COMP5121
Data Mining and Data Warehousing Applications

Week 9: Advanced Pattern Mining

Dr. Fengmei Jin

- Email: fengmei.jin@polyu.edu.hk
- Office: PQ747 (+852 3400 3327)
- Consultation Hours: 2.30-4.30pm every Thursday

Outline

- Sequential Association Rules
- Multi-level Association Rules
- Multi-dimensional Association Rules
- Quantitative Association Rules
- Rare Patterns and Negative Patterns
- Spatial Association Rules
- Compressed Patterns

MINING SEQUENTIAL ASSOCIATION RULES

Examples of Sequential Patterns

- A customer first buy an iPad, then an Apple Pencil, and then a magic keyboard, within a month.
- If a customer buys a car, he is likely to buy insurance within one week.
- If the stock price of Apple goes up, it is likely that Samsung, Google, or Microsoft up within two days.
- Other examples
  - Web access patterns
  - Weather prediction

Basic Concepts: Sequence

- Each transaction is itemset (a set of items): $s = \{i_1, i_2, \ldots, i_m\}$
- Each customer's multiple transactions:
  - s_1 = {Digital Camera, SD Memory card} on 1 Mar
  - s_2 = {Digital Video, Tripod} on 10 Mar
  - s_3 = {TV, PS5} on 20 Mar

→ Form a sequence $S =< s_1, s_2, s_3 >$ ordered by time

- A sequence consists of a list of transactions in temporal order, denoted as $S =< s_1, s_2, \ldots, s_n >$
  - Sequence length: n, i.e., the number of transactions.
  - **Sequence database**: a set of all sequences in the form $< SID, S >$.

Basic Concepts: Sequence Support

- The support of a sequence α in a sequence database D is defined as
  - the fraction of total tuples (NOT individual events) that support this sequence in D.
- Example:
  - The support of <a> is 4
    - S1 contains the item 'a' in three transactions, but S1 contributes only one to the support.
  - The support of <(ab)(c)> is 2 (in S1 and S3)

SID	Sequence
S1	<(a)(abc)(ac)(d)(cf)>
S2	<(ad)(c)(bc)(ae)>
S3	<(ef)(ab)(df)(cb)>
S4	<(e)(g)(af)(c)(bc)>

- <(a)(bc)(d)(f)> is the subsequence of S1
- <(a)(b)(c)(e)> is NOT the subsequence of S2

A sequence $\alpha =< a_1, a_2, \ldots, a_n >$ is a subsequence of $\beta =< b_1, b_2, \ldots, b_m >$, denoted as $\alpha \subseteq \beta$, if it exists n integers $j_1, j_2, \ldots, j_n \in [1, m]$ such that $a_1 \subseteq b_{j_1}, a_2 \subseteq b_{j_2}, \ldots, a_n \subseteq b_{j_n}$.

An Example of Sequence Support

Transaction DB: D		
Customer ID	**Transaction Time**	**Item IDs**
1	Jun 25	30
1	Jun 30	90
2	Jun 10	10, 20
2	Jun 15	30
2	Jun 20	40, 60, 70
3	Jun 25	30, 50, 70
4	Jun 25	30
4	Jun 30	40, 70
4	July 25	90
5	Jun 12	90

Sequential version of D	
Customer ID	**Customer Sequence**
1	<(30), (90)>
2	<(10 20), (30), (40 60 70)>
3	<(30 50 70)>
4	<(30), (40 70), (90)>
5	<(90)>

Customer sequence: all the transactions of a customer is a sequence ordered by transaction time.

Given a minimum support = 25%, we can find a frequent subsequence:

- <(30), (90)> with support of 40%: two customers (C1 and C4) bought the item 30 in an earlier transaction and then item 90 in a later one.

Sequential Pattern Mining: The mining of frequently occurring subsequences as *sequential patterns.*

Sequential Association Rule Mining Steps

1. **Sort Phase:** Convert D into a D' of customer sequences
   - D is sorted with customer-ID as the *major key* and time as the *minor key*.
2. **Frequent Itemset Phase:** Find all frequent itemsets L (using Apriori)
3. **Transformation Phase:** Transform each customer sequence into the frequent itemset representation, i.e., $< s_1, s_2, \dots, s_n > \rightarrow < l_1, l_2, \dots, l_n >$ where $l_i \in L$
4. **Sequence Phase:** Use *Apriori-like* algorithms to find all frequent sequences based on the mined frequent itemsets
5. **Maximal Phase:** Find the maximal sequences (the longest sequences that cannot be extended) among the set of frequent sequences

An Example of Step 1-3

Step 1: Concert to D'

CustomerID	Customer Sequence
1	<(30),(90)>
2	<(10 20),(30),(40 60 70)>
3	<(30 50 70)>
4	<(30),(40 70),(90)>
5	<(90)>

Step 2: Frequent Itemsets

Freq. Itemsets†	Mapped to
(30)	1
(40)	2
(70)	3
(40 70)	4
(90)	5

minsup=25%

Again, "support" is defined against customer (NOT transaction)!

Step 3: Transformation

CID	Customer Sequence	Transformed Sequence	Mapping
1	<(30),(90)>	<{(30)}{(90)}>	<{1}{5}>
2	<(10 20),(30),(40 60 70)>	<{(30)}{(40),(70),(40 70)}>	<{1}{2,3,4}>
3	<(30 50 70)>	<{(30),(70)}>	<{1,3}>
4	<(30),(40 70),(90)>	<{(30)}{(40),(70),(40 70)}{(90)}>	<{1}{2,3,4}{5}>
5	<(90)>	<{(90)}>	<{5}>

Step 4: AprioriAll Algorithm

L_k = {all frequent 1-itemsets};

$k = 2$; /* k represents the pass number. */

While $(L_{k-1} \neq \emptyset)$ {

$F = F \cup L_k$;

C_k = New candidates of size k generated from L_{k-1};

For each sequence $s \in D'$:

calculate the count of all candidates in C_k that are contained in s;

