Faculty of Computer Science and Engineering
Ho Chi Minh City University of Technology

Chapter 6
Association Rule

TRAN MINH QUANG
quangtran@hcmut.edu.vn
http://researchmap.jp/quang

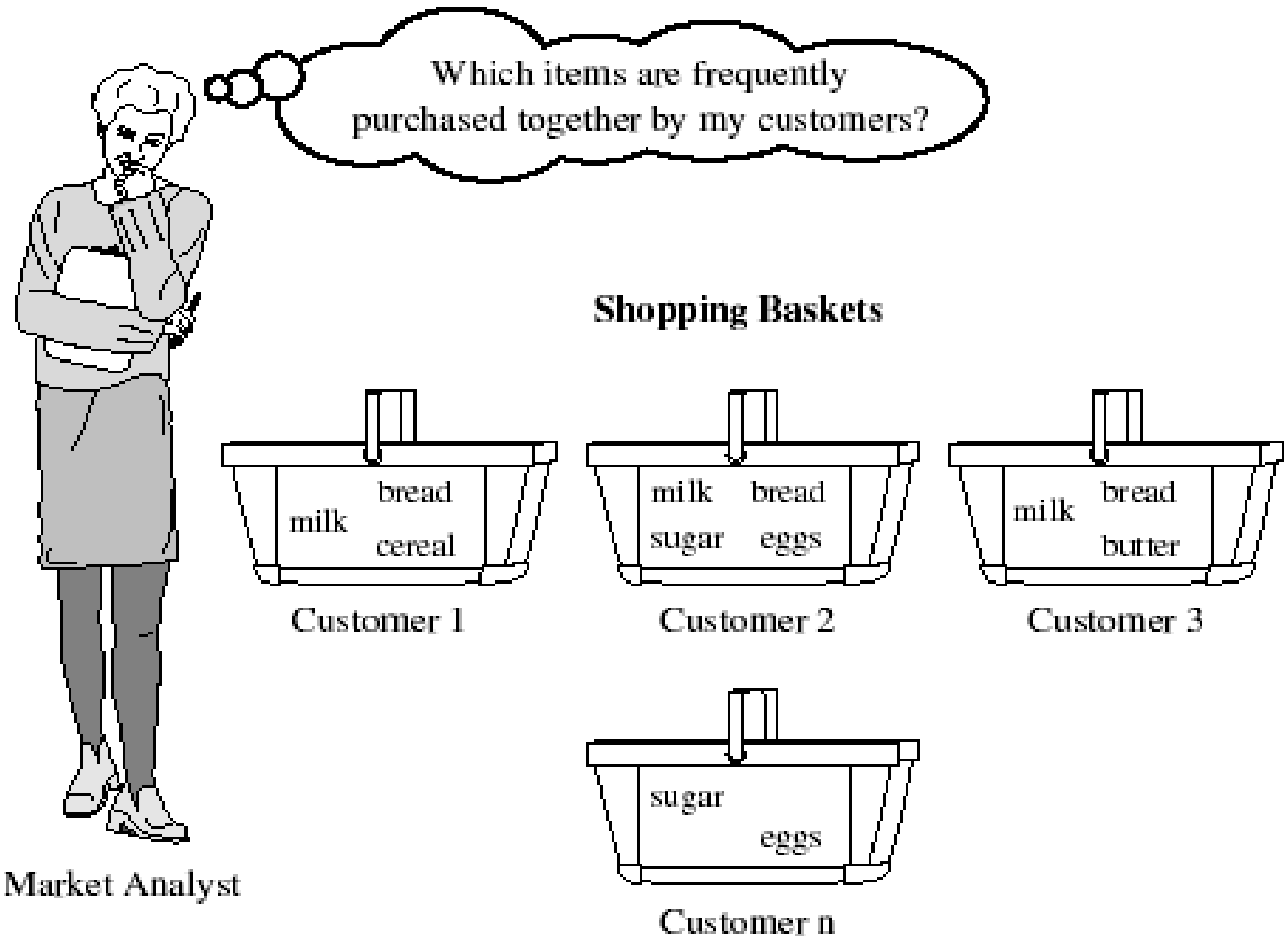
2020/5/22

CONTENT

1. Situations
2. Overview on association rules
3. Association rule representation
4. Frequent itemset mining
    1. Apriori
    2. FP-Growth
5. Mining association rules from frequent itemsets
6. Mining association rules based on constraints
7. Correlation analysis
8. Summary

Dr. Tran Minh Quang – quangtran@hcmut.edu.vn

1. Situation 1 – Basket Analysis



Dr. Tran Minh Quang – quangtran@hcmut.edu.vn

1. Situation 2 – Recommendation

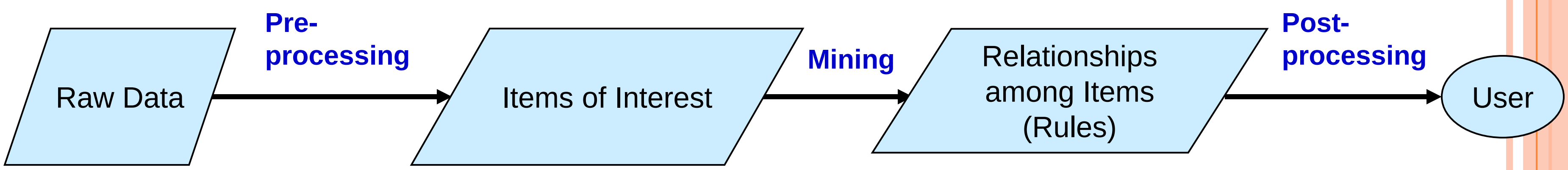
1. Situation 2 – Recommendation

1. SITUATION …

- Basket data analysis
- Cross-marketing, recommendation,…
- Catalog design
- Classification and clustering with frequent patterns
- …

