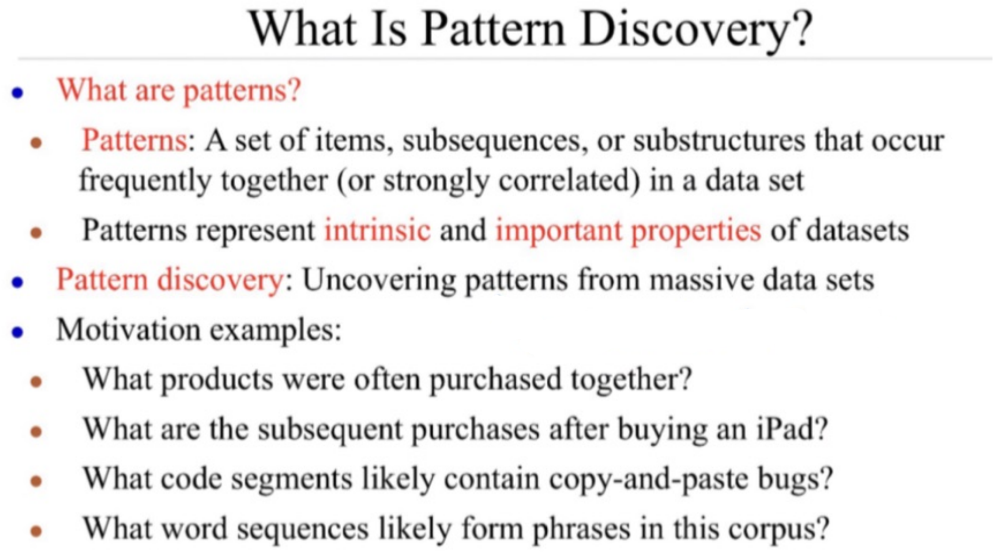
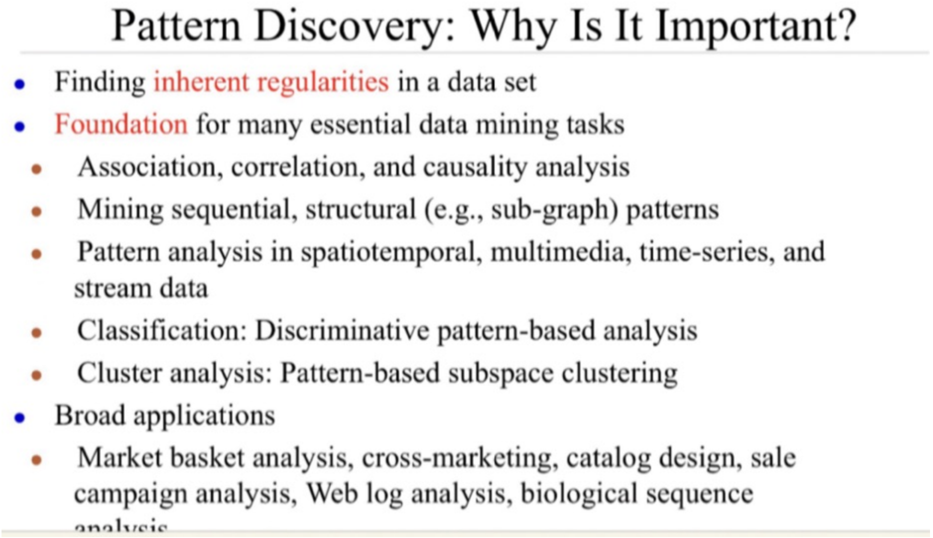


