

# **Data Mining: Concepts and Techniques**

---

**(3<sup>rd</sup> ed.)**

## **— Chapter 7 —**

Jiawei Han, Micheline Kamber, and Jian Pei  
University of Illinois at Urbana-Champaign &  
Simon Fraser University

©2010 Han, Kamber & Pei. All rights reserved.



# Chapter 7 : Advanced Frequent Pattern Mining

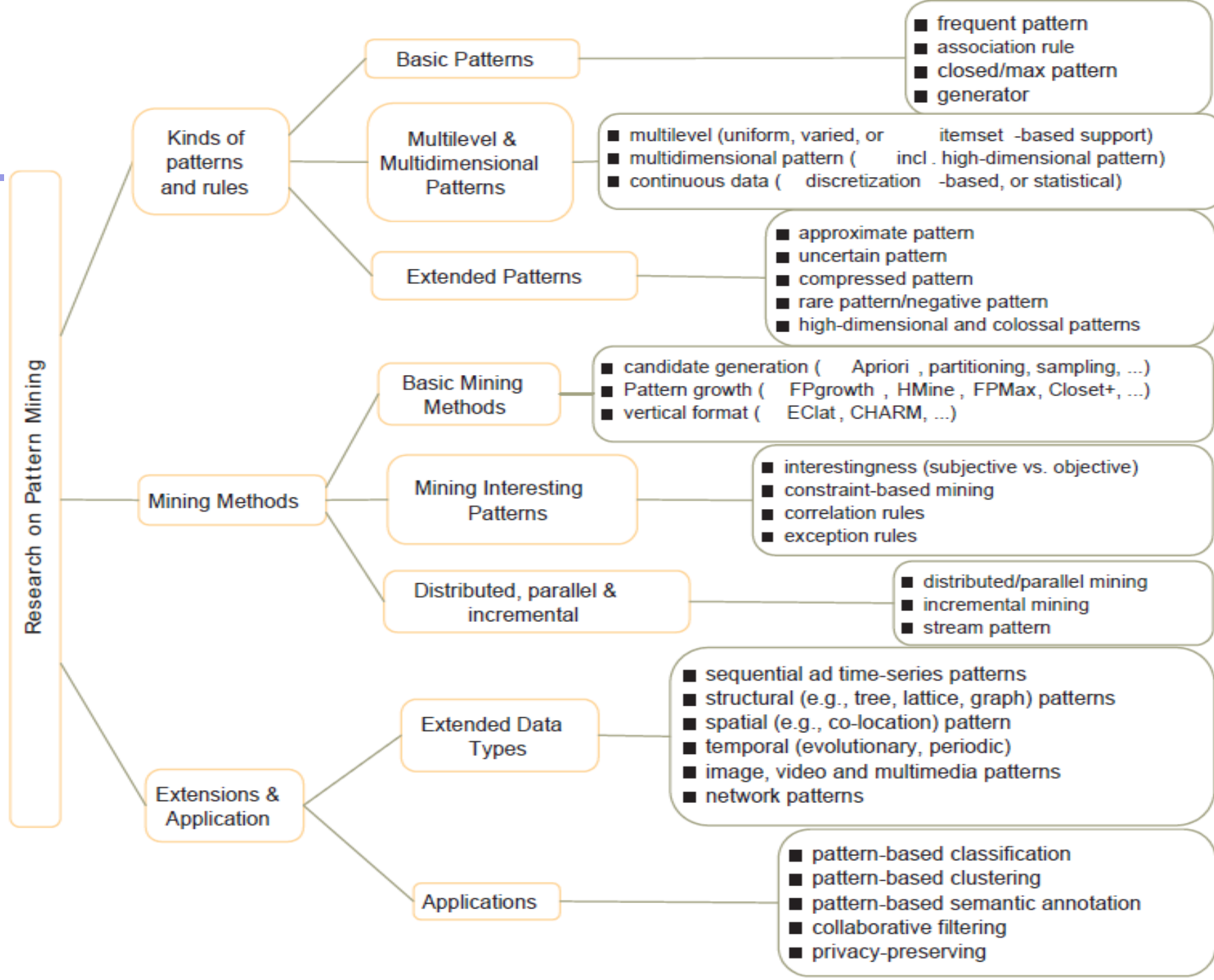
---



- Pattern Mining: A Road Map
- Pattern Mining in Multi-Level, Multi-Dimensional Space
- Constraint-Based Frequent Pattern Mining
- Mining High-Dimensional Data and Colossal Patterns
- Mining Compressed or Approximate Patterns
- Pattern Exploration and Application
- Summary



# Research on Pattern Mining: A Road Map



# Chapter 7 : Advanced Frequent Pattern Mining

---

- Pattern Mining: A Road Map
- Pattern Mining in Multi-Level, Multi-Dimensional Space 
  - Mining Multi-Level Association 
  - Mining Multi-Dimensional Association
  - Mining Quantitative Association Rules
  - Mining Rare Patterns and Negative Patterns
- Constraint-Based Frequent Pattern Mining
- Mining High-Dimensional Data and Colossal Patterns
- Mining Compressed or Approximate Patterns
- Pattern Exploration and Application
- Summary

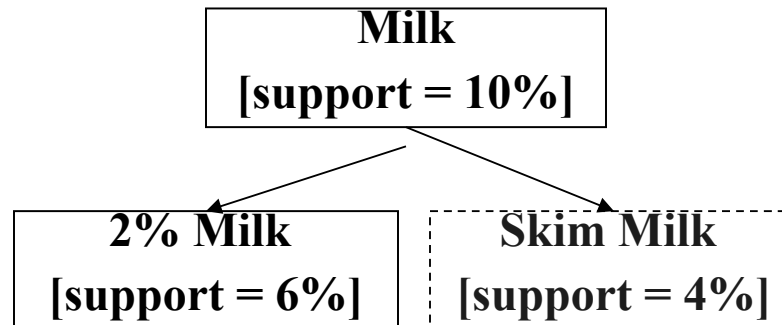
# Mining Multiple-Level Association Rules

- 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%

# Multi-level Association: Flexible Support and Redundancy filtering

- Flexible min-support thresholds: Some items are more valuable but less frequent
  - Use non-uniform, group-based min-support
  - E.g., {diamond, watch, camera}: 0.05%; {bread, milk}: 5%; ...
- Redundancy Filtering: Some rules may be redundant due to “ancestor” relationships between items
  - milk  $\Rightarrow$  wheat bread [support = 8%, confidence = 70%]
  - 2% milk  $\Rightarrow$  wheat bread [support = 2%, confidence = 72%]

The first rule is an ancestor of the second rule
- A rule is *redundant* if its support is close to the “expected” value, based on the rule’s ancestor

# Chapter 7 : Advanced Frequent Pattern Mining

---

- Pattern Mining: A Road Map
- Pattern Mining in Multi-Level, Multi-Dimensional Space 
  - Mining Multi-Level Association
  - Mining Multi-Dimensional Association 
  - Mining Quantitative Association Rules
  - Mining Rare Patterns and Negative Patterns
- Constraint-Based Frequent Pattern Mining
- Mining High-Dimensional Data and Colossal Patterns
- Mining Compressed or Approximate Patterns
- Pattern Exploration and Application
- Summary



# Mining Multi-Dimensional Association

- Single-dimensional rules:  
 $\text{buys}(X, \text{"milk"}) \Rightarrow \text{buys}(X, \text{"bread"})$
- Multi-dimensional rules:  $\geq 2$  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

# Chapter 7 : Advanced Frequent Pattern Mining

---

- Pattern Mining: A Road Map
- Pattern Mining in Multi-Level, Multi-Dimensional Space 
  - Mining Multi-Level Association
  - Mining Multi-Dimensional Association
  - Mining Quantitative Association Rules 
  - Mining Rare Patterns and Negative Patterns
- Constraint-Based Frequent Pattern Mining
- Mining High-Dimensional Data and Colossal Patterns
- Mining Compressed or Approximate Patterns
- Pattern Exploration and Application
- Summary

# Mining Quantitative Associations

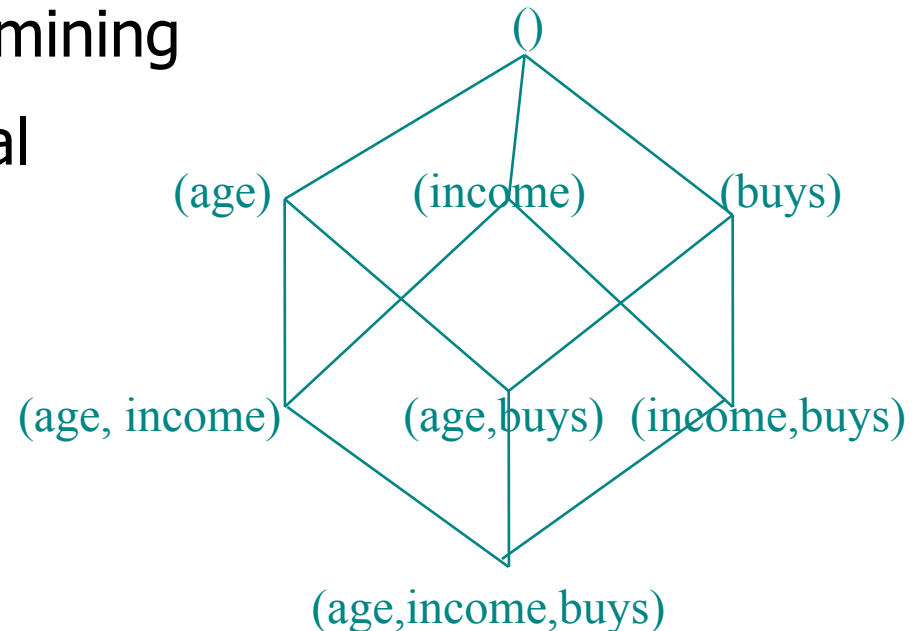
---

Techniques can be categorized by how numerical attributes, such as **age** or **salary** are treated