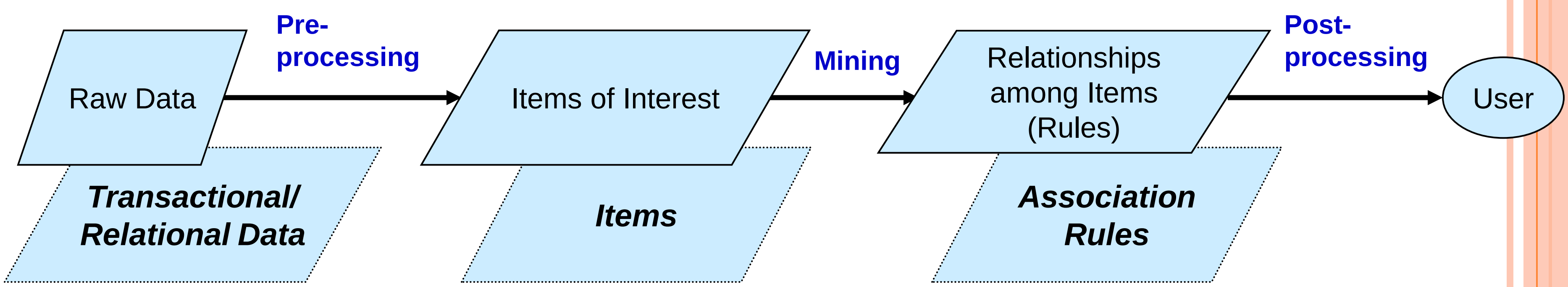
2. Overview on Association Rule Mining

- Association rule mining process



2. OVERVIEW ON ASSOCIATION RULE MINING

- Association rule mining process



Transaction	Items_bought
2000	A, B, C
1000	A, C
4000	A, D
5000	B, E, F
…	

A, B, C, D, F, …

$A \rightarrow C$ (50%, 66.6%)
…

Basket-based analysis

Dr. Tran Minh Quang – quangtran@hcmut.edu.vn

2. OVERVIEW ON ASSOCIATION RULE MINING

- Basic concepts
  - Item
  - Itemset
  - Transaction
  - Association and association rules
  - Support
  - Confidence
  - Frequent itemset
  - Strong association rules

Dr. Tran Minh Quang – quangtran@hcmut.edu.vn

2. OVERVIEW ON ASSOCIATION RULE MINING

- AllElectronics dataset (after preprocessing)

TID	*List of item_IDs*
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

2. Overview on Association Rule Mining

- AllElectronics dataset (after preprocessing)

TID	List of item_IDs
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

Itemsets:
{I1, I2, I5}, {I2}, …

Item: I4

Transaction: T800

2. Overview on Association Rule Mining

- Basic concepts
  - Item: pattern, sample, object of interest
  - $J = \{I_1, I_2, \ldots, I_m\}$: a set of all m items in the dataset
  - Itemset
    - A set of items
    - k-itemset: itemset of k items
  - Transaction
    - A record in a transactional dataset
    - A set of T items in the same transaction/record

Dr. Tran Minh Quang – quangtran@hcmut.edu.vn

2. Overview on Association Rule Mining

- Basic concepts
  - Association vs. association rule
    - Association: occurrence of items in the same transaction(s)
    - Represents the linkage/association between items/itemsets
  - Association rule: a criteria/rule for the association between itemsets
    - Represents the linkage (with some conditions) between itemsets
    - Let A and B be itemsets, an association rule between A and B is **A → B** (B occurs in the condition that A occurs)

2. Overview on Association Rule Mining

- Basic concept
  - Support: measure the occurrence frequency of items/itemsets
  - Minimum support threshold (MinSup): is the minimum support value defined by user to validate the "support" of a frequent itemset/association
  - Confidence: to measure the occurrence frequency of an itemset upon the occurrence of another itemset
  - Minimum confidence threshold (MinConf): is the minimum confidence value defined by user to validate the "confidence" of an association rule.

Dr. Tran Minh Quang – quangtran@hcmut.edu.vn

2. Overview on Association Rule Mining

- Basic concepts
  - Frequent itemset: is an itemset whose support satisfies the minimum support threshold

    A is a frequent itemset iff

    $$\text{support}(A) \geq \text{MinSup}$$

  - Strong association rule: is an association rule whose support and confidence satisfy MinSup and MinConf
    - Give A→B, where A and B are itemsets
    - A→B is a strong association rule iff

      $$\text{support}(A \rightarrow B) \geq \text{MinSup} \text{ and}$$

      $$\text{confidence}(A \rightarrow B) \geq \text{MinConf}$$

2. OVERVIEW ON ASSOCIATION RULE MINING

- Association rule categories
  - Boolean association rule vs. quantitative association rule
  - Single-dimensional association rule vs. multidimensional association rule
  - Single-level association rule vs. multilevel association rule
  - Association rule vs. correlation rule

Dr. Tran Minh Quang – quangtran@hcmut.edu.vn

2. Overview on Association Rule Mining

- Boolean vs. quantitative association rules
  - Boolean association rule: represents the association of occurrence/absence of itemsets

    Computer → Financial_management_software [support=50%, confidence=60%]

  - Quantitative association rule: represents the association of between quantitative items/features

    Age(X, "30..39") ∧ Income(X, "42K..48K") → buys(X, high resolution TV)

    [support=50%, confidence=60%]

2. OVERVIEW ON ASSOCIATION RULE MINING

- Single-dimensional vs. multidimensional association rules
  - Single-dimensional association rule: is the rule that relates to only one dimension of items

    Buys(X, "computer") → Buys(X, "financial_management_software")

  - Multidimensional association rule: is the rule that relates to multiple dimensions of items

    Age(X, "30..39") → Buys(X, "computer")

2. Overview on Association Rule Mining

- Single-level vs. multilevel association rules
  - Single-level association rule: is the rule that relates to items/features in one level of abstraction

    Age(X, "30..39") → Buys(X, "computer")

    Age(X, "18..29") → Buys(X, "camera")

  - Multilevel association rule: is the rule that relates to items/features in multiples levels of abstraction

    Age(X, "30..39") → Buys(X, "laptop computer")

    Age(X, "30..39") → Buys(X, "computer")

2. OVERVIEW ON ASSOCIATION RULE MINING

- Association rule vs. correlation rule
  - Association rule: strong association rule A→B is that satisfies minimum support threshold and minimum confidence threshold
  - Correlation rule: strong correlation rules A → B is that satisfies correlation conditions in statistics (using different statistical correlation measures)

