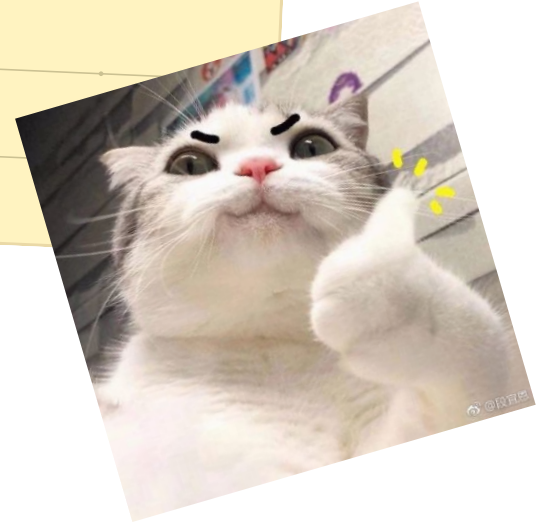
Chapter 6: Mining Frequent Patterns, Association and Correlations: Basic Concepts and Methods

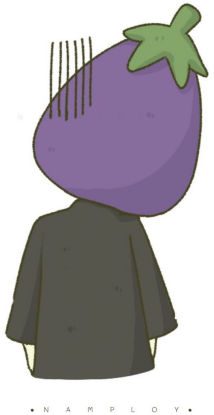
- Basic Concepts
- Efficient Pattern Mining Methods
- Pattern Evaluation
- Summary

Mining frequent patterns
↳ หรา patterns ที่มันซ่อนอยู่ใน Data



What Is Pattern Discovery?

- What are patterns?
  - Patterns: A set of items, subsequences, or substructures that occur frequently together (or strongly correlated) in a data set
  - Patterns represent intrinsic and important properties of datasets
- Pattern discovery: Uncovering patterns from massive data sets
- Motivation examples:
  - What products were often purchased together?
  - What are the subsequent purchases after buying an iPad?
  - What code segments likely contain copy-and-paste bugs?
  - What word sequences likely form phrases in this corpus?



Basic Concepts: k-Itemsets and Their Supports

- Itemset: A set of one or more items
- k-itemset: $X = \{x_1, \ldots, x_k\}$
  - Ex. {Beer, Nuts, Diaper} is a 3-itemset
- (*absolute*) *support* (*count*) of X, sup{X}: Frequency or the number of occurrences of an itemset X
  - Ex. sup{Beer} = 3
  - Ex. sup{Diaper} = 4
  - Ex. sup{Beer, Diaper} = 3
  - Ex. sup{Beer, Eggs} = 1
- (*relative*) *support*, *s*{*X*}: The fraction of transactions that contains X (i.e., the probability that a transaction contains X)
  - Ex. s{Beer} = 3/5 = 60%
  - Ex. s{Diaper} = 4/5 = 80%
  - Ex. s{Beer, Eggs} = 1/5 = 20%

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

Basic Concepts: Frequent Itemsets (Patterns)

- An itemset (or a pattern) X is *frequent* if the support of X is no less than a *minsup* threshold σ
- Let σ = *50%* (σ: *minsup* threshold)

For the given 5-transaction dataset

- All the frequent 1-itemsets:
  - Beer: 3/5 (60%); Nuts: 3/5 (60%)
  - Diaper: 4/5 (80%); Eggs: 3/5 (60%)
- All the frequent 2-itemsets:
  - {Beer, Diaper}: 3/5 (60%)
- All the frequent 3-itemsets?
  - None

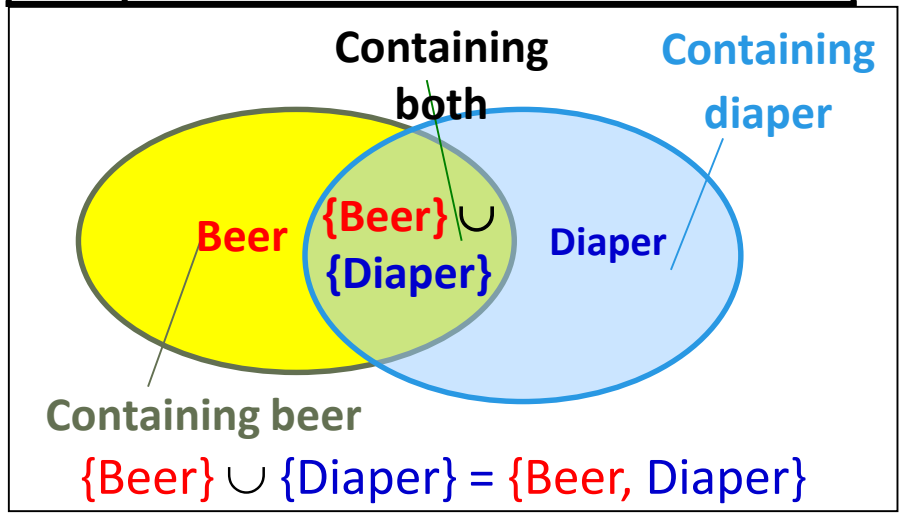
Tid	Items bought
10	Beer, Nuts, Diaper
20	Beer, Coffee, Diaper
30	Beer, Diaper, Eggs
40	Nuts, Eggs, Milk
50	Nuts, Coffee, Diaper, Eggs, Milk