1. Static discretization based on predefined concept hierarchies (data cube methods)
2. Dynamic discretization based on data distribution (quantitative rules, e.g., Agrawal & Srikant@SIGMOD96)
3. Clustering: Distance-based association (e.g., Yang & Miller@SIGMOD97)
  - One dimensional clustering then association
1. Deviation: (such as Aumann and Lindell@KDD99)
  - Sex = female => Wage: mean=\$7/hr (overall mean = \$9)

# Static Discretization of Quantitative Attributes

- Discretized prior to mining using concept hierarchy.
- Numeric values are replaced by ranges
- In relational database, finding all frequent  $k$ -predicate sets will require  $k$  or  $k+1$  table scans
- Data cube is well suited for mining
- The cells of an  $n$ -dimensional cuboid correspond to the predicate sets
- Mining from data cubes can be much faster



# Quantitative Association Rules Based on Statistical Inference Theory [Aumann and Lindell@DMKD'03]

---

- Finding extraordinary and therefore interesting phenomena, e.g.,  
(Sex = female)  $\Rightarrow$  Wage: mean=\$7/hr (overall mean = \$9)
  - LHS: a subset of the population
  - RHS: an extraordinary behavior of this subset
- The rule is accepted only if a statistical test (e.g., Z-test) confirms the inference with high confidence
- Subrule: highlights the extraordinary behavior of a subset of the pop. of the super rule
  - E.g., (Sex = female)  $\wedge$  (South = yes)  $\Rightarrow$  mean wage = \$6.3/hr
- Two forms of rules
  - Categorical  $\Rightarrow$  quantitative rules, or Quantitative  $\Rightarrow$  quantitative rules
  - E.g., Education in [14-18] (yrs)  $\Rightarrow$  mean wage = \$11.64/hr
- Open problem: Efficient methods for LHS containing two or more quantitative attributes



# Chapter 7 : Advanced Frequent Pattern Mining

---

- Pattern Mining: A Road Map
- Pattern Mining in Multi-Level, Multi-Dimensional Space 
  - Mining Multi-Level Association
  - Mining Multi-Dimensional Association
  - Mining Quantitative Association Rules
  - Mining Rare Patterns and Negative Patterns 
- Constraint-Based Frequent Pattern Mining
- Mining High-Dimensional Data and Colossal Patterns
- Mining Compressed or Approximate Patterns
- Pattern Exploration and Application
- Summary

# Negative and Rare Patterns

---

- Rare patterns: Very low support but interesting
  - E.g., buying Rolex watches
  - Mining: Setting individual-based or special group-based support threshold for valuable items
- Negative patterns
  - Since it is unlikely that one buys Ford Expedition (an SUV car) and Toyota Prius (a hybrid car) together, Ford Expedition and Toyota Prius are likely negatively correlated patterns
- Negatively correlated patterns that are infrequent tend to be more interesting than those that are frequent

# Defining Negative Correlated Patterns (I)

- Definition 1 (support-based)
  - If itemsets X and Y are both frequent but rarely occur together, i.e.,
$$\text{sup}(X \cup Y) < \text{sup}(X) * \text{sup}(Y)$$
  - Then X and Y are negatively correlated
- Problem: A store sold two needle 100 packages A and B, only one transaction containing both A and B.
  - When there are in total 200 transactions, we have
$$s(A \cup B) = 0.005, s(A) * s(B) = 0.25, s(A \cup B) < s(A) * s(B)$$
  - When there are  $10^5$  transactions, we have
$$s(A \cup B) = 1/10^5, s(A) * s(B) = 1/10^3 * 1/10^3, s(A \cup B) > s(A) * s(B)$$
  - Where is the problem? —Null transactions, i.e., the support-based definition is not null-invariant!


# Defining Negative Correlated Patterns (II)

- Definition 2 (negative itemset-based)
  - $X$  is a *negative itemset* if (1)  $X = \bar{A} \cup B$ , where  $B$  is a set of positive items, and  $\bar{A}$  is a set of negative items,  $|\bar{A}| \geq 1$ , and (2)  $s(X) \geq \mu$
  - Itemsets  $X$  is negatively correlated, if
$$s(X) < \prod_{i=1}^k s(x_i), \text{ where } x_i \in X, \text{ and } s(x_i) \text{ is the support of } x_i$$
- This definition suffers a similar null-invariant problem
- Definition 3 (Kulczynski measure-based) If itemsets  $X$  and  $Y$  are frequent, but  $(P(X|Y) + P(Y|X))/2 < \epsilon$ , where  $\epsilon$  is a negative pattern threshold, then  $X$  and  $Y$  are negatively correlated.
- Ex. For the same needle package problem, when no matter there are 200 or  $10^5$  transactions, if  $\epsilon = 0.01$ , we have

$$(P(A|B) + P(B|A))/2 = (0.01 + 0.01)/2 < \epsilon$$

# Chapter 7 : Advanced Frequent Pattern Mining

---

- Pattern Mining: A Road Map
- Pattern Mining in Multi-Level, Multi-Dimensional Space
- Constraint-Based Frequent Pattern Mining 
- Mining High-Dimensional Data and Colossal Patterns
- Mining Compressed or Approximate Patterns
- Pattern Exploration and Application
- Summary



# Constraint-based (Query-Directed) Mining

---

- Finding **all** the patterns in a database **autonomously**? — unrealistic!
  - The patterns could be too many but not focused!
- Data mining should be an **interactive** process
  - User directs what to be mined using a **data mining query language** (or a graphical user interface)
- Constraint-based mining
  - User flexibility: provides **constraints** on what to be mined
  - Optimization: explores such constraints for efficient mining — **constraint-based mining**: constraint-pushing, similar to push selection first in DB query processing
  - Note: still find all the answers satisfying constraints, not finding some answers in “heuristic search”

# Constraints in Data Mining

---

- Knowledge type constraint:
  - classification, association, etc.
- Data constraint — using SQL-like queries
  - find product pairs sold together in stores in Chicago this year
- Dimension/level constraint
  - in relevance to region, price, brand, customer category
- Rule (or pattern) constraint
  - small sales (price < \$10) triggers big sales (sum > \$200)
- Interestingness constraint
  - strong rules:  $\text{min\_support} \geq 3\%$ ,  $\text{min\_confidence} \geq 60\%$

# Meta-Rule Guided Mining

- Meta-rule can be in the rule form with partially instantiated predicates and constants

$$P_1(X, Y) \wedge P_2(X, W) \Rightarrow \text{buys}(X, \text{"iPad"})$$

- The resulting rule derived can be

$$\text{age}(X, \text{"15-25"}) \wedge \text{profession}(X, \text{"student"}) \Rightarrow \text{buys}(X, \text{"iPad"})$$

- In general, it can be in the form of

$$P_1 \wedge P_2 \wedge \dots \wedge P_l \Rightarrow Q_1 \wedge Q_2 \wedge \dots \wedge Q_r$$

- Method to find meta-rules

- Find frequent (l+r) predicates (based on min-support threshold)
- Push constants deeply when possible into the mining process (see the remaining discussions on constraint-push techniques)
- Use confidence, correlation, and other measures when possible

# Constraint-Based Frequent Pattern Mining

- Pattern space pruning constraints
  - **Anti-monotonic**: If constraint  $c$  is violated, its further mining can be terminated
  - **Monotonic**: If  $c$  is satisfied, no need to check  $c$  again
  - **Succinct**:  $c$  must be satisfied, so one can start with the data sets satisfying  $c$
  - **Convertible**:  $c$  is not monotonic nor anti-monotonic, but it can be converted into it if items in the transaction can be properly ordered
- Data space pruning constraint
  - **Data succinct**: Data space can be pruned at the initial pattern mining process
  - **Data anti-monotonic**: If a transaction  $t$  does not satisfy  $c$ ,  $t$  can be pruned from its further mining

# Pattern Space Pruning with Anti-Monotonicity Constraints

- A constraint  $C$  is *anti-monotone* if the super pattern satisfies  $C$ , all of its sub-patterns do so too
- In other words, *anti-monotonicity*: If an itemset  $S$  **violates** the constraint, so does any of its superset
- Ex. 1.  $\text{sum}(S.\text{price}) \leq v$  is **anti-monotone**
- Ex. 2.  $\text{range}(S.\text{profit}) \leq 15$  is **anti-monotone**
  - Itemset  $ab$  violates  $C$
  - So does every superset of  $ab$
- Ex. 3.  $\text{sum}(S.\text{Price}) \geq v$  is **not anti-monotone**
- Ex. 4. *support count* is anti-monotone: core property used in Apriori

TDB (min\_sup=2)

TID	Transaction
10	a, b, c, d, f
20	b, c, d, f, g, h
30	a, c, d, e, f
40	c, e, f, g

Item	Profit
a	40
b	0
c	-20
d	10
e	-30
f	30
g	20
h	-10



# Pattern Space Pruning with Monotonicity Constraints

- A constraint  $C$  is *monotone* if the pattern satisfies  $C$ , we do not need to check  $C$  in subsequent mining
- Alternatively, monotonicity: *If an itemset  $S$  **satisfies** the constraint, so does any of its superset*
- Ex. 1.  $\text{sum}(S.\text{Price}) \geq v$  is **monotone**
- Ex. 2.  $\text{min}(S.\text{Price}) \leq v$  is **monotone**
- Ex. 3.  $C: \text{range}(S.\text{profit}) \geq 15$ 
  - Itemset  $ab$  satisfies  $C$
  - So does every superset of  $ab$

TDB (min\_sup=2)

TID	Transaction
10	a, b, c, d, f
20	b, c, d, f, g, h
30	a, c, d, e, f
40	c, e, f, g

Item	Profit
a	40
b	0
c	-20
d	10
e	-30
f	30
g	20
h	-10

# Data Space Pruning with Data Anti-monotonicity

- A constraint  $c$  is *data anti-monotone* if for a pattern  $p$  cannot satisfy a transaction  $t$  under  $c$ ,  $p$ 's superset cannot satisfy  $t$  under  $c$  either
- The key for data anti-monotone is *recursive data reduction*
- Ex. 1.  $\text{sum}(S.\text{Price}) \geq v$  is data anti-monotone
- Ex. 2.  $\text{min}(S.\text{Price}) \leq v$  is data anti-monotone
- Ex. 3.  $C: \text{range}(S.\text{profit}) \geq 25$  is data anti-monotone
  - Itemset  $\{b, c\}$ 's projected DB:
    - $T10'$ :  $\{d, f, h\}$ ,  $T20'$ :  $\{d, f, g, h\}$ ,  $T30'$ :  $\{d, f, g\}$
  - since  $C$  cannot satisfy  $T10'$ ,  $T10'$  can be pruned

