Data mining

Frequent Pattern Mining

Outline

1

Frequent Pattern Mining

1. Definition
2. Application
3. Concepts

2

Methods

1. Apriori Algorithm
2. Improving Apriori Efficiency
3. ECLAT Algorithm

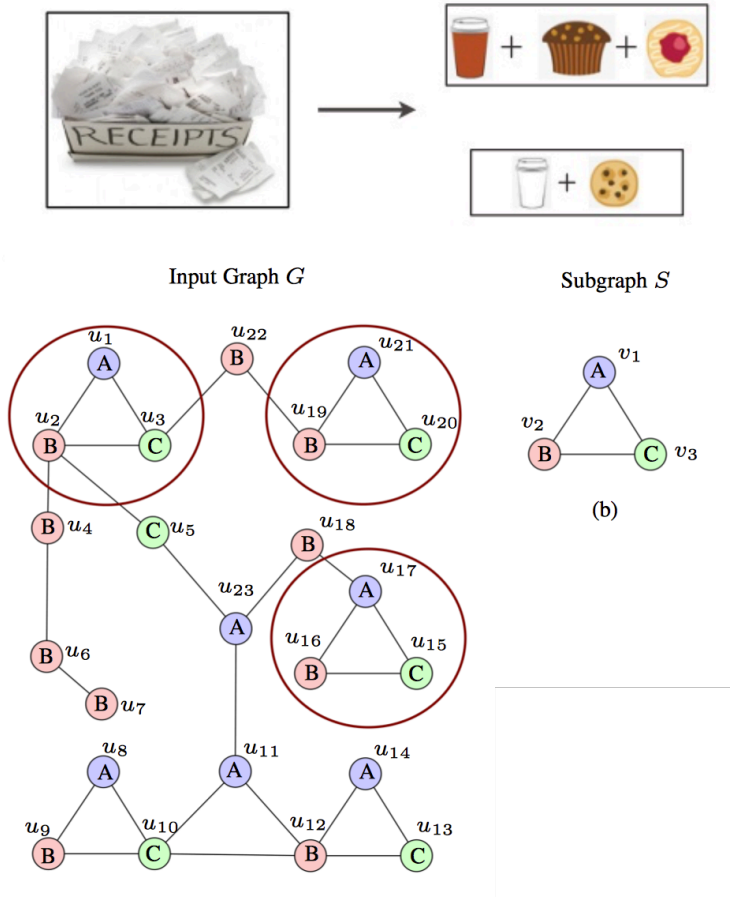
3

Association Rule

1. Support and Confidence
2. Mining Association Rules
3. Correlation Measures

Frequent Pattern Mining

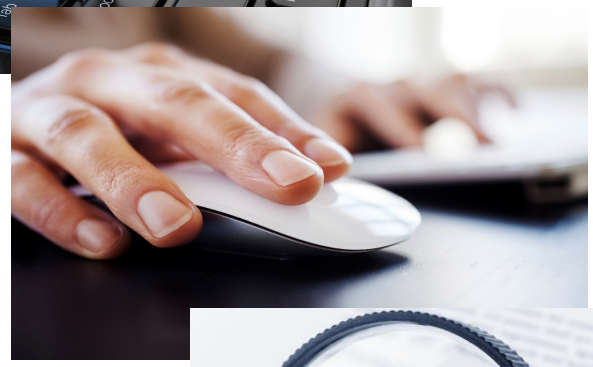
- Frequent Pattern: a pattern (a set of items, subsequences, substructures, etc.) that occurs frequently in a dataset.
  - What products were often purchased together?
  - What are the subsequent purchases after buying a PC?
  - What kinds of DNA are sensitive to this new drug?
- First proposed in 1993 in the context of frequent itemset mining.



Abdelhamid, Ehab, et al. "Incremental Frequent Subgraph Mining on Large Evolving Graphs." *IEEE Transactions on Knowledge and Data Engineering* (2017).