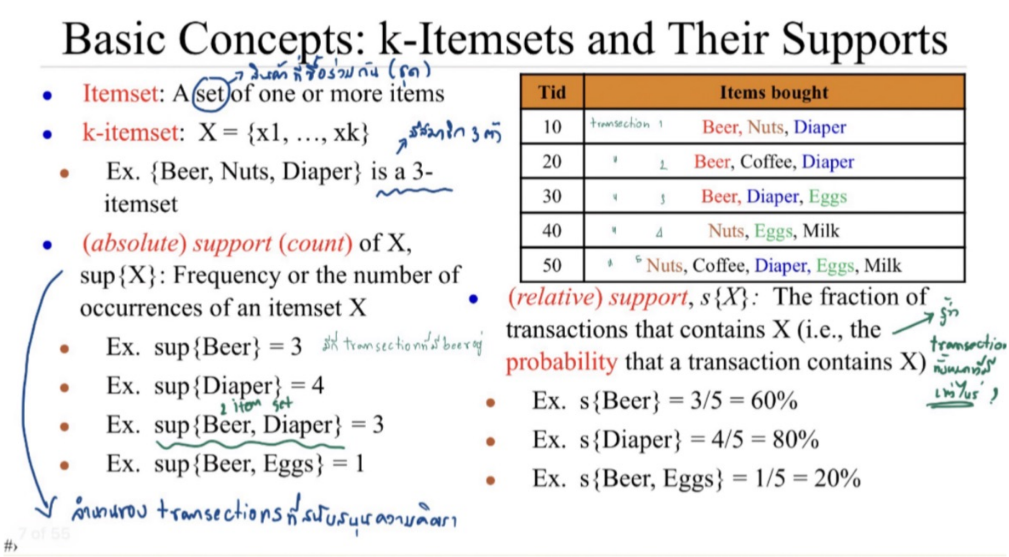
What Is Pattern Discovery?

- 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?



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



Basic Concepts: k-Itemsets and Their Supports

- Itemset: A set of one or more items → สินค้าที่ซื้อร่วมกัน (ชุด)
- k-itemset: X = {x1, …, xk} ↗ มีสินค้า 3 ตัว
  - 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 มีที่ transectionที่มีbeerอยู่
  - Ex. sup{Diaper} = 4
  - Ex. sup{Beer, Diaper} = 3 (2 item set)
  - Ex. sup{Beer, Eggs} = 1

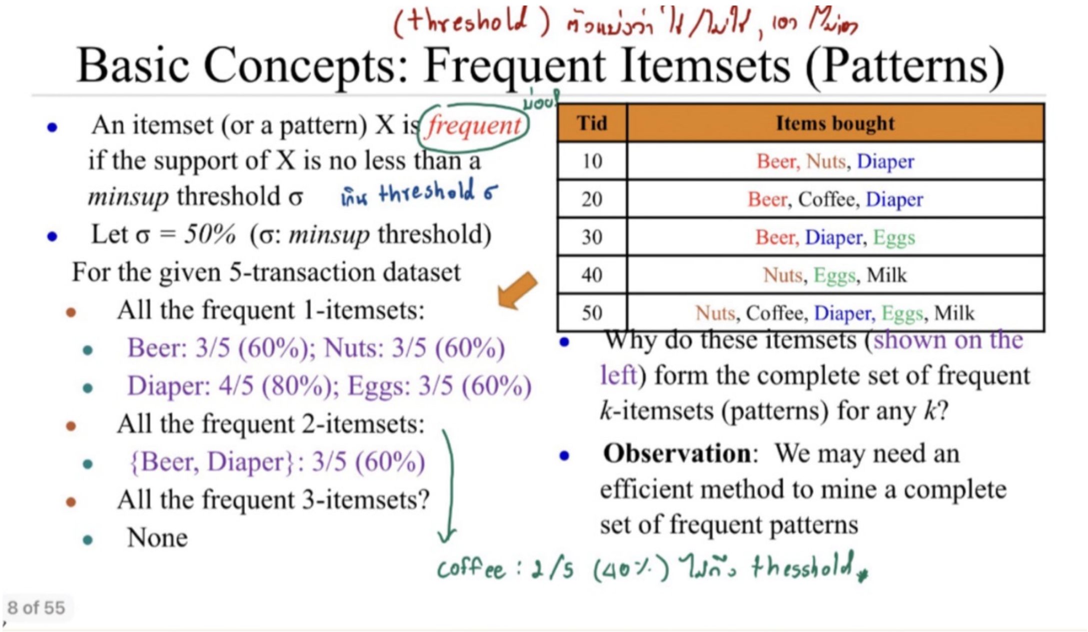
จำนวนของ transections ที่สนับสนุนความคิดเรา