TDB (min\_sup=2)

TID	Transaction
10	a, b, c, d, f, h
20	b, c, d, f, g, h
30	b, c, d, f, g
40	c, e, f, g

Item	Profit
a	40
b	0
c	-20
d	-15
e	-30
f	-10
g	20
h	-5

# Pattern Space Pruning with Succinctness

- Succinctness:
  - Given  $A_1$ , the set of items satisfying a succinctness constraint  $C$ , then any set  $S$  satisfying  $C$  is based on  $A_1$ , i.e.,  $S$  contains a subset belonging to  $A_1$
  - Idea: Without looking at the transaction database, whether an itemset  $S$  satisfies constraint  $C$  can be determined based on the selection of items
  - $\min(S.Price) \leq v$  is succinct
  - $\sum(S.Price) \geq v$  is not succinct
- Optimization: If  $C$  is succinct,  $C$  is pre-counting pushable

# Naïve Algorithm: Apriori + Constraint

Database D

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

Scan D

$C_1$

itemset	sup.
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

$L_1$

itemset	sup.
{1}	2
{2}	3
{3}	3
<del>{5}</del>	<del>3</del>

$C_2$

itemset	sup
{1 2}	1
{1 3}	2
{1 5}	1
{2 3}	2
{2 5}	3
{3 5}	2

Scan D

$C_2$

itemset
{1 2}
{1 3}
{1 5}
{2 3}
{2 5}
{3 5}

$L_2$

itemset	sup
{1 3}	2
<del>{2 3}</del>	<del>2</del>
<del>{2 5}</del>	<del>3</del>
<del>{3 5}</del>	<del>2</del>

$C_3$

itemset
{2 3 5}

Scan D

$L_3$

itemset	sup
<del>{2 3 5}</del>	<del>2</del>

**Constraint:**

**Sum{S.price} < 5**

# Constrained Apriori : Push a Succinct Constraint Deep

Database D

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

Scan D

$C_1$

itemset	sup.
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

$L_1$

itemset	sup.
{1}	2
{2}	3
{3}	3
{5}	3

$C_2$

itemset	sup
{1 2}	1
{1 3}	2
{1 5}	1
<del>{2 3}</del>	<del>2</del>
<del>{2 5}</del>	<del>3</del>
<del>{3 5}</del>	<del>2</del>

Scan D

$C_2$

itemset
{1 2}
{1 3}
{1 5}
<del>{2 3}</del>
<del>{2 5}</del>
<del>{3 5}</del>

not immediately  
to be used

$L_2$

itemset	sup
{1 3}	2
<del>{2 3}</del>	<del>2</del>
<del>{2 5}</del>	<del>3</del>
<del>{3 5}</del>	<del>2</del>

$C_3$

itemset
<del>{2 3 5}</del>

Scan D

$L_3$

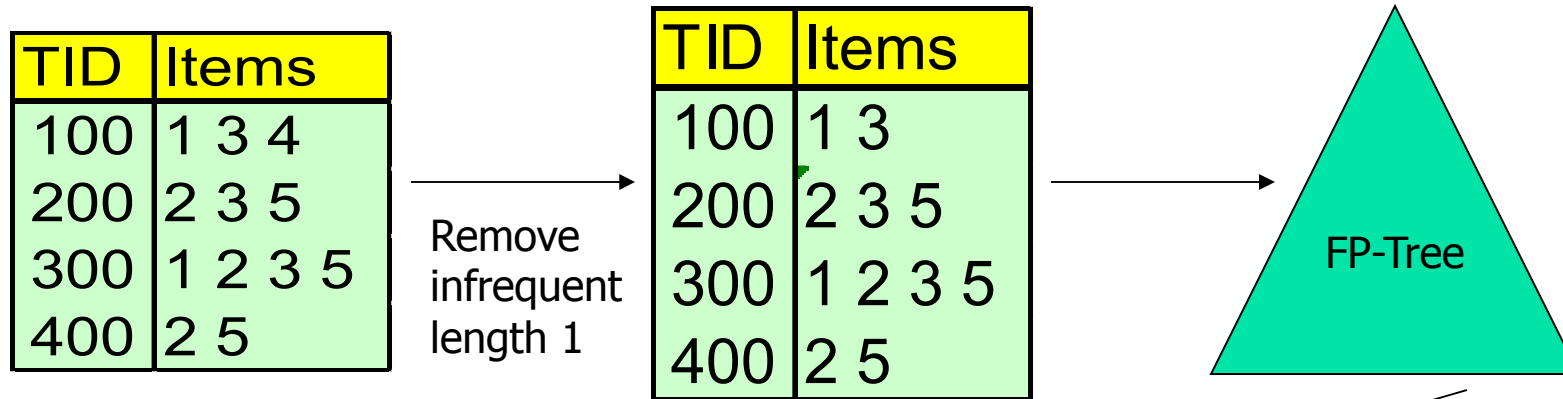
itemset	sup
<del>{2 3 5}</del>	<del>2</del>

**Constraint:**

**$\min\{S.\text{price}\} \leq 1$**



# Constrained FP-Growth: Push a Succinct Constraint Deep



1-Projected DB

TID	Items
100	3 4
300	2 3 5

No Need to project on 2, 3, or 5

**Constraint:**

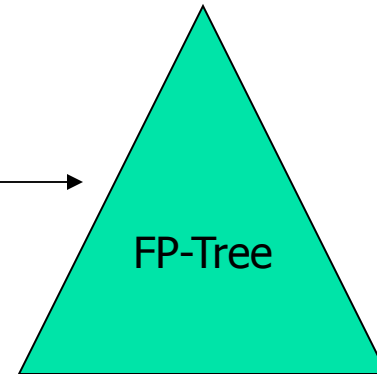
**$\min\{S.\text{price}\} \leq 1$**

# Data Anti-monotonic Constraint Deep

Remove from data

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

TID	Items
100	1 3
300	1 3



Single branch, we are done

**Constraint:**

**$\min\{S.\text{price}\} \leq 1$**

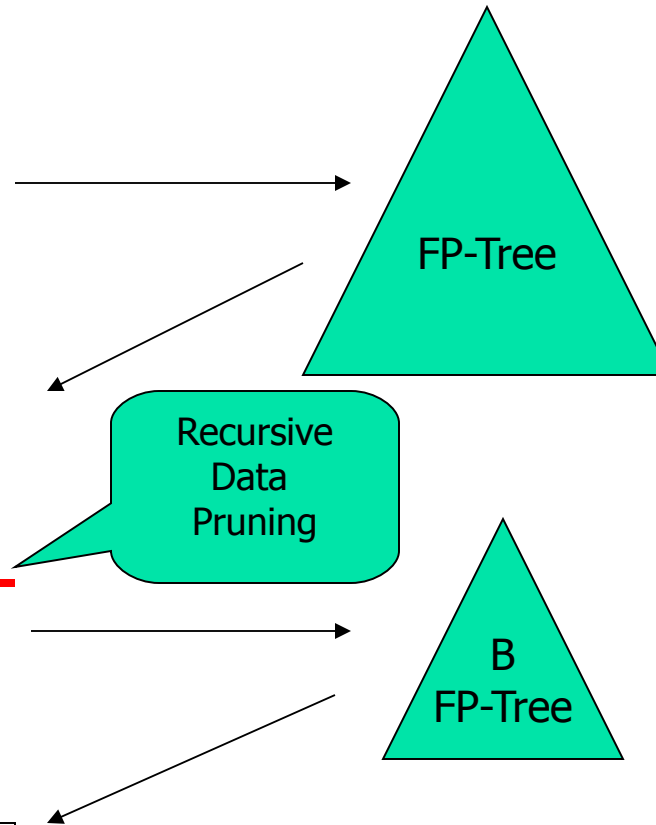
# Constrained FP-Growth: Push a Data Anti-monotonic Constraint Deep

TID	Transaction
10	a, b, c, d, f, h
20	b, c, d, f, g, h
30	b, c, d, f, g
40	a, c, e, f, g

B-Projected DB

TID	Transaction
<del>10</del>	<del>a, c, d, f, h</del>
20	c, d, f, g, <del>h</del>
30	c, d, f, g

Single branch:  
bcdfg: 2



TID	Transaction
10	a, b, c, d, f, h
20	b, c, d, f, g, h
30	b, c, d, f, g
40	a, c, e, f, g

Item	Profit
a	40
b	0
c	-20
d	-15
e	-30
f	-10
g	20
h	-5

**Constraint:**  
**range{S.price } > 25**  
**min\_sup >= 2**

# Convertible Constraints: Ordering Data in Transactions

- Convert tough constraints into anti-monotone or monotone by properly ordering items
- Examine C:  $\text{avg}(S.\text{profit}) \geq 25$ 
  - Order items in value-descending order
    - $\langle a, f, g, d, b, h, c, e \rangle$
  - If an itemset  $afb$  violates C
    - So does  $afbh, afb^*$
    - It becomes **anti-monotone!**

TDB (min\_sup=2)

TID	Transaction
10	a, b, c, d, f
20	b, c, d, f, g, h
30	a, c, d, e, f
40	c, e, f, g

Item	Profit
a	40
b	0
c	-20
d	10
e	-30
f	30
g	20
h	-10

# Strongly Convertible Constraints

- $\text{avg}(X) \geq 25$  is convertible anti-monotone w.r.t. item **value descending** order  $R$ :  $\langle a, f, g, d, b, h, c, e \rangle$ 
  - If an itemset  $af$  violates a constraint  $C$ , so does every itemset with  $af$  as prefix, such as  $afd$
- $\text{avg}(X) \geq 25$  is convertible monotone w.r.t. item **value ascending** order  $R^{-1}$ :  $\langle e, c, h, b, d, g, f, a \rangle$ 
  - If an itemset  $d$  satisfies a constraint  $C$ , so does itemsets  $df$  and  $dfa$ , which having  $d$  as a prefix
- Thus,  $\text{avg}(X) \geq 25$  is **strongly convertible**

Item	Profit
a	40
b	0
c	-20
d	10
e	-30
f	30
g	20
h	-10

# Can Apriori Handle Convertible Constraints?

- A convertible, not monotone nor anti-monotone nor succinct constraint cannot be pushed deep into the an Apriori mining algorithm
  - Within the level wise framework, no direct pruning based on the constraint can be made
  - Itemset df violates constraint C:  $\text{avg}(X) \geq 25$
  - Since adf satisfies C, Apriori needs df to assemble adf, df cannot be pruned
- But it can be pushed into frequent-pattern growth framework!

