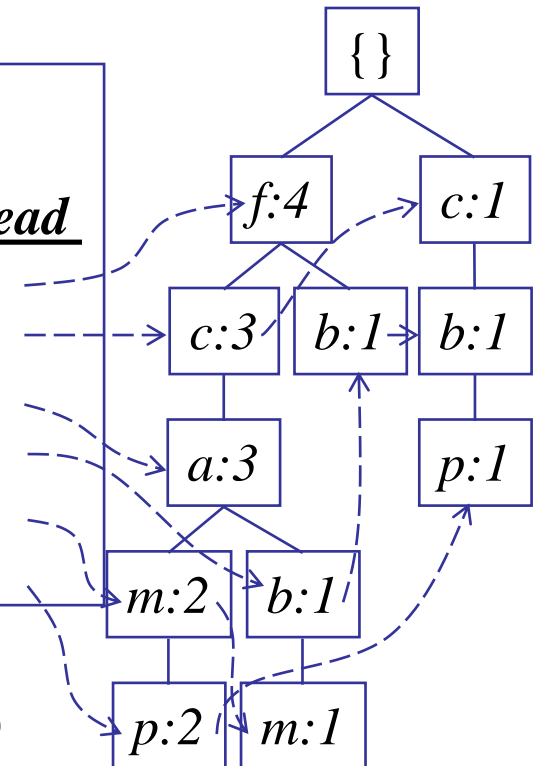
min_support = 3

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

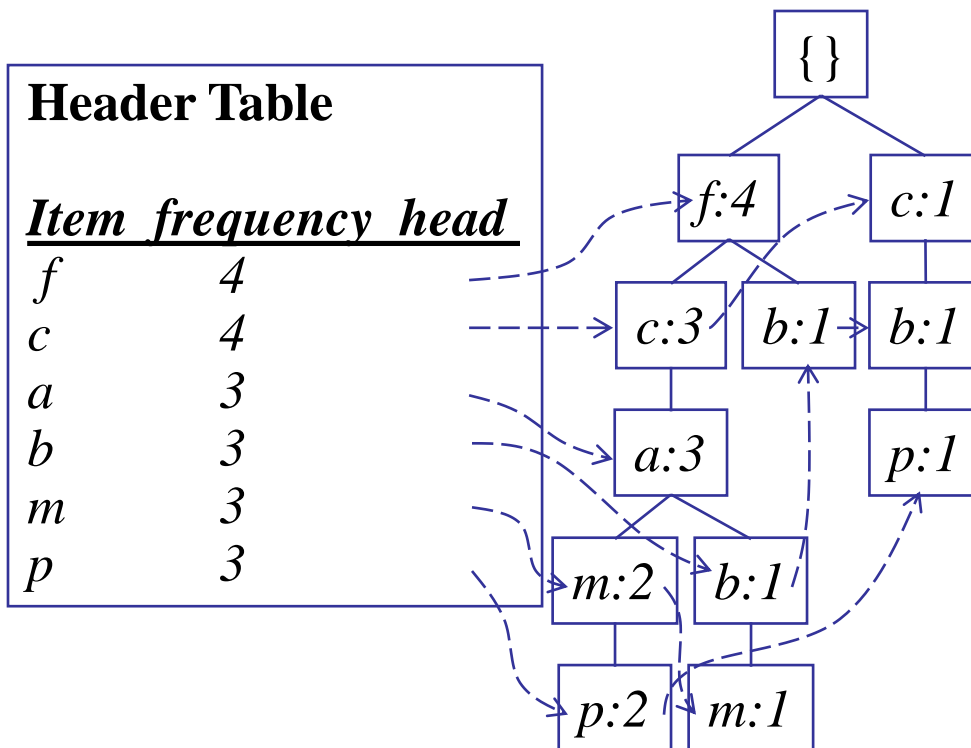


Partition Patterns and Databases

- 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



Conditional pattern bases

item cond. pattern base

c $f:3$

a $fc:3$

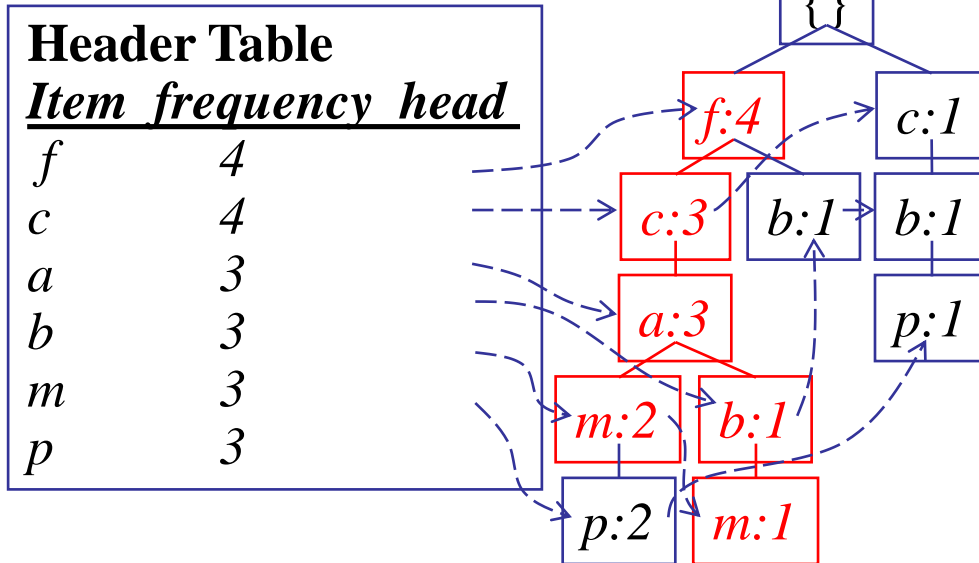
b $fca:1, f:1, c:1$

m $fca:2, fcab:1$

p $fcam:2, cb:1$

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



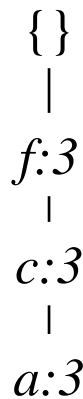
m-conditional pattern base:
fca:2, fcab:1



$\begin{array}{c} \{\} \\ | \\ f:3 \\ | \\ c:3 \\ | \\ a:3 \end{array} \rightarrow \begin{array}{l} \text{All frequent} \\ \text{patterns relate to } m \\ m, \\ fm, cm, am, \\ fcm, fam, cam, \\ fcam \end{array}$