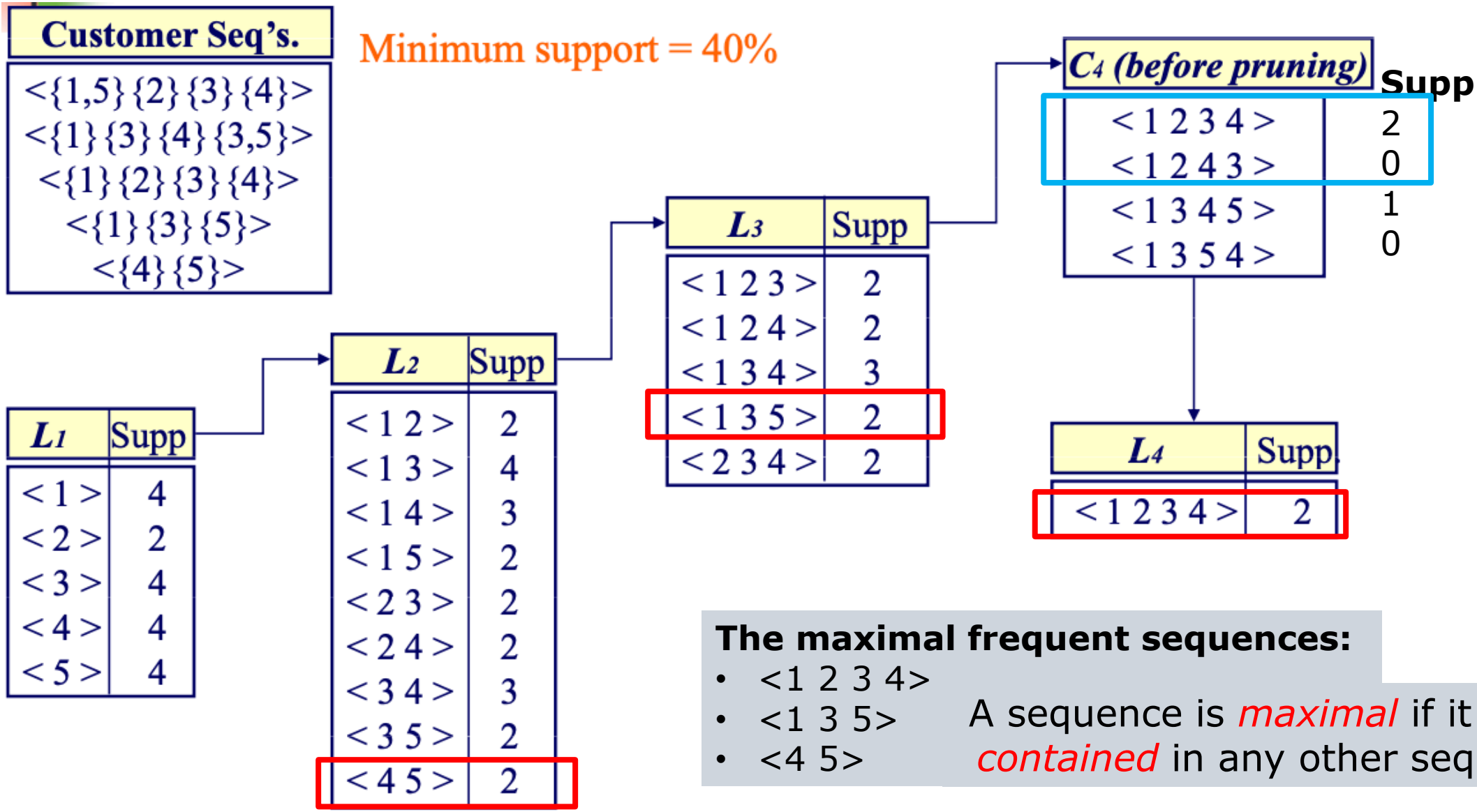
L_k = All candidates in C_k with minimum support ;

k += 1;

}

Return (F);

An Example of Step 4 (Sequence Phase)



The maximal frequent sequences:

- <1 2 3 4>
- <1 3 5>
- <4 5>

A sequence is *maximal* if it is *NOT contained* in any other sequence.

Candidate Sequence Generation

- Similar to non-sequential association rule mining, i.e.,

Step 1: self-joining L_{k-1}

insert into C_k

select $p.itemset_1$, $p.itemset_2$, …, $p.itemset_{k-1}$, $q\_itemset_{k-1}$

from L_{k-1} as p, L_{k-1} as q

where $p.itemset_1 = q.itemset_1$, …, $p.itemset_{k-2} = q.itemset_{k-2}$

~~, and $p.itemset_{k-1} < q.itemset_{k-1}$~~

Step 2: pruning

for all sequence c in C_k do

for all $(k-1)$-subsequence s of c do

if (s is not in L_{k-1}) then delete c from C_k

Example:
L3 = {*abc*, *abd*, *acd*, *ace*, *bcd*}

Self-joining: L3*L3
- *abcd* and *abdc* from *abc* and *abd*
- *acde* and *aced* from *acd* and *ace*

Prunining:
- *abdc* is removed as *adc*/*bdc* is not in L3
- *acde* is removed as *ade*/*cde* is not in L3
- *aced* is removed as *aed*/*ced* is not in L3

C4={*abcd*}

Maximal Sequence

- **Subsequence**: A sequence $< a_1, a_2, \ldots, a_n >$ is *contained* in another sequence $< b_1, b_2, \ldots, b_m >$ if there exists integers $1 \leq j_1 < j_2 < \cdots < j_n \leq m$ such that $a_1 \subseteq b_{j_1}, a_2 \subseteq b_{j_2}, \ldots, a_n \subseteq b_{j_n}$
  - <(3), (4 5), (8)> is contained in <(7), (3 8), (9), (4 5 6), (8)> ? **TRUE**
  - <(3), (5)> is contained in <(3 5)> ? **FALSE**
    - Customers bought “iPad” in Jan (3), then “Apple Pencil” in Feb (5)
    - Customers bought “iPad” and “Apple Pencil” together (3 5)

A sequence s is *maximal* if s *is not contained* in any other sequence.