Item	Value
a	40
b	0
c	-20
d	10
e	-30
f	30
g	20
h	-10

# Pattern Space Pruning w. Convertible Constraints

- C:  $\text{avg}(X) \geq 25$ ,  $\text{min\_sup}=2$
- List items in every transaction in value descending order R:  $\langle a, f, g, d, b, h, c, e \rangle$ 
  - C is convertible anti-monotone w.r.t. R
- Scan TDB once
  - remove infrequent items
    - Item h is dropped
  - Itemsets a and f are good, ...
- Projection-based mining
  - Imposing an appropriate order on item projection
  - Many tough constraints can be converted into (anti)-monotone

Item	Value
a	40
f	30
g	20
d	10
b	0
h	-10
c	-20
e	-30

TDB ( $\text{min\_sup}=2$ )

TID	Transaction
10	a, f, d, b, c
20	f, g, d, b, c
30	a, f, d, c, e
40	f, g, h, c, e

# Handling Multiple Constraints

---

- Different constraints may require different or even conflicting item-ordering
- If there exists an order  $R$  s.t. both  $C_1$  and  $C_2$  are convertible w.r.t.  $R$ , then there is no conflict between the two convertible constraints
- If there exists conflict on order of items
  - Try to satisfy one constraint first
  - Then using the order for the other constraint to mine frequent itemsets in the corresponding projected database



# What Constraints Are Convertible?

Constraint	Convertible anti-monotone	Convertible monotone	Strongly convertible
$\text{avg}(S) \leq , \geq v$	Yes	Yes	Yes
$\text{median}(S) \leq , \geq v$	Yes	Yes	Yes
$\text{sum}(S) \leq v$ (items could be of any value, $v \geq 0$ )	Yes	No	No
$\text{sum}(S) \leq v$ (items could be of any value, $v \leq 0$ )	No	Yes	No
$\text{sum}(S) \geq v$ (items could be of any value, $v \geq 0$ )	No	Yes	No
$\text{sum}(S) \geq v$ (items could be of any value, $v \leq 0$ )	Yes	No	No
.....			

# Constraint-Based Mining — A General Picture

Constraint	Anti-monotone	Monotone	Succinct
$v \in S$	no	yes	yes
$S \supseteq V$	no	yes	yes
$S \subseteq V$	yes	no	yes
$\min(S) \leq v$	no	yes	yes
$\min(S) \geq v$	yes	no	yes
$\max(S) \leq v$	yes	no	yes
$\max(S) \geq v$	no	yes	yes
$\text{count}(S) \leq v$	yes	no	weakly
$\text{count}(S) \geq v$	no	yes	weakly
$\text{sum}(S) \leq v \ (a \in S, a \geq 0)$	yes	no	no
$\text{sum}(S) \geq v \ (a \in S, a \geq 0)$	no	yes	no
$\text{range}(S) \leq v$	yes	no	no
$\text{range}(S) \geq v$	no	yes	no
$\text{avg}(S) \theta \ v, \theta \in \{=, \leq, \geq\}$	convertible	convertible	no
$\text{support}(S) \geq \xi$	yes	no	no
$\text{support}(S) \leq \xi$	no	yes	no

# Chapter 7 : Advanced Frequent Pattern Mining

---

- Pattern Mining: A Road Map
- Pattern Mining in Multi-Level, Multi-Dimensional Space
- Constraint-Based Frequent Pattern Mining
- Mining High-Dimensional Data and Colossal Patterns
- Mining Compressed or Approximate Patterns
- Pattern Exploration and Application
- Summary



# Mining Colossal Frequent Patterns

- F. Zhu, X. Yan, J. Han, P. S. Yu, and H. Cheng, “Mining Colossal Frequent Patterns by Core Pattern Fusion”, ICDE'07.
- We have many algorithms, but can we mine large (i.e., colossal) patterns? — such as just size around 50 to 100? Unfortunately, not!
- Why not? — the curse of “downward closure” of frequent patterns
  - The “downward closure” property
    - Any sub-pattern of a frequent pattern is frequent.
  - Example. If  $(a_1, a_2, \dots, a_{100})$  is frequent, then  $a_1, a_2, \dots, a_{100}, (a_1, a_2), (a_1, a_3), \dots, (a_1, a_{100}), (a_1, a_2, a_3), \dots$  are all frequent! There are about  $2^{100}$  such frequent itemsets!
  - No matter using breadth-first search (e.g., Apriori) or depth-first search (FPgrowth), we have to examine so many patterns
- Thus the downward closure property leads to explosion!

# Colossal Patterns: A Motivating Example

Let's make a set of 40 transactions

**T1 = 1 2 3 4 ..... 39 40**

**T2 = 1 2 3 4 ..... 39 40**

⋮

.

⋮

.

⋮

.

⋮

.

**T40=1 2 3 4 ..... 39 40**

Then delete the items on the diagonal

**T<sub>1</sub> = 2 3 4 ..... 39 40**

**T<sub>2</sub> = 1 3 4 ..... 39 40**

⋮

.

⋮

.

⋮

.

⋮

.

**T<sub>40</sub>=1 2 3 4 ..... 39**

Closed/maximal patterns may partially alleviate the problem but not really solve it: We often need to mine scattered large patterns!

Let the minimum support threshold  $\sigma = 20$

There are  $\binom{40}{20}$  frequent patterns of size 20

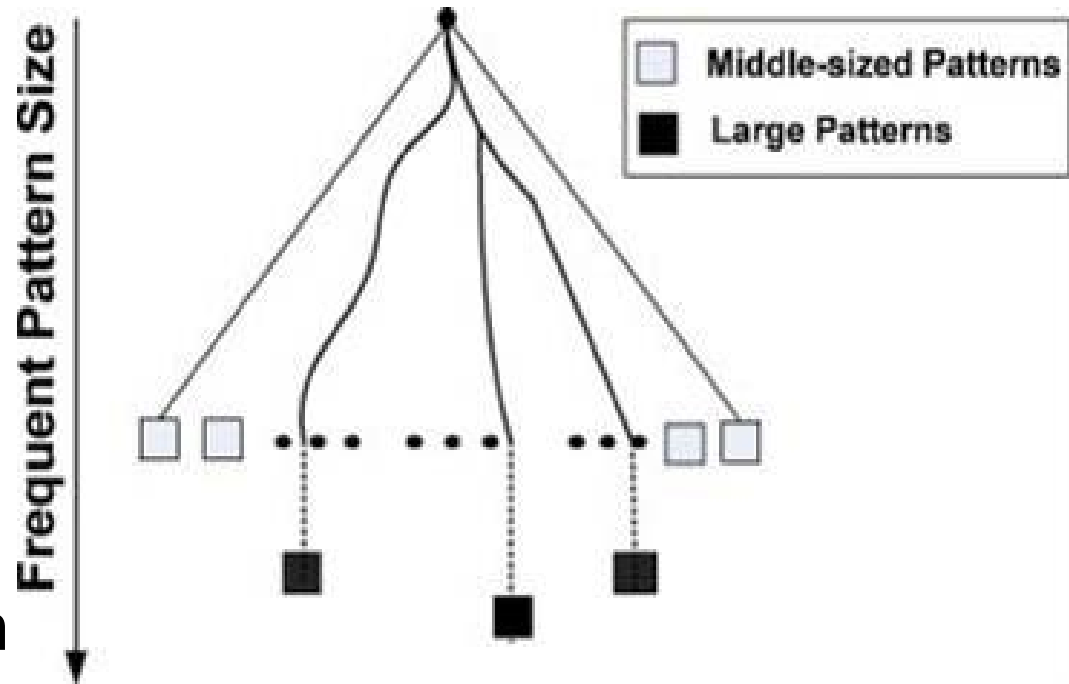
Each is closed and maximal

$$\# \text{ patterns} = \binom{n}{n/2} \approx \sqrt{2/\pi} \frac{2^n}{\sqrt{n}}$$

The size of the answer set is exponential to  $n$

# Colossal Pattern Set: Small but Interesting

- It is often the case that only a small number of patterns are colossal, i.e., of large size
- Colossal patterns are usually attached with greater importance than those of small pattern sizes



# Mining Colossal Patterns: Motivation and Philosophy

---

- Motivation: Many real-world tasks need mining colossal patterns
  - Micro-array analysis in bioinformatics (when support is low)
  - Biological sequence patterns
  - Biological/sociological/information graph pattern mining
- *No hope for completeness*
  - If the mining of mid-sized patterns is explosive in size, there is no hope to find colossal patterns efficiently by insisting “complete set” mining philosophy
- *Jumping out of the swamp of the mid-sized results*
  - What we may develop is a philosophy that may jump out of the swamp of mid-sized results that are explosive in size and jump to reach colossal patterns
- *Striving for mining almost complete colossal patterns*
  - The key is to develop a mechanism that may quickly reach colossal patterns and discover most of them

# Alas, A Show of Colossal Pattern Mining!

**T<sub>1</sub> = 2 3 4 ..... 39 40**  
**T<sub>2</sub> = 1 3 4 ..... 39 40**  
:  
:  
:  
:  
**T<sub>40</sub> = 1 2 3 4 ..... 39**  
**T<sub>41</sub> = 41 42 43 ..... 79**  
**T<sub>42</sub> = 41 42 43 ..... 79**  
:  
:  
**T<sub>60</sub> = 41 42 43 ... 79**

