Chapter 5: Mining Frequent Patterns, Association and Correlations

Dong-Kyu Chae

PI of the Data Intelligence Lab @HYU
Department of Computer Science & Data Science
Hanyang University

Data
Intelligence
LAB

MaxMiner: Mining Max-patterns

- **Review: max pattern**
  - An itemset X is a max-pattern if X is frequent and there exists no **frequent** super-pattern Y ⊃ X
  - i.e., no such a Y
    - Y is a super-pattern of X
    - The support of Y is greater than **minSup (=frequent)**
      - The support of Y can be smaller than the support of X

Cf) Closed pattern : The super-pattern Y should be the same with the support of X 인경우
Y는 closed pattern. X는 Not closed pattern

- **MaxMiner is based on the Apriori algorithm**
  - It tries to find only max-patterns, not all the frequent patterns
  - Naïve approach: too much costly!

MaxMiner: Mining Max-patterns

- **1st scan: find frequent items and sort them (ascending order)**
  - Assume the order is A, B, C, D, E (E is most frequently occurring)
- **2nd scan: find support for 2-itemsets with potential max-patterns**
  - AB, AC, AD, AE, ABCDE
  - BC, BD, BE, BCDE (~A)
  - CD, CE, CDE (~A,~B)
  - DE

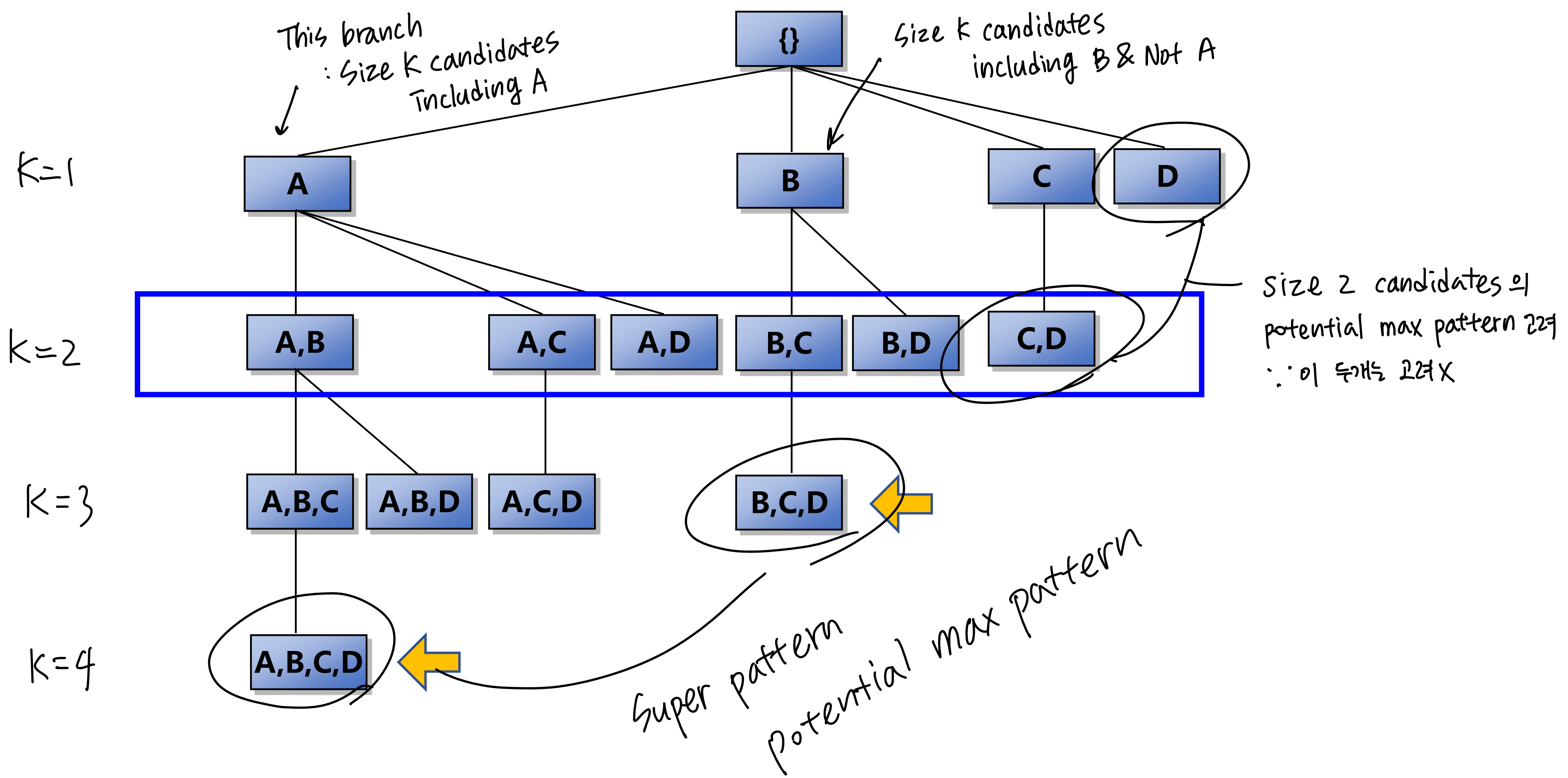
Potential max-patterns for each category of candidates