Forming the Sequential Association Rules

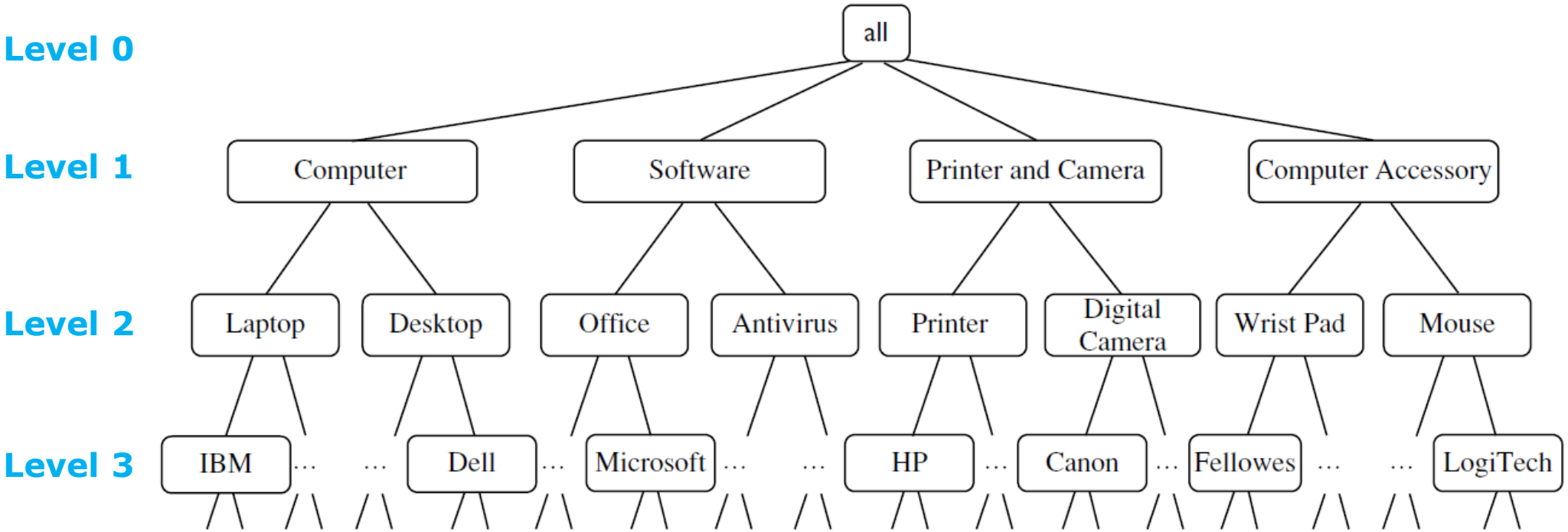
- Again, the user specified confidence is used for rule generation to qualify the strength of sequential association rules.
- The rule generation step is rather simple:
  - For each frequent sequence S, divide the sequence into two nonempty sequential parts S_f and S_l and generate the rule $R: S_f \rightarrow S_l$
  - If R satisfies the threshold, $conf(S_f \rightarrow S_l) = \frac{support(S)}{support(S_f)} \geq min\_conf$, then R is a strong rule and should be output.
- Example: <1 2 3 4> will form 1 → 2 3 4, 1 2 → 3 4, 1 2 3 → 4
  - Rules like 1 3 → 2 4, 1 2 4 → 3 cannot be formed as the temporal order has been distorted!

MINING MULTI-LEVEL ASSOCIATION RULES

Concept Hierarchy for Electronics

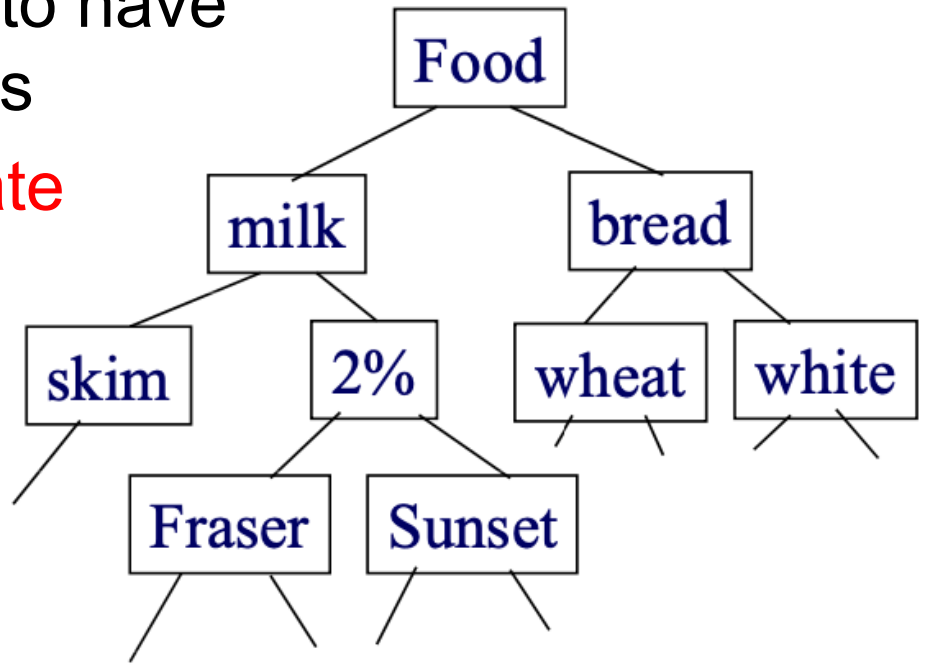
TID	Items Purchased
T100	Apple 17″ MacBook Pro Notebook, HP Photosmart Pro b9180
T200	Microsoft Office Professional 2010, Microsoft Wireless Optical Mouse
T300	Logitech VX Nano Cordless Laser Mouse, Fellowes GEL Wrist Rest
T400	Dell Studio XPS 16 Notebook, Canon PowerShot SD1400
T500	Lenovo ThinkPad X200 Tablet PC, Symantec Norton Antivirus 2010
…	…

TID	Items (Level 2)
T100	Laptop, Printer
T200	Office, Mouse
T300	Mouse, Wrist Pad
T400	Laptop, Digital Camera
T500	Desktop, Antivirus



Multiple-Level Association Rules

- Items often form hierarchy!
  - Strong rules discovered at high abstraction levels – could be *commonsense* knowledge
  - Items at the lower level are expected to have lower support, or trivial at higher levels
  - Rules regarding itemsets at appropriate levels could be more useful.
    - milk → bread
    - milk → wheat bread
    - 2% milk → wheat bread
    - …



Mining Multilevel Association Rules – *top-down*

- Regarding support, counts are accumulated for the calculation of frequent itemsets at each concept level.
  - Start at Level 1 and work downward in the hierarchy toward more specific concept levels, until no more frequent itemsets can be found.
  - For each level, any algorithm for discovering frequent itemsets may be used, e.g., Apriori or its variations.