Let the min-support threshold  $\sigma = 20$

Then there are  $\binom{40}{20}$  closed/maximal frequent patterns of size 20

However, there is only one with size greater than 20, (*i.e.*, colossal):

$\alpha = \{41, 42, \dots, 79\}$  of size 39

The existing fastest mining algorithms (*e.g.*, FPClose, LCM) fail to complete running

Our algorithm outputs this colossal pattern in seconds

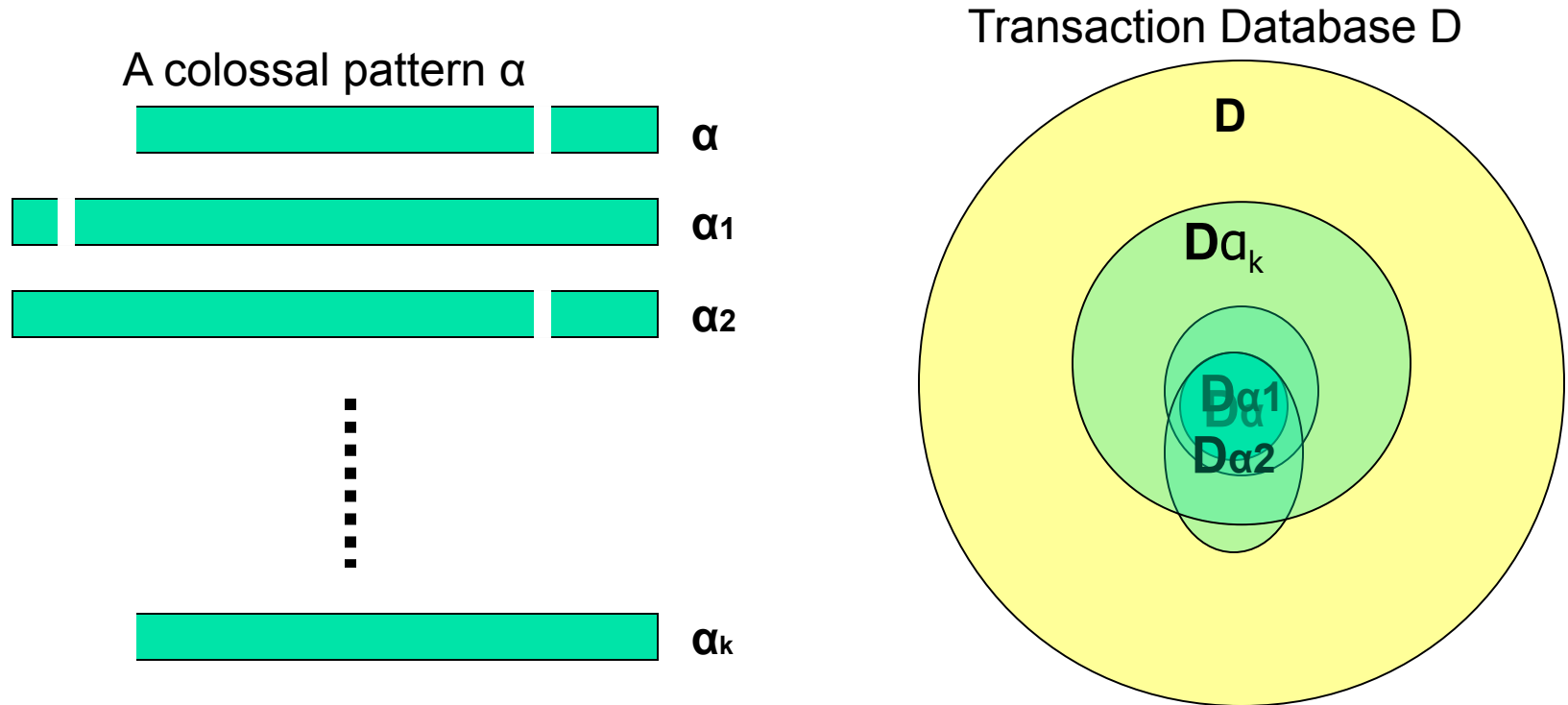


# Methodology of Pattern-Fusion Strategy

---

- Pattern-Fusion traverses the tree in a bounded-breadth way
  - Always pushes down a frontier of a bounded-size candidate pool
  - Only a fixed number of patterns in the current candidate pool will be used as the starting nodes to go down in the pattern tree — thus avoids the exponential search space
- Pattern-Fusion identifies “shortcuts” whenever possible
  - Pattern growth is not performed by single-item addition but by leaps and bounded: agglomeration of multiple patterns in the pool
  - These shortcuts will direct the search down the tree much more rapidly towards the colossal patterns

# Observation: Colossal Patterns and Core Patterns



Subpatterns  $\alpha_1$  to  $\alpha_k$  cluster tightly around the colossal pattern  $\alpha$  by sharing a similar support. We call such subpatterns *core patterns* of  $\alpha$

# Robustness of Colossal Patterns

---

- Core Patterns

Intuitively, for a frequent pattern  $\alpha$ , a subpattern  $\beta$  is a  $\tau$ -core pattern of  $\alpha$  if  $\beta$  shares a similar support set with  $\alpha$ , i.e.,

$$\frac{|D_{\alpha}|}{|D_{\beta}|} \geq \tau \quad 0 < \tau \leq 1$$

where  $\tau$  is called the core ratio

- Robustness of Colossal Patterns

A colossal pattern is robust in the sense that it tends to have much more core patterns than small patterns

# Example: Core Patterns

- A colossal pattern has far more core patterns than a small-sized pattern
- A colossal pattern has far more core descendants of a smaller size  $c$
- A random draw from a complete set of pattern of size  $c$  would more likely to pick a core descendant of a colossal pattern
- A colossal pattern can be generated by merging a set of core patterns

Transaction (# of Ts)	Core Patterns ( $\tau = 0.5$ )
(abe) (100)	(abe), (ab), (be), (ae), (e)
(bcf) (100)	(bcf), (bc), (bf)
(acf) (100)	(acf), (ac), (af)
(abcef) (100)	(ab), (ac), (af), (ae), (bc), (bf), (be), (ce), (fe), (e), (abc), (abf), (abe), (ace), (acf), (afe), (bcf), (bce), (bfe), (cfe), (abcf), (abce), (bcfe), (acfe), (abfe), (abcef)

# Robustness of Colossal Patterns

- $(d, \tau)$ -robustness: A pattern  $\alpha$  is  $(d, \tau)$ -robust if  $d$  is the maximum number of items that can be removed from  $\alpha$  for the resulting pattern to remain a  $\tau$ -core pattern of  $\alpha$
- For a  $(d, \tau)$ -robust pattern  $\alpha$ , it has  $\Omega(2^d)$  core patterns
  - A colossal patterns tend to have a large number of core patterns
- Pattern distance: For patterns  $\alpha$  and  $\beta$ , the pattern distance of  $\alpha$  and  $\beta$  is defined to be

$$Dist(\alpha, \beta) = 1 - \frac{|D_\alpha \cap D_\beta|}{|D_\alpha \cup D_\beta|}$$

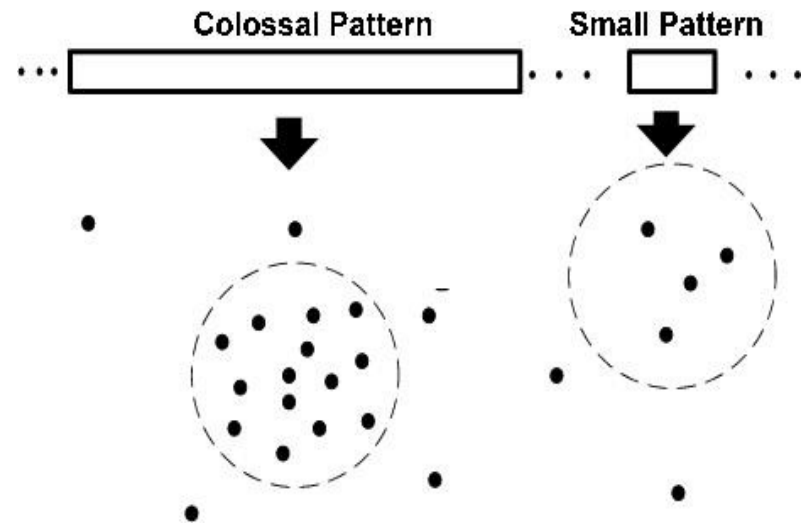
- If two patterns  $\alpha$  and  $\beta$  are both core patterns of a same pattern, they would be bounded by a “ball” of a radius specified by their core ratio  $\tau$

$$Dist(\alpha, \beta) \leq 1 - \frac{1}{2 / \tau - 1} = r(\tau)$$

- Once we identify one core pattern, we will be able to find all the other core patterns by a bounding ball of radius  $r(\tau)$

# Colossal Patterns Correspond to Dense Balls

- Due to their robustness, colossal patterns correspond to dense balls
  - $\Omega(2^d)$  in population
- A random draw in the pattern space will hit somewhere in the ball with high probability



# Idea of Pattern-Fusion Algorithm

---

- Generate a complete set of frequent patterns up to a small size
- Randomly pick a pattern  $\beta$ , and  $\beta$  has a high probability to be a core-descendant of some colossal pattern  $\alpha$
- Identify all  $\alpha$ 's descendants in this complete set, and merge all of them — This would generate a much larger core-descendant of  $\alpha$
- In the same fashion, we select  $K$  patterns. This set of larger core-descendants will be the candidate pool for the next iteration