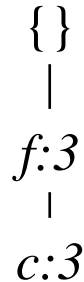
m-conditional FP-tree

Recursion: Mining Each Conditional FP-tree



m-conditional FP-tree

Cond. pattern base of "am": (fc:3)



am-conditional FP-tree

Cond. pattern base of "cm": (f:3)



cm-conditional FP-tree

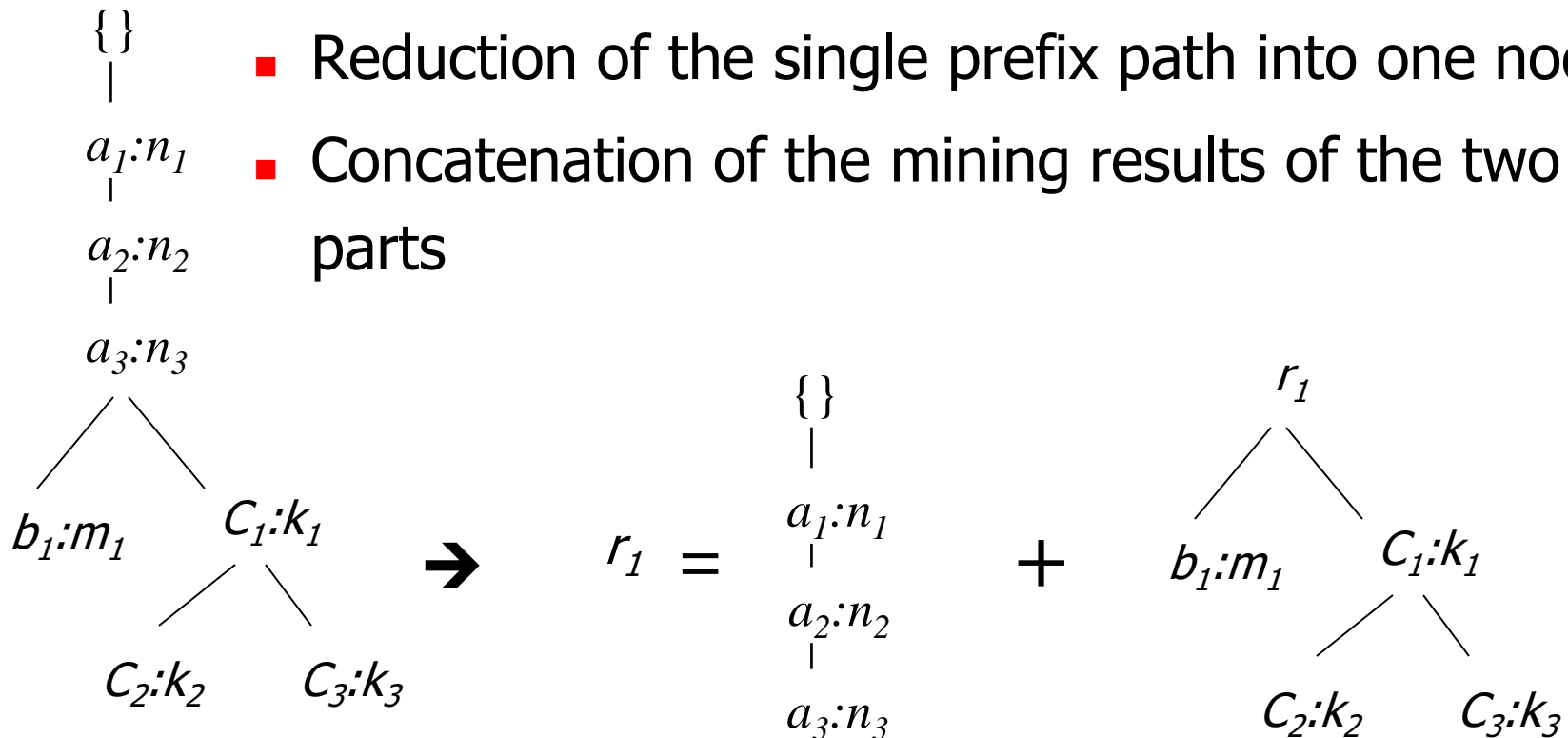
Cond. pattern base of "cam": (f:3)



cam-conditional FP-tree

A Special Case: Single Prefix Path in FP-tree

- Suppose a (conditional) FP-tree T has a shared single prefix-path P
- Mining can be decomposed into two parts
 - Reduction of the single prefix path into one node
 - Concatenation of the mining results of the two parts



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)