- **Choosing the right support threshold:**
  - Uniform support for all levels
  - Reduced support at lower levels: *the deeper, the smaller*
  - …

Redundancy Problem

- A rule is redundant if its support is close to the "expected" value based on the rule's ancestors.
  - **R1: milk → wheat bread** [*support* = 8%, *confidence* = 70%]
  - **R2: 2% milk → wheat bread** [*support* = 2%, *confidence* = 72%]
  - R1 is an ancestor of R2 → R2 is redundant!
- An additional step to handle redundancy :
  - Add all the ancestors of each item in the original transactions T to T, and call it **extended transaction T'**
  - Run any association rule mining algorithm (e.g. Apriori) on the extended transactions

An Example

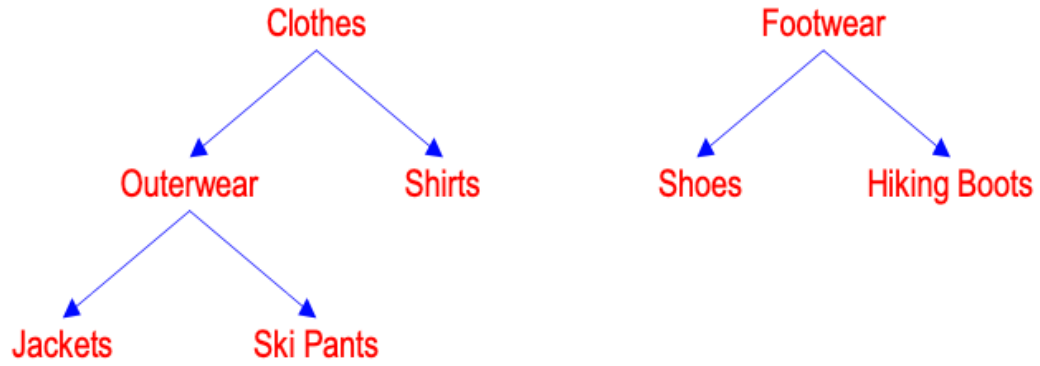
Original Transactions T

Transaction	Items bought
100	Shirt
200	Jacket, Hiking Boots
300	Ski Pants, Hiking Boots
400	Shoes
500	Shoes
600	Jacket

Form the extended transactions

Extended Transactions T'

Transaction	Items bought
100	Shirt, (Clothes)
200	Jacket, (Outerwear, Clothes,), Hiking Boots, (Footwear)
300	Ski Pants, (Outerwear, Clothes,) Hiking Boots, (Footwear)
400	Shoes, (Footwear)
500	Shoes, (Footwear)
600	Jacket, (Outerwear, Clothes)



Find frequent itemsets

Frequent Itemsets

Itemset	Support
{Jacket}	2
{Outerwear}	3
{Clothes}	4
{Shoes}	2
{Hiking Boots}	2
{Footwear}	4
{Outwear, Hiking Boots}	2
{Clothes, Hiking Boots}	2
{Outerwear, Footwear}	2
{Clothes, Footwear}	2

minsup = 30%

Find the rules

Rules

minconf = 60%

Rule	Support	Conf.
{Outerwear}⇒{Hiking Boots}	33%	66%
{Outerwear}⇒{Footwear}	33%	66%
{Hiking Boots}⇒{Outerwear}	33%	100%
{Hiking Boots}⇒{Clothes}	33%	100%

multi-dimensional rules, quantitative rules, rare/negative patterns, compressed patterns

OTHER ASSOCIATION RULES

Multi-Dimensional Associations

- **Single-dimensional Rules**: *a single predicate "buys"*
  - *buys*(X, "milk") → *buys*(X, "bread")

- **Multi-dimensional Rules**: ≥ 2 dimensions or predicates
  - Inter-dimension association rules (*no repeated predicates*)
    - *age*(X, "19-25") → *occupation*(X, "student") → *buys*(X, "Laptop")
  - Hybrid-dimension association rules (*repeated predicates*)
    - *age*(X, "19-25") → *buys*(X, "chips") → *buys*(X, "Laptop")

Mining Quantitative Associations

- Techniques can be categorized by how quantitative attributes, such as age or salary, are treated:
  1. **Static discretization** based on predefined concept hierarchies
     - Partition salary as “0..20K”, “21..30K”, … → each interval is a category
  2. **Dynamic discretization** based on data distribution
     - To satisfy some mining criteria, such as maximizing the confidence or compactness of the rules mined.
  3. **Clustering**: Distance-based association
     - one dimensional clustering, then association
     - e.g., *age*(X, “30-40”) ∧ *salary*(X, “30..40K”) → *buys*(X, “TV”)