Tid	Items bought
10	transection 1 Beer, Nuts, Diaper
20	" 2 Beer, Coffee, Diaper
30	" 3 Beer, Diaper, Eggs
40	" 4 Nuts, Eggs, Milk
50	" 5 Nuts, Coffee, Diaper, Eggs, Milk

- *(relative) support*, *s{X}*: The fraction of transactions that contains X (i.e., the probability that a transaction contains X) → รู้ว่า transaction ทั้งหมดที่มี เท่าไหร่ ?
  - Ex. s{Beer} = 3/5 = 60%
  - Ex. s{Diaper} = 4/5 = 80%
  - Ex. s{Beer, Eggs} = 1/5 = 20%

7 of 55

#›



(threshold) ตัวแบ่งว่าใช่/ไม่ใช่, เอา/ไม่เอา

Basic Concepts: Frequent Itemsets (Patterns)

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

For the given 5-transaction dataset

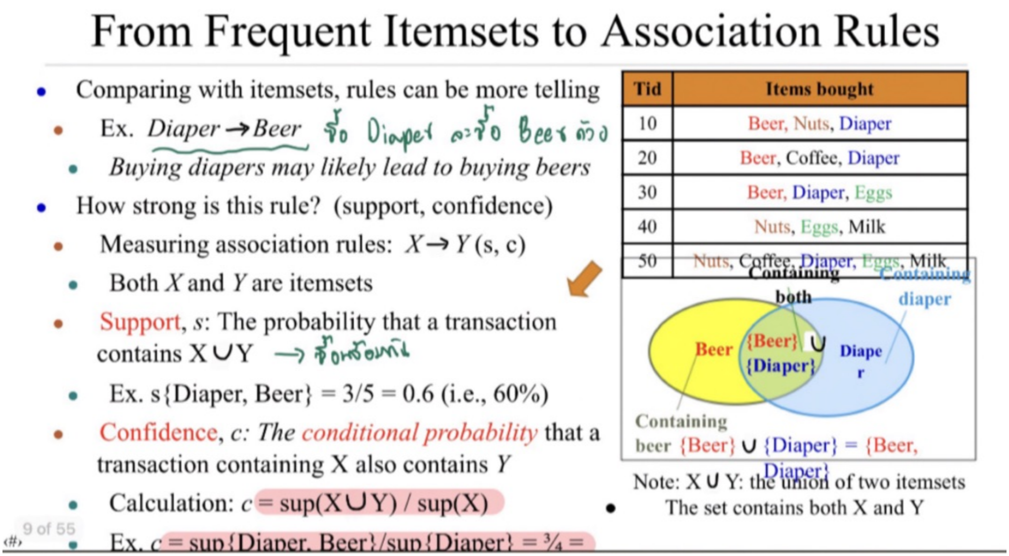
- All the frequent 1-itemsets:
  - Beer: 3/5 (60%); Nuts: 3/5 (60%)
  - Diaper: 4/5 (80%); Eggs: 3/5 (60%)
- All the frequent 2-itemsets:
  - {Beer, Diaper}: 3/5 (60%)
- All the frequent 3-itemsets?
  - None

coffee : 2/5 (40%) ไม่ถึง thesshold*

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

8 of 55



From Frequent Itemsets to Association Rules

- Comparing with itemsets, rules can be more telling
  - Ex. *Diaper → Beer* ซื้อ Diaper และซื้อ Beer ด้วย
    - *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 → ซื้อพร้อมกัน
    - 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 =

