Association Mining

วัตถุประสงค์ของการทำ Association Mining คือการค้นหาความสัมพันธ์ที่น่าสนใจ เช่น
ลูกค้าที่ซื้อผ้าห่มส่วนใหญ่ จะซื้อ เนยร์ ด้วย

เขียนกฎความสัมพันธ์ได้เป็น

{ผ้าห่ม} ⟶ {เนยร์}

{a,b,c} → {d}
{v,w} → {x,y,z}
{x} → {y}

ในข้อมูล tracsaction จริงๆ จะมีรายการสินค้าที่ถูกขายมากมาย ซึ่งการจับคู่ หรือจับกลุ่ม

รายการสินค้าเหล่านี้ทั้งหมด เพื่อหากฎความสัมพันธ์
เป็นความคิดที่ไม่ค่อยดีนัก เพราะจะต้องประมวลข้อมูลจำนวน
มหาศาล $(2^n - n - 1)$; n = จำนวนรายการสินค้าทั้งหมด
และกฎทั้งหมดที่ได้มาส่วนใหญ่ จะมีค่า Confidence < minimum Confidence

★ เพื่อหลีกเลี่ยงการประมวลผลข้อมูลจำนวนมหาศาล ★
เราจำเป็นต้องคัดรายการสินค้าที่ปรากฏน้อย ก่อน นำไปสร้างกฎความสัมพันธ์

Q: ทำไมต้องคัดเฉพาะสินค้าที่ปรากฏน้อย ?
A: เพราะการหากฎความสัมพันธ์ของสินค้าที่มีการขายน้อยครั้ง จะให้กฎความสัมพันธ์ที่เป็น noise (กฎความสัมพันธ์ใช้งานจริงไม่ได้)

ขั้นตอน Association Mining

1. หาเซตของรายการสินค้าที่ปรากฏน้อย ⟶ 2. หากฎความสัมพันธ์ของเซตเหล่านั้น

(1)

- How to หาเซตของรายการสินค้าที่ปรากฏน้อย
  Q: เราจะแยกอย่างไรว่า เซตของรายการสินค้าปรากฏน้อย หรือไม่น้อย ?
  A: เซตของรายการสินค้าที่มีค่า Support ≥ minimum Support
  จะเป็นเซตของรายการสินค้าที่ปรากฏน้อย

NO.
DATE / /

ค่า Support สามารถหาได้จาก

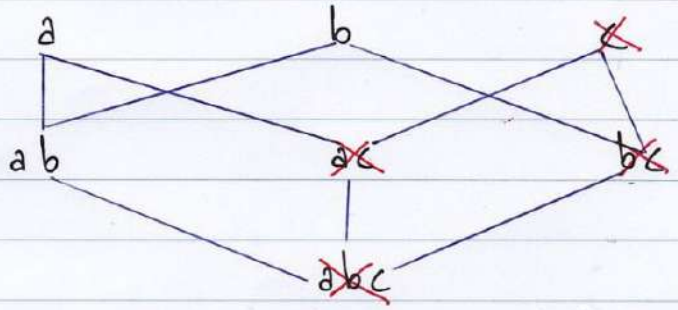
$$support(X \rightarrow Y) = \frac{\text{จำนวน transaction ที่มี } X \text{ และ } Y \text{ ปรากฏร่วมกัน}}{\text{จำนวน transaction ทั้งหมด}}$$

เขียนเป็นสูตร ดังนี้ $$support(X \rightarrow Y) = \frac{tran\_count(X \cup Y)}{tran\_count(D)}$$

* minimum support เป็น ค่าที่เราต้องเลือกเอง
โดยที่ $0 \leq$ minimum support ≤ 1

การหาเซตทั้งหมดของรายการสินค้าที่ปรากฏบ่อย ไม่ใช่ความคิดที่ดีนัก เนื่องจากจะต้องประมวลผลข้อมูลจำนวนมหาศาล $(2^n - 1)$

วิธีการแก้ก็คือ เราไม่ต้องสร้างเซตของรายการสินค้าที่มีสมาชิกเป็นเซตที่ไม่ปรากฏบ่อย