Dr. Tran Minh Quang – quangtran@hcmut.edu.vn

3. ASSOCIATION RULE REPRESENTATION

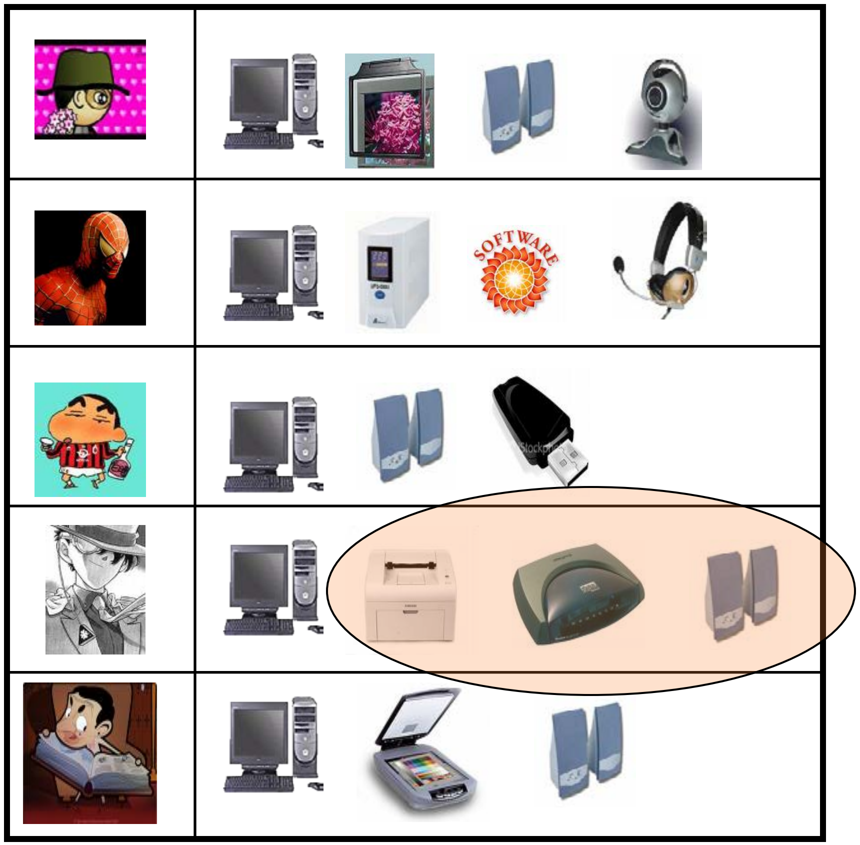
Rule form:

A→B [support, confidence]

Where,

- **A, B are frequent itemsets**
- Support(A→B) = Support(A U B) ≥ min_sup
- Confidence(A→B) = P(B|A) = Support(A U B)/Support(A) ≥ min_conf

3. Association rule representation



 + ?

- Support

Sup = 1

Sup = 1

- Confidence

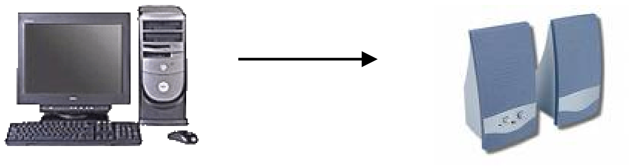
C( → ) = $\frac{4}{5}$

4. MINING FREQUENT ITEMSETS/PATTERNS

- 2 steps in association rule mining
  - Finding frequent itemsets/patterns

 Sup = 5,  Sup = 4

  - Mining association rules from frequent itemsets/patterns



4. Mining frequent itemsets/patterns

- Apriori method: mining frequent itemsets with candidate itemsets using prior knowledge
  - R. Agrawal, R. Srikant. Fast algorithms for mining association rules. In VLDB 1994, pp. 487-499
- FP-Growth: using FP-tree
  - J. Han, J. Pei, Y. Yin. Mining frequent patterns without candidate generation. In SIGMOD 2000, pp. 1-12

4.1. MINING FREQUENT ITEMSETS - APRIORI

- Use the prior knowledge about the characteristics of frequent itemsets
- Iterate the searching process to find frequent itemsets at each level (level-wise search)
  - k+1-itemsets: generated from k-itemsets
  - At each level, check all the data set to identify frequent itemsets

- Apriori property (to reduce the search space): All subsets of a frequent itemset are frequent itemsets
- Anti-monotone: If X is NOT a frequent itemset then **neither** does {X U Y}

4.1. Mining frequent itemsets - Apriori

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

Scan D → C_2

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

L_2

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

C_3

itemset
{2 3 5}

Scan D → L_3

itemset	sup
{2 3 5}	2

4.1. Mining frequent itemsets - Apriori

- AllElectronics dataset (after preprocessing)

TID	*List of item_IDs*
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

4.1. MINING FREQUENT ITEMSETS - APRIORI

min_sup = 2/9

minimum support count = 2

Scan D for count of each candidate

C_1

Itemset	Sup. count
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

Compare candidate support count with minimum support count

L_1

Itemset	Sup. count
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

Generate C_2 candidates from L_1

C_2

Itemset
{I1, I2}
{I1, I3}
{I1, I4}
{I1, I5}
{I2, I3}
{I2, I4}
{I2, I5}
{I3, I4}
{I3, I5}
{I4, I5}

Scan D for count of each candidate

C_2

Itemset	Sup. count
{I1, I2}	4
{I1, I3}	4
{I1, I4}	1
{I1, I5}	2
{I2, I3}	4
{I2, I4}	2
{I2, I5}	2
{I3, I4}	0
{I3, I5}	1
{I4, I5}	0

Compare candidate support count with minimum support count

L_2

Itemset	Sup. count
{I1, I2}	4
{I1, I3}	4
{I1, I5}	2
{I2, I3}	4
{I2, I4}	2
{I2, I5}	2

Generate C_3 candidates from L_2

C_3

Itemset
{I1, I2, I3}
{I1, I2, I5}

Scan D for count of each candidate

C_3

Itemset	Sup. count
{I1, I2, I3}	2
{I1, I2, I5}	2

Compare candidate support count with minimum support count

L_3

Itemset	Sup. count
{I1, I2, I3}	2
{I1, I2, I5}	2

4.1. Mining frequent itemsets - Apriori

- Pseudo-code:

  C_k: Candidate itemset of size k

  L_k : frequent itemset of size k

  L_1 = {frequent items};

  While (L_k !=∅; 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 $\cup_k L_k$;

Dr. Tran Minh Quang – quangtran@hcmut.edu.vn

4.1. MINING FREQUENT ITEMSETS - APRIORI

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

4.1. Mining frequent itemsets - Apriori

- Suppose the items in L_{k-1} are listed in an order
- Step 1: self-joining L_{k-1}

  insert into $\boldsymbol{C_k}$

  select $\boldsymbol{p.item_1, p.item_2, \ldots, p.item_{k-1}, q.item_{k-1}}$

  from $\boldsymbol{L_{k-1}\ p, L_{k-1}\ q}$

  where $\boldsymbol{p.item_1=q.item_1, \ldots, p.item_{k-2}=q.item_{k-2}, p.item_{k-1} < q.item_{k-1}}$
- Step 2: pruning

  forall ***itemsets c in*** $\boldsymbol{C_k}$ do

  forall ***(k-1)-subsets s of c*** do

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

4.1. Mining frequent itemsets - Apriori

- Apriori's main characteristics
  - Create many candidates for frequent itemsets
    - 10^4 frequent 1-itemsets → more than 10^7 ($\approx 10^4(10^4-1)/2$) 2-itemsets as candidates
    - For each k-itemset, it examines 2^k -1 itemsets of candidates
  - Scan the original data (D) many times for calculating the supports
    - Huge cost when the size of candidate itemsets increase
    - If k-itemsets is mined D is scanned k+1 times

Dr. Tran Minh Quang – quangtran@hcmut.edu.vn

4.1. Mining frequent itemsets - Apriori

- Apriori's improvement techniques
  - Techniques based on hashing: An k-itemset whose hashing bucket count is smaller than minimum support threshold **is not** a frequent itemset.
  - Reduce the scan time: A transaction that does not contain **any** frequent k-itemset are not needed to be examined at later level (i.e., for identify k+1-itemset).
  - Partitioning: An itemset must be frequent in at least one partition to be considered as frequent in the whole dataset.
  - Sampling: Mine the frequent itemsets from samples (via sampling) with a smaller support threshold. It is necessary to have a method for validating the completeness.
  - Dynamic itemset counting: Add an itemset to the candidates only iff all of its subsets are **predicted** to be "frequent".