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

Containing both

Containing diaper

Beer

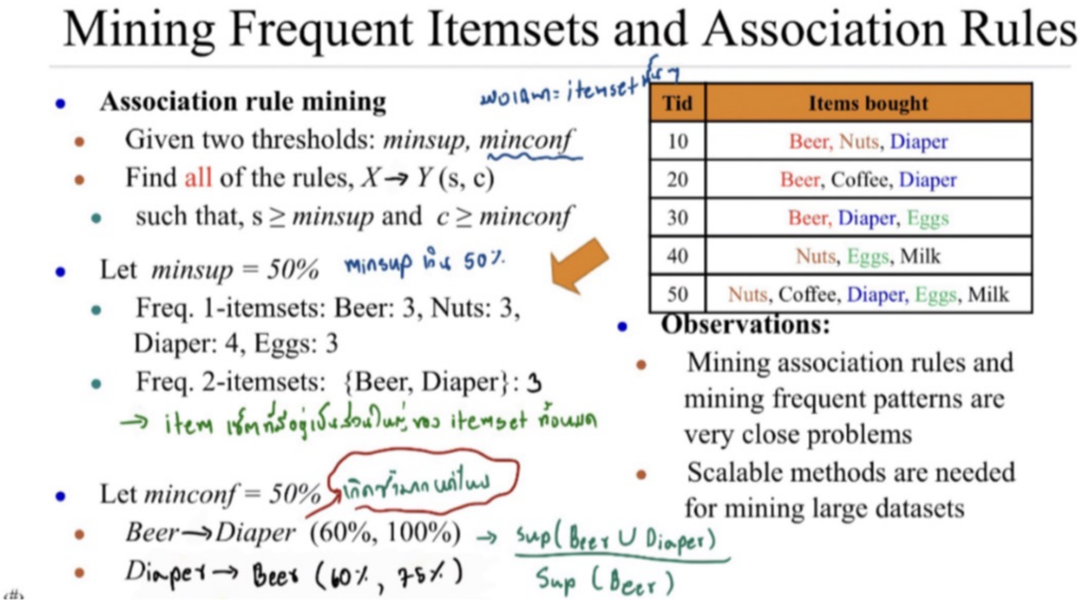
{Beer} ∪ {Diaper}

Diape r

Containing beer {Beer} ∪ {Diaper} = {Beer, Diaper}

Note: X ∪ Y: the union of two itemsets

- The set contains both X and Y



Mining Frequent Itemsets and Association Rules

- **Association rule mining** พอเหมาะ: itemset ที่ดีๆ
  - Given two thresholds: *minsup, minconf*
  - Find all of the rules, *X* → *Y* (s, c)
    - such that, s ≥ *minsup* and *c* ≥ *minconf*
- Let *minsup = 50%* minsup เกิน 50%
  - Freq. 1-itemsets: Beer: 3, Nuts: 3, Diaper: 4, Eggs: 3
  - Freq. 2-itemsets: {Beer, Diaper}: 3

→ item เซ็ตที่อยู่เป็นส่วนใหญ่ ของ itemset ทั้งหมด

- Let *minconf = 50%* เกิดซ้ำมากแค่ไหน
  - *Beer → Diaper* (60%, 100%) → $\frac{\text{sup(Beer} \cup \text{Diaper)}}{\text{Sup (Beer)}}$
  - Diaper → Beer (60%, 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

