Frequent Pattern Analysis

`Definition`

A pattern that occurs **frequently** in a data set

`Importance`

Disclose an intrinisic and important property of data sets

Types of data or knowledge

- associative pattern
- sequential pattern
- Sub-Grapgh pattern
- Iceberg cube

Main operations

- read / write / point

Other Methods

- closed / max pattern
- compression method
- pruning method
- constraints

Association Rule

概念

- 规则（**Rule**）：

$$\{x_1, x_2, x_3, \ldots, x_n\} \to Y$$

- 可信度（**Confidence**）和最小可信度
    - 购买$x_1, x_2, \ldots x_n$的情况下购买Y的可能性，条件概率
    - $Confidence(A \to B) = P(B|A)$
- 支持度（**Support**）和最小支持度
    - 同时购买$x_1, x_2, \ldots x_n$和Y的可能性

    - $Support(A \to B) = P(A \cup B)$
- 频繁项目集

  满足最小支持度的项目集

Example for calculate Support & Confidence

$\{ABC, ACD, BCD, ADE, BCE\}$

Rule	Support	Confidence
A -> D	2/5	2/3
C -> A	2/5	2/4
A -> C	2/5	2/3
B & C -> D	1/5	1/3

PS： 注意因果关系

Evolution of AR (Association Rule)

1. AR Model
2. Apriori（层次算法产生候选集）
3. FP-Growth

Sub Problems of AR

1. 依据最小支持度，寻找频繁项目集
2. 依据最小可信度，产生关联规则

重要公理

如果一个项目集S是频繁的（项目集S的出现频度大于最小支持度），那么S的任意子集是频繁的

Eg. $\{a, b, c\}$ 其子集 $\{a, b\}$

其逆否命题

如果一个项目集合S是不频繁的，那么它的任何超集是不频繁的

Eg. $\{a\}$ 其超集 $\{a, b\}$

算法

1. 分层挖掘（每一层需要对数据做一次扫描）

   我们只需将精力放在大小为2的频繁项目集上

2. 对数据做1、2次扫描就找出频繁项目集（利用公理）

> Apriori
>
> 1. self-joining L_k
> 2. pruning

Apriori Example (找出频繁集,建立关联规则)

$\{ABC, AC, BCD, DE, ABCD\}$ （$Min_s = 2[\text{常忽略分母}], Min_c$ = 80%）

C_1

item	Freq
A	3
B	3
C	4
D	3
E	1

L_1

item	Freq
A	3
B	3
C	4
D	3

C_2

item	Freq

AB	2
AC	3
AD	1
BC	3
BD	2
CD	2

L_2

item	Freq
AB	2
AC	3
BC	3
BD	2
CD	2

C_3

item	Freq
ABC	2
BCD	2

L_3

item	Freq
ABC	2
BCD	2

Based on L_2 We can get

Rule	Confidence

A -> B	2/3
B -> A	2/3
A -> C	1
C -> A	3/4
...	...

Based on L_3 We can get

Rule	Confidence
A -> BC	2/3
B -> AC	2/3
C -> AB	2/4
AB -> C	1
...	...

算法缺陷:候选集的生成耗费太大

算法改进

- 基于Hash
- 基于Partition
- 基于Sample

Graph Mining

A (sub)graph is *frequent* if its support in a given dataset is no less than a minimum support threshold

> Subgraph Explosion Problem