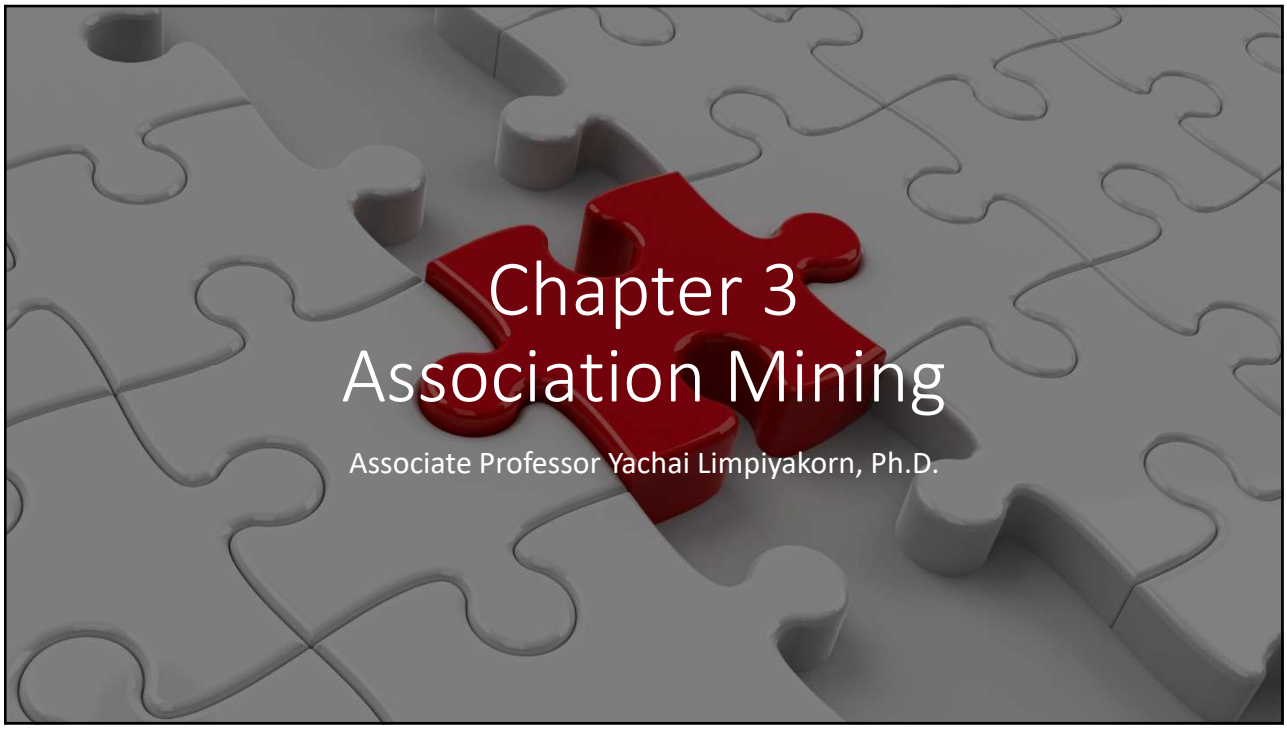


Chapter 3
Association Mining

Associate Professor Yachai Limpiyakorn, Ph.D.

Basic Concepts

TID	Produce
1	MILK, BREAD, EGGS
2	BREAD, SUGAR
3	BREAD, CEREAL
4	MILK, BREAD, SUGAR
5	MILK, CEREAL
6	BREAD, CEREAL
7	MILK, CEREAL
8	MILK, BREAD, CEREAL, EGGS
9	MILK, BREAD, CEREAL

- Given:
  - (1) database of transactions/ transactional database
  - (2) each transaction is a list of items purchased
- Find:

  ความสัมพันธ์ที่น่าสนใจระหว่างไอเทมเซต (itemset) ในชุดข้อมูล ความสัมพันธ์ที่ได้เขียนอยู่ในรูปกฎความสัมพันธ์ (Association Rule) ของเซตของไอเทมที่เป็นเหตุ (Antecedent) ไปสู่เซตของไอเทมที่เป็นผล (Consequent)

  {Cheese, Milk} → Bread [S=5%, C=80%]

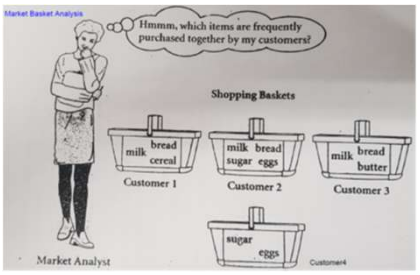
  80% of customers who buy cheese and milk also buy bread and 5% of customers buy all these products together