Dr. Tran Minh Quang – quangtran@hcmut.edu.vn

4.2. Mining frequent itemsets - FPGrowth

- Compact the data into a Frequent Pattern tree (FP-tree)
  - Reduce the size of the examined dataset -> reduce the mining cost
    - **Infrequent items are discarded early in the procees**
- Divide-and-conquer approach
  - The mining process is divided into smaller tasks
    1. Build the FP-tree
    2. Mine frequent itemsets using the FP-tree
- Avoid creating large candidate itemsets
  - Each mining task examines a small portion of the dataset

4.2. Mining frequent itemsets - FPGrowth

- 1. Building the FP-tree
  - 1.1. Scan D to find *frequent 1-itemsets*
  - 1.2. Order *frequent 1-itemsets* in the descending of support count (frequency)
  - 1.3. Scan D again (2^{nd} scan), build the FP-tree
    - Initiate with root node, label it with "null" or {}
    - Each transaction/tuples will correspond to a branch in the FP-tree
    - Each node contains an item of the transaction
      - Items in a transaction are shorted in descending order
      - Each node associates with support count of the corresponding item
    - Transactions contain the same items will create shared prefix branches

4.2. Mining frequent itemsets - FPGrowth

Input:

- D, a transaction database;
- *min_sup*, the minimum support count threshold.

Output: The complete set of frequent patterns.

Method:

1. The FP-tree is constructed in the following steps:

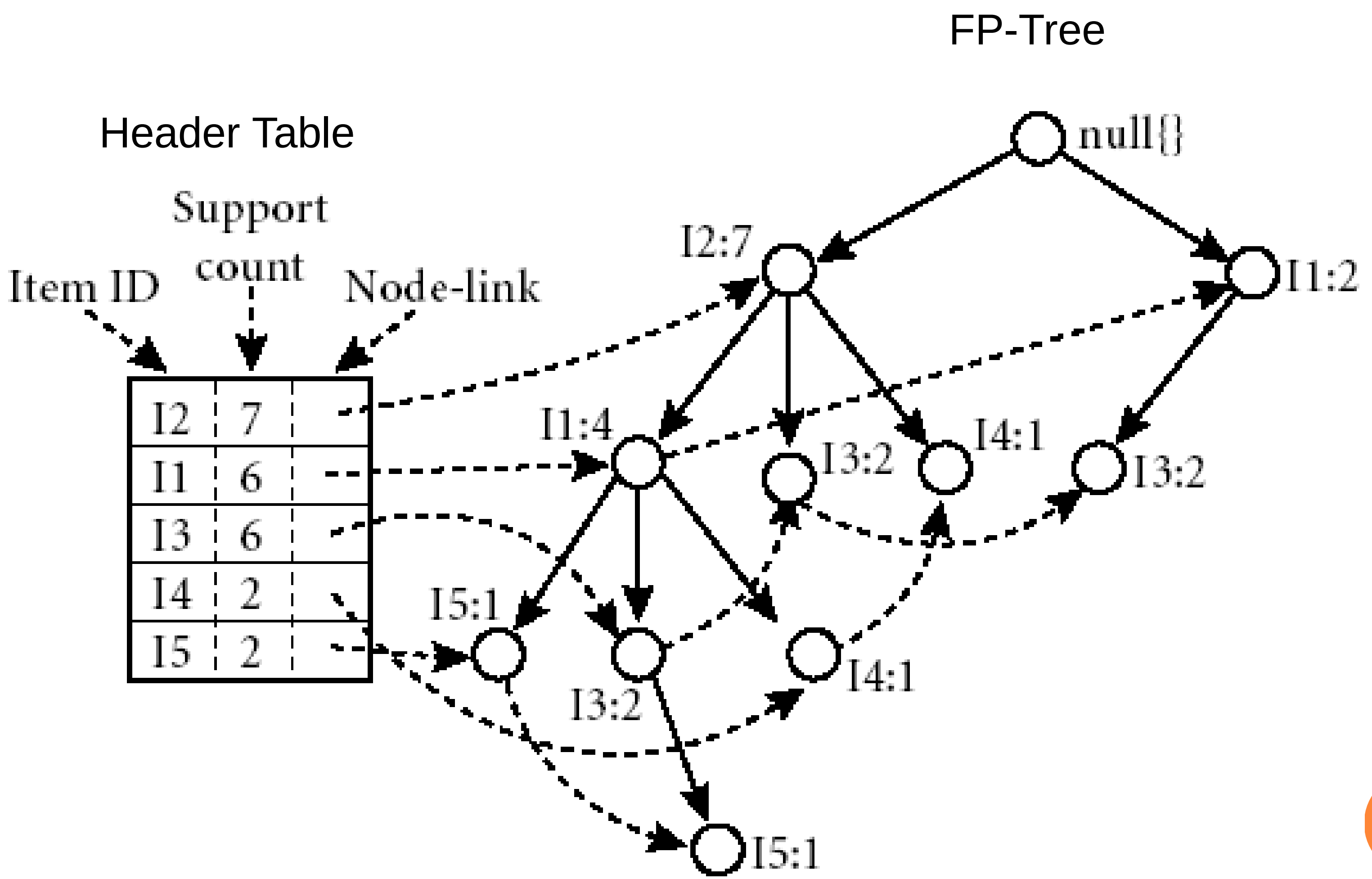
   (a) Scan the transaction database D once. Collect F, the set of frequent items, and their support counts. Sort F in support count descending order as L, the *list* of frequent items.

   (b) Create the root of an FP-tree, and label it as "null." For each transaction *Trans* in D do the following.

   Select and sort the frequent items in *Trans* according to the order of L. Let the sorted frequent item list in *Trans* be $[p|P]$, where p is the first element and P is the remaining list. Call insert_tree($[p|P]$, T), which is performed as follows. If T has a child N such that $N.item\text{-}name = p.item\text{-}name$, then increment N's count by 1; else create a new node N, and let its count be 1, its parent link be linked to T, and its node-link to the nodes with the same *item-name* via the node-link structure. If P is nonempty, call insert_tree(P, N) recursively.