จากรูป ให้ เซตของสินค้า c เป็นเซตที่มีค่า support น้อยกว่า minimum support
เมื่อ support(c) < minimum support แสดงว่า เซตของสินค้า c เป็นเซตที่ไม่ปรากฏบ่อย
ดังนั้น เซตของ ac, bc และ abc ก็จะเป็นเซตที่ปรากฏไม่บ่อยด้วย {ปรากฏไม่บ่อย}
จึงไม่จำเป็นต้องสร้างเซตเหล่านี้ขึ้นมา

วิธีการนี้มีชื่อว่า APRIORI Algorithm

NO.
DATE / /

Example of APRIORI Algorithm

Tran_ID	List of Item
T1	I1, I2, I5
T2	I2, I4
T3	I2, I3
T4	I1, I2, I4
T5	I1, I3
T6	I2, I3
T7	I1, I3
T8	I1, I2, I3, I5
T9	I1, I2, I3

ขั้นตอนที่ 1

นับจำนวนรายการสินค้าทั้งหมด

ในที่นี้จะได้ว่า มี รายการสินค้าทั้งหมด 5 รายการ ได้แก่ I1, I2, I3, I4, I5

ขั้นตอนที่ 2

สร้าง Candidate itemset (C_1)

ทำการนับว่า รายการสินค้า แต่ละตัวในขั้นตอนที่ 1 ปรากฏกี่ครั้ง? จะได้ว่า {ใน transaction

I1 ปรากฏ 6 ครั้ง
I2 ปรากฏ 7 ครั้ง
I3 ปรากฏ 6 ครั้ง
I4 ปรากฏ 2 ครั้ง
I5 ปรากฏ 2 ครั้ง

ต่อมา เอาข้อมูลเหล่านี้ไปสร้าง Candidate itemset

Candidate itemset ประกอบด้วย

ชื่อ itemset และ ค่า support ของ itemset นั้น นั่นคือ (จำนวน transaction ทั้งหมด = 9)

C_1 =

Itemset	Support
{I1}	6/9
{I2}	7/9
{I3}	6/9
{I4}	2/9
{I5}	2/9

= L_1

ขั้นตอนที่ 3

สร้าง frequent itemset (L_1)

โดยการพิจารณา แต่ละ itemset ใน C_1 ว่ามีค่า support(itemset) ≥ minimum support ไหม ถ้าค่า support ของแต่ละ itemset ใน C_1 ≥ minimum support itemset นั้นก็จะผ่านการคัดเลือก ไปใช้สร้าง frequent itemset (ในที่นี้ minimum support = 2/9)

ดังนั้น ทุก itemset ใน C_1 จึงผ่านการคัดเลือกไปใช้สร้าง L_1

NO.
DATE / /

ขั้นตอนที่ 4

สร้าง itemset C_2

เราสามารถสร้าง itemset C_2 ได้จาก frequent itemset (L_1)

เพราะว่า itemset L_1 = {I1}, {I2}, {I3}, {I4}, {I5}

ดังนั้นจะสามารถสร้าง itemset C_2 ได้ทั้งหมด 10 แบบ ได้แก่

{I1,I2}, {I1,I3}, {I1,I4}, {I1,I5}, {I2,I3}, {I2,I4}, {I2,I5}, {I3,I4}, {I3,I5}, {I4,I5}

แต่ในการใช้งานจริง เราไม่ต้องสร้าง Itemset C_2 ขึ้นมาครบทุกแบบตามที่เขียนด้านบน เราทำการสร้างเฉพาะ itemset ที่มีการปรากฏใน transaction ก็เพียงพอ (เพื่อลดขั้นตอนในการประมวลผลในลำดับถัดๆไป)

จากการพิจารณา พบว่า

{I1,I2}	ปรากฏใน	T1, T4, T8, T9
{I1,I3}	"	T5, T7, T8, T9
{I1,I4}	"	T4
{I1,I5}	"	T1, T8
{I2,I3}	"	T3, T8, T9
{I2,I4}	"	T2, T4
{I2,I5}	"	T1, T8
{I3,I4}		ไม่ปรากฏใน transaction ใด
{I3,I5}	"	T8
{I4,I5}		ไม่ปรากฏใน transaction ใด

จากข้อมูลด้านบนจะเห็นว่า {I3,I4}, {I4,I5} ไม่มีปรากฏใน transaction ใดๆ จะได้ว่า

Itemset
{I1,I2}
{I1,I3}
{I1,I4}
{I1,I5}
{I2,I3}
{I2,I4}
{I2,I5}
{I3,I5}

*** จะเห็นว่า การสร้าง Itemset C_2 ในขั้นตอนนี้ เป็นเพียงการพิจารณาเท่านั้น ว่าต้องสร้าง itemset ใดขึ้นมาบ้าง (เป็นประโยชน์ต่อการลดขั้นตอนในลำดับถัดๆไป) ***