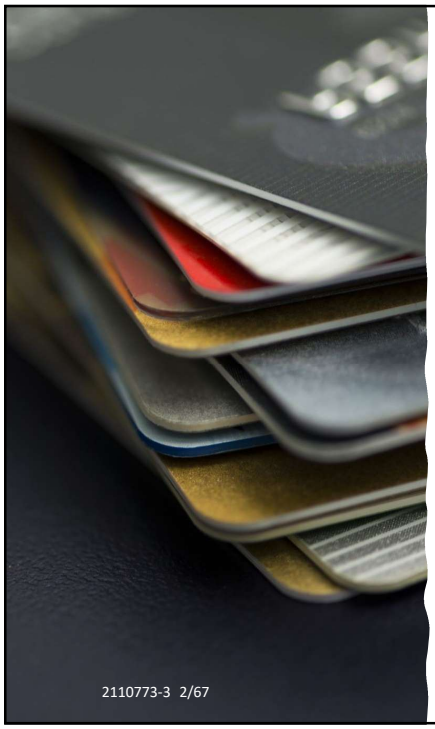
How can association rules be used?

Application 1

ส่วนใหญ่มักประยุกต์ใช้เทคนิคการทำเหมืองความสัมพันธ์กับ**การวิเคราะห์ทางการตลาด (Market Basket Analysis: MBA)** ซึ่งเป็นรูปแบบการจัดกลุ่ม (Clustering) แบบหนึ่ง ที่ใช้เพื่อหากลุ่มสิ่งของที่น่าจะปรากฏร่วมกันในทรานแซกชันหนึ่งๆ มักเป็นทรานแซกชัน ณ จุดขาย (point-of-sale) ผลลัพธ์หรือแบบจำลองที่ได้สามารถแสดงได้ด้วยกฎซึ่งบอกความเป็นไปได้ของการซื้อผลิตภัณฑ์ต่างๆร่วมกัน การวิเคราะห์ทางการตลาดมีบทบาทสำคัญต่ออุตสาหกรรมการค้าปลีก (Retail industry) เพื่อให้ทราบถึงพฤติกรรมการซื้อสินค้าของลูกค้าซึ่งเป็นประโยชน์ในการ

- จัดพื้นที่ร้านค้า (Store layout)
- ทำตลาดเพื่อส่งเสริมการขายสินค้าหรือบริการซึ่งกันและกัน (Cross-marketing)
- ออกแบบหนังสือแคตตาลอกสินค้า (Catalog design)
- วางแผนการส่งเสริมการขายและการตั้งราคาผลิตภัณฑ์ (Product pricing and promotion)





Application 2

นอกจากนี้ สามารถประยุกต์ใช้การวิเคราะห์ทางการตลาดกับกิจกรรมใกล้เคียงที่ลูกค้ามักกระทำด้วยกัน เพื่อก่อให้เกิดรายได้สูงสุดจากการจัดประเภทผลิตภัณฑ์หรือบริการเข้าด้วยกัน ได้แก่

- การใช้จ่ายผ่านบัตรเครดิตของลูกค้าในการเข้าพักโรงแรม เช่ารถ ทำให้สามารถทำนายค่าใช้จ่ายต่อไปของลูกค้า
- แพ็กเกจการให้บริการการสื่อสารโทรคมนาคม เพื่อก่อให้เกิดรายได้สูงสุด
- การให้บริการทางธนาคารที่ลูกค้ามักซื้อด้วยกัน เพื่อก่อให้เกิดประโยชน์สูงสุด เช่น ประเภทบัญชีที่ลูกค้ามักเปิดด้วยกัน (account bundle) การให้บริการการลงทุนครบวงจร และแพ็กเกจสินเชื่อการซื้อรถ เป็นต้น

นิยามพื้นฐานการทำเหมืองความสัมพันธ์

- **ไอเทมเซต** (itemset - I) คือเซตที่มีไอเทมทั้งหมดเป็นสมาชิก ซึ่งไอเทมในที่นี้อาจเป็นชื่อสินค้า หรือชื่อใดๆ ที่เป็นหน่วยพื้นฐานที่จะนำมาทำการเรียนรู้
- **ทรานแซกชัน** (transaction - T) เป็นเซตของไอเทม โดยที่ $T \subseteq I$
- **เซตข้อมูล** (data set - D) คือเซตที่มีทรานแซกชันทุกตัวเป็นสมาชิก

เรากล่าวว่าทรานแซกชัน T บรรจุเซตย่อยของไอเทม X ก็ต่อเมื่อ $X \subseteq T$

เพราะฉะนั้นจึงนิยามกฎความสัมพันธ์ได้ว่า

- **กฎความสัมพันธ์** (Association Rule) คือการอุปนัยในรูปแบบ $X \rightarrow Y$ เมื่อ $X \subset I$, $Y \subset I$ และ $X \cap Y = \phi$

