Association Rules

Created	@December 28, 2021 3:33 AM
Tags	

💡 在一個 Transaction table 中找出 attribute 的 co-occurrence（僅探討共同出現的可能性、而不是因果關係）

Example：超商的購物紀錄

Transaction Table

TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Support

$$s = \frac{\sigma(Milk, Diaper, Beer)}{|Transactions|} = 0.4$$

Confidence

$$c = \frac{\sigma(Milk, Diaper, Beer)}{\sigma(Milk, Diaper)} = 0.67$$

💡 這邊套討出的 confidence value 其實並不具有統計意義，單純是一個絕對的計量值而已

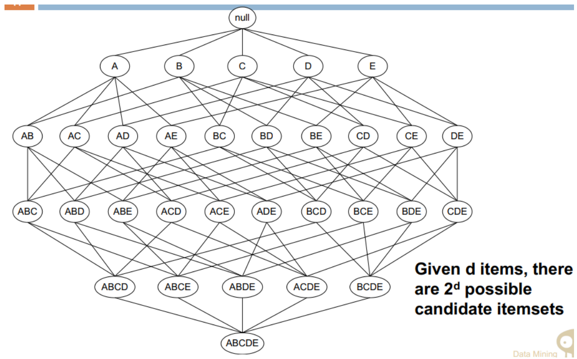
一、Simple Association Rules Mining

(1) Brute-force

演算法說明：

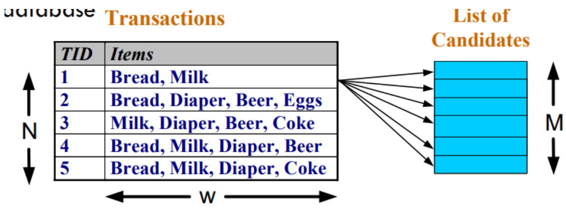
- 依序列舉出所有可能的 k itemset

- Scan transaction table 來驗證其是否為 frequent itemset



分析：

> 💡 如右圖所示，時間複雜度為 $O(NMw)$，其中 M 為 candidates 的數量將高達 2^d



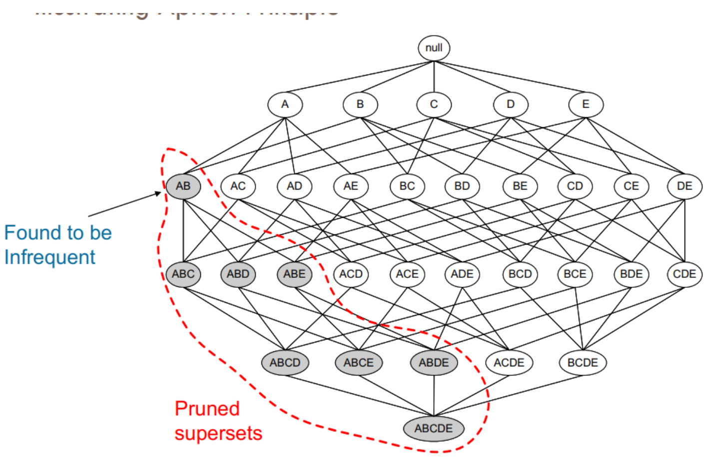
TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

此方法有兩個明顯可優化的地方：

1. 窮舉出的 candidates 之間有繼承性，並不需要每一個都列出來驗證
2. Table 被不斷 scan，顯然不必要

(2) Apriori algorithm

> 💡 核心概念：
> Frequent itemset 的 subset 必定也會是 frequent itemset（因為條件必定更寬鬆），這代表 **non-frequent itemset 的 superset 必定也不會是 frequent itemset**




```
L1={frequent 1-itemsets};
for (k=2; Lk-1 ≠ 0; k++) do begin
    Ck=apriori-gen(Lk-1);
    for each transactions t ∈D do begin //scan DB
     Ct=subset(Ck, t) //get the subsets of t that are candidates
     for each candidate c ∈Ct do
         c.count++;
    end
    Lk={c ∈Ck | c.count ≥ minsup}
end
Answer=∪kLk;
```