2. The FP-tree is mined by calling FP_growth(*FP_tree*, *null*), which is implemented as follows.

4.2. MINING FREQUENT ITEMSETS - FPGROWTH



4.2. Mining frequent itemsets - FPGrowth

- 2. Mining frequent itemsets from the FP-tree
  - 2.1. Generate conditional pattern base for each node of the FP-tree
    - Accumulate the prefix paths with frequency of that node
  - 2.2. Create conditional FP-tree from each conditional pattern base
    - Accumulate frequency for each item in each base
    - Build conditional FP-tree for frequent items of that base
  - 2.3. Mine the conditional FP-tree for frequent itemsets recursively
    - If conditional FP-tree has a single path then list all itemsets

4.2. Mining frequent itemsets - FPGrowth

- FP-Growth algorithm

procedure FP_growth(*Tree*, α)

    if ***Tree*** contains a single path ***P*** **then**

        for each combination (denoted as β) of the nodes in the path ***P***

            generate pattern $\beta \cup \alpha$ with $support\_count = minimum\ support\ count\ of\ nodes\ in\ \beta$;

    else for each a_i in the header of ***Tree*** {

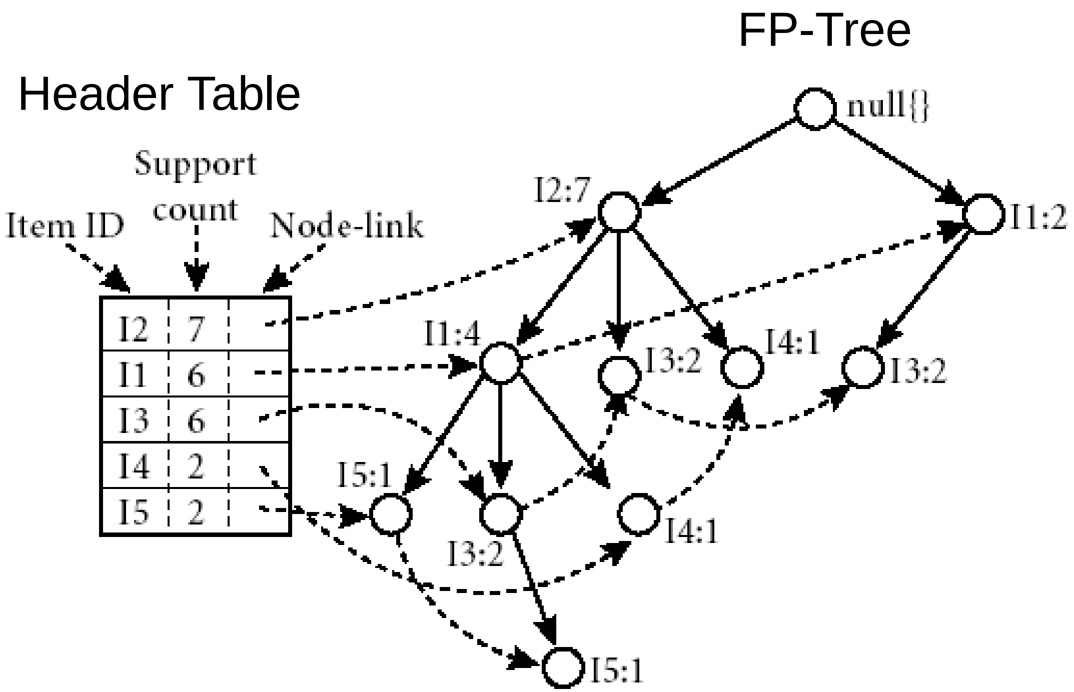
        generate pattern $\beta = a_i \cup \alpha$ with $support\_count = a_i.support\_count$;

        construct β's conditional pattern base and then β's conditional FP_tree $Tree_\beta$;

        if $Tree_\beta \neq \emptyset$ **then**

            call FP_growth($Tree_\beta$, β); }

4.2. Mining frequent itemsets - FPGrowth



Item	Conditional Pattern Base	Conditional FP-tree	Frequent Patterns Generated
I5	{{I2, I1: 1}, {I2, I1, I3: 1}}	⟨I2: 2, I1: 2⟩	{I2, I5: 2}, {I1, I5: 2}, {I2, I1, I5: 2}
I4	{{I2, I1: 1}, {I2: 1}}	⟨I2: 2⟩	{I2, I4: 2}
I3	{{I2, I1: 2}, {I2: 2}, {I1: 2}}	⟨I2: 4, I1: 2⟩, ⟨I1: 2⟩	{I2, I3: 4}, {I1, I3: 4}, {I2, I1, I3: 2}
I1	{{I2: 4}}	⟨I2: 4⟩	{I2, I1: 4}

4.2. MINING FREQUENT ITEMSETS – FPGROWTH (ANOTHER EXAMPLE)

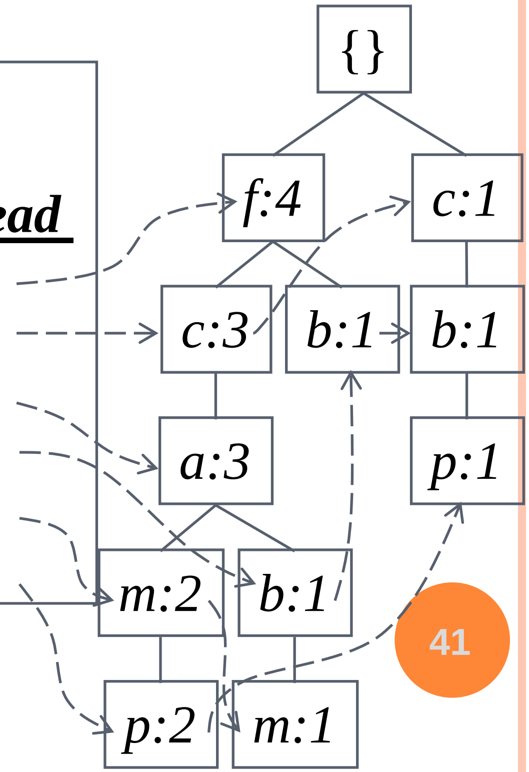
TID	*Items bought*	*(ordered) frequent items*
100	{*f, a, c, d, g, i, m, p*}	{*f, c, a, m, p*}
200	{*a, b, c, f, l, m, o*}	{*f, c, a, b, m*}
300	{*b, f, h, j, o, w*}	{*f, b*}
400	{*b, c, k, s, p*}	{*c, b, p*}
500	{*a, f, c, e, l, p, m, n*}	{*f, c, a, m, p*}