# Pattern-Fusion: The Algorithm

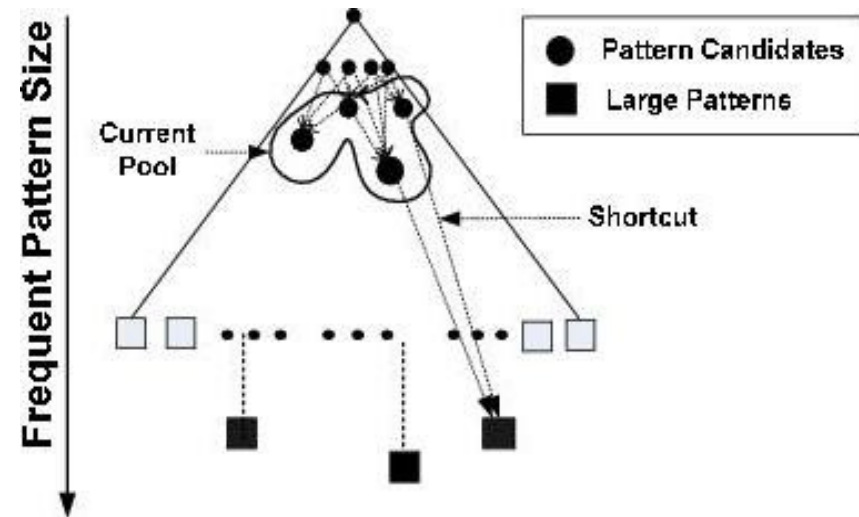
---

- Initialization (Initial pool): Use an existing algorithm to mine all frequent patterns up to a small size, e.g., 3
- Iteration (Iterative Pattern Fusion):
  - At each iteration,  $k$  seed patterns are randomly picked from the current pattern pool
  - For each seed pattern thus picked, we find all the patterns within a bounding ball centered at the seed pattern
  - All these patterns found are fused together to generate a set of super-patterns. All the super-patterns thus generated form a new pool for the next iteration
- Termination: when the current pool contains no more than  $K$  patterns at the beginning of an iteration



# Why Is Pattern-Fusion Efficient?

- A bounded-breadth pattern tree traversal
  - It avoids explosion in mining mid-sized ones
  - Randomness comes to help to stay on the right path
- Ability to identify “short-cuts” and take “leaps”
  - fuse small patterns together in one step to generate new patterns of significant sizes
  - Efficiency



# Pattern-Fusion Leads to Good Approximation

---

- Gearing toward colossal patterns
  - The larger the pattern, the greater the chance it will be generated
- Catching outliers
  - The more distinct the pattern, the greater the chance it will be generated

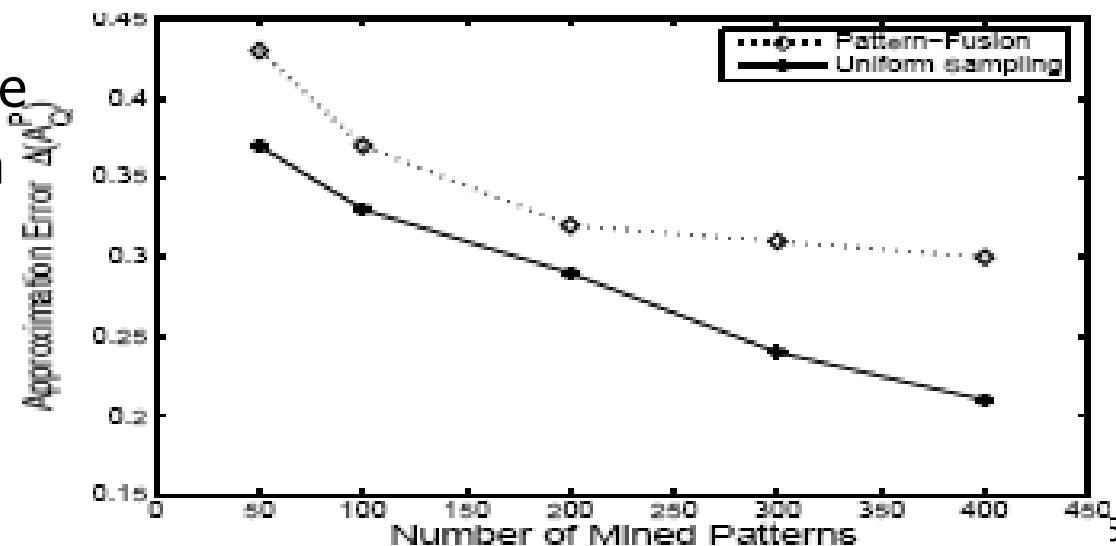
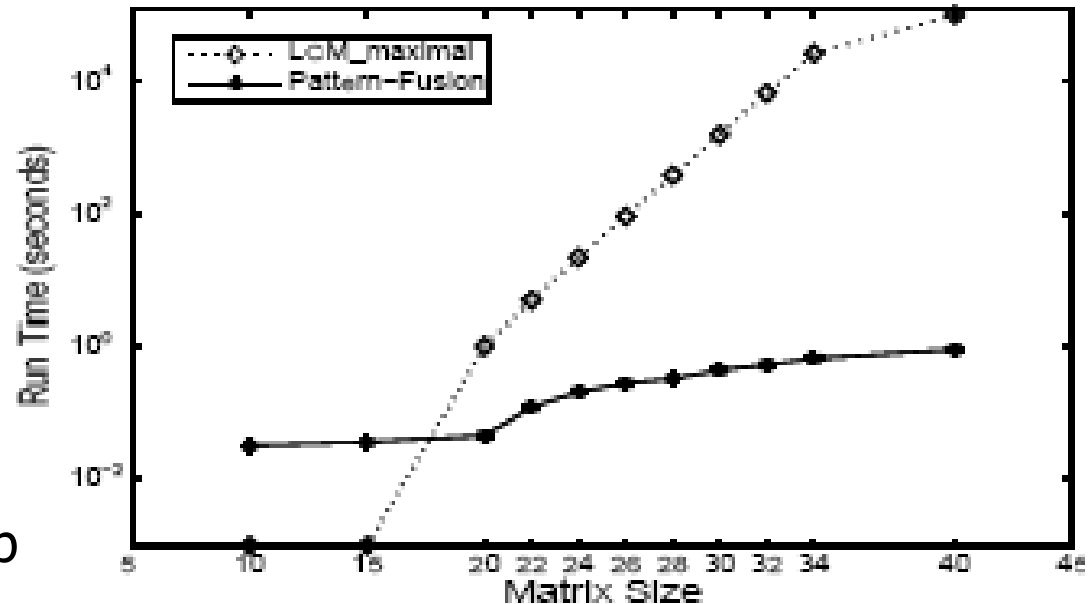
# Experimental Setting

---

- Synthetic data set
  - $\text{Diag}_n$  an  $n \times (n-1)$  table where  $i^{\text{th}}$  row has integers from 1 to  $n$  except  $i$ . Each row is taken as an itemset.  $\text{min\_support}$  is  $n/2$ .
- Real data set
  - Replace: A program trace data set collected from the “replace” program, widely used in software engineering research
  - ALL: A popular gene expression data set, a clinical data on ALL-AML leukemia ([www.broad.mit.edu/tools/data.html](http://www.broad.mit.edu/tools/data.html)).
    - Each item is a column, representing the activity level of gene/protein in the same
    - Frequent pattern would reveal important correlation between gene expression patterns and disease outcomes

# Experiment Results on $\text{Diag}_n$

- LCM run time increases exponentially with pattern size  $n$
- Pattern-Fusion finishes efficiently
- The approximation error of Pattern-Fusion (with min-sup 20) in comparison with the complete set) is rather close to uniform sampling (which randomly picks  $K$  patterns from the complete answer set)



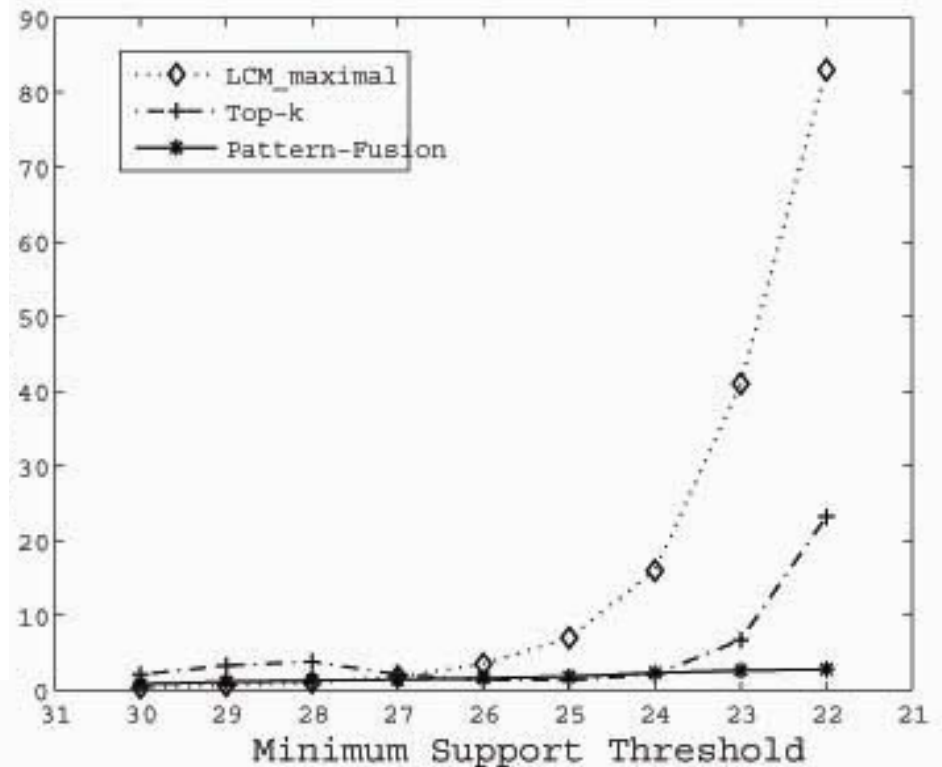
# Experimental Results on ALL

- ALL: A popular gene expression data set with 38 transactions, each with 866 columns
  - There are 1736 items in total
  - The table shows a high frequency threshold of 30

Pattern Size	110	107	102	91	86	84	83
The complete set	1	1	1	1	1	2	6
Pattern-Fusion	1	1	1	1	1	1	4

Pattern Size	82	77	76	75	74	73	71
The complete set	1	2	1	1	1	2	1
Pattern-Fusion	0	2	0	1	1	1	1