缺點：

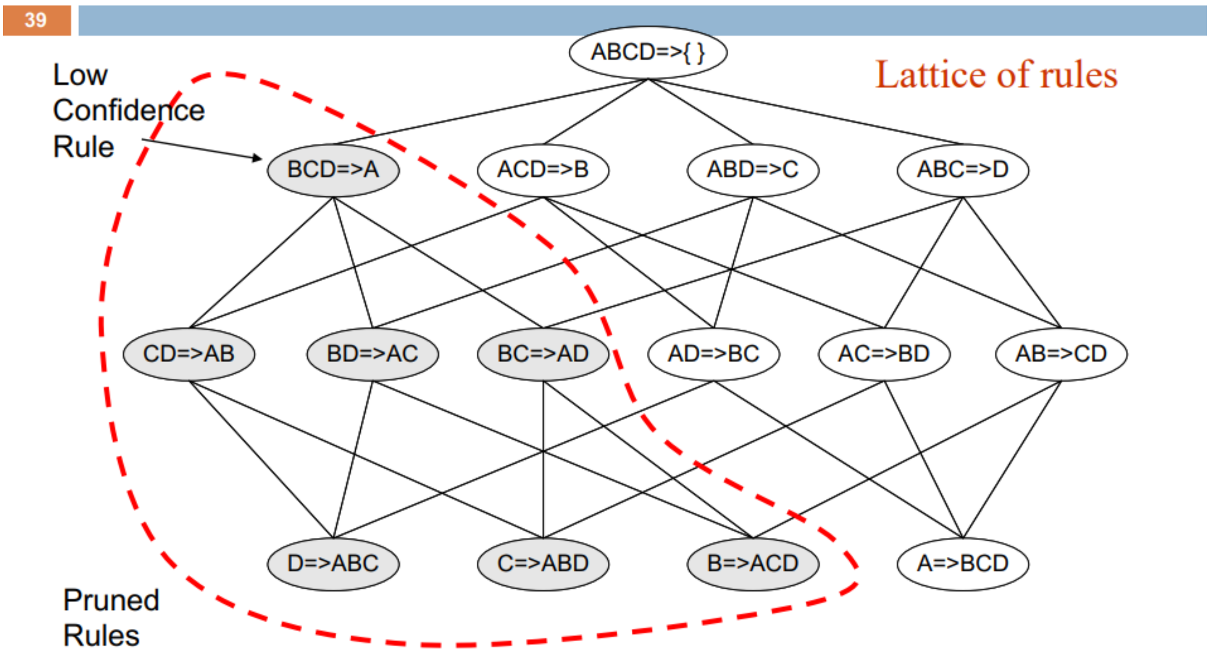
1. 每個 candidate 還是要對 table scan 才能驗證，顯然不必要
2. Candidates 的數量依舊龐大

(3) FP-growth

💡 從 frequency 最小的往回找 ⇒ 是因為他們最有可能被 prune 掉，讓每次 conditional FP-tree 的 size 盡可能縮小

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/1a2d881f-b5dd-4c99-9b0f-db68594ec58e/FPG_example.pdf

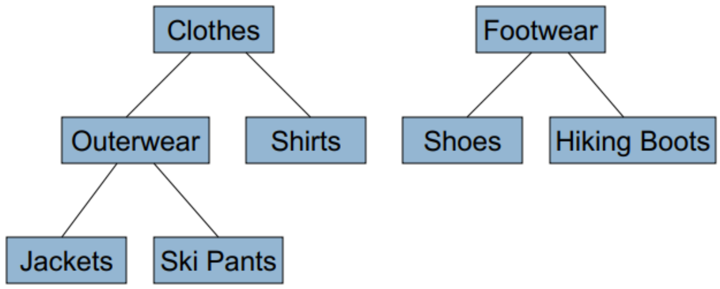
Rule Generation (by apriori algorithm)



二、Multilevel Association Rules Mining

💡 通常人類只會對更 high-level 的 association 感興趣，太過 detail 的資訊反而很難理解

Tx	Items bought
100	Shirt
200	Jacket, Hiking Boots
300	Ski Pants, Hiking Boots
400	Shoes
500	Shoes
600	Jacket