NO.
DATE / /

ขั้นตอนที่ 5 (เหมือนขั้นตอนที่ 2)

สร้าง Candidate itemset (C_2)

ทำการนับค่า itemset แต่ละตัวในขั้นตอนที่ 4 ปรากฏกี่ครั้งใน transaction จะได้ค่า

{I1, I2}	ปรากฏ	4	ครั้ง
{I1, I3}	"	4	ครั้ง
{I1, I4}	"	1	ครั้ง
{I1, I5}	"	2	ครั้ง
{I2, I3}	"	3	ครั้ง
{I2, I4}	"	2	ครั้ง
{I2, I5}	"	2	ครั้ง
{I3, I5}	"	1	ครั้ง

ต่อมานำข้อมูลเหล่านี้ไปสร้าง Candidate itemset นั่นคือ

C_2 =

Itemset	Support
{I1, I2}	4/9
{I1, I3}	4/9
{I1, I4}	1/9
{I1, I5}	2/9
{I2, I3}	3/9
{I2, I4}	2/9
{I2, I5}	2/9
{I3, I5}	1/9

NO.
DATE / /

ขั้นตอนที่ 6 (เหมือนขั้นตอนที่ 3)

สร้าง frequent itemset (L_2)

โดยการพิจารณาแต่ละ itemset ใน C_2 ว่ามีค่า support(itemset) ≥ minimum support ในหรือไม่ (ในที่นี้ minimum support = 2/9)

จากการพิจารณาจะได้ว่า

L_2 =

Itemset	Support
{I1, I2}	4/9
{I1, I3}	4/9
{I1, I5}	2/9
{I2, I3}	4/9
{I2, I4}	2/9
{I2, I5}	2/9

ขั้นตอนที่ 7 (เหมือนขั้นตอนที่ 4)

สร้าง itemset C_3

เราสามารถสร้าง itemset C_3 ได้จาก frequent itemset (L_2)

เพราะว่า itemset L_2 = {I1,I2}, {I1,I3}, {I1,I5}, {I2,I3}, {I2,I4}, {I2,I5}

ดังนั้นเราสามารถสร้าง itemset C_3 ได้ทั้งหมด 4 แบบ ได้แก่

{I1,I2,I3}, {I1,I2,I5}, {I2,I3,I4}, {I2,I3,I5}

แต่ในการใช้งานจริง เราไม่ต้องสร้าง itemset C_3 ขึ้นมาครบทุกแบบตามที่เขียนด้านบน เราทำการสร้างเฉพาะ itemset ที่มีการปรากฏใน transaction ก็พอ

จากการพิจารณาพบว่า

{I1, I2, I3}	ปรากฏใน	T8, T9
{I1, I2, I5}	ปรากฏใน	T1, T9
{I2, I3, I4}		ไม่ปรากฏใน transaction ใด
{I2, I3, I5}		ไม่ปรากฏใน transaction ใด

จากข้อมูลด้านบนจะเห็นว่า {I2,I3,I4}, {I2,I3,I5} ไม่มีปรากฏใน transaction ใดๆ จะได้ว่า

Itemset
{I1, I2, I3}
{I1, I2, I5}

ขั้นตอนที่ 8 (เหมือนขั้นตอนที่ 2, 5)

สร้าง Candidate itemset (C_3)

ทำการพิจารณาว่า itemset แต่ละตัวในขั้นตอนที่ 7 ปรากฏกี่ครั้งใน transaction จะได้ว่า

{I1, I2, I3} ปรากฏ 2 ครั้ง

{I1, I2, I5} ปรากฏ 2 ครั้ง

ต่อมานำข้อมูลเหล่านี้ไปสร้าง Candidate itemset นั่นคือ

C_3 =

Itemset	Support
{I1, I2, I3}	2/9
{I1, I2, I5}	2/9

ขั้นตอนที่ 9 (เหมือนขั้นตอนที่ 3, 6)

สร้าง frequent itemset (L_3)

โดยการพิจารณาแต่ละ itemset ใน C_3 ว่ามีค่า support(itemset) ≥ minimum support ใหม่ จากการพิจารณาจะได้ว่า (ในที่นี้ minimum support = 2/9)