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

- **Reduce a lot of candidates in later stages**
  - If **BCDE** is a max-pattern, **no need** to check **BCD**, **BDE**, **CDE** in later scan

MaxMiner: Mining Max-patterns

- **Construct set-enumeration tree over four items**
  - Assume order is A B C D
  - It helps the process of **finding max pattern candidates**



Mining Closed Patterns: CLOSET

- **Review: closed pattern**
  - An itemset X is closed if X is *frequent* and there exists *no super-pattern* Y ⊃ X, ***with the same support*** as X
  - i.e., no such a Y
    - Y is a super-pattern of X
    - The support of Y is should **be the same** as that of X

- **CLOSET finds the closed patterns while running FP-growth**
  - The algorithm of FP-growth naturally supports finding closed patterns

Mining Closed Patterns: CLOSET

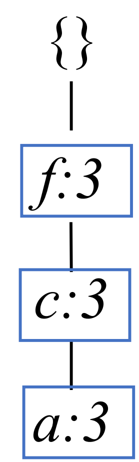
- **Naïve approach: again, too much costly!**
- **Find only the closed itemsets recursively during the mining process of FP-growth**
  - The all transactions having **m** also has **fca** => **fcam** is a frequent **closed** pattern!

Conditional **pattern bases**

item	*cond. pattern base*		*freq*
f	{}		4
c	*f:3*	cf: potential closed pattern	4
a	*fc:3*	afc	3
b	*fca:1, f:1, c:1*		3
m	*fca:2, fcab:1*		3
p	*fcam:2, cb:1*	cp	3

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

min_sup = 3



m-conditional **FP-tree**

All frequent patterns related to *m*

m,
fm, cm, am,
fcm, fam, cam,
fcam

CHARM: Mining by Exploring Vertical Data Format

a	T1, ..
b	
c	T1,T2,T3,...

row: itemsets,
columns: Transaction ID

- **Vertical format: $t(A) = \{T_{11}, T_{25}, \ldots\}$**
  - Represent each itemset as **a list of transaction IDs** containing the corresponding itemset

T1	a c e f k
T2	c f g k i
T3	c g z

- **Algorithm**
  - Transform a horizontally formatted data to a **vertical format** by scanning the dataset once, considering only frequent size-1 itemsets.
    - Easy: the number of items is much smaller than that of transactions
  - Starting with k=1, find (k+1) size frequent itemsets from frequent k-size itemsets
    - Using the **intersection operation** and **Apriori property**
  - Repeat this process until no frequent itemsets can be found in the next k+1 step

CHARM: Mining by Exploring Vertical Data Format

ITEMSET	TID SET
I1	{T101,T401,T501,T701,T801,T901}
I2	{T101,T201,T301,T401,T601,T801,T901}
I3	{T301,T501,T601,T701,T801,T901}
I4	{T201,T401}
I5	{T101,T801}

→ Transection ID 길이 = frequency of each itemset

Size 1

DB갈 필요없이
itemsets
intersection

Size 2

ITEMSET	TID SET
{I1,I2}	{T101,T401,T801,T901}
{I1,I3}	{T501,T701,T801,T901}
~~{I1,I4}~~	~~{T401}~~
{I1,I5}	{T301,T601,T801,T901}
{I2,I3}	{T201,T401}
{I2,I5}	{T101,T801}
~~{I3,I5}~~	~~{T801}~~

....

Min_sup = 2

CHARM: Mining by Exploring Vertical Data Format

- **Effectiveness:**
  - The **frequency** of each itemset is equal to **the length of its TID list**
  - The simple **intersection operation** brings us the k+1 length candidates

ITEMSET	TID SET
{I1,I2}	{T101,T401,T801,T901}
{I1,I3}	{T501,T701,T801,T901}
~~{I1,I4}~~	~~{T401}~~
{I1,I5}	{T301,T601,T801,T901}
{I2,I3}	{T201,T401}
{I2,I5}	{T101,T801}
~~{I3,I5}~~	~~{T801}~~

....

Size 2

Size 3

ITEMSET	TID SET
{I1,I2,I3}	{T801,T901}
{I1,I2,I5}	{T101,T901}

....