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

Item	*frequency*	*head*
f	*4*	
c	*4*	
a	*3*	
b	*3*	
m	*3*	
p	*3*	



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

4.2. FP-Growth Example: 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

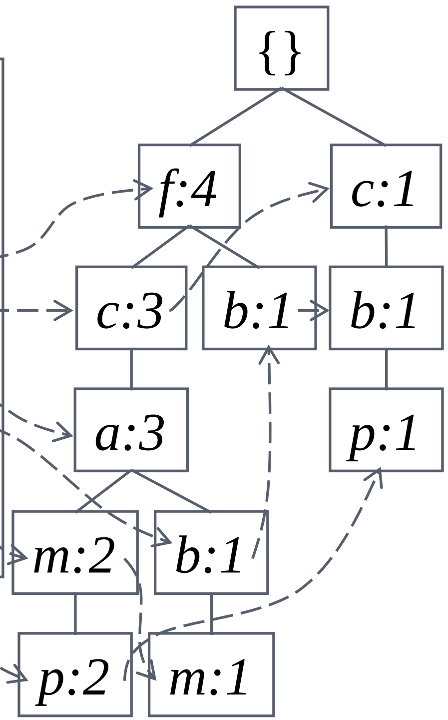
  Pattern f
- Completeness and non-redundance

4.2. FP-GROWTH EXAMPLE: 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

Header Table

Item	*frequency*	*head*
f	4	
c	4	
a	3	
b	3	
m	3	
p	3	



***Conditional* pattern bases**

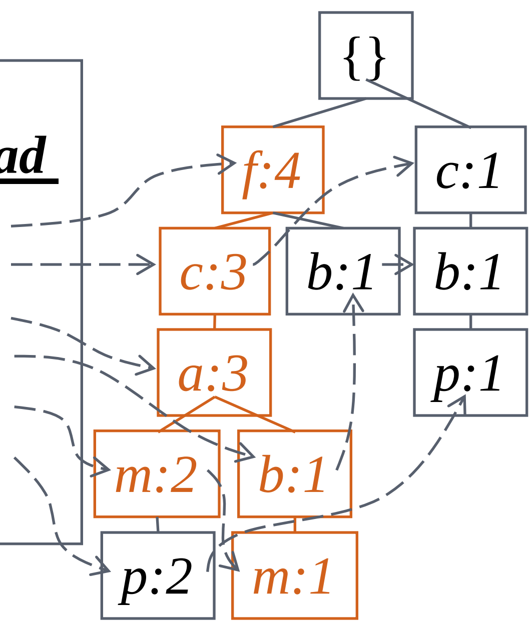
item	*cond. pattern base*
c	***f:3***
a	***fc:3***
b	***fca:1, f:1, c:1***
m	***fca:2, fcab:1***
p	***fcam:2, cb:1***

4.2. FP-Growth example: 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

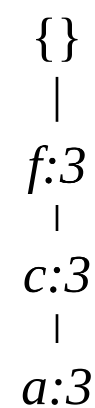
Header Table

Item	***frequency***	***head***
f	4	
c	4	
a	3	
b	3	
m	3	
p	3	



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

➔



***m-conditional* FP-tree**

➔

All frequent patterns relate to *m*

m,

fm, cm, am,

fcm, fam, cam,

fcam

-> associations

4.2. FP-Growth example: Recursion: Mining Each Conditional FP-tree

{}
|
f:3
|
c:3
|
a:3

m-conditional **FP-tree**

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

{}
|
f:3
|
c:3

am-conditional **FP-tree**

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

{}
|
f:3

cm-conditional **FP-tree**

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

{}
|
f:3

cam-conditional **FP-tree**

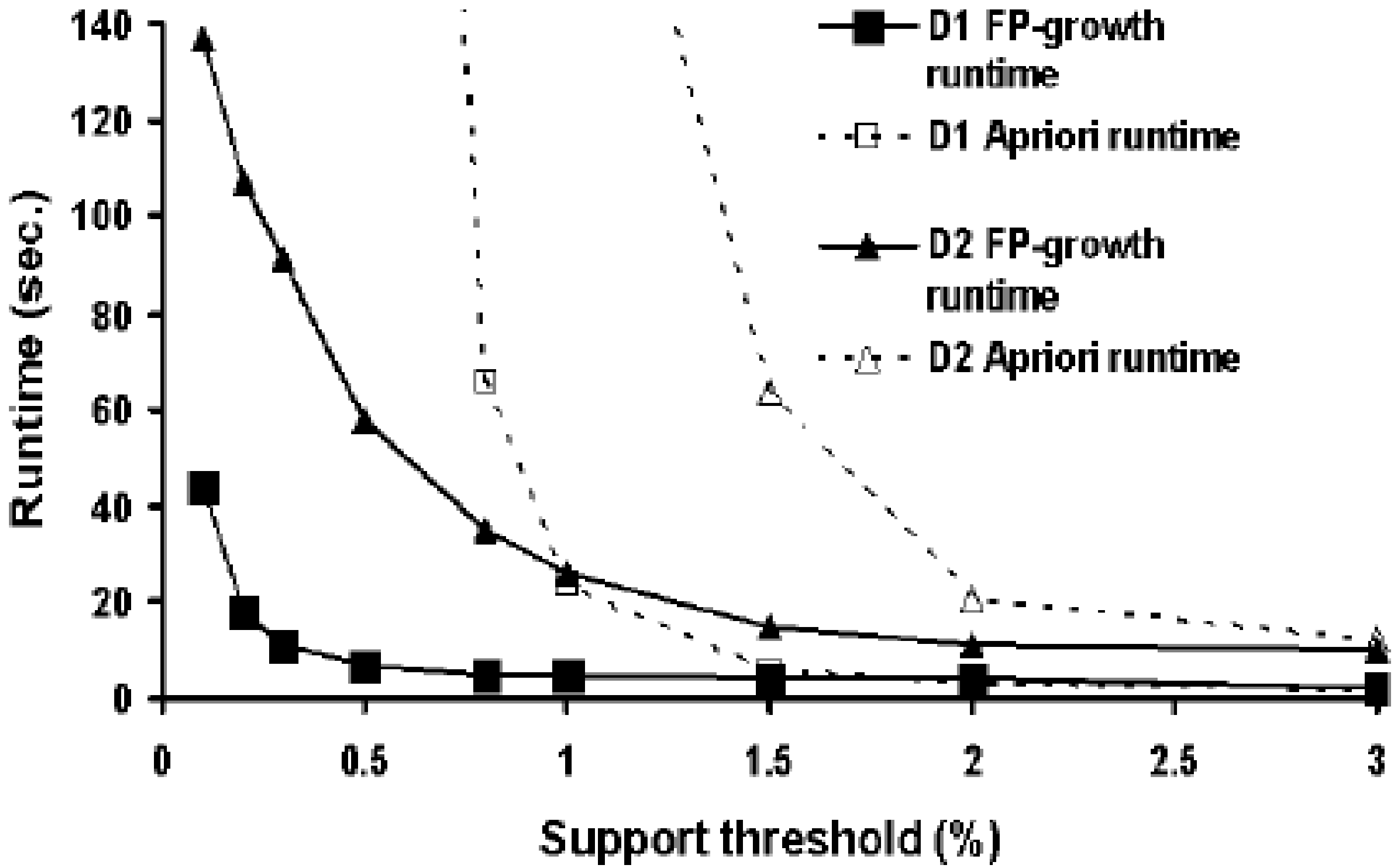
4.2. Mining frequent itemsets - FPGrowth