Rare Patterns and Negative Patterns

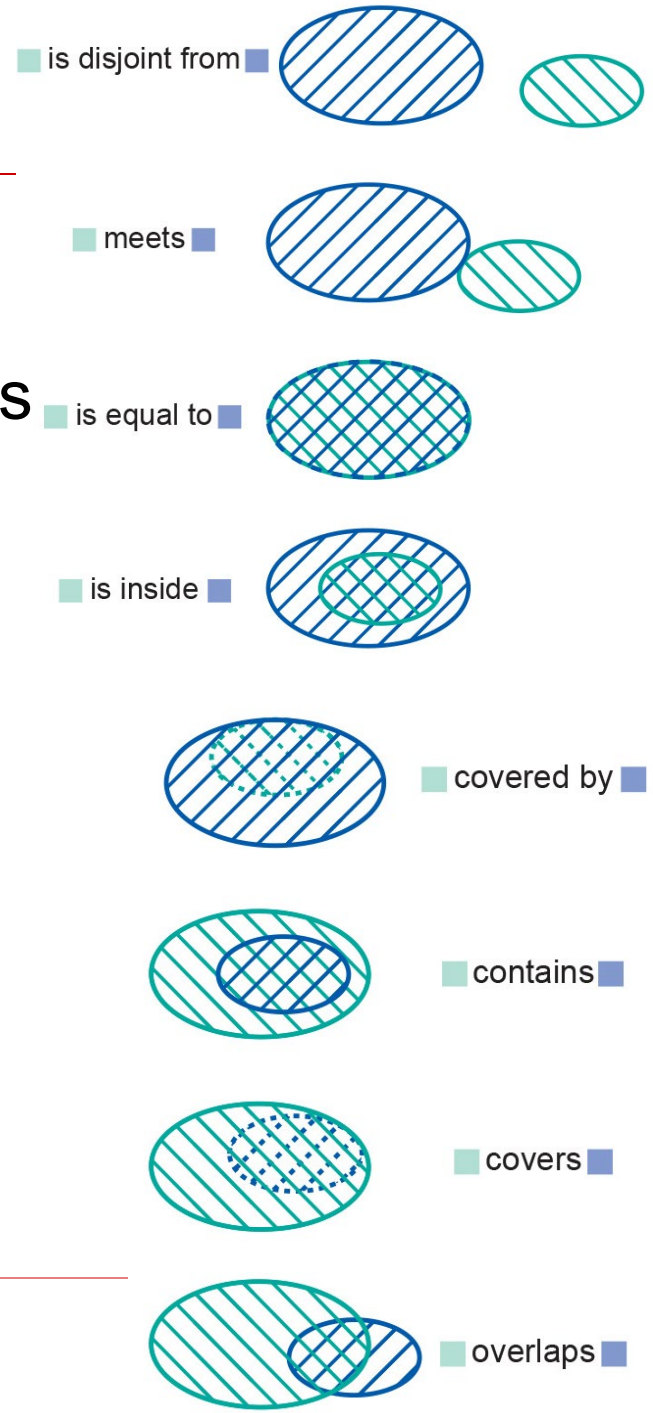
- It is interesting to discover and examine exceptional cases:
  - A **rare pattern** occurs when an item or combination of items appears infrequently in the data (threshold-based, user-specific)
    - In jewelry sales, diamond watches might sell rarely
  - A **negative pattern** occurs when two items are individually frequent but rarely purchased together, i.e., $\sup(X \cup Y) < \sup(X) \times \sup(Y)$
    - In supermarket data, if we find that customers frequently buy diet coke or classic coke but not both, then buying them together is considered a negative (correlated) pattern.

Spatial Association Rules

- Spatial association rule: $A \rightarrow B$ [s%, c%]
  - A and B are sets of spatial or nonspatial predicates
    - Topological relations: *disjoint*, meets, inside, etc.
    - Spatial orientations: *left_of*, *west_of*, *under*, etc.
    - Distance information: *close_to*, *within_distance*, etc.
- Examples

  is_a(X, urban_district) ∧ *intersect*(X, MTR_line)
  → *adjacent_to*(X, Victoria_Harbour) [7%, 85%]

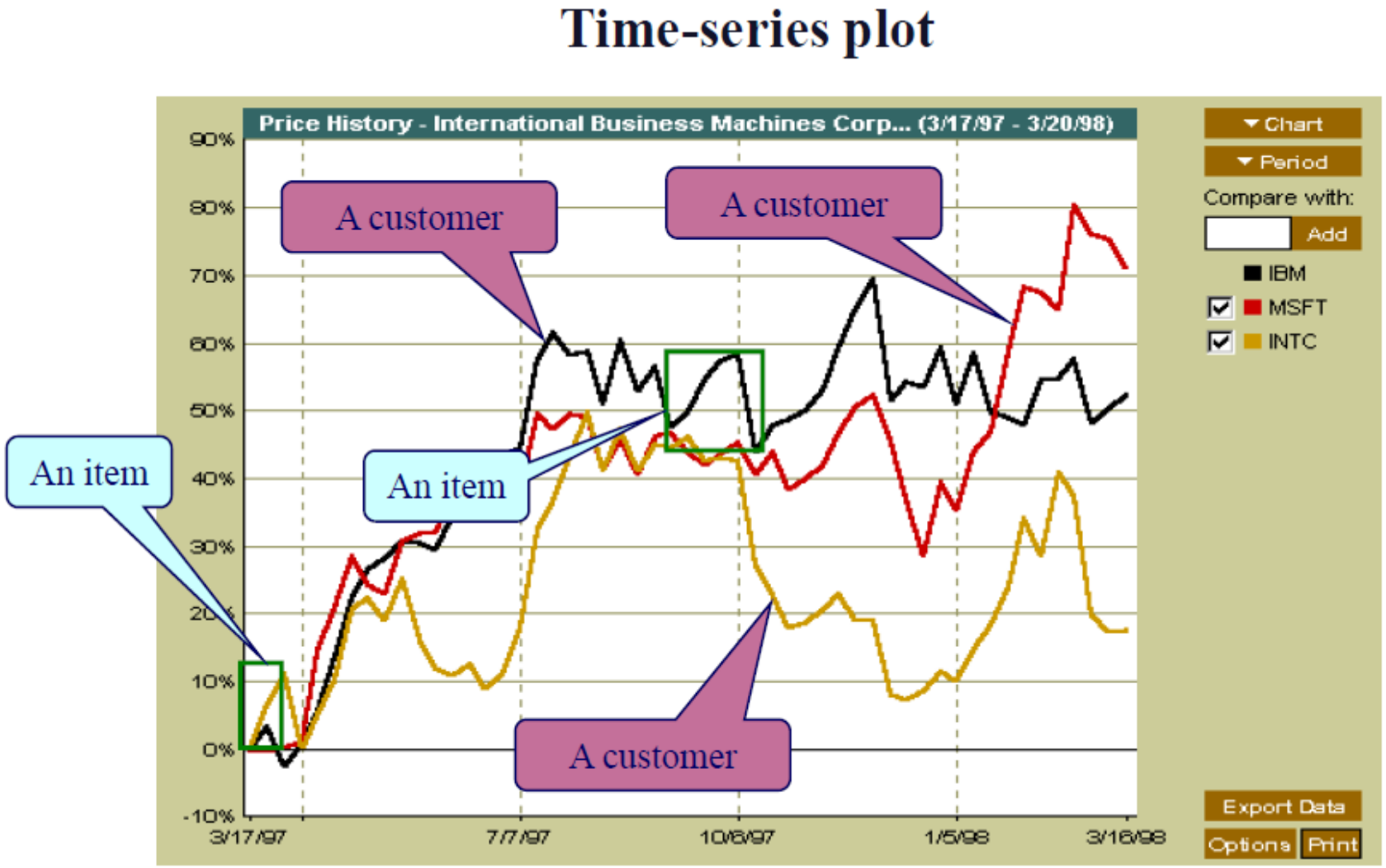
  is_a(X, tourist_attraction) ∧ *within*(X, Sichuan_province)
  → *close_to*(X, panda_reserve) [2%, 88%]