The Frequent Pattern Growth Mining Method

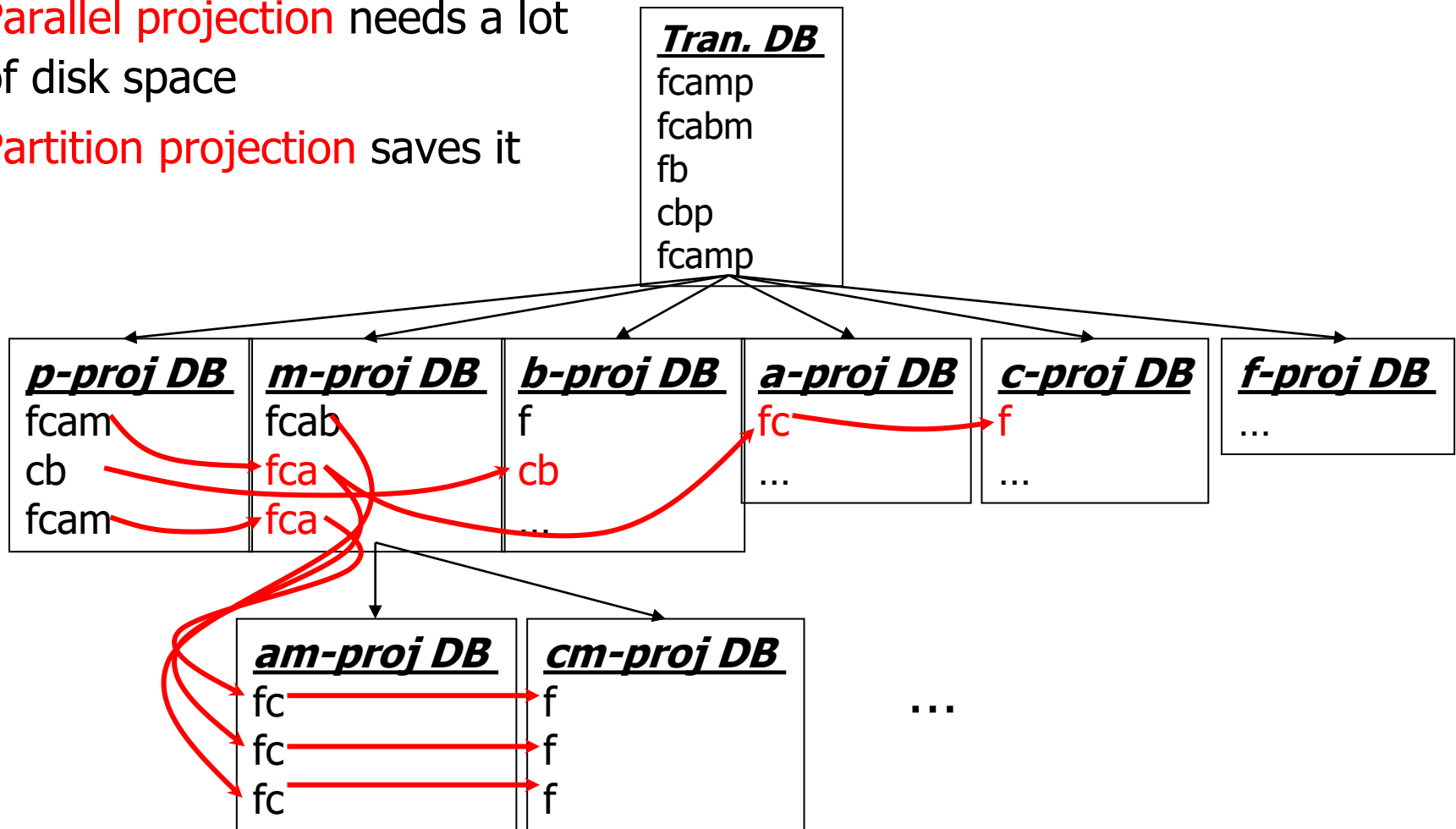
- Idea: Frequent pattern growth
 - Recursively grow frequent patterns by pattern and database partition
- Method
 - For each frequent item, construct its conditional pattern-base, and then its conditional FP-tree
 - Repeat the process on each newly created conditional FP-tree
 - Until the resulting FP-tree is empty, or it contains only one path—single path will generate all the combinations of its sub-paths, each of which is a frequent pattern

Scaling FP-growth by Database Projection

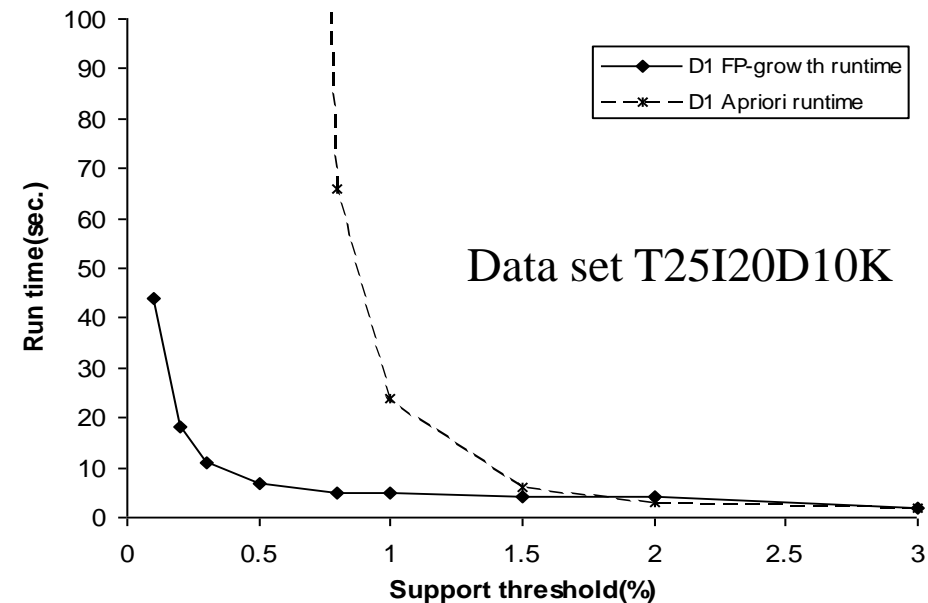
- What about if FP-tree cannot fit in memory?
 - DB projection
- First partition a database into a set of projected DBs
- Then construct and mine FP-tree for each projected DB
- **Parallel projection** vs. **partition projection** techniques
 - Parallel projection
 - Project the DB in parallel for each frequent item
 - Parallel projection is space costly
 - All the partitions can be processed in parallel
 - Partition projection
 - Partition the DB based on the ordered frequent items
 - Passing the unprocessed parts to the subsequent partitions

Partition-Based Projection

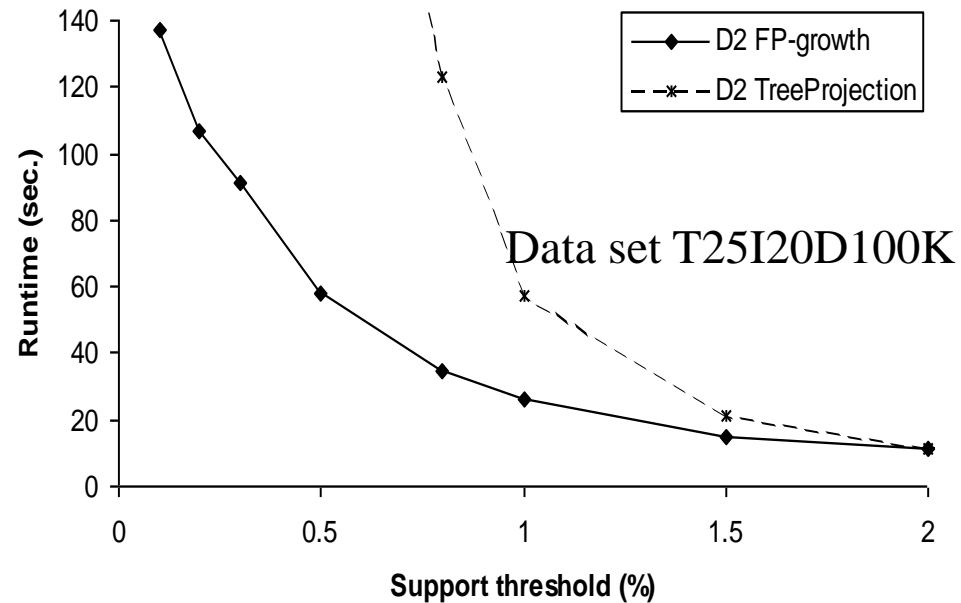
- **Parallel projection** needs a lot of disk space
- **Partition projection** saves it