- FP-Growth's main characteristics
  - Does not create candidate itemsets
  - FP-tree is a compact data structure
  - Reduce the cost for scanning the original dataset
  - Main costs come from building FP-tree (at the initial) and mining it

  → Effective and scalable for mining long and sort frequent itemsets

4.2. Mining frequent itemsets - FPGrowth

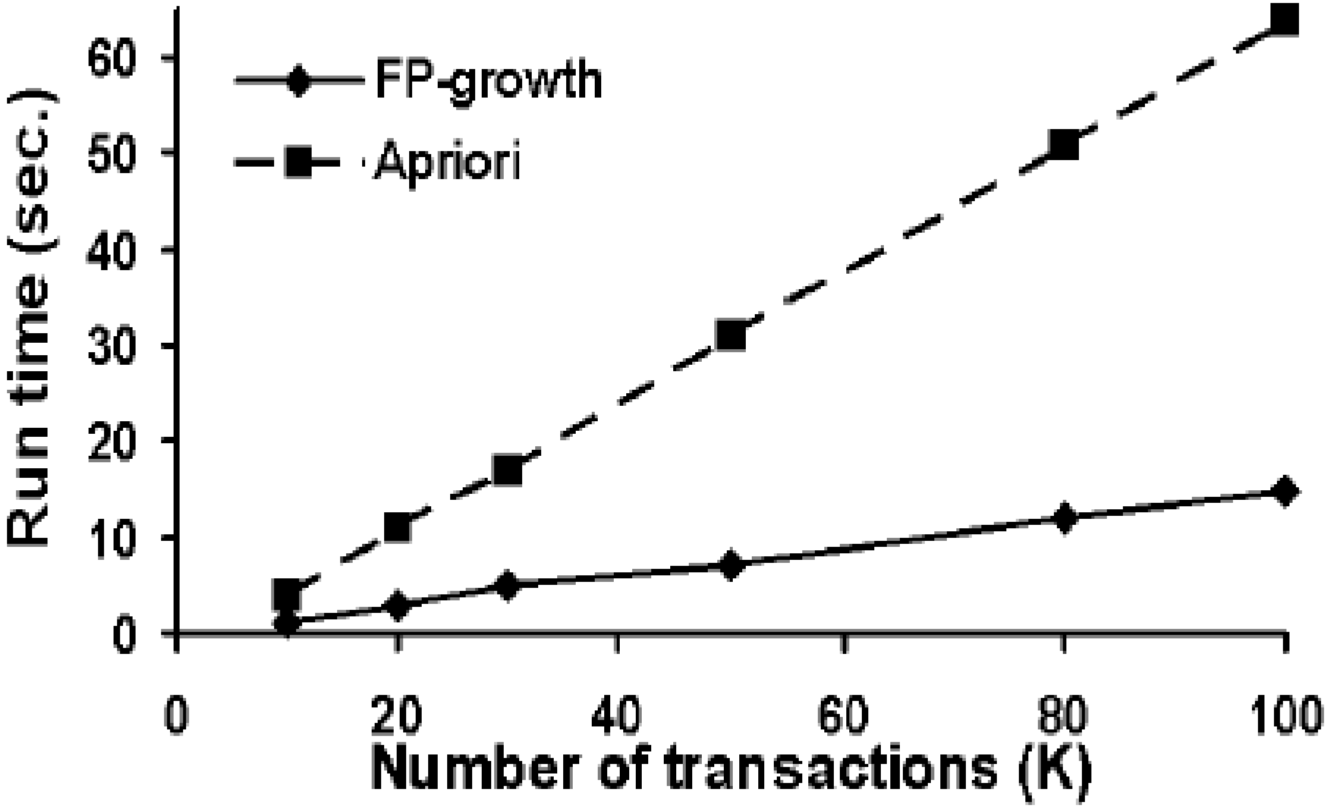
- Comparison between Apriori and FP-Growth



FP-growth scales well with support threshold

4.2. Mining frequent itemsets - FPGrowth

- Comparison between Apriori and FP-Growth



Linearly scales with number of transactions

4.3 Variance of frequent itemset mining

- Some variances of frequent itemsets
  - Frequent itemsets/subsequences/substructures
  - Closed frequent itemsets
  - Maximal frequent itemsets
  - Constrained frequent itemsets
  - Approximate frequent itemsets
  - Top-k frequent itemsets

5. Mining association rules from frequent itemsets

- Strong association rules $A \rightarrow B$
  - Support($A \rightarrow B$) = Support(A U B) >= min_sup
  - Confidence($A \rightarrow B$) = Support(A U B)/Support(A) = P(B|A) >= min_conf

→ Support($A \rightarrow B$) = Support_count(A U B) >= min_sup

→ Confidence($A \rightarrow B$) = P(B|A) = Support_count(AUB)/Support_count(A) >= min_conf

Dr. Tran Minh Quang – quangtran@hcmut.edu.vn

5. MINING ASSOCIATION RULES FROM FREQUENT ITEMSETS

- Process to generate strong association rules from frequent itemsets

  - For each frequent itemset *l*, create non-empty subsets of *l*.

$$Support\_count(l) >= min\_sup$$

  - For each non-empty subset *s* of *l*, create rule "*s* → (*l*-*s*)" if:

$$\boldsymbol{Support\_count(l)/Support\_count(s) >= min\_conf}$$

5. Mining association rules from frequent itemsets

Frequent Patterns Generated
{I2, I5: 2}, {I1, I5: 2}, {I2, I1, I5: 2}
{I2, I4: 2}
{I2, I3: 4}, {I1, I3: 4}, {I2, I1, I3: 2}
{I2, I1: 4}