Contributions of Association Rule Mining

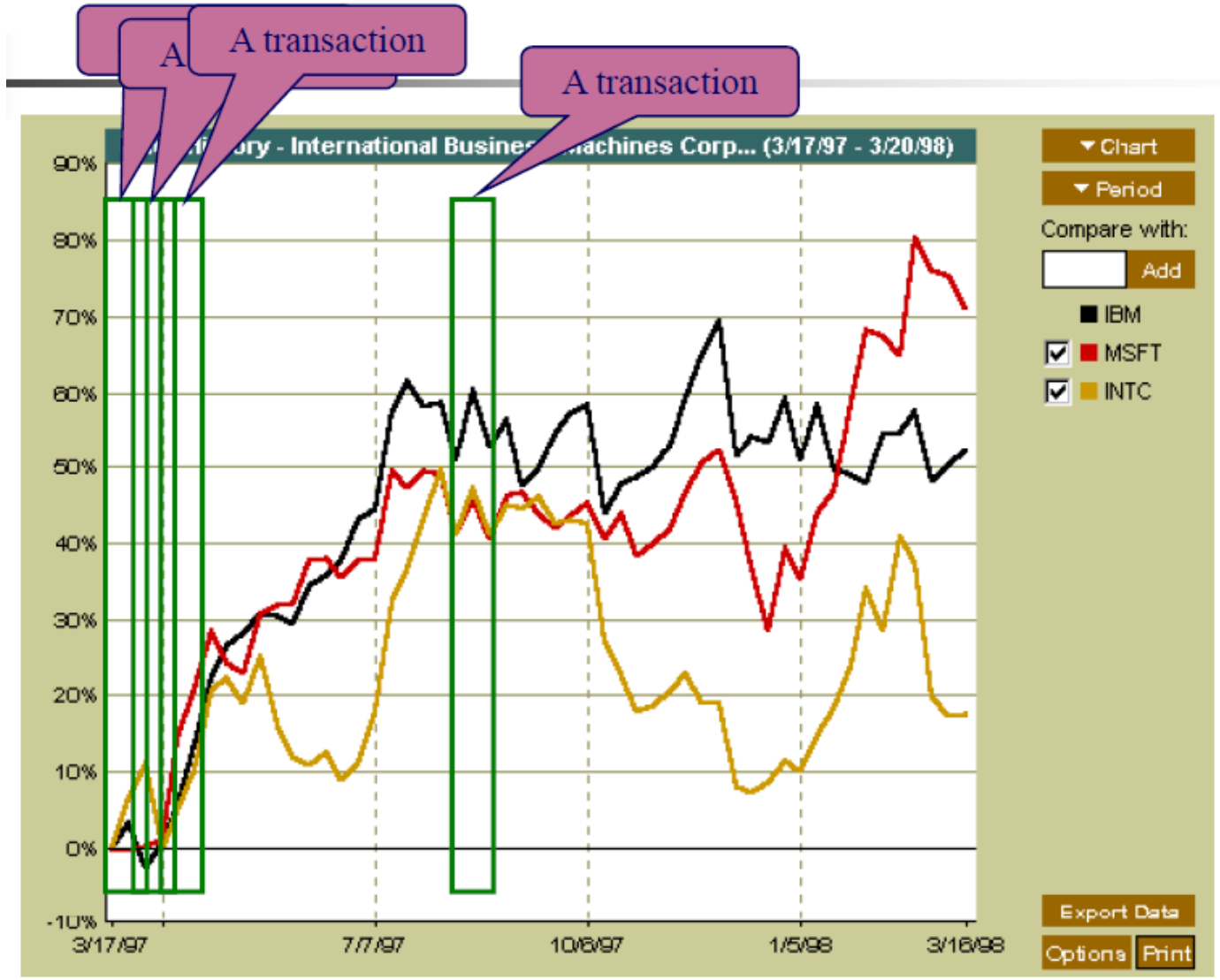
- Basically, it is more like a statistical technique, i.e., not too much intelligence is embedded!
- But it is easy and flexible to apply, offering efficient DB solutions to very large database applications.
  - Keep track of the supposedly huge number of candidate itemsets through limited RAM.
  - Can be applied to virtually all applications and all available data.
    - What is a transaction in your application?
    - What is an item in your application? – a basic unit
    - What is a customer in your application? (*for sequential association rules*)