Performance of FPGrowth in Large Datasets



FP-Growth vs. Apriori



FP-Growth vs. Tree-Projection

Advantages of the Pattern Growth Approach

- Divide-and-conquer:
 - Decompose both the mining task and DB according to the frequent patterns obtained so far
 - Lead to focused search of smaller databases
- Other factors
 - No candidate generation, no candidate test
 - Compressed database: FP-tree structure
 - No repeated scan of entire database
 - Basic ops: counting local freq items and building sub FP-tree, no pattern search and matching
- A good open-source implementation and refinement of FPGrowth
 - FPGrowth+ (Grahne and J. Zhu, FIMI'03)

Further Improvements of Mining Methods

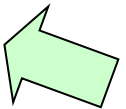
- AFOPT (Liu, et al. @ KDD'03)
 - A “push-right” method for mining condensed frequent pattern (CFP) tree
- Carpenter (Pan, et al. @ KDD'03)
 - Mine data sets with small rows but numerous columns
 - Construct a row-enumeration tree for efficient mining
- FPgrowth+ (Grahne and Zhu, FIMI'03)
 - Efficiently Using Prefix-Trees in Mining Frequent Itemsets, Proc. ICDM'03 Int. Workshop on Frequent Itemset Mining Implementations (FIMI'03), Melbourne, FL, Nov. 2003
- TD-Close (Liu, et al, SDM'06)

Extension of Pattern Growth Mining Methodology

- Mining closed frequent itemsets and max-patterns
 - CLOSET (DMKD'00), FPclose, and FPMMax (Grahne & Zhu, Fimi'03)
- Mining sequential patterns
 - PrefixSpan (ICDE'01), CloSpan (SDM'03), BIDE (ICDE'04)
- Mining graph patterns
 - gSpan (ICDM'02), CloseGraph (KDD'03)
- Constraint-based mining of frequent patterns
 - Convertible constraints (ICDE'01), gPrune (PAKDD'03)
- Computing iceberg data cubes with complex measures
 - H-tree, H-cubing, and Star-cubing (SIGMOD'01, VLDB'03)
- Pattern-growth-based Clustering
 - MaPle (Pei, et al., ICDM'03)
- Pattern-Growth-Based Classification
 - Mining frequent and discriminative patterns (Cheng, et al, ICDE'07)

Scalable Frequent Itemset Mining Methods

- Apriori: A Candidate Generation-and-Test Approach
- Improving the Efficiency of Apriori
- FPGrowth: A Frequent Pattern-Growth Approach
- ECLAT: Frequent Pattern Mining with Vertical Data Format
- Mining Close Frequent Patterns and Maxpatterns

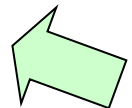


ECLAT: Mining by Exploring Vertical Data Format

- Vertical format: $t(AB) = \{T_{11}, T_{25}, \dots\}$
 - tid-list: list of trans.-ids containing an itemset
- Deriving frequent patterns based on vertical intersections
 - $t(X) = t(Y)$: X and Y always happen together
 - $t(X) \subset t(Y)$: transaction having X always has Y
- Using **diffset** to accelerate mining
 - Only keep track of differences of tids
 - $t(X) = \{T_1, T_2, T_3\}$, $t(XY) = \{T_1, T_3\}$
 - $\text{Diffset}(XY, X) = \{T_2\}$
- Eclat (Zaki et al. @KDD'97)
- Mining Closed patterns using vertical format: CHARM (Zaki & Hsiao@SDM'02)

Scalable Frequent Itemset Mining Methods

- Apriori: A Candidate Generation-and-Test Approach
- Improving the Efficiency of Apriori
- FPGrowth: A Frequent Pattern-Growth Approach
- ECLAT: Frequent Pattern Mining with Vertical Data Format
- Mining Close Frequent Patterns and Maxpatterns



Mining Frequent Closed Patterns: CLOSET

- Flist: list of all frequent items in support ascending order
 - Flist: d-a-f-e-c
- Divide search space
 - Patterns having d
 - Patterns having d but no a, etc.
- Find frequent closed pattern recursively
 - Every transaction having d also has *cfa* → *cfad* is a frequent closed pattern
- J. Pei, J. Han & R. Mao. "CLOSET: An Efficient Algorithm for Mining Frequent Closed Itemsets", DMKD'00.

Min_sup=2

| TID | Items |
|-----|---------------|
| 10 | a, c, d, e, f |
| 20 | a, b, e |
| 30 | c, e, f |
| 40 | a, c, d, f |
| 50 | c, e, f |

CLOSET+: Mining Closed Itemsets by Pattern-Growth

- Itemset merging: if Y appears in every occurrence of X , then Y is merged with X
- Sub-itemset pruning: if $Y \supset X$, and $\text{sup}(X) = \text{sup}(Y)$, X and all of X 's descendants in the set enumeration tree can be pruned
- Hybrid tree projection
 - Bottom-up physical tree-projection
 - Top-down pseudo tree-projection
- Item skipping: if a local frequent item has the same support in several header tables at different levels, one can prune it from the header table at higher levels
- Efficient subset checking