Applications

- Market basket analysis
  - Cross-marketing
  - Shelf management
- Web log (click stream) analysis
- DNA sequence analysis
- Word association mining
- …



Basic Concepts

- Itemset: A set of items
- k-itemset: An itemset with k items.
- Support count: Number of transactions that contain an itemset
- Support ratio: Fraction of transactions that contain an itemset
- Frequent itemset: An itemset whose support is greater than or equal to a minsup threshold

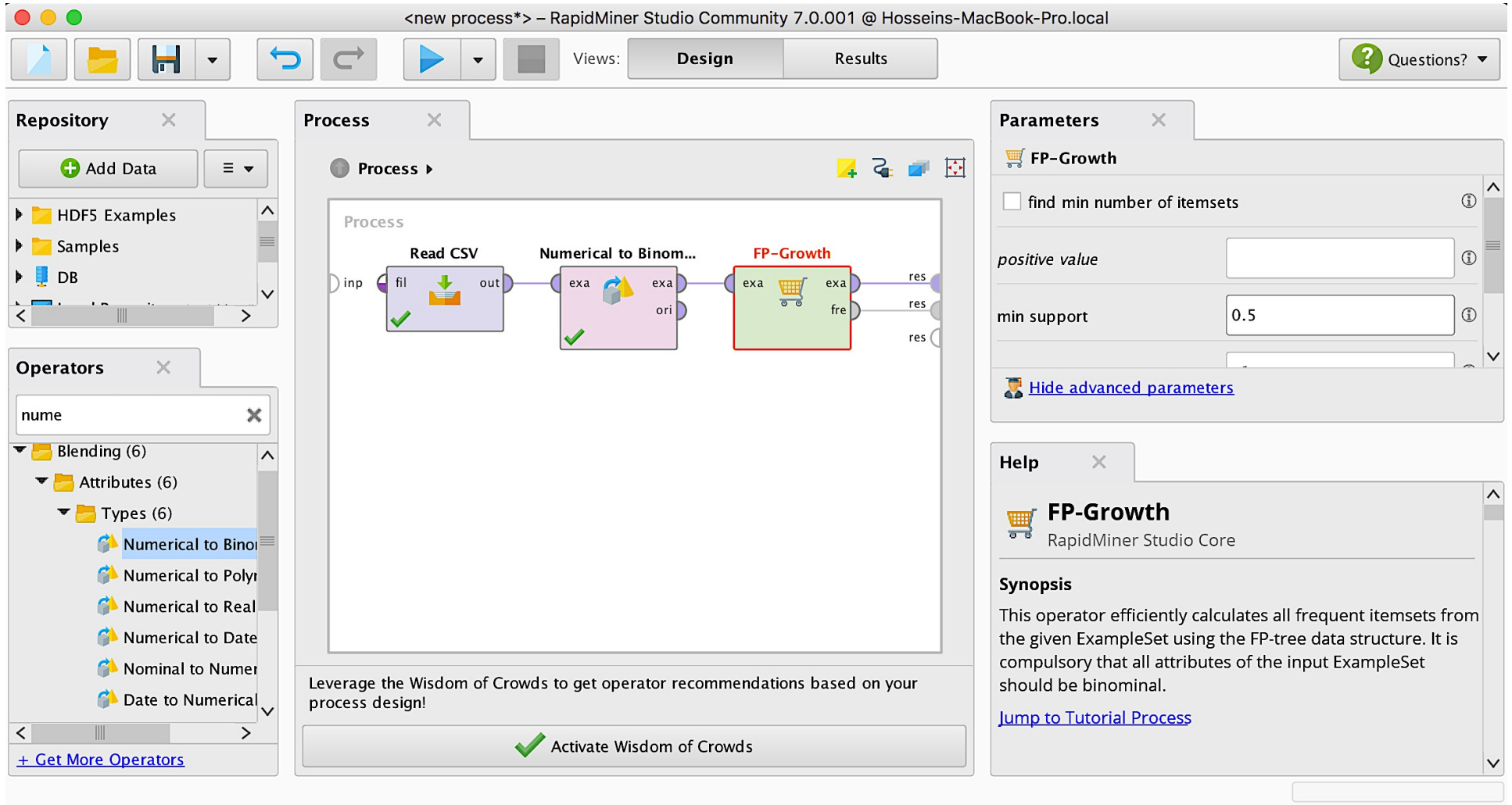
Basic Concepts

TID	Transaction
T_{10}	A, C, D
T_{20}	B, C, E
T_{30}	A, B, C, E
T_{40}	B, E

1-itemset	2-itemset	3-itemset
Support count ({C}) = 3 Support ratio ({C})= 3/4	Support count ({B, C}) = 2 Support ratio ({B,C})= 2/4	Support count ({B, C, E}) = 2 Support ratio({B,C,E}) = 2/4