Objective measures of rule interest

- Support
- Confidence or strength
- Lift or Interest or Correlation
- Conviction
- Leverage or Piatetsky-Shapiro
- Coverage

Association Rule

- Rule form

  Antecedent ⟶ Consequent [*support, confidence*]

  Note: *support* and *confidence* are user defined measures of interestingness

- Examples

  buys(x, “computer”) ⟶ buys(x, “financial management software”) [0.5%, 60%]

  age(x, “30..39”) ^ income(x, “42..48K”) ⟶ buys(x, “car”) [1%,75%]

Rule basic Measures: Support and Confidence

$A \Rightarrow B$ [s, c]

Support: denotes the frequency of the rule within transactions. A high value means that the rule involve a great part of database.

$$support(A \Rightarrow B) = p(A \cup B)$$

Confidence: denotes the percentage of transactions containing A which contain also B. It is an estimation of conditioned probability .

$$confidence(A \Rightarrow B\ [\ s, c\ ]) = p(B|A) = sup(A,B)/sup(A)$$

Calculation of Support and Confidence

- Support

คำนวณหาค่าสนับสนุน ได้จากจำนวนทรานแซกชันที่มีรายการ *X* และ *Y* เกิดร่วมกัน หารด้วยจำนวนทรานแซกชันทั้งหมด

$$support\ (X \rightarrow Y)$$
$$= P(X \cup Y)$$
$$= tran\_count\ (X \cup Y) / tran\_count\ (D)$$

- Confidence

คำนวณค่าความเชื่อมั่นได้จากจำนวน ทรานแซกชันที่มีรายการ *X* และ *Y* เกิดร่วมกัน หารด้วยจำนวนทรานแซกชันที่มีรายการ *X*

$$confidence\ (X \rightarrow Y)$$
$$= P(Y|X)$$
$$= tran\_count\ (X \cup Y) / tran\_count\ (X)$$

Practice Calculating Support and Confidence

Transaction ID	Items Bought
2000	A,B,C
1000	A,C
4000	A,D
5000	B,E,F

ก. ให้คำนวณหาค่า support และ confidence ของความสัมพันธ์ A→C และ C→A

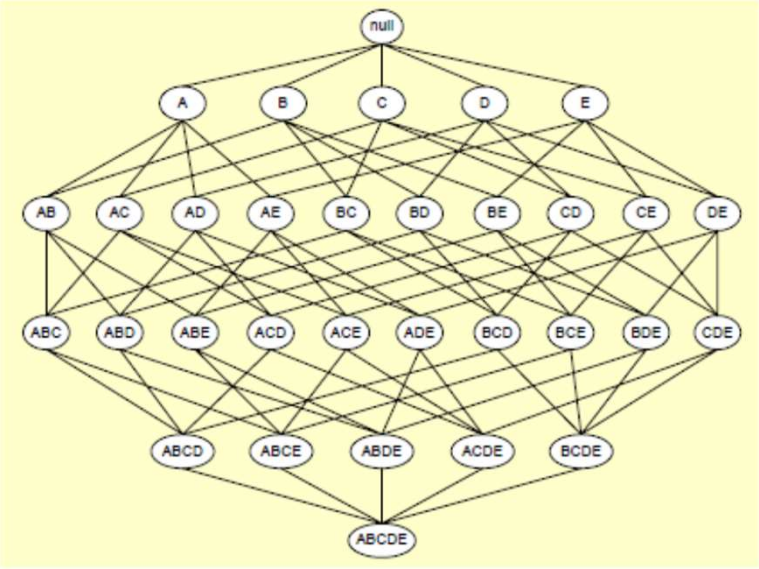
ข. กำหนดให้ minimum support = 50% และ minimum confidence = 80% อยากทราบว่าความสัมพันธ์ A→C และ C→A ความสัมพันธ์ใดเป็นกฎความสัมพันธ์

Association Mining

เป็นปัญหาการค้นหากฎความสัมพันธ์ นิยามได้ดังนี้

- **การค้นหากฎความสัมพันธ์** คือ การหากฎความสัมพันธ์ทั้งหมดในทรานแซกชันทุกตัวของเซตข้อมูลที่กำหนดให้ โดยกฎความสัมพันธ์ที่หาได้ทั้งหมดจะต้องมี*ค่าสนับสนุน* (support) ไม่ต่ำกว่าค่าสนับสนุนน้อยสุด (minimum support) ที่ผู้ใช้กำหนดไว้ และมี*ค่าความเชื่อมั่น* (confidence) ไม่ต่ำกว่าค่าความเชื่อมั่นน้อยสุด (minimum confidence) ที่ผู้ใช้ได้กำหนดไว้
- การค้นหากฎความสัมพันธ์สามารถแบ่งย่อยได้เป็นสองขั้นตอน คือ
  1. ค้นหาเซตของไอเทมปรากฏบ่อย **(frequent itemset) หรือไอเทมเซตที่มีค่าสนับสนุนไม่ต่ำกว่าค่าสนับสนุนน้อยสุดที่กำหนดให้**
  2. นำไอเทมเซตปรากฏบ่อยเหล่านั้นมาสร้างเป็นกฎความสัมพันธ์ต่อไป