MaxMiner: Mining Max-Patterns

- 1st scan: find frequent items

- A, B, C, D, E

- 2nd scan: find support for

- AB, AC, AD, AE, ABCDE

- BC, BD, BE, BCDE

- CD, CE, CDE, DE

Potential
max-patterns



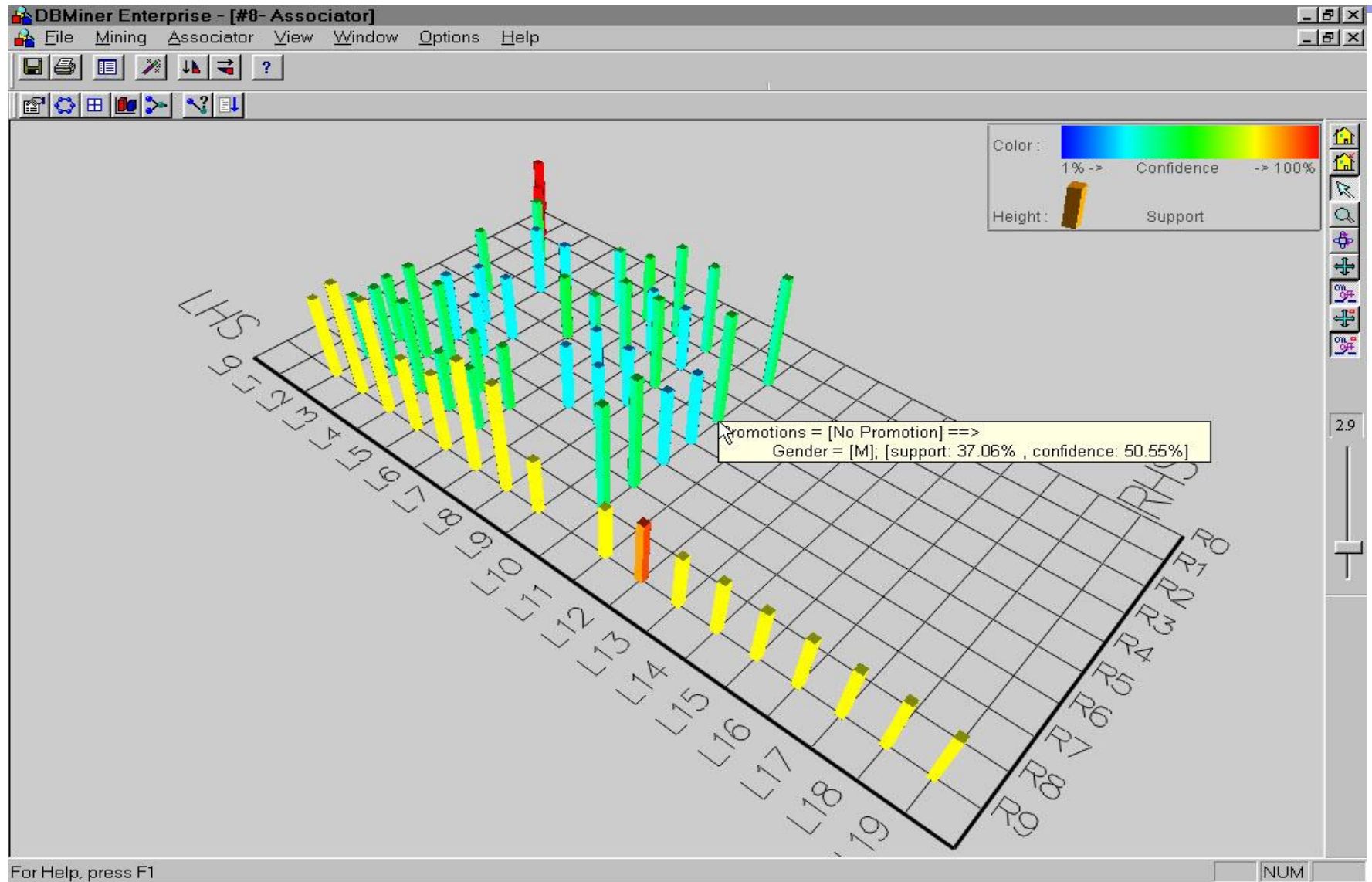
- Since BCDE is a max-pattern, no need to check BCD, BDE, CDE in later scan
- R. Bayardo. Efficiently mining long patterns from databases. *SIGMOD'98*

| Tid | Items |
|-----|---------------|
| 10 | A, B, C, D, E |
| 20 | B, C, D, E, |
| 30 | A, C, D, F |

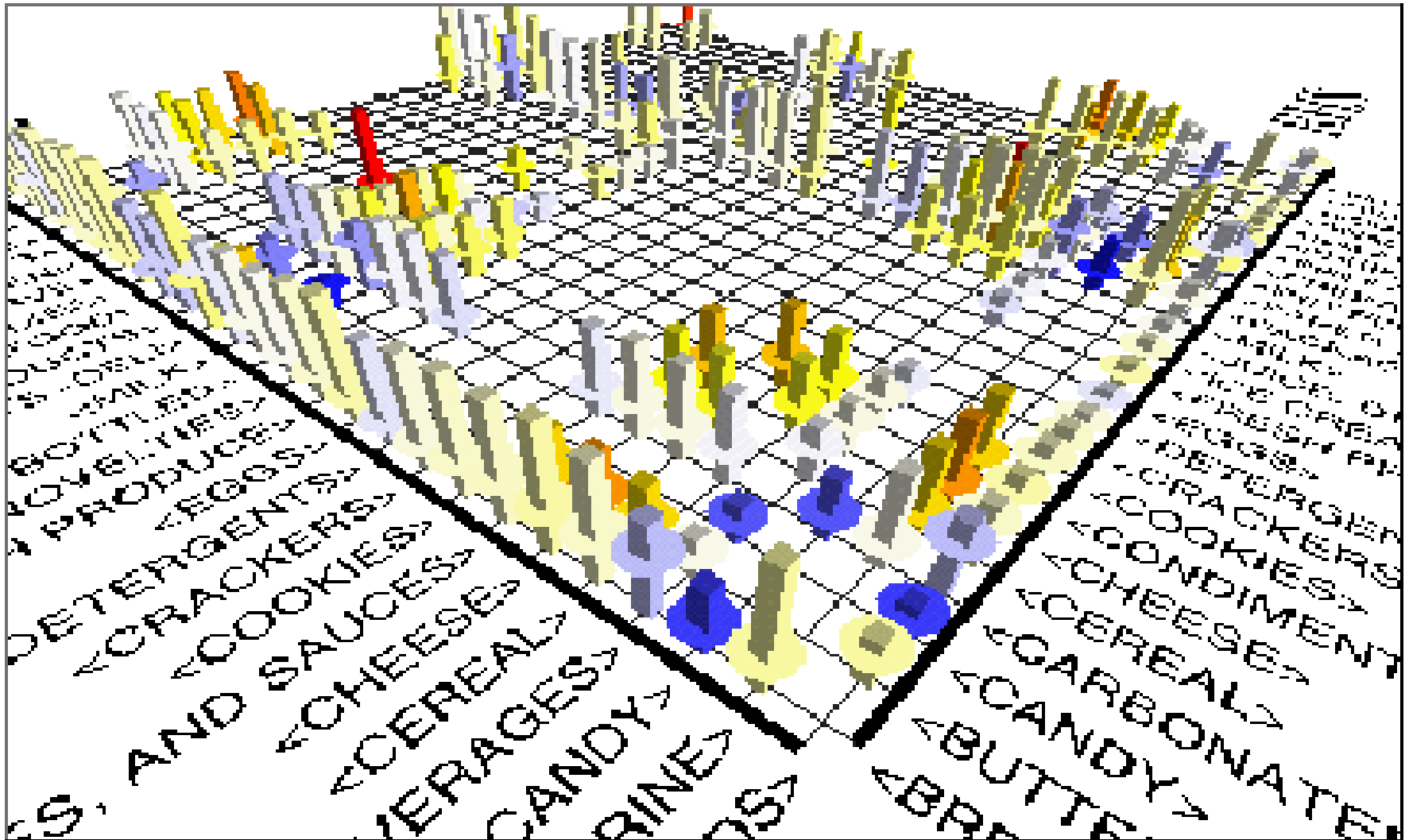
CHARM: Mining by Exploring Vertical Data Format