Mining Stock Data – Intra-stock Mining



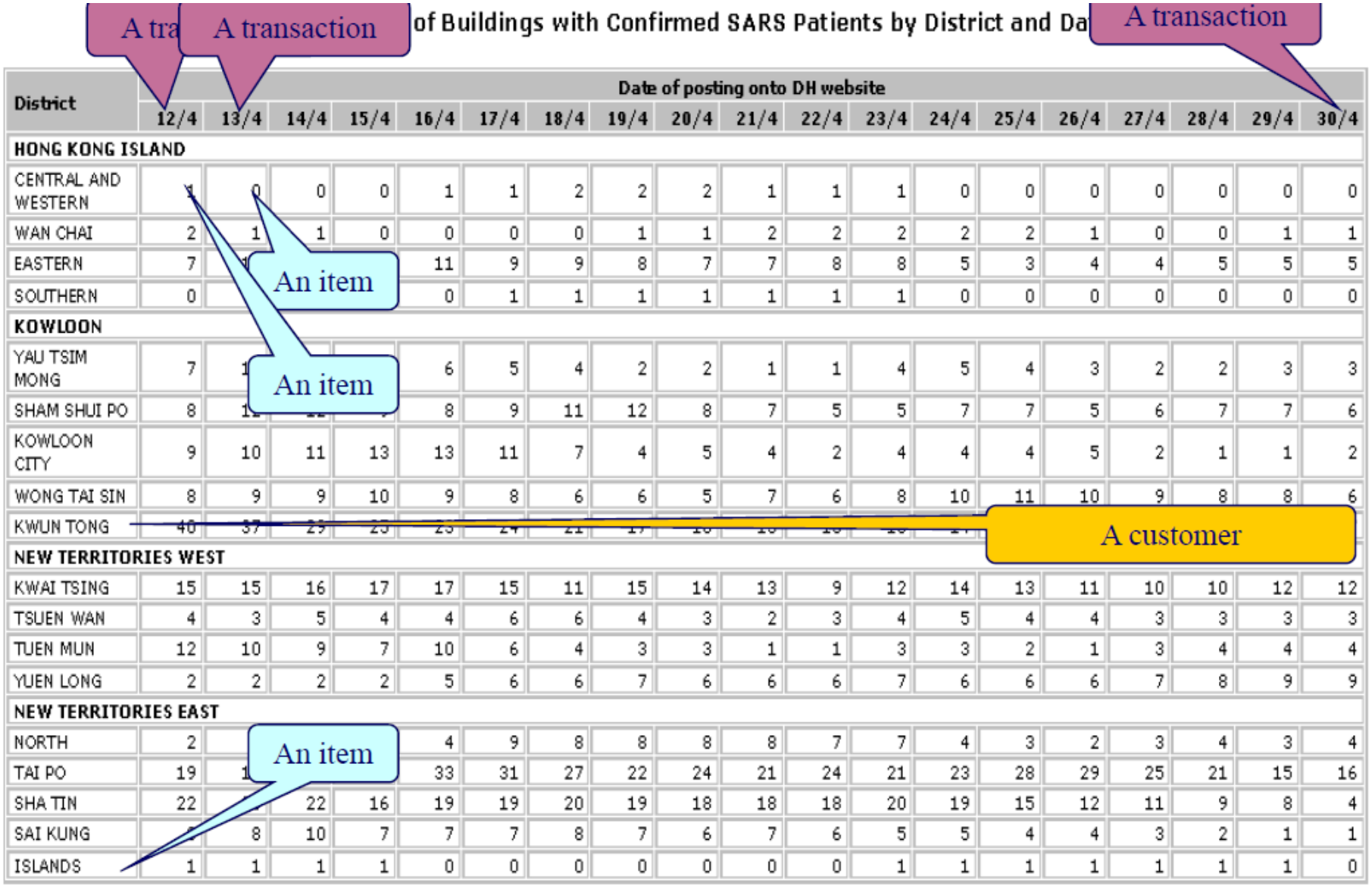
- ***Customer***: Stock, e.g., IBM, MSFC, INTC, …
- ***Item***: Trends (each price movement), e.g., Go-up, Go-down, …
- ***Rules***: e.g., "go-up, go-up, and then go-down"

Mining Stock Data – Inter-stock Mining



- Each time window is treated as a "***Transaction***"
- To analyze the behavior of *different stocks* within a time window
  - open-high-close-low
  - open-low-close-low
  - …

Mining the Data of Confirmed SARS Patients



District	Date of posting onto DH website																		
	12/4	13/4	14/4	15/4	16/4	17/4	18/4	19/4	20/4	21/4	22/4	23/4	24/4	25/4	26/4	27/4	28/4	29/4	30/4
HONG KONG ISLAND																			
CENTRAL AND WESTERN	1	0	0	0	1	1	2	2	2	1	1	1	0	0	0	0	0	0	0
WAN CHAI	2	1	1	0	0	0	0	1	1	2	2	2	2	2	1	0	0	1	1
EASTERN	7	1			11	9	9	8	7	7	8	8	5	3	4	4	5	5	5
SOUTHERN	0				0	1	1	1	1	1	1	1	0	0	0	0	0	0	0
KOWLOON																			
YAU TSIM MONG	7	1			6	5	4	2	2	1	1	4	5	4	3	2	2	3	3
SHAM SHUI PO	8	11			8	9	11	12	8	7	5	5	7	7	5	6	7	7	6
KOWLOON CITY	9	10	11	13	13	11	7	4	5	4	2	4	4	4	5	2	1	1	2
WONG TAI SIN	8	9	9	10	9	8	6	6	5	7	6	8	10	11	10	9	8	8	6
KWUN TONG	40	37	29	25	25	24	21	17	16	15	15	16	17						
NEW TERRITORIES WEST																			
KWAI TSING	15	15	16	17	17	15	11	15	14	13	9	12	14	13	11	10	10	12	12
TSUEN WAN	4	3	5	4	4	6	6	4	3	2	3	4	5	4	4	3	3	3	3
TUEN MUN	12	10	9	7	10	6	4	3	3	1	1	3	3	2	1	3	4	4	4
YUEN LONG	2	2	2	2	5	6	6	7	6	6	6	7	6	6	6	7	8	9	9
NEW TERRITORIES EAST																			
NORTH	2				4	9	8	8	8	8	7	7	4	3	2	3	4	3	4
TAI PO	19	1			33	31	27	22	24	21	24	21	23	28	29	25	21	15	16
SHA TIN	22		22	16	19	19	20	19	18	18	18	20	19	15	12	11	9	8	4
SAI KUNG		8	10	7	7	7	8	7	6	7	6	5	5	4	4	3	2	1	1
ISLANDS	1	1	1	1	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0

- ***Item***: # patients
- ***Transaction***: Daily record
- ***Customer***: each region