Min_sup = 2

Association Rules

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

Mining Multiple-Level Association Rules

- **Items often form hierarchies (assume that it is given)**
- **Flexible support settings will be needed**

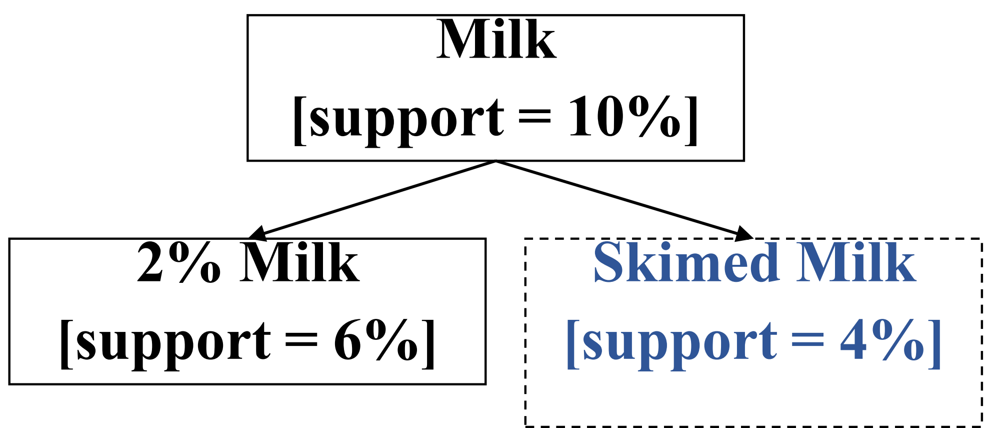
  minimum

  - Items at the **lower level** are expected to have **lower support**
  - Make the probability of lower-level items and higher-level items being included in frequent patterns as similar as possible

uniform setting

flexible setting

Level 1
min_sup = 5%



Level 1
min_sup = 5%

Level 2
min_sup = 5%

Level 2
min_sup = 3%

Data Intelligence LAB

Multi-level Association: Redundancy Filtering

- Some rules may be redundant due to "ancestor" relationships between items.
- Example
  - milk ⇒ wheat bread [support = 8%, confidence = 70%]

    비슷

  - ~~2% milk ⇒ wheat bread [support = 2%, confidence = 72%]~~
- We say the first rule is an ancestor of the second rule.
- A (descendent) rule is redundant if
  - Its support is close to the *"expected" value*, based on the ancestor rule's support value → 전체 milk 카테고리에서 2% milk의 비율로 support 예측가능 할때
  - Its confidence is close to that of ancestor rule

$\fallingdotseq 8\% \times \frac{25}{100} = 2\%$

Mining Multi-Dimensional Associations

- **Single-dimensional rules:**

  Customer   dimension

  buys(X, "milk") ⇒ buys(X, "bread"): milk ⇒ bread

  then, highly likely

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

    age(X,"19-25") ∧ occupation(X, "student") ⇒ buys(X, "coke")

  - Hybrid-dimension assoc. rules (*repeated dimensions*)

    age(X,"19-25") ∧ buys(X, "popcorn") ⇒ buys(X, "coke")

Attribute ≠ (Feature, = Dimension) Types

Attribute

Variable

- age(X," 19-25") ∧ occupation(X, "student") ⇒ buys(X, "coke")