- Vertical format: $t(AB) = \{T_{11}, T_{25}, \dots\}$
 - tid-list: list of trans.-ids containing an itemset
- Deriving closed patterns based on vertical intersections
 - $t(X) = t(Y)$: X and Y always happen together
 - $t(X) \subset t(Y)$: transaction having X always has Y
- Using **diffset** to accelerate mining
 - Only keep track of differences of tids
 - $t(X) = \{T_1, T_2, T_3\}$, $t(XY) = \{T_1, T_3\}$
 - $\text{Diffset}(XY, X) = \{T_2\}$
- Eclat/MaxEclat (Zaki et al. @KDD'97), VIPER(P. Shenoy et al. @SIGMOD'00), CHARM (Zaki & Hsiao @SDM'02)

Visualization of Association Rules: Plane Graph

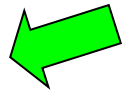


Visualization of Association Rules (SGI/MineSet 3.0)



Mining Frequent Patterns, Association and Correlations: Basic Concepts and Methods

- Basic Concepts
- Frequent Itemset Mining Methods
- Which Patterns Are Interesting?—Pattern
Evaluation Methods
- Summary



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**

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

$$lift(B, C) = \frac{2000 / 5000}{3000 / 5000 * 3750 / 5000} = 0.89$$

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

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

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 indicate correlations
- Over 20 interestingness measures have been proposed (see Tan, Kumar, Sritastava @KDD'02)
- Which are good ones?

| 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 | Klogsen'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 | $\max(P(A, B) \log(\frac{P(A B)}{P(A)}) + P(\bar{A}B) \log(\frac{P(\bar{A} B)}{P(\bar{A})}),$ $P(B)[P(A B)^2 + P(\bar{A} B)^2] + P(\bar{A})[P(B \bar{A})^2 + P(\bar{B} \bar{A})^2] - P(B)^2 - P(\bar{B})^2,$ |
| 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 | $\frac{P(A,B)}{\max(P(A), P(B))}$ |
| 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 ... ∞ | $\max(\frac{P(A)P(\bar{B})}{P(A\bar{B})}, \frac{P(B)P(\bar{A})}{P(\bar{B}A)})$ |
| 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 ... ∞ | $\sum_i \frac{(P(A_i) - E_i)^2}{E_i}$ |

Null-Invariant Measures

Table 6: Properties of interestingness measures. Note that none of the measures satisfies all the properties.

| Symbol | Measure | Range | P1 | P2 | P3 | O1 | O2 | O3 | O3' | O4 |
|-----------|---------------------|--|------|-----|-----|------|-----|------|-----|-----|
| ϕ | ϕ -coefficient | $-1 \dots 0 \dots 1$ | Yes | Yes | Yes | Yes | No | Yes | Yes | No |
| λ | Goodman-Kruskal's | $0 \dots 1$ | Yes | No | No | Yes | No | No* | Yes | No |
| α | odds ratio | $0 \dots 1 \dots \infty$ | Yes* | Yes | Yes | Yes | Yes | Yes* | Yes | No |
| Q | Yule's Q | $-1 \dots 0 \dots 1$ | Yes | Yes | Yes | Yes | Yes | Yes | Yes | No |
| Y | Yule's Y | $-1 \dots 0 \dots 1$ | Yes | Yes | Yes | Yes | Yes | Yes | Yes | No |
| κ | Cohen's | $-1 \dots 0 \dots 1$ | Yes | Yes | Yes | Yes | No | No | Yes | No |
| M | Mutual Information | $0 \dots 1$ | Yes | Yes | Yes | No** | No | No* | Yes | No |
| J | J-Measure | $0 \dots 1$ | Yes | No | No | No** | No | No | No | No |
| G | Gini index | $0 \dots 1$ | Yes | No | No | No** | No | No* | Yes | No |
| s | Support | $0 \dots 1$ | No | Yes | No | Yes | No | No | No | No |
| c | Confidence | $0 \dots 1$ | No | Yes | No | No** | No | No | No | Yes |
| L | Laplace | $0 \dots 1$ | No | Yes | No | No** | No | No | No | No |
| V | Conviction | $0.5 \dots 1 \dots \infty$ | No | Yes | No | No** | No | No | Yes | No |
| I | Interest | $0 \dots 1 \dots \infty$ | Yes* | Yes | Yes | Yes | No | No | No | No |
| IS | Cosine | $0 \dots \sqrt{P(A, B)} \dots 1$ | No | Yes | Yes | Yes | No | No | No | Yes |
| PS | Piatetsky-Shapiro's | $-0.25 \dots 0 \dots 0.25$ | Yes | Yes | Yes | Yes | No | Yes | Yes | No |
| F | Certainty factor | $-1 \dots 0 \dots 1$ | Yes | Yes | Yes | No** | No | No | Yes | No |
| AV | Added value | $-0.5 \dots 0 \dots 1$ | Yes | Yes | Yes | No** | No | No | No | No |
| S | Collective strength | $0 \dots 1 \dots \infty$ | No | Yes | Yes | Yes | No | Yes* | Yes | No |
| ζ | Jaccard | $0 \dots 1$ | No | Yes | Yes | Yes | No | No | No | Yes |
| K | Klosgen's | $(\frac{2}{\sqrt{3}} - 1)^{1/2} [2 - \sqrt{3} - \frac{1}{\sqrt{3}}] \dots 0 \dots \frac{2}{3\sqrt{3}}$ | Yes | Yes | Yes | No** | No | No | No | No |

where: P1: $O(M) = 0$ if $\det(M) = 0$, i.e., whenever A and B are statistically independent.