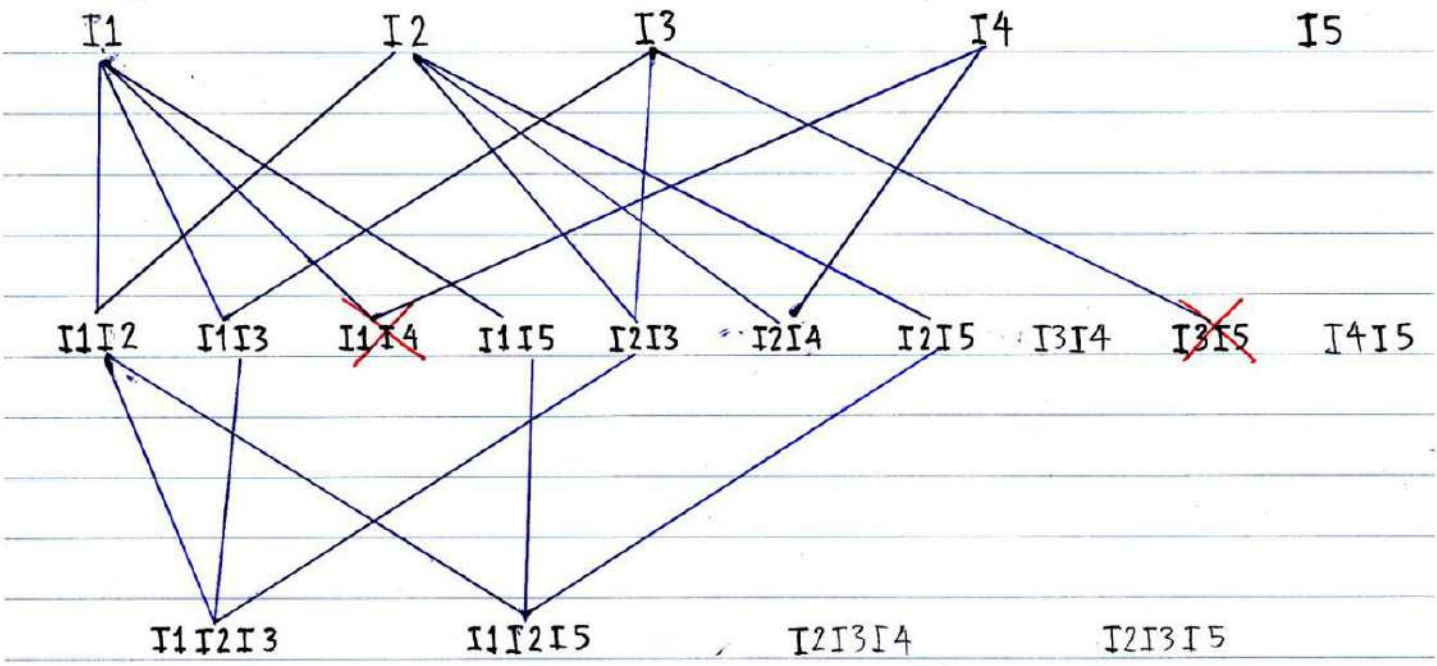
L_3 =

Itemset	Support
{I1, I2, I3}	2/9
{I1, I2, I5}	2/9

ถ้าเราพิจารณา itemset C_4 จะพบว่า itemset C_4 = {I1, I2, I3, I5} ซึ่งไม่มีปรากฏใน transaction ใดๆ

ดังนั้น

"END"



รูปแสดงการสร้าง frequent itemset
ด้วย APRIORI Algorithm

หลังจากที่เราหา เซตของรายการสินค้าที่ปรากฏบ่อย (frequent itemset) เสร็จแล้ว
ต่อมา เราก็ทำการหา กฎ ความสัมพันธ์ ได้เลย

(2)

- How to หากฎความสัมพันธ์

Q: เราจะทราบได้อย่างไรว่า เซตของรายการสินค้า มีความสัมพันธ์กัน ?