例如：association rule 的結果告訴我買了外套的人會買布鞋，但我只想知道買了衣服的人會不會順便買鞋子

(1) Mining Methods

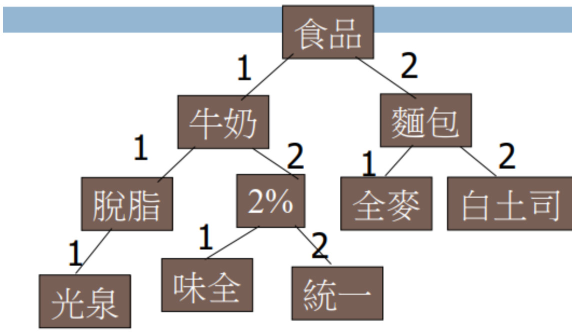
1. 為所有 high level class 都額外新增一筆 transaction

Freq. pattern	Support
Jacket	2
Outerwear	3
Clothes	4
Shoes	2
Hiking Boots	2
Footwear	4
OW, HB	2
Clothes, HB	2
OW, FW	2
Clothes, FW	2

	sup(30%)	conf(60%)
OW -> HB	33%	66%
OW -> FW	33%	66%
HB -> OW	33%	100%
HB -> Clothes	33%	100%
Jacket -> HB	16%	50%
Ski Pants -> HB	16%	100%

💡 會多算出很多我們並不感興趣的 rule，這個方法顯然不理想

2. 將 class level 置入 item 編號中



TID	Items
T1	{111, 121, 211, 221}
T2	{111, 211, 222, 323}
T3	{112, 122, 221, 411}
T4	{111, 121}
T5	{111, 122, 211, 221, 413}

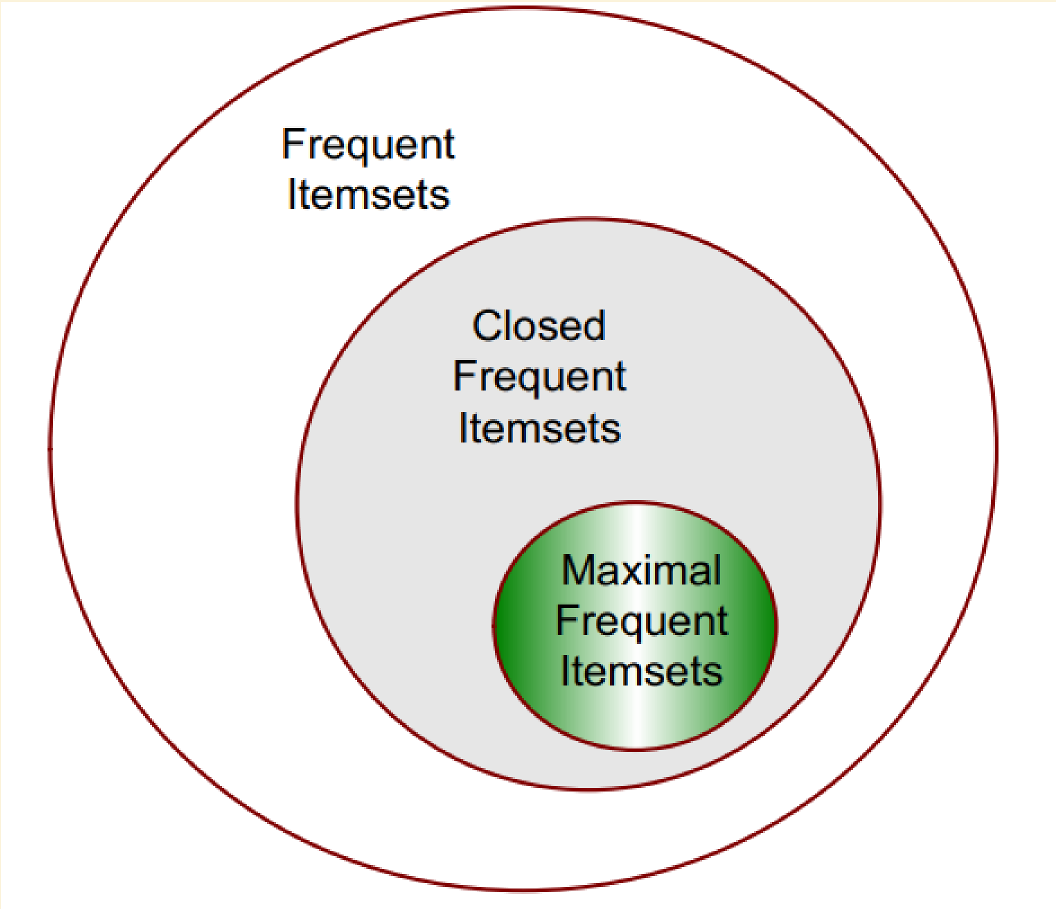
(2) Uniform Support v.s. Reduced Support