P2: $O(M_2) > O(M_1)$ if $M_2 = M_1 + [k \ -k; \ -k \ k]$.

P3: $O(M_2) < O(M_1)$ if $M_2 = M_1 + [0 \ k; \ 0 \ -k]$ or $M_2 = M_1 + [0 \ 0; \ k \ -k]$.

O1: Property 1: Symmetry under variable permutation.

O2: Property 2: Row and Column scaling invariance.

O3: Property 3: Antisymmetry under row or column permutation.

O3': Property 4: Inversion invariance.

O4: Property 5: Null invariance.

Yes*: Yes if measure is normalized.

No*: Symmetry under row or column permutation.

No**: No unless the measure is symmetrized by taking $\max(M(A, B), M(B, A))$.

Comparison of Interestingness Measures

- Null-(transaction) invariance is crucial for correlation analysis
- Lift and χ^2 are not null-invariant
- 5 null-invariant measures

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

| Measure | Definition | Range | Null-Invariant |
|-------------------|--|---------------|----------------|
| $\chi^2(a, b)$ | $\sum_{i,j=0,1} \frac{(e(a_i, b_j) - o(a_i, b_j))^2}{e(a_i, b_j)}$ | $[0, \infty]$ | No |
| $Lift(a, b)$ | $\frac{P(ab)}{P(a)P(b)}$ | $[0, \infty]$ | No |
| $AllConf(a, b)$ | $\frac{sup(ab)}{\max\{sup(a), sup(b)\}}$ | $[0, 1]$ | Yes |
| $Coherence(a, b)$ | $\frac{sup(ab)}{sup(a) + sup(b) - sup(ab)}$ | $[0, 1]$ | Yes |
| $Cosine(a, b)$ | $\frac{sup(ab)}{\sqrt{sup(a)sup(b)}}$ | $[0, 1]$ | Yes |
| $Kulc(a, b)$ | $\frac{sup(ab)}{2} \left(\frac{1}{sup(a)} + \frac{1}{sup(b)} \right)$ | $[0, 1]$ | Yes |
| $MaxConf(a, b)$ | $\max\left\{ \frac{sup(ab)}{sup(a)}, \frac{sup(ab)}{sup(b)} \right\}$ | $[0, 1]$ | Yes |

Null-transactions
w.r.t. m and c

Kulczynski
measure (1927)

Table 3. Interestingness measure definitions.

Null-invariant

| Data set | mc | $\bar{m}\bar{c}$ | $m\bar{c}$ | $\bar{m}c$ | χ^2 | $Lift$ | $AllConf$ | $Coherence$ | $Cosine$ | $Kulc$ | $MaxConf$ |
|----------|--------|------------------|------------|------------|----------|--------|-----------|-------------|----------|--------|-----------|
| D_1 | 10,000 | 1,000 | 1,000 | 100,000 | 90557 | 9.26 | 0.91 | 0.83 | 0.91 | 0.91 | 0.91 |
| D_2 | 10,000 | 1,000 | 1,000 | 100 | 0 | 1 | 0.91 | 0.83 | 0.91 | 0.91 | 0.91 |
| D_3 | 100 | 1,000 | 1,000 | 100,000 | 670 | 8.44 | 0.09 | 0.05 | 0.09 | 0.09 | 0.09 |
| D_4 | 1,000 | 1,000 | 1,000 | 100,000 | 24740 | 25.75 | 0.5 | 0.33 | 0.5 | 0.5 | 0.5 |
| D_5 | 1,000 | 100 | 10,000 | 100,000 | 8173 | 9.18 | 0.09 | 0.09 | 0.29 | 0.5 | 0.91 |
| D_6 | 1,000 | 10 | 100,000 | 100,000 | 965 | 1.97 | 0.01 | 0.01 | 0.10 | 0.5 | 0.99 |

Table 2. Example data sets.

Subtle: They disagree

Analysis of DBLP Coauthor Relationships

Recent DB conferences, removing balanced associations, low sup, etc.

| ID | Author <i>a</i> | Author <i>b</i> | <i>sup(ab)</i> | <i>sup(a)</i> | <i>sup(b)</i> | <i>Coherence</i> | <i>Cosine</i> | <i>Kulc</i> |
|----|----------------------|----------------------|----------------|---------------|---------------|------------------|---------------|-------------|
| 1 | Hans-Peter Kriegel | Martin Ester | 28 | 146 | 54 | 0.163 (2) | 0.315 (7) | 0.355 (9) |
| 2 | Michael Carey | Miron Livny | 26 | 104 | 58 | 0.191 (1) | 0.335 (4) | 0.349 (10) |
| 3 | Hans-Peter Kriegel | Joerg Sander | 24 | 146 | 36 | 0.152 (3) | 0.331 (5) | 0.416 (8) |
| 4 | Christos Faloutsos | Spiros Papadimitriou | 20 | 162 | 26 | 0.119 (7) | 0.308 (10) | 0.446 (7) |
| 5 | Hans-Peter Kriegel | Martin Pfeifle | 18 | 146 | 18 | 0.123 (6) | 0.351 (2) | 0.562 (2) |
| 6 | Hector Garcia-Molina | Wilburt Labio | 16 | 144 | 18 | 0.110 (9) | 0.314 (8) | 0.500 (4) |
| 7 | Divyakant Agrawal | Wang Hsiung | 16 | 120 | 16 | 0.133 (5) | 0.365 (1) | 0.567 (1) |
| 8 | Elke Rundensteiner | Murali Mani | 16 | 104 | 20 | 0.148 (4) | 0.351 (3) | 0.477 (6) |
| 9 | Divyakant Agrawal | Oliver Po | 12 | 120 | 12 | 0.100 (10) | 0.316 (6) | 0.550 (3) |
| 10 | Gerhard Weikum | Martin Theobald | 12 | 106 | 14 | 0.111 (8) | 0.312 (9) | 0.485 (5) |

Table 5. Experiment on DBLP data set.