# Experimental Results on REPLACE

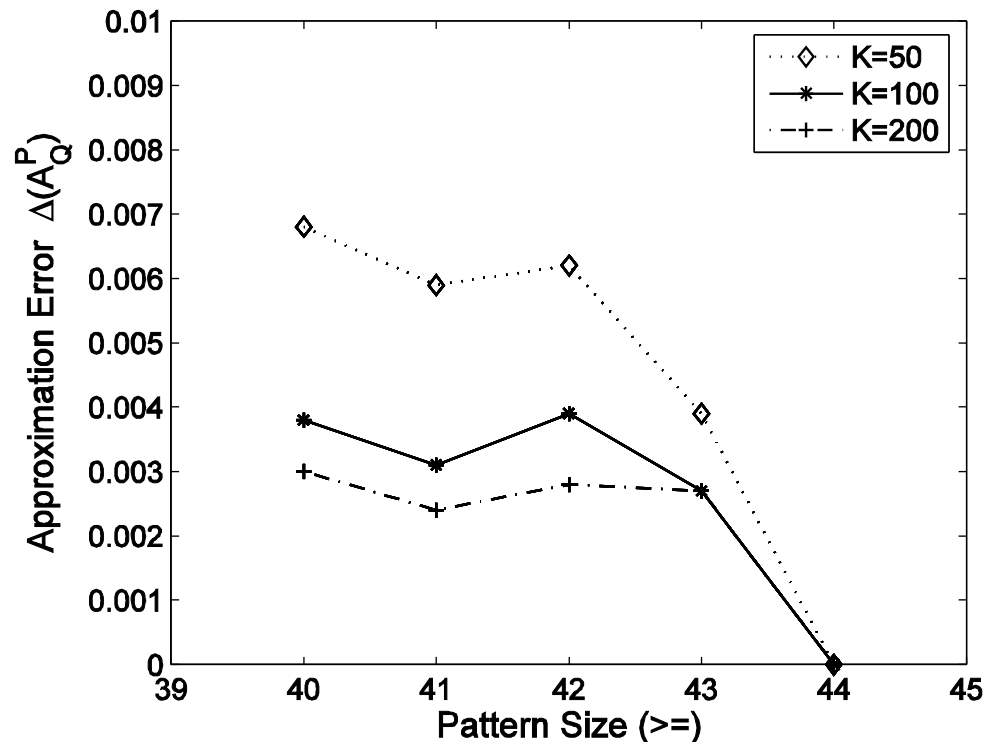
---

- REPLACE

- A program trace data set, recording 4395 calls and transitions
- The data set contains 4395 transactions with 57 items in total
- With support threshold of 0.03, the largest patterns are of size 44
- They are all discovered by Pattern-Fusion with different settings of  $K$  and  $\tau$ , when started with an initial pool of 20948 patterns of size  $\leq 3$


# Experimental Results on REPLACE

- Approximation error when compared with the complete mining result
- Example. Out of the total 98 patterns of size  $\geq 42$ , when  $K=100$ , Pattern-Fusion returns 80 of them
- A good approximation to the colossal patterns in the sense that any pattern in the complete set is on average at most 0.17 items away from one of these 80 patterns



# Chapter 7 : Advanced Frequent Pattern Mining

---

- Pattern Mining: A Road Map
- Pattern Mining in Multi-Level, Multi-Dimensional Space
- Constraint-Based Frequent Pattern Mining
- Mining High-Dimensional Data and Colossal Patterns
- Mining Compressed or Approximate Patterns 
- Pattern Exploration and Application
- Summary



# Mining Compressed Patterns: $\delta$ -clustering

- Why compressed patterns?
  - too many, but less meaningful
- Pattern distance measure

$$D(P_1, P_2) = 1 - \frac{|T(P_1) \cap T(P_2)|}{|T(P_1) \cup T(P_2)|}$$

- $\delta$ -clustering: For each pattern P, find all patterns which can be expressed by P and their distance to P are within  $\delta$  ( $\delta$ -cover)
- All patterns in the cluster can be represented by P
- Xin et al., "Mining Compressed Frequent-Pattern Sets", VLDB'05

ID	Item-Sets	Support
P1	{38,16,18,12}	205227
P2	{38,16,18,12,17}	205211
P3	{39,38,16,18,12,17}	101758
P4	{39,16,18,12,17}	161563
P5	{39,16,18,12}	161576

- Closed frequent pattern
  - Report P1, P2, P3, P4, P5
  - Emphasize too much on support
  - no compression
- Max-pattern, P3: info loss
- A desirable output: P2, P3, P4

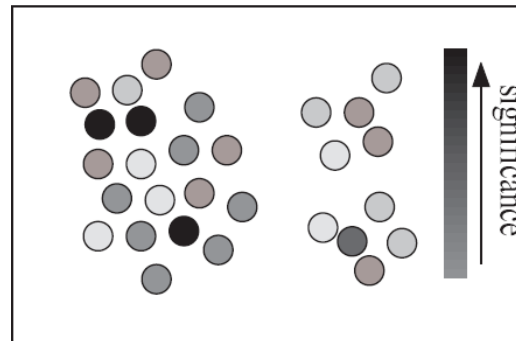
# Redundancy-Award Top-k Patterns

- Why redundancy-aware top-k patterns?

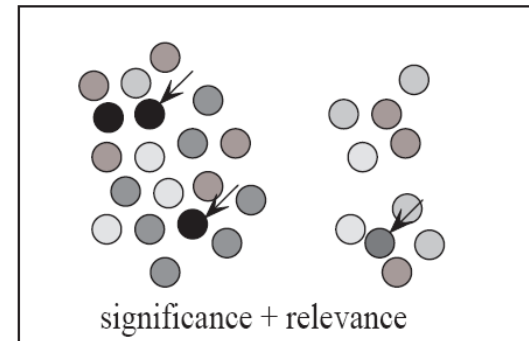
- Desired patterns: high significance & low redundancy

- Propose the MMS (Maximal Marginal Significance) for measuring the combined significance of a pattern set

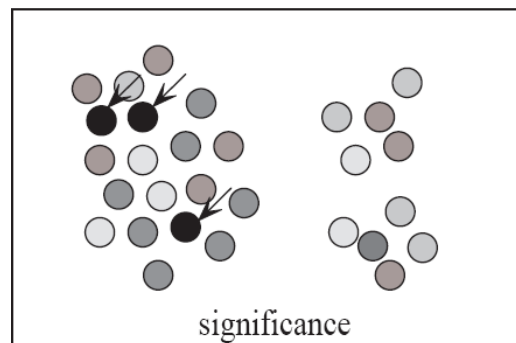
- Xin et al., Extracting Redundancy-Aware Top-K Patterns, KDD'06



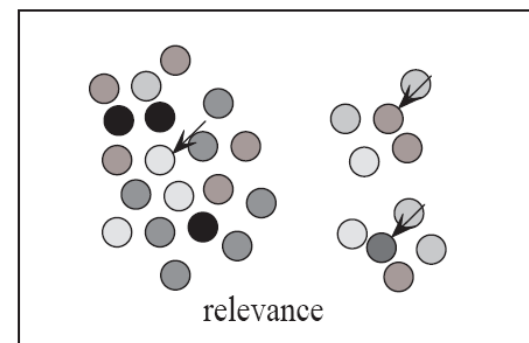
(a) a set of patterns



(b) redundancy-aware top-k




(c) traditional top-k



(d) summarization

# Chapter 7 : Advanced Frequent Pattern Mining

---

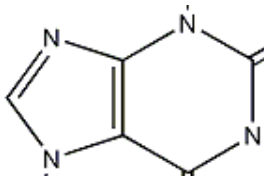
- Pattern Mining: A Road Map
- Pattern Mining in Multi-Level, Multi-Dimensional Space
- Constraint-Based Frequent Pattern Mining
- Mining High-Dimensional Data and Colossal Patterns
- Mining Compressed or Approximate Patterns
- Pattern Exploration and Application 
- Summary

# How to Understand and Interpret Patterns?

diaper beer

- Do they all make sense?
- What do they mean?
- How are they useful?

female sterile (2) tekele



**morphological info. and simple statistics**



**Semantic Information**

*Not all frequent patterns are useful, only meaningful ones ...*



**Annotate patterns with semantic information**

# A Dictionary Analogy

## Word: "pattern" – from Merriam-Webster

Main Entry: **1** *pat-tern*

Pronunciation: 'pa-tern

Function: *noun*

Etymology: Middle English *patron*, from Middle French, from *patron* 'patron',  
Latin *patronus*

Date: 14th century

**Non-semantic info.**

**Definitions indicating semantics**

1 : a form

2 : something

*pattern*

3 : a model

4 : an artist

5 : a nature

Main Entry:

**pattern**

Function:

*noun*

1

**Synonyms** *nonyns* MODEL 2, archetype, beau ideal, ensample, example, exemplar, ideal,

error, paradigm, standard

**Related Word** *original*

2

**Synonyms** FIGURE 3, design, device, motif, motive

**Related Word** *patterning*

*-pat-tern*

**Synonyms** ORDER 8, method, orderliness, plan, system

**Related Word** arrangement, *constellation*

**Synonyms**

**Related Words**

# Semantic Analysis With Context Models

---

- Task1: Model the context of a frequent pattern

*Based on the Context Model...*

- Task2: Extract strongest context indicators
- Task3: Extract representative transactions
- Task4: Extract semantically similar patterns

# Annotating DBLP Co-authorship & Title Pattern

## Database:

Authors	Title
X.Yan, P. Yu, J. Han	Substructure Similarity Search in Graph Databases
...	...
...	...

## Frequent

$P_1: \{x\_yan, j\_han\}$

Frequent Itemset

$P_2: \text{"substructure search"}$

## Semantic Annotations

Pattern	$\{x\_yan, j\_han\}$
Non	Sup = ...
CI	$\{p\_yu\}$ , graph pattern, ...
Trans.	gSpan: graph-base.....
SSPs	$\{j\_wang\}$ , $\{j\_han, p\_yu\}$ , ...