A: เซตของรายการสินค้าที่มีค่า Confidence ≥ minimum Confidence
จะเป็นเซตของรายการสินค้าที่มีความสัมพันธ์กัน

ค่า Confidence สามารถหาได้จาก

$$confidence(x \rightarrow y) = \frac{\text{จำนวน transaction ที่มี x และ y ปรากฏร่วมกัน}}{\text{จำนวน transaction ที่มี x ปรากฏ}}$$

เขียนเป็นสูตรได้ดังนี้ $confidence(x \rightarrow y) = \frac{tran\_count(X \cup Y)}{tran\_count(X)}$

NO.
DATE / /

* minimum confidence เป็นค่าที่เราต้องเลือกเอง
โดยที่ $0 \leq$ minimum confidence ≤ 1

เราเรียก confidence $(x \rightarrow y)$ ที่ $\geq$ minimum confidence
ว่า "Strong Association Rule"

Example of Confidence Calculation

เราจะใช้ frequent itemset ที่ได้มาจากขั้นตอนที่ 1 (ตัวอย่างที่แล้ว)
มาหาค่า confidence

เราไม่สนใจ frequent itemset ที่มีสมาชิกเพียง 1 ตัว ในการหากฎความสัมพันธ์
จะได้ว่า

$$\text{confidence}(I1 \rightarrow I2) = \frac{\text{tran\_count}(I1 \cup I2)}{\text{tran\_count}(I1)} = \frac{4}{6}$$

$$\text{confidence}(I2 \rightarrow I1) = \frac{\text{tran\_count}(I1 \cup I2)}{\text{tran\_count}(I2)} = \frac{4}{7}$$

* จากตัวอย่าง เราสังเกตได้ว่า confidence $(x \rightarrow y)$ ไม่จำเป็น
ต้องเท่ากับ confidence $(y \rightarrow x)$

$$\text{confidence}(I1, I2 \rightarrow I3) = \frac{\text{tran\_count}(I1 \cup I2 \cup I3)}{\text{tran\_count}(I1 \cup I2)} = \frac{2}{4}$$

$$\text{confidence}(I1 \rightarrow I2, I3) = \frac{\text{tran\_count}(I1 \cup I2 \cup I3)}{\text{tran\_count}(I1)} = \frac{2}{6}$$

NO.
DATE / /

$$confidence(I3 \rightarrow I1, I2) = \frac{tran\_count(I1 \cup I2 \cup I3)}{tran\_count(I3)} = \frac{2}{6}$$

$$confidence(I1, I3 \rightarrow I2) = \frac{tran\_count(I1 \cup I2 \cup I3)}{tran\_count(I1 \cup I3)} = \frac{2}{4}$$

$$confidence(I2 \rightarrow I1, I3) = \frac{tran\_count(I1 \cup I2 \cup I3)}{tran\_count(I2)} = \frac{2}{7}$$

$$confidence(I2, I3 \rightarrow I1) = \frac{tran\_count(I1 \cup I2 \cup I3)}{tran\_count(I2 \cup I3)} = \frac{2}{3}$$

---

* ขั้นตอนสุดท้าย หลังจากได้กฎความสัมพันธ์ (Strong Association Rule) แล้ว ก็คือ การทดสอบว่า Strong Association Rule สามารถนำไปใช้งานจริงได้ไหม?

ยกตัวอย่าง เช่น

confidence(A → B) = 0.7 นั่นหมายความว่า มีโอกาส 70% ที่ลูกค้าซื้อสินค้า A แล้วจะซื้อสินค้า B ด้วย

แต่ถ้า เราทราบว่า ความน่าจะเป็นที่ลูกค้าซื้อสินค้า B เป็น 90% จากจำนวน transaction ทั้งหมด แล้ว Strong Association Rule ที่เราได้มา หมายถึงอะไรกับเรา?

(3)

- วิธีการตรวจสอบว่ากฎความสัมพันธ์ที่ได้มา สามารถนำไปใช้ได้จริงไหม มีดังนี้

① หาค่าสหสัมพันธ์ ซึ่งมีสูตร $corr_{A,B} = \frac{P(A \cup B)}{P(A)P(B)}$

② พิจารณาค่า $corr_{A,B}$ ที่ได้

- ถ้า $corr_{A,B} \geq 1$ แล้ว Strong Association Rule สามารถใช้งานได้จริง

- แต่ถ้า $corr_{A,B} < 1$ แล้ว Strong Association Rule ไม่สามารถใช้งานได้จริง (การซื้อสินค้าชนิดหนึ่ง จะลดโอกาสที่จะซื้อสินค้าอีกชนิดหนึ่ง)

(10)