If minsup = 0.7 → {C} is a Frequent itemset

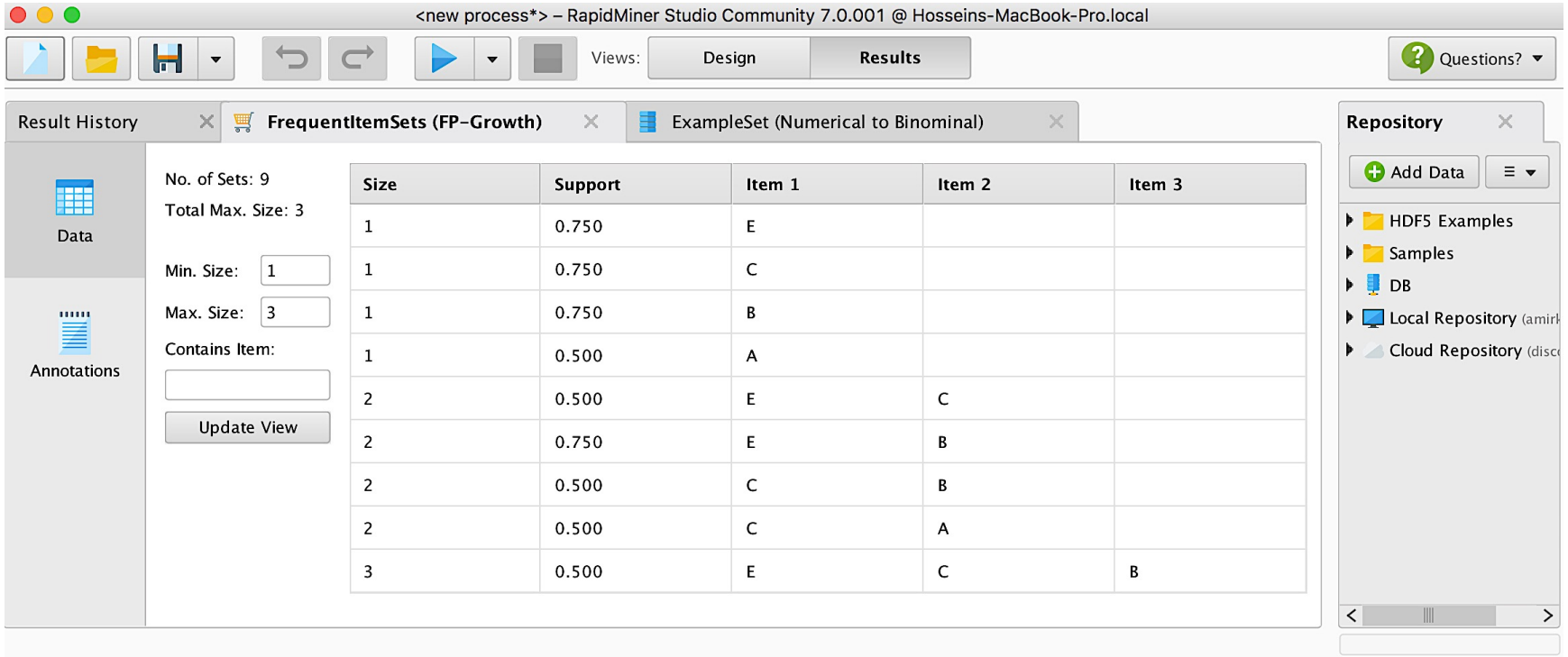
Rapidminer example



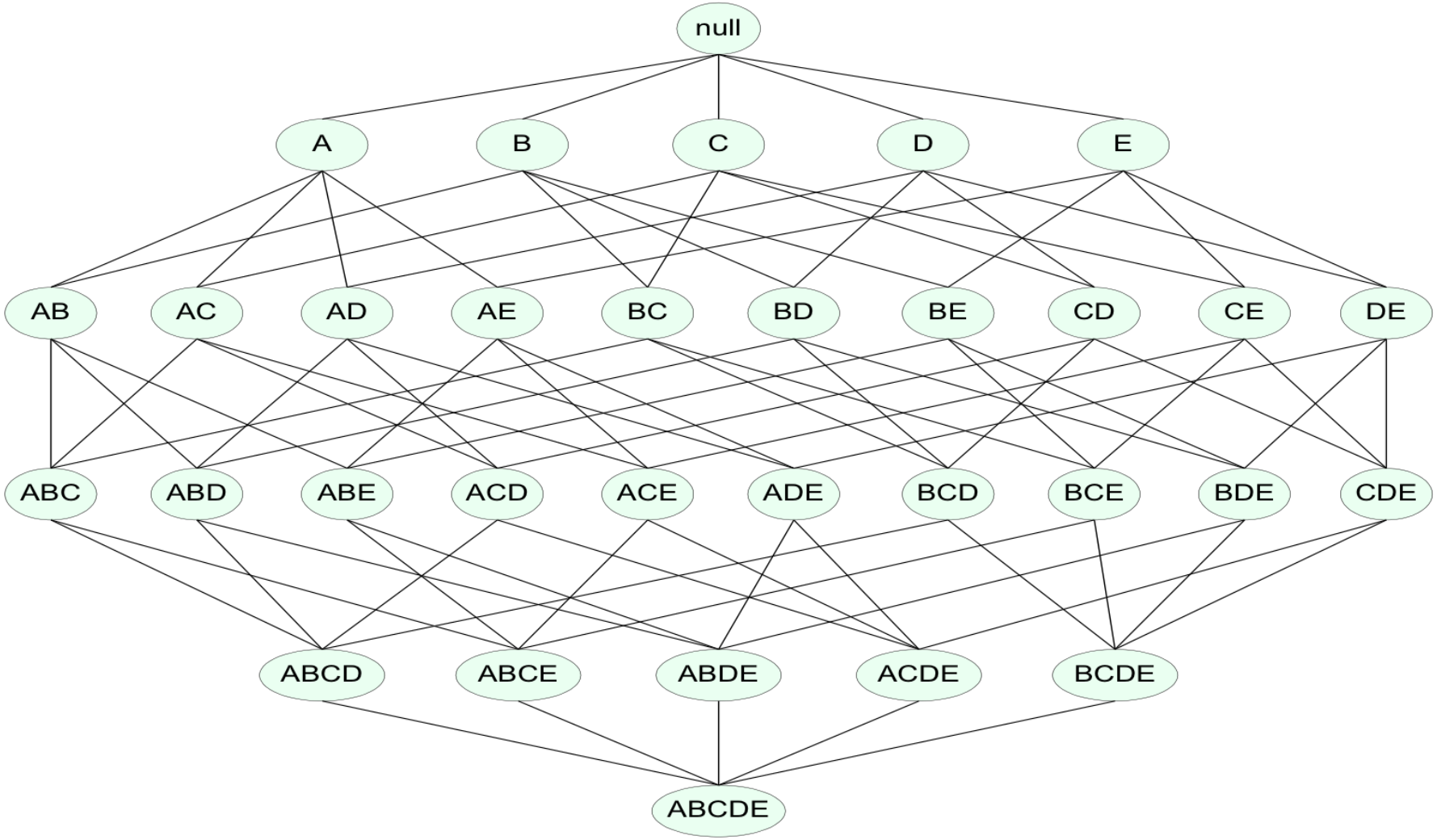
Rapidminer example

TID	Transaction
T_{10}	A, C, D
T_{20}	B, C, E
T_{30}	A, B, C, E
T_{40}	B, E

```
A,B,C,D,E
1,0,1,1,0
0,1,1,0,1
1,1,1,0,1
0,1,0,0,1
```



Frequent Itemset Mining

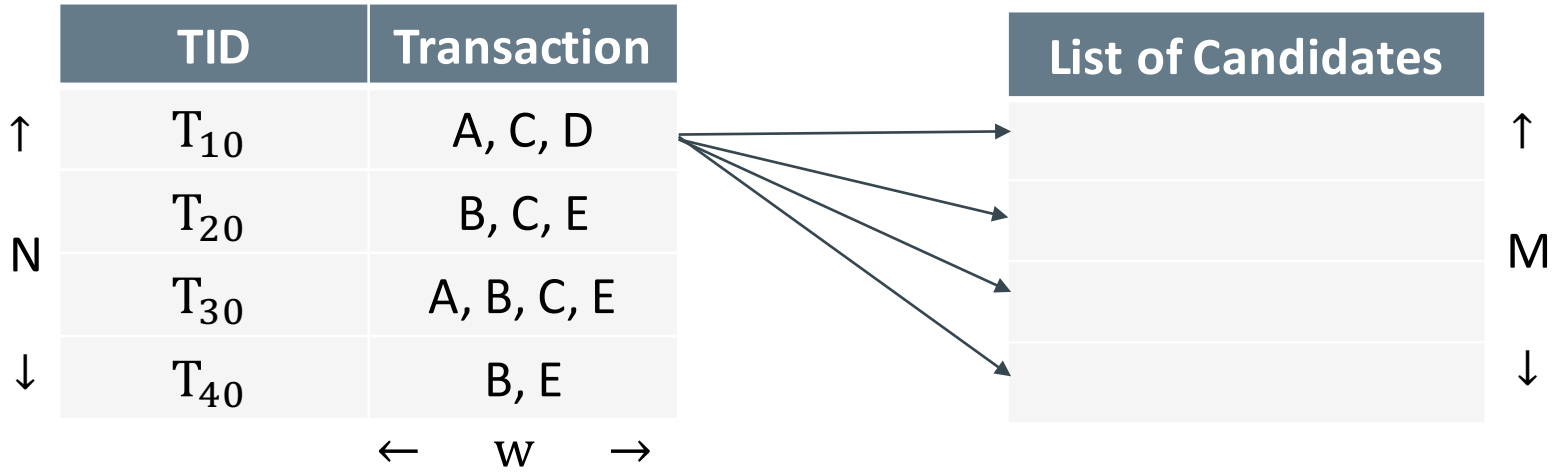


Frequent Itemset Mining

- Brute-force approach:

– Each itemset in the lattice is a candidate frequent itemset

– Count the support of each candidate by scanning the database



– Match each transaction against every candidate

– Complexity ~ O(NMw) => Expensive since M = 2^d ☹

Apriori Algorithm

Input: DB, minsup

- Initially, scan DB once to get frequent 1-itemset
- Set k=1
- Generate length (k+1) candidate itemsets from length k frequent itemsets
- Test the candidates against DB to get frequent (k+1)-itemsets
- k += 1
- Terminate when no frequent or candidate set can be generated

Initialization

Iteration body

Stop condition

Apriori Algorithm – Example I

TID	Transaction
T_{10}	A, C, D
T_{20}	B, C, E
T_{30}	A, B, C, E
T_{40}	B, E
minsup = 2	

Scan D to count each item →

C_1

Itemset	Sup.count
A	2
B	3
C	3
D	1
E	3

Compare supports with minsup →

L_1

Itemset	Sup.count
A	2
B	3
C	3
E	3

Apriori Algorithm – Example I

Generate C_2 candidates from L_1 →

C_2

Itemset
A B
A C
A E
B C
B E
C E

Scan D for count of each candidate →

C_2

Itemset	Sup. count
A B	1
A C	2
A E	1
B C	2
B E	3
C E	2

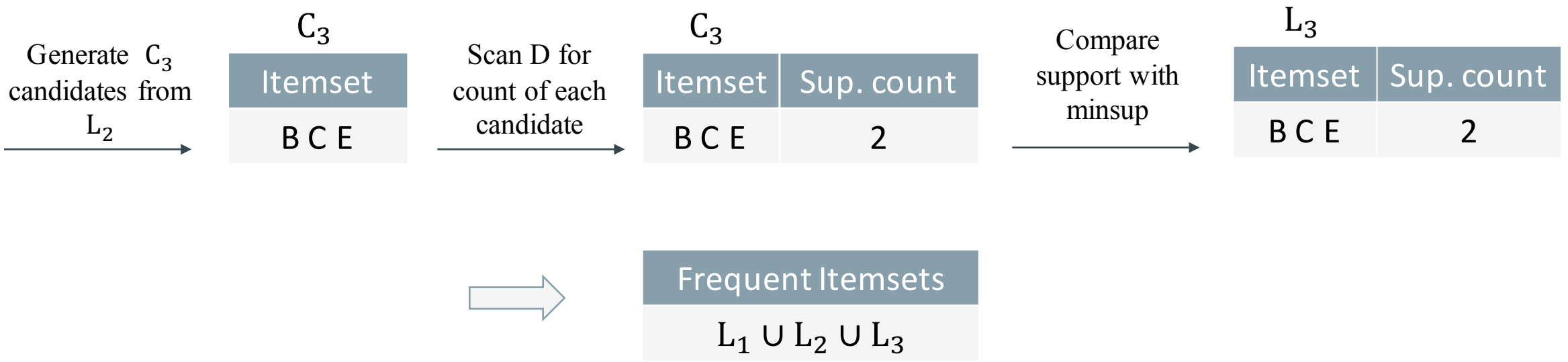
Compare supports with minsup →

L_2

Itemset	Sup.count
A C	2
B C	2
B E	3
C E	2

TID	Transaction
T_{10}	A, C, D
T_{20}	B, C, E
T_{30}	A, B, C, E
T_{40}	B, E

Apriori Algorithm – Example I



How to Generate Candidates?

1. Self joining L_{K-1}: C_k is generated by joining L_{k-1} with itself

2. Pruning: Any (k-1)-itemset that is not frequent cannot be a subset of a frequent k-itemset: The apriori prunning principle

C_k: Candidate itemset of size k

L_k : frequent itemset of size k

Example

L_3={abc, abd, acd, ace, bcd}

Self-joining: $L_3 * L_3$

abcd from abc and abd

acde from acd and ace

Pruning:

acde is removed because ade is not in L_3

C_4 = {abcd}

Apriori Algorithm – Example II

TID	Items
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
minsup = 2	

Scan D for count of each candidate →

C_1

Itemset	Sup.count
I1	6
I2	7
I3	6
I4	2
I5	2

Compare Candidate support with minsup count →

L_1

Itemset	Sup.count
I1	6
I2	7
I3	6
I4	2
I5	2

Apriori Algorithm – Example II

Generate C_2 candidates from L_1

C_2

Itemset
I1, I2
I1, I3
I1, I4
I1, I5
I2, I3
I2, I4
I2, I5
I3, I4
I3, I5
I4, I5

Scan D for count of each candidate

C_2

Itemset	Sup. count
I1, I2	4
I1, I3	4
I1, I4	1
I1, I5	2
I2, I3	4
I2, I4	2
I2, I5	2
I3, I4	0
I3, I5	1
I4, I5	0

Compare Candidate support with minsup count

L_2

Itemset	Sup.count
I1, I2	4
I1, I3	4
I1, I5	2
I2, I3	4
I2, I4	2
I2, I5	2

Apriori Algorithm – Example II

Generate C_3 candidates from L_2 →

C_3

Itemset
I1, I2, I3
I1, I2, I5

Scan D for count of each candidate →

C_3

Itemset	Sup. count
I1, I2, I3	2
I1, I2, I5	2

Compare Candidate support with minsup count →

L_3

Itemset	Sup. count
I1, I2, I3	2
I1, I2, I5	2

Generate C_4 candidates from L_3 →

C_4

Itemset
∅

⇒

Frequent Itemsets
$L_1 \cup L_2 \cup L_3$

Methods to Improve Apriori's Efficiency

Sampling

Mining on a subset of given data with a lower support threshold

Transaction reduction

A transaction that does not contain any frequent k-itemset is useless in subsequent scans

Methods to Improve Apriori's Efficiency

Direct hashing

A k-itemset whose corresponding hashing bucket count is below the threshold cannot be frequent

Partitioning

Any itemset that is potentially frequent in DB must be frequent in at least one of the partitions of DB

Improve Apriori: Direct Hashing