## Context Units

$\langle \{p\_yu, j\_han\}, \{d\_xin\}, \dots, \text{"graph pattern"}, \dots \text{"substructure similarity"}, \dots \rangle$

Pattern = {xifeng\_yan, jiawei\_han}

## Annotation Results:

Context Indicator (CI)	graph; {philip_yu}; mine close; graph pattern; sequential pattern; ...
Representative Transactions (Trans)	> gSpan: graph-base substructure pattern mining; > mining close relational graph connect constraint; ...
Semantically	{jiawei_han, philip_yu}; {jian_pei, jiawei_han}; {jiong_yang, philip_yu,

# Chapter 7 : Advanced Frequent Pattern Mining

---

- Pattern Mining: A Road Map
- Pattern Mining in Multi-Level, Multi-Dimensional Space
- Constraint-Based Frequent Pattern Mining
- Mining High-Dimensional Data and Colossal Patterns
- Mining Compressed or Approximate Patterns
- Pattern Exploration and Application
- Summary 



# Summary

---

- Roadmap: Many aspects & extensions on pattern mining
- Mining patterns in multi-level, multi dimensional space
- Mining rare and negative patterns
- Constraint-based pattern mining
- Specialized methods for mining high-dimensional data and colossal patterns
- Mining compressed or approximate patterns
- Pattern exploration and understanding: Semantic annotation of frequent patterns

# Ref: Mining Multi-Level and Quantitative Rules

---

- R. Srikant and R. Agrawal. Mining generalized association rules. VLDB'95.
- J. Han and Y. Fu. Discovery of multiple-level association rules from large databases. VLDB'95.
- R. Srikant and R. Agrawal. Mining quantitative association rules in large relational tables. SIGMOD'96.
- T. Fukuda, Y. Morimoto, S. Morishita, and T. Tokuyama. Data mining using two-dimensional optimized association rules: Scheme, algorithms, and visualization. SIGMOD'96.
- K. Yoda, T. Fukuda, Y. Morimoto, S. Morishita, and T. Tokuyama. Computing optimized rectilinear regions for association rules. KDD'97.
- R.J. Miller and Y. Yang. Association rules over interval data. SIGMOD'97.
- Y. Aumann and Y. Lindell. A Statistical Theory for Quantitative Association Rules KDD'99.

# Ref: Mining Other Kinds of Rules

---

- R. Meo, G. Psaila, and S. Ceri. A new SQL-like operator for mining association rules. VLDB'96.
- B. Lent, A. Swami, and J. Widom. Clustering association rules. ICDE'97.
- A. Savasere, E. Omiecinski, and S. Navathe. Mining for strong negative associations in a large database of customer transactions. ICDE'98.
- D. Tsur, J. D. Ullman, S. Abitboul, C. Clifton, R. Motwani, and S. Nestorov. Query flocks: A generalization of association-rule mining. SIGMOD'98.
- F. Korn, A. Labrinidis, Y. Kotidis, and C. Faloutsos. Ratio rules: A new paradigm for fast, quantifiable data mining. VLDB'98.
- F. Zhu, X. Yan, J. Han, P. S. Yu, and H. Cheng, "Mining Colossal Frequent Patterns by Core Pattern Fusion", ICDE'07.

# Ref: Constraint-Based Pattern Mining

---

- R. Srikant, Q. Vu, and R. Agrawal. Mining association rules with item constraints. KDD'97
- R. Ng, L.V.S. Lakshmanan, J. Han & A. Pang. Exploratory mining and pruning optimizations of constrained association rules. SIGMOD'98
- G. Grahne, L. Lakshmanan, and X. Wang. Efficient mining of constrained correlated sets. ICDE'00
- J. Pei, J. Han, and L. V. S. Lakshmanan. Mining Frequent Itemsets with Convertible Constraints. ICDE'01
- J. Pei, J. Han, and W. Wang, Mining Sequential Patterns with Constraints in Large Databases, CIKM'02
- F. Bonchi, F. Giannotti, A. Mazzanti, and D. Pedreschi. ExAnte: Anticipated Data Reduction in Constrained Pattern Mining, PKDD'03
- F. Zhu, X. Yan, J. Han, and P. S. Yu, "gPrune: A Constraint Pushing Framework for Graph Pattern Mining", PAKDD'07

# Ref: Mining Sequential and Structured Patterns

---

- R. Srikant and R. Agrawal. Mining sequential patterns: Generalizations and performance improvements. EDBT'96.
- H. Mannila, H Toivonen, and A. I. Verkamo. Discovery of frequent episodes in event sequences. DAMI:97.
- M. Zaki. SPADE: An Efficient Algorithm for Mining Frequent Sequences. Machine Learning:01.
- J. Pei, J. Han, H. Pinto, Q. Chen, U. Dayal, and M.-C. Hsu. PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth. ICDE'01.
- M. Kuramochi and G. Karypis. Frequent Subgraph Discovery. ICDM'01.
- X. Yan, J. Han, and R. Afshar. CloSpan: Mining Closed Sequential Patterns in Large Datasets. SDM'03.
- X. Yan and J. Han. CloseGraph: Mining Closed Frequent Graph Patterns. KDD'03.

# Ref: Mining Spatial, Multimedia, and Web Data

---

- K. Koperski and J. Han, Discovery of Spatial Association Rules in Geographic Information Databases, SSD'95.
- O. R. Zaiane, M. Xin, J. Han, Discovering Web Access Patterns and Trends by Applying OLAP and Data Mining Technology on Web Logs. ADL'98.
- O. R. Zaiane, J. Han, and H. Zhu, Mining Recurrent Items in Multimedia with Progressive Resolution Refinement. ICDE'00.
- D. Gunopulos and I. Tsoukatos. Efficient Mining of Spatiotemporal Patterns. SSTD'01.

# Ref: Mining Frequent Patterns in Time-Series Data

---

- B. Ozden, S. Ramaswamy, and A. Silberschatz. Cyclic association rules. ICDE'98.
- J. Han, G. Dong and Y. Yin, Efficient Mining of Partial Periodic Patterns in Time Series Database, ICDE'99.
- H. Lu, L. Feng, and J. Han. Beyond Intra-Transaction Association Analysis: Mining Multi-Dimensional Inter-Transaction Association Rules. TOIS:00.
- B.-K. Yi, N. Sidiropoulos, T. Johnson, H. V. Jagadish, C. Faloutsos, and A. Biliris. Online Data Mining for Co-Evolving Time Sequences. ICDE'00.
- W. Wang, J. Yang, R. Muntz. TAR: Temporal Association Rules on Evolving Numerical Attributes. ICDE'01.
- J. Yang, W. Wang, P. S. Yu. Mining Asynchronous Periodic Patterns in Time Series Data. TKDE'03.

# Ref: FP for Classification and Clustering

---

- G. Dong and J. Li. Efficient mining of emerging patterns: Discovering trends and differences. KDD'99.
- B. Liu, W. Hsu, Y. Ma. Integrating Classification and Association Rule Mining. KDD'98.
- W. Li, J. Han, and J. Pei. CMAR: Accurate and Efficient Classification Based on Multiple Class-Association Rules. ICDM'01.
- H. Wang, W. Wang, J. Yang, and P.S. Yu. Clustering by pattern similarity in large data sets. SIGMOD' 02.
- J. Yang and W. Wang. CLUSEQ: efficient and effective sequence clustering. ICDE'03.
- X. Yin and J. Han. CPAR: Classification based on Predictive Association Rules. SDM'03.
- H. Cheng, X. Yan, J. Han, and C.-W. Hsu, Discriminative Frequent Pattern Analysis for Effective Classification", ICDE'07.



# Ref: Stream and Privacy-Preserving FP Mining

---

- A. Evfimievski, R. Srikant, R. Agrawal, J. Gehrke. Privacy Preserving Mining of Association Rules. KDD'02.
- J. Vaidya and C. Clifton. Privacy Preserving Association Rule Mining in Vertically Partitioned Data. KDD'02.
- G. Manku and R. Motwani. Approximate Frequency Counts over Data Streams. VLDB'02.
- Y. Chen, G. Dong, J. Han, B. W. Wah, and J. Wang. Multi-Dimensional Regression Analysis of Time-Series Data Streams. VLDB'02.
- C. Giannella, J. Han, J. Pei, X. Yan and P. S. Yu. Mining Frequent Patterns in Data Streams at Multiple Time Granularities, Next Generation Data Mining:03.
- A. Evfimievski, J. Gehrke, and R. Srikant. Limiting Privacy Breaches in Privacy Preserving Data Mining. PODS'03.

# Ref: Other Freq. Pattern Mining Applications

---

- Y. Huhtala, J. Kärkkäinen, P. Porkka, H. Toivonen. Efficient Discovery of Functional and Approximate Dependencies Using Partitions. ICDE'98.
- H. V. Jagadish, J. Madar, and R. Ng. Semantic Compression and Pattern Extraction with Fascicles. VLDB'99.
- T. Dasu, T. Johnson, S. Muthukrishnan, and V. Shkapenyuk. Mining Database Structure; or How to Build a Data Quality Browser. SIGMOD'02.
- K. Wang, S. Zhou, J. Han. Profit Mining: From Patterns to Actions. EDBT'02.





# Chapter 7 : Advanced Frequent Pattern Mining

---

- Frequent Pattern and Association Mining: A Road Map
- Pattern Mining in Multi-Level, Multi-Dimensional Space
  - Mining Multilevel Association
  - Mining Multi-Dimensional Association
  - Mining Quantitative Association Rules
- Exploring Alternative Approaches to Improve Efficiency and Scalability
  - Mining Closed and Max Patterns
  - Scalable Pattern Mining in High-Dimensional Data
  - Mining Colossal Patterns
- Mining Beyond Typical Frequent Patterns
  - Mining Infrequent and Negative Patterns
  - Mining Compressed and Approximate Patterns
- Constraint-Based Frequent Pattern Mining
  - Metarule-Guided Mining of Association Rules
  - Constraint-Based Pattern Generation: Monotonicity, Anti-monotonicity, Succinctness, and Data Anti-monotonicity
  - Convertible Constraints: Ordering Data in Transactions
- Advanced Applications of Frequent Patterns
  - Towards pattern-based classification and cluster analysis
  - Context Analysis: Generating Semantic Annotations for Frequent Patterns
- Summary