CS 412 Intro. to Data Mining

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

Jiawei Han, Computer Science, Univ. Illinois at Urbana-Champaign, 2017

การทำ mining เพื่อหา Patterns ที่เกิดขึ้นซ้ำๆ

What Is Pattern Discovery?

- What are patterns? การค้นหา patterns ที่ซ่อนอยู่
  - 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: เอาไปใช้ประโยชน์อะไรได้บ้าง?
  - What products were often purchased together? สินค้าใดที่ลูกค้ามักจะซื้อคู่กันเสมอ? เพราะถ้าร้านรู้ ร้านจะได้เตรียมสินค้าสต็อกไว้เท่าๆกัน
  - What are the subsequent purchases after buying an iPad? เมื่อลูกค้าซื้อ ipad ไปแล้ว จะกลับมาซื้อปากกา/เคส เป็นต้น ครั้งหน้าร้านอาจจัดโปรโมชั่นให้ลูกค้า
  - What code segments likely contain copy-and-paste bugs?
  - What word sequences likely form phrases in this corpus?

ลูกค้ามักซื้ออะไรคู่กันเสมอ (set of items)

ถ้าเราพิมพ์ค้นหาแบบยังไม่เต็มคำ เช่น  มันก็จะขึ้นคำที่เราต้องการค้นหามา ซึ่งเรียนรู้จาก patterns ที่คนพิมพ์

Pattern Discovery: Why Is It Important?

- Finding inherent regularities in a data set
- Foundation for many essential data mining tasks
  - Association, correlation, and causality analysis
  - Mining sequential, structural (e.g., sub-graph) patterns
  - Pattern analysis in spatiotemporal, multimedia, time-series, and stream data
  - Classification: Discriminative pattern-based analysis
  - Cluster analysis: Pattern-based subspace clustering
- Broad applications
  - Market basket analysis, cross-marketing, catalog design, sale campaign analysis, Web log analysis, biological sequence analysis

ตาราง, รูปภาพ, วิดีโอ, แผนที่

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

เราสามารถเติมเลขต่อท้าย k ได้

- (*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

จำนวน transaction ที่มาสนับสนุน

Absolute ไม่ดี เพราะไม่รู้ transaction ทั้งหมด

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

มักใช้ relative แทน absolute

- (*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%

จุดมุ่งหมายคือ เราต้องทำมือได้

เข้าใจการทำงานของ Data mining

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

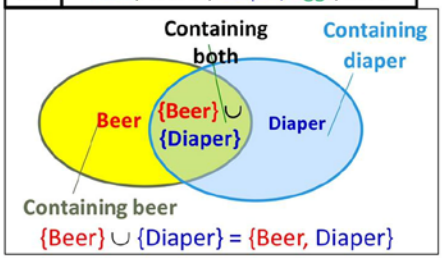
Tid	Items bought
10	Beer, Nuts, Diaper
20	Beer, Coffee, Diaper
30	Beer, Diaper, Eggs
40	Nuts, Eggs, Milk
50	Nuts, Coffee, Diaper, Eggs, Milk

- 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*
    - *Buying diapers may likely lead to buying beers*
- How strong is this rule? (support, confidence)
  - Measuring association rules: *X → Y* (s, c)
    - Both *X* and *Y* are itemsets
  - Support, *s*: The probability that a transaction contains X ∪ Y
    - Ex. s{Diaper, Beer} = 3/5 = 0.6 (i.e., 60%)
  - Confidence, *c*: *The conditional probability* that a transaction containing X also contains *Y*
    - Calculation: *c* = sup(X ∪ Y) / sup(X)
    - Ex. *c* = sup{Diaper, Beer}/sup{Diaper} = ¾ = 0.75

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** ก่อนจะหา ต้องกำหนด minsup,minconf
  - 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%*
  - Freq. 1-itemsets: Beer: 3, Nuts: 3, Diaper: 4, Eggs: 3
  - Freq. 2-itemsets: {Beer, Diaper}: 3
- Let *minconf = 50%*
  - *Beer* → *Diaper* (60%, 100%)
  - *Diaper* → *Beer* (60%, 75%)

อาจเลือกจากตัวที่มี support และ confident สูงสุด

สูตร

$c = \sup(X \cup Y) / \sup(X)$

(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

10

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

15

Apriori Pruning and Scalable Mining Methods

หลักการสำคัญ

- 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)

The Apriori Algorithm (Pseudo-Code)

C_k: Candidate itemset of size k

F_k : Frequent itemset of size k

ซูโดโค้ด = เป็นโค้ดโปรแกรม แต่ไม่ได้เป็นภาษาใดภาษาหนึ่ง เขียนให้เราเข้าใจ แปลงเป็นภาษาที่เราใช้ได้

```
K := 1;
F_k := {frequent items};  // frequent 1-itemset
While (F_k != ∅) do {     // when F_k is non-empty
   C_{k+1} := candidates generated from F_k;  // candidate generation
   Derive F_{k+1} by counting candidates in C_{k+1} with respect to TDB at minsup;
   k := k + 1
   }
return ∪_k F_k            // return F_k generated at each level
```

The Apriori Algorithm—An Example