Compressed Patterns

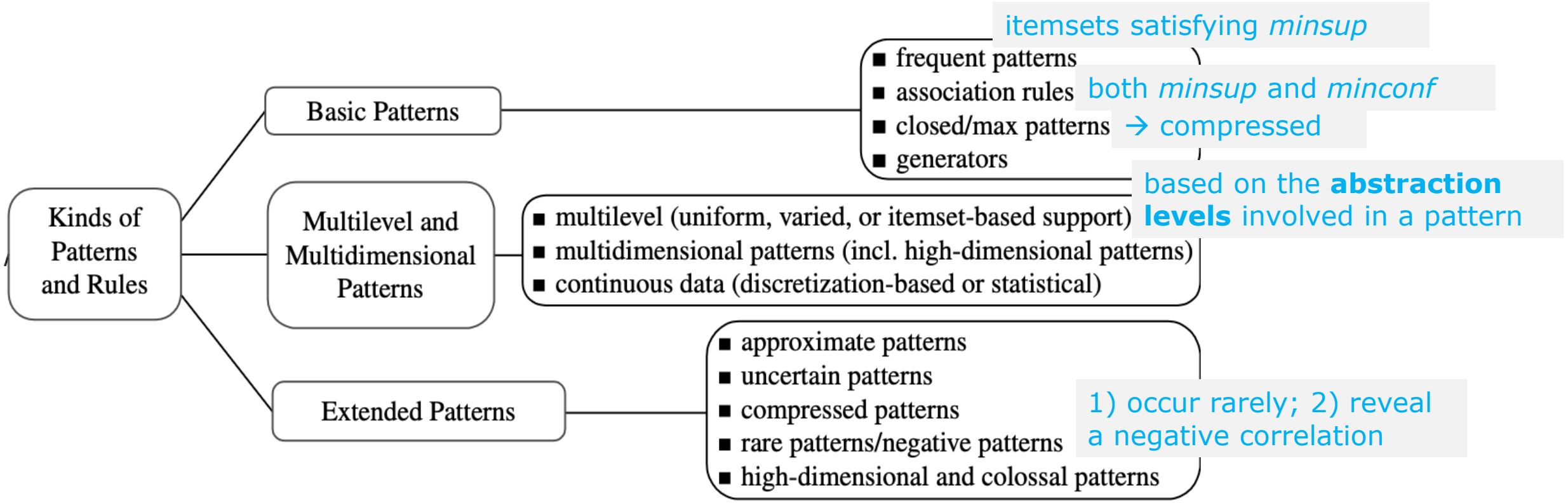
- Using *thresholds* to control # rules has limited effect.
  - Too low → an explosive number of output patterns
  - Too high → the discovery of only commonsense patterns

- X is a closed frequent itemset in a dataset D if X is frequent and there exists no proper super-itemset Y of X such that Y has the same support count as X in D.
- X is a maximal frequent itemset in data set D if X is frequent and there exists no super-itemset Y such that $X \subset Y$ and Y is frequent.

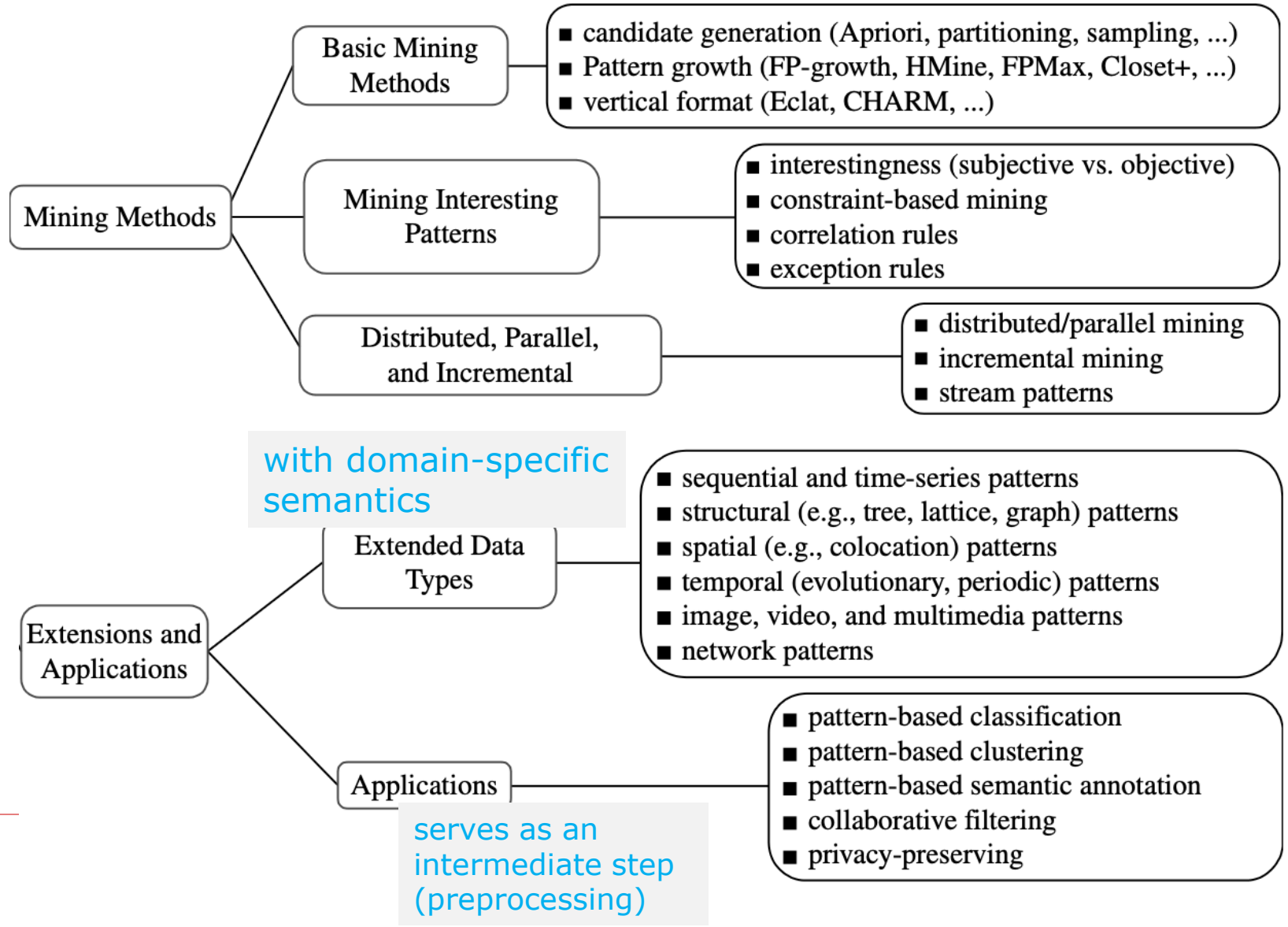
Item	Support
A	5
B	7
C	10
AB	5
AC	5
BC	7
ABC	5

Item	Support
A	5
B	7
C	10
AB	5
AC	5
BC	7
ABC	4

Summary: Patterns and Rules



Summary: Methods and Applications



Email: fengmei.jin@polyu.edu.hk

Office: PQ747

THANK YOU!