Itemset Lattice



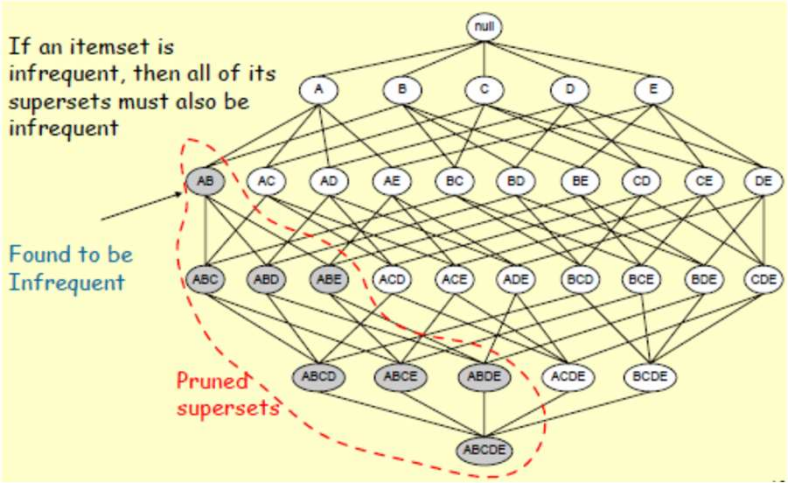
Apriori Principle

Any subset of a frequent itemset must also be frequent

No superset of any infrequent itemset should be generated or tested

- Many item combinations can be pruned

Apriori Principle for Pruning Candidates



Association Mining: 2 key steps

1. Find all Frequent Itemsets: the sets of items that pass minimum support
   - ❖ Apriori Algorithm
   - มีการจัดเรียงลำดับของไอเทมในแต่ละทรานแซกชันก่อนประมวลผล
   - การสร้างไอเทมเซตจะสร้างตามระดับชั้น จากชั้นที่ k, k+1, k+2 .....
   - ใช้ความรู้ก่อนหน้าคือคุณสมบัติของไอเทมเซตเกิดบ่อยในการตัดเล็ม
2. For every frequent itemset *X*, generate all non-empty subset *S* of *X*

   $S \rightarrow (X-S)$

   Output the rule $S \rightarrow (X-S)$

   If confidence >= min_confidence

Apriori Algorithm

Algorithm: Apriori. Find frequent itemsets using an iterative level-wise approach based on candidate generation.

Input:

- D, a database of transactions;
- min_sup, the minimum support count threshold.

Output: L, frequent itemsets in D.

Method:

```
(1)   L1 = find_frequent_1-itemsets(D);
(2)   for (k = 2; Lk−1 ≠ ϕ; k++) {
(3)       Ck = apriori_gen(Lk−1);
(4)       for each transaction t ∈ D { // scan D for counts
(5)           Ct = subset(Ck, t); // get the subsets of t that are candidates
(6)           for each candidate c ∈ Ct
(7)               c.count++;
(8)       }
(9)       Lk = {c ∈ Ck | c.count ≥ min_sup}
(10)  }
(11)  return L = ∪k Lk;

procedure apriori_gen(Lk−1:frequent (k − 1)-itemsets)
(1)   for each itemset l1 ∈ Lk−1
(2)       for each itemset l2 ∈ Lk−1
(3)           if (l1[1] = l2[1]) ∧ (l1[2] = l2[2]) ∧ ... ∧ (l1[k − 2] = l2[k − 2]) ∧ (l1[k − 1] < l2[k − 1]) then {
(4)               c = l1 ⋈ l2; // join step: generate candidates
(5)               if has_infrequent_subset(c, Lk−1) then
(6)                   delete c; // prune step: remove unfruitful candidate
(7)               else add c to Ck;
(8)           }
(9)   return Ck;

procedure has_infrequent_subset(c: candidate k-itemset;
          Lk−1: frequent (k − 1)-itemsets); // use prior knowledge
(1)   for each (k − 1)-subset s of c
(2)       if s ∉ Lk−1 then
(3)           return TRUE;
(4)   return FALSE;
```