Advisor-advisee relation: Kulc: high,
coherence: low, cosine: middle

- Tianyi Wu, Yuguo Chen and Jiawei Han, "[Association Mining in Large Databases: A Re-Examination of Its Measures](#)", Proc. 2007 Int. Conf. Principles and Practice of Knowledge Discovery in Databases (PKDD'07), Sept. 2007

Which Null-Invariant Measure Is Better?

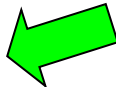
- IR (Imbalance Ratio): measure the imbalance of two itemsets A and B in rule implications

$$IR(A, B) = \frac{|sup(A) - sup(B)|}{sup(A) + sup(B) - sup(A \cup B)}$$

- Kulczynski and Imbalance Ratio (IR) together present a clear picture for all the three datasets D_4 through D_6
 - D_4 is balanced & neutral
 - D_5 is imbalanced & neutral
 - D_6 is very imbalanced & neutral

| <i>Data</i> | <i>mc</i> | \overline{mc} | $m\overline{c}$ | $\overline{m\overline{c}}$ | <i>all_conf.</i> | <i>max_conf.</i> | <i>Kulc.</i> | <i>cosine</i> | IR |
|-------------|-----------|-----------------|-----------------|----------------------------|------------------|------------------|--------------|---------------|------|
| D_1 | 10,000 | 1,000 | 1,000 | 100,000 | 0.91 | 0.91 | 0.91 | 0.91 | 0.0 |
| D_2 | 10,000 | 1,000 | 1,000 | 100 | 0.91 | 0.91 | 0.91 | 0.91 | 0.0 |
| D_3 | 100 | 1,000 | 1,000 | 100,000 | 0.09 | 0.09 | 0.09 | 0.09 | 0.0 |
| D_4 | 1,000 | 1,000 | 1,000 | 100,000 | 0.5 | 0.5 | 0.5 | 0.5 | 0.0 |
| D_5 | 1,000 | 100 | 10,000 | 100,000 | 0.09 | 0.91 | 0.5 | 0.29 | 0.89 |
| D_6 | 1,000 | 10 | 100,000 | 100,000 | 0.01 | 0.99 | 0.5 | 0.10 | 0.99 |

Mining Frequent Patterns, Association and Correlations: Basic Concepts and Methods

- Basic Concepts
- Frequent Itemset Mining Methods
- Which Patterns Are Interesting?—Pattern Evaluation Methods
- Summary 

Summary

- Basic concepts: association rules, support-confident framework, closed and max-patterns
- Scalable frequent pattern mining methods
 - Apriori (Candidate generation & test)
 - Projection-based (FPgrowth, CLOSET+, ...)
 - Vertical format approach (ECLAT, CHARM, ...)
- Which patterns are interesting?
 - Pattern evaluation methods

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

Ref: Depth-First, Projection-Based FP Mining

- R. Agarwal, C. Aggarwal, and V. V. V. Prasad. A tree projection algorithm for generation of frequent itemsets. *J. Parallel and Distributed Computing*, 2002.
- G. Grahne and J. Zhu, Efficiently Using Prefix-Trees in Mining Frequent Itemsets, *Proc. FIMI'03*
- B. Goethals and M. Zaki. An introduction to workshop on frequent itemset mining implementations. *Proc. ICDM'03 Int. Workshop on Frequent Itemset Mining Implementations (FIMI'03)*, Melbourne, FL, Nov. 2003
- J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. *SIGMOD' 00*
- J. Liu, Y. Pan, K. Wang, and J. Han. Mining Frequent Item Sets by Opportunistic Projection. *KDD'02*
- J. Han, J. Wang, Y. Lu, and P. Tzvetkov. Mining Top-K Frequent Closed Patterns without Minimum Support. *ICDM'02*
- J. Wang, J. Han, and J. Pei. CLOSET+: Searching for the Best Strategies for Mining Frequent Closed Itemsets. *KDD'03*

Ref: Vertical Format and Row Enumeration Methods

- M. J. Zaki, S. Parthasarathy, M. Ogihara, and W. Li. Parallel algorithm for discovery of association rules. DAMI:97.
- M. J. Zaki and C. J. Hsiao. CHARM: An Efficient Algorithm for Closed Itemset Mining, SDM'02.
- C. Bucila, J. Gehrke, D. Kifer, and W. White. DualMiner: A Dual-Pruning Algorithm for Itemsets with Constraints. KDD'02.
- F. Pan, G. Cong, A. K. H. Tung, J. Yang, and M. Zaki , CARPENTER: Finding Closed Patterns in Long Biological Datasets. KDD'03.
- H. Liu, J. Han, D. Xin, and Z. Shao, Mining Interesting Patterns from Very High Dimensional Data: A Top-Down Row Enumeration Approach, SDM'06.

Ref: Mining Correlations and Interesting Rules

- S. Brin, R. Motwani, and C. Silverstein. Beyond market basket: Generalizing association rules to correlations. SIGMOD'97.
- M. Klemettinen, H. Mannila, P. Ronkainen, H. Toivonen, and A. I. Verkamo. Finding interesting rules from large sets of discovered association rules. CIKM'94.
- R. J. Hilderman and H. J. Hamilton. *Knowledge Discovery and Measures of Interest*. Kluwer Academic, 2001.
- C. Silverstein, S. Brin, R. Motwani, and J. Ullman. Scalable techniques for mining causal structures. VLDB'98.
- P.-N. Tan, V. Kumar, and J. Srivastava. Selecting the Right Interestingness Measure for Association Patterns. KDD'02.
- E. Omiecinski. Alternative Interest Measures for Mining Associations. TKDE'03.
- T. Wu, Y. Chen, and J. Han, "Re-Examination of Interestingness Measures in Pattern Mining: A Unified Framework", Data Mining and Knowledge Discovery, 21(3):371-397, 2010