💡 越 high level 的 class 必定會有較的 support ⇒ min-support 值應要隨不同的 level class 而動態調整，其調整策略在不同的 data 上會有不同的適配性

- Level-by-level independent
  - Each node is examined, regardless of whether or not its parent node is found to be frequent.
- Level-cross filtering by single item
  - An item at the i-th level is examined iff its parent node at the (i-1)-th level is frequent
- Level-cross filtering by k-itemset
  - Check parent k-itemset
- Controlled level-cross filtering by single item

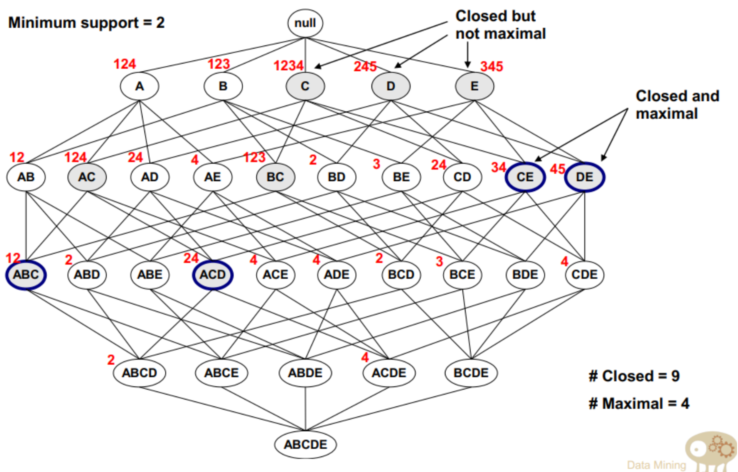
Maximal Itemset v.s. Closed Itemset

👉 在所有 frequent-itemsets 中，找出「具有代表性的」的幾個 rules 就好



(1) Closed-patterns

TID	Items
1	ABC
2	ABCD
3	BCE
4	ACDE
5	DE



(2) Max-patterns

保留「最長」的 rules 就好，其訊號強度會 dominate 其他 rules

即不存在任何 frequent itemset 為其 superset

Min_sup=2

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

- BCDE, ACD are max-patterns
- BCD is not a max-pattern

Quantitative Association Rules

> 💡 資料的特性並非 binary 時，有可能是 multi-class 或 continuous

將連續的資料離散化，且讓資料分佈數量盡可能平衡即可

People

Record ID	Age	Married	NumCars
100	23	No	1
200	25	Yes	1
300	29	No	0
400	34	Yes	2
500	38	yes	2

Age	Married
20..24:1 25..29:2 30..34:3 35..39:4	Yes:1 No:2

After Mapping attributes

Record ID	Age	Married	NumCars
100	1	2	1
200	2	1	1
300	2	2	0
400	3	1	2
500	4	1	2

Frequent Itemset (Sample)	Support
{Age:25..29}	2
{Age:30..39}	2
{Married:Yes}	3
{Married:No}	2
{NumCars:1}	2
{NumCars:2}	2
{<Age:30..39>,<Married:Yes>}	2

Rules: Sample

Rule	Support	Confidence
<Age:30..39>and<Married:Yes>=><NumCars:2>	40%	100%
<Age:20..29>=><NumCars:0..1>	60%	100%

Mining Association Rules with Weighted Items

💡 有時產出的 association rules 彼此之間需要 ranking，找出效益最大的前 n 個 rules 就好

- Weighted items
- Weighted support
- Association rule with minimum weighted support
- Given minimum weighted support 0.4
  => {B,E} ((0.3+0.9)*5/7=0.86)

code	Item	Profit	Weight
A	Apple	100	0.1
B	Orange	300	0.3
C	Banana	400	0.4
D	Milk	800	0.8
E	Coca	900	0.9

TID	Items
100	A, B, D, E
200	A, D, E
300	B, D, E
400	A, B, D, E
500	A, C, E
600	B, D, E
700	B, C, D, E