$l = \{I1, I2, I5\}$

nonempty subsets of l are {I1, I2}, {I1, I5}, {I2, I5}, {I1}, {I2}, and {I5}

$I1 \wedge I2 \Rightarrow I5,\quad confidence = 2/4 = 50\%$
$I1 \wedge I5 \Rightarrow I2,\quad confidence = 2/2 = 100\%$
$I2 \wedge I5 \Rightarrow I1,\quad confidence = 2/2 = 100\%$
$I1 \Rightarrow I2 \wedge I5,\quad confidence = 2/6 = 33\%$
$I2 \Rightarrow I1 \wedge I5,\quad confidence = 2/7 = 29\%$
$I5 \Rightarrow I1 \wedge I2,\quad confidence = 2/2 = 100\%$

$\xrightarrow{\text{Min\_conf} = 50\%}$

$I1 \wedge I2 \Rightarrow I5$
$I1 \wedge I5 \Rightarrow I2$
$I2 \wedge I5 \Rightarrow I1$
$I5 \Rightarrow I1 \wedge I2$

6. MINING ASSOCIATION RULES BASED ON CONSTRAINTS

- Constraints
  - Instruct the processes of mining patterns and rules
  - Limit the search space of mining process
  - Some common constraints
    - Knowledge type constraints
    - Data constraints
    - Level/dimension constraints
    - Interestingness constraints
    - Rule constraints

6. MINING ASSOCIATION RULES BASED ON CONSTRAINTS

- Knowledge type constraints
  - What kind of knowledge we want to consider when mining rules

  Ex. Mining association rules or correlation rules
- Data constraints
  - The characteristics of the data to be mined
- Level/dimension constraints
  - What kind of dimensions/features or level of abstractions that we want to know when mining association rules
- Interestingness constraints: Definition of the measure, what is the threshold,...
- Rule constraints: The types of rules to be mined

6. MINING ASSOCIATION RULES BASED ON CONSTRAINTS

- The mining process becomes more effective and efficient
  - Rules are mined based on business needs/user requirements -> More effective
  - Optimizers can be utilized to improve the efficiency

7. CORRELATION ANALYSIS

- Strong association rules $A \Rightarrow B$ are mined based on:
  - Occurrence frequency of A and B (*min_sup*)
  - The conditional probability of B based on A (*min_conf*)
  - → *minsupport* and *minconfidence* are set in accordance with the subjective views of users
    - → A large amount of rules will be returned.
  - → Within 10,000 transactions, 6,000 ones for *computer games*, 7,500 for *videos*, and 4,000 for both *computer games* and *videos*
    - → Buys(X, "*computer games*") $\Rightarrow$ Buys (X, "*videos*") [support = 40%, confidence = 66%]

7. Correlation Analysis

- Correlation analysis for association rule A ⇒ B
  - Examine the correlation and dependency between A and B
  - Based on data statistics
  - Measures are objective, do not subjectively depend on users' views

→ Withing 10,000 transactions, 6,000 for *computer games*, 7,500 for *videos*, and 4,000 for both *computer games* and *videos*

→ Buys(X, "*computer games*") ⇒ Buys (X, "*videos*") [support = 40%, confidence = 66%]

→ P("*videos*") = 75% > 66%: "*computer games*" and "*videos*" are negatively correlated with each other.

Dr. Tran Minh Quang – quangtran@hcmut.edu.vn

7. Correlation analysis

- Correlation rules: A $\Rightarrow$ B [support, confidence, correlation]
  - correlation: measuring the correlation between A and B.
    - Common correlation measures: *lift*, χ^2 (Chi-square), *all_confidence*, *cosine*
      - *lift*: Validate the independent occurrence between A and B based on probability
      - χ^2 (Chi-square): Validate the independence between A and B based on expected value and the observed value.
      - *all_confidence*: Validate rules based on maximum support
      - *cosine*: similar to *lift* but it help to mitigate the dependency to the total number of transactions (i.e., the size of the dataset)

    → *all_confidence* and *cosine* are good for large datasets, do not dependd on transactions that do not contain any itemsets being validated (null-transactions)

Dr. Tran Minh Quang – quangtran@hcmut.edu.vn

7. CORRELATION ANALYSIS

- *lift*
  - *lift*(A, B) < 1: A and B are negatively correlated
  - *lift*(A, B) > 1: A and B are positively correlated
  - *lift*(A, B) = 1: A and B are independent

$$lift(A,B) = \frac{P(A \cup B)}{P(A)P(B)} = P(B \mid A) / P(B) = confidence(A => B) / support(B)$$

	$game$	$\overline{game}$	Σ_{row}
$video$	4,000	3,500	7,500
$\overline{video}$	2,000	500	2,500
Σ_{col}	6,000	4,000	10,000

$P(\{game\}) = 0.60$,

$P(\{video\}) = 0.75$

$P(\{game, video\}) = 0.40$.

$P(\{game, video\}) / (P(\{game\}) \times P(\{video\})) = 0.40/(0.60 \times 0.75) = 0.89$.

lift({game}=>{video}) = 0.89 < 1 → {game} and {video} are negatively correlated.

8. SUMMARY

- Association rule mining
  - Considered as one of the most important contributions from database communities in KDD
- Rule types: logical/quantitative rules, single dimension/multiple dimensions, single level/multiple levels of abstraction, association/correlation rules
- Forms of item/pattern: Frequent itemsets/subsequences/substructures, Closed frequent itemsets, Maximal frequent itemsets, Constrained frequent itemsets, Approximate frequent itemsets, Top-k frequent itemsets
- Mine frequent itemsets: Apriori and FP-Growth algorithms

REFERENCES

- **[1] Jiawei Han, Micheline Kamber, "Data Mining: Concepts and Techniques", Second Edition, Morgan Kaufmann Publishers, 2006.**
- **[2] David Hand, Heikki Mannila, Padhraic Smyth, "Principles of Data Mining", MIT Press, 2001.**
- **[3] David L. Olson, Dursun Delen, "Advanced Data Mining Techniques", Springer-Verlag, 2008.**
- **[4] Graham J. Williams, Simeon J. Simoff, "Data Mining: Theory, Methodology, Techniques, and Applications", Springer-Verlag, 2006.**
- **[5] ZhaoHui Tang, Jamie MacLennan, "Data Mining with SQL Server 2005", Wiley Publishing, 2005.**
- **[6] Oracle, "Data Mining Concepts", B28129-01, 2008.**
- [7] Oracle, "Data Mining Application Developer's Guide", B28131-01, 2008.
- [8] Ian H.Witten, Eibe Frank, "Data mining : practical machine learning tools and techniques", Second Edition, Elsevier Inc, 2005.
- [9] Florent Messeglia, Pascal Poncelet & Maguelonne Teisseire, "Successes and new directions in data mining", IGI Global, 2008.
- [10] Oded Maimon, Lior Rokach, "Data Mining and Knowledge Discovery Handbook", Second Edition, Springer Science + Business Media, LLC 2005, 2010.

Q&A

quangtran@hcmut.edu.vn