Attribute values

❑ Categorical Attributes

- ❑ Finite number of possible values, no ordering among values

ex) occupation, buys

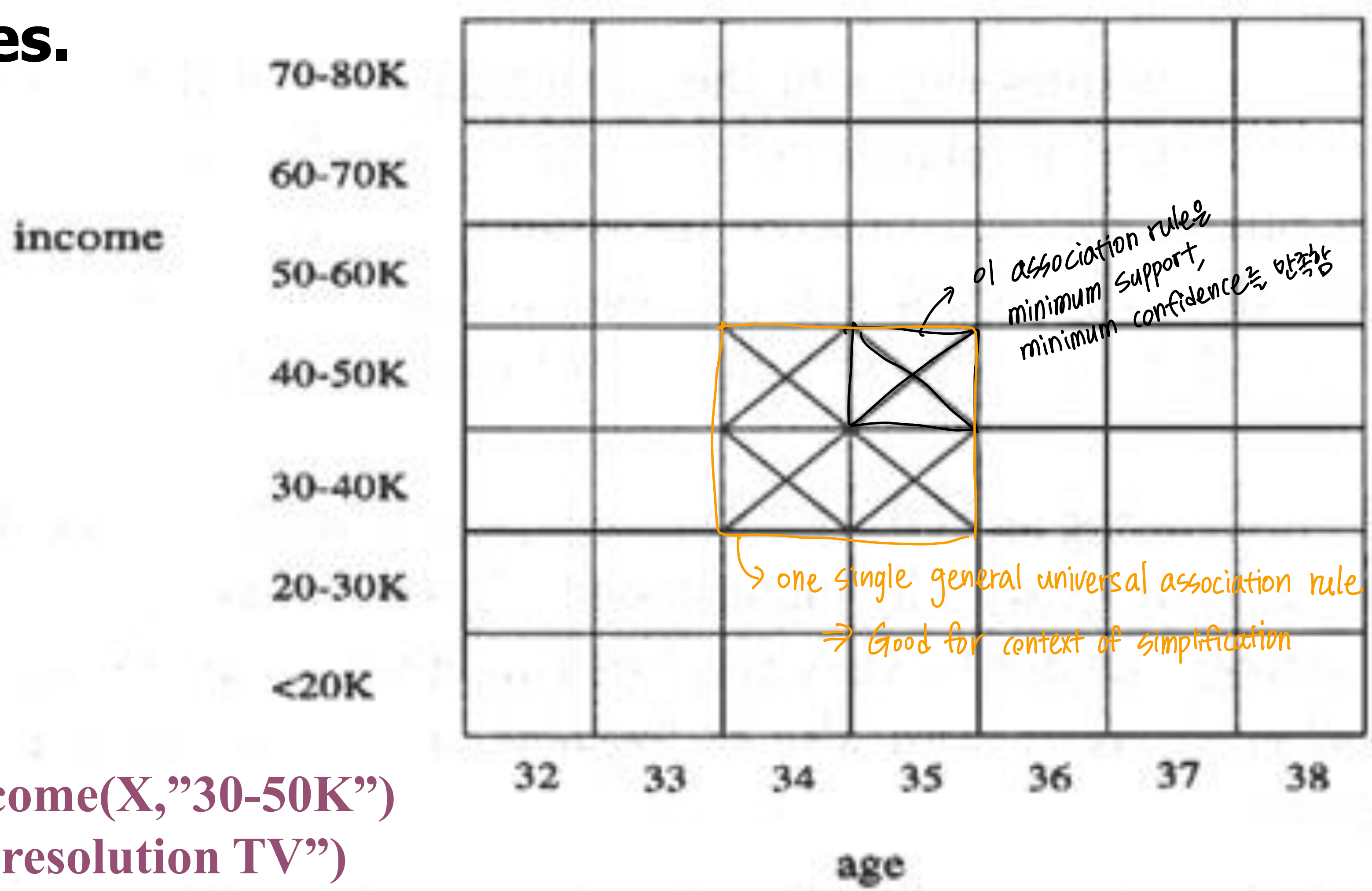
❑ Quantitative Attributes

- ❑ Numeric, implicit ordering among values
- ❑ Typically discretization is required (will be explained in the next several chapters)

ex) age

Quantitative Association Rules

- Numeric attributes are discretized
- Skeleton of our association rules: $\overbrace{A_{quan1}}^{\text{quantitative}} \land A_{quan2} \Rightarrow \overbrace{A_{cat}}^{\text{catagorical}}$
- If needed, we **cluster *adjacent* association** rules to form **general rules.**
- Example



age(X,"34-35") ∧ income(X,"30-50K")
⇒ buys(X,"high resolution TV")

From Association Mining to Correlation Analysis

- ***play basketball* ⇒ *eat cereal* [40%, 66.7%] is misleading!** ← Not meaningful!
  - Because, the overall % of students eating cereal is 75% (> 66.7%)
    : general trend
- ***play basketball* ⇒ *not eat cereal* [20%, 33.3%] is more meaningful,**
  common sense로 학생들이 시리얼을 많이 먹는다고 알고있는 상황에서 → 이 %는 높음 . 의미있음
  although it has lower support and confidence (limitation of the support & confidence framework)
- **Measure of interestingness of correlated events: lift**

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

lift = 1 : Event A,B는 completely independent
No correlation at all
lift < 1 : Two events have negative correlation
\> 1 : 〃 positive 〃

Contingency table

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

→ negative correlation compare to the general trend

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

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

Summary

- **Frequent pattern & association rule mining—an important task in data mining**
- **Scalable frequent pattern mining methods**
  - **Apriori** (the basic candidate generation & test approach)
  - **FP-growth** (without the candidate generation & test)
  - Minor improvements (**DIC**, **Partition**, **Sampling**, **MaxMiner**, **CHARM**, etc…)
- **Mining a variety of rules and interesting patterns**
  - Hierarchy and multi-attribute settings, Lift, etc…

Thank You