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

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

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

Efficient Pattern Mining Methods

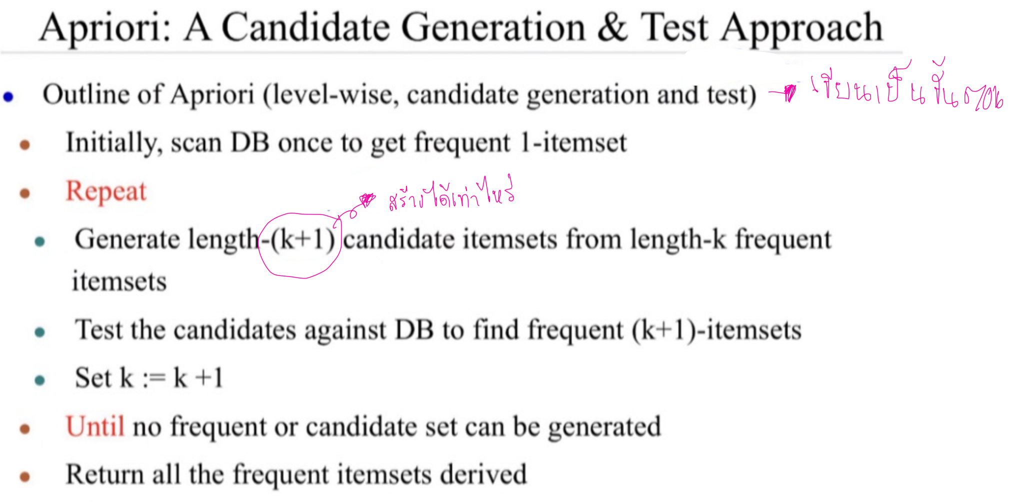
- The Downward Closure Property of Frequent Patterns
- The Apriori Algorithm
- Extensions or Improvements of Apriori
- Mining Frequent Patterns by Exploring Vertical Data Format
- FPGrowth: A Frequent Pattern-Growth Approach
- Mining Closed Patterns

Apriori Pruning and Scalable Mining Methods

ตัดแต่ง

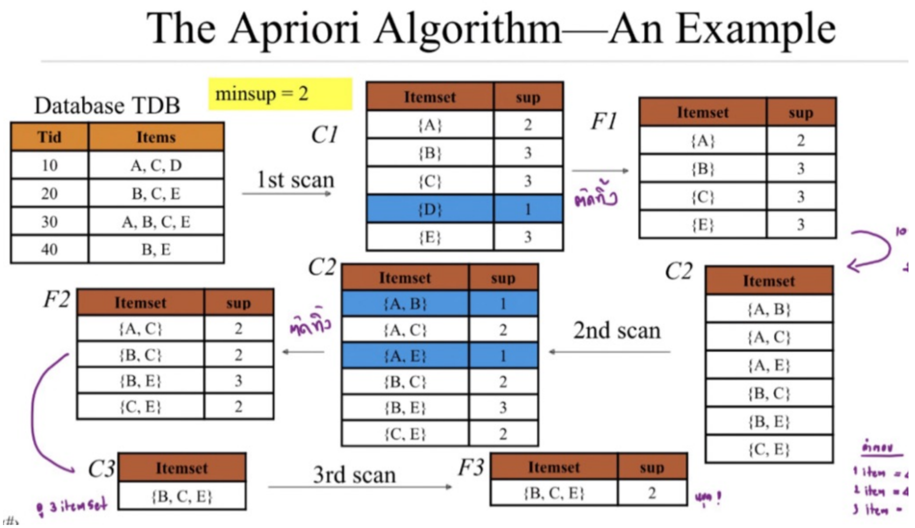
- Apriori pruning principle: If there is any itemset which is infrequent, its superset should not even be generated! (Agrawal & Srikant @VLDB'94, Mannila, et al. @ KDD' 94)
- Scalable mining Methods: Three major approaches
  - Level-wise, join-based approach: Apriori (Agrawal & Srikant@VLDB'94)
  - Vertical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

ถ้าเกิด item set
ที่เล็กกว่าไม่ผ่าน
จะตัดทิ้ง



Apriori: A Candidate Generation & Test Approach

- Outline of Apriori (level-wise, candidate generation and test) → เขียนเป็นขั้นตอน
  - Initially, scan DB once to get frequent 1-itemset
  - Repeat
    - Generate length-(k+1) candidate itemsets from length-k frequent itemsets → สร้างได้เท่าไหร่
    - 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

1st scan

C1

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

ตัดทิ้ง

F1

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

C2

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

2nd scan

C2

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

ตัดทิ้ง

F2

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

C3

Itemset
{B, C, E}

3 itemset

3rd scan

F3

Itemset	sup
{B, C, E}	2

หยุด !

ตัดออก

1 item = 4

2 item = 4

3 item =