- Why do these itemsets (shown on the left) form the complete set of frequent *k*-itemsets (patterns) for any *k*?
- **Observation**: We may need an efficient method to mine a complete set of frequent patterns

From Frequent Itemsets to Association Rules

- Comparing with itemsets, rules can be more telling
  - Ex. *Diaper* → *Beer* → คนซื้อ Diaper แล้วไปสุ่มๆไปซื้อ Beer
    - *Buying diapers may likely lead to buying beers*
- How strong is this rule? (support, confidence) ผ่านสูตรกฎขั้นต่ำแต่สูงมากๆไม่มีอะไรให้ดูเลยก็ได้
  - Measuring association rules: $X \rightarrow Y$ (s, c)
    - Both *X* and *Y* are itemsets
  - Support, *s*: The probability that a transaction contains X ∪ Y เริ่มจาก หา Support ของ x และ y
    - Ex. s{Diaper, Beer} = 3/5 = 0.6 (i.e., 60%) หา sup ได้ 0.6
  - Confidence, *c: The conditional probability* that a transaction containing X also contains *Y*
    - Calculation: $c = \sup(X \cup Y) / \sup(X)$ เอา sup ของ x อันหารด้วย sup ตัวหน้า
    - Ex. c = sup{Diaper, Beer}/sup{Diaper} = 3/4 = 0.75 (Diaper/Beer, Diaper ทั้งหมด)

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



Note: X ∪ Y: the union of two itemsets
- The set contains both X and Y

Mining Frequent Itemsets and Association Rules

- **Association rule mining**
  - Given two thresholds: *minsup, minconf*
  - Find all of the rules, $X \rightarrow Y$ (s, c)
    - such that, s ≥ *minsup* and $c \geq minconf$

- Let *minsup = 50%* (minsupport) → เป็นไอเท็มเซ็ตที่มีอยู่เป็นส่วนใหญ่ของการทำแร็กชั่น
  - Freq. 1-itemsets: Beer: 3, Nuts: 3, Diaper: 4, Eggs: 3
  - Freq. 2-itemsets: {Beer, Diaper}: 3

- Let *minconf = 50%* (เกิดขึ้นซ้ำๆ กันมากแค่ไหน) sup(Beer / Diaper) / sup(Beer)
  - *Beer → Diaper* (60%, 100%) (sup, conf) → เอา Diaper ไปวางชั้นขาย Beer
  - *Diaper → Beer* (60%, 75%)

(กฎ 2 ตัว)

(Q: Are these all rules?)

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

- **Observations:**
  - Mining association rules and mining frequent patterns are very close problems
  - Scalable methods are needed for mining large datasets

Efficient Pattern Mining Methods

- The Downward Closure Property of Frequent Patterns
- The Apriori Algorithm
- Extensions or Improvements of Apriori
- Mining Frequent Patterns by Exploring Vertical Data Format
- FPGrowth: A Frequent Pattern-Growth Approach
- Mining Closed Patterns

Apriori Pruning and Scalable Mining Methods

ตัดแต่ง

ถ้าเกิดว่าไอเท็มเซ็ตที่ต่ำกว่าไม่ผ่าน minsup ไอเท็มเซ็ตที่สูงกว่าก็ไม่มีทางผ่าน Minsup (ขี้เกียจคูณ)

- Apriori pruning principle: If there is any itemset which is infrequent, its superset should not even be generated! (Agrawal & Srikant @VLDB'94, Mannila, et al. @ KDD' 94)
- Scalable mining Methods: Three major approaches
  - Level-wise, join-based approach: Apriori (Agrawal & Srikant@VLDB'94)
  - Vertical data format approach: Eclat (Zaki, Parthasarathy, Ogihara, Li @KDD'97)
  - Frequent pattern projection and growth: FPgrowth (Han, Pei, Yin @SIGMOD'00)

Apriori: A Candidate Generation & Test Approach

การเขียนให้เป็นขั้นตอน

- Outline of Apriori (level-wise, candidate generation and test)
  - Initially, scan DB once to get frequent 1-itemset → สแกนเริ่มจากดู 1 ไอเท็มเซ็ทก่อน
  - Repeat
    - Generate length-(k+1) candidate itemsets from length-k frequent itemsets
    - Test the candidates against DB to find frequent (k+1)-itemsets
    - Set k := k +1
  - Until no frequent or candidate set can be generated
  - Return all the frequent itemsets derived

The Apriori Algorithm—An Example

minsup = 2

Database TDB

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

C_1 → 1st scan → F_1

Itemset	sup
{A}	2
{B}	3
{C}	3
~~{D}~~	~~1~~
{E}	3

F_1

Itemset	sup
{A}	2
{B}	3
{C}	3
{E}	3

C_2 (candidate 2-itemsets)

Itemset
{A, B}
{A, C}
{A, E}
{B, C}
{B, E}
{C, E}

2nd scan → C_2

Itemset	sup
~~{A, B}~~	~~1~~
{A, C}	2
~~{A, E}~~	~~1~~
{B, C}	2
{B, E}	3
{C, E}	2

F_2

Itemset	sup
{A, C}	2
{B, C}	2
{B, E}	3
{C, E}	2

C_3 → 3rd scan → F_3

Itemset
{B, C, E}

Itemset	sup
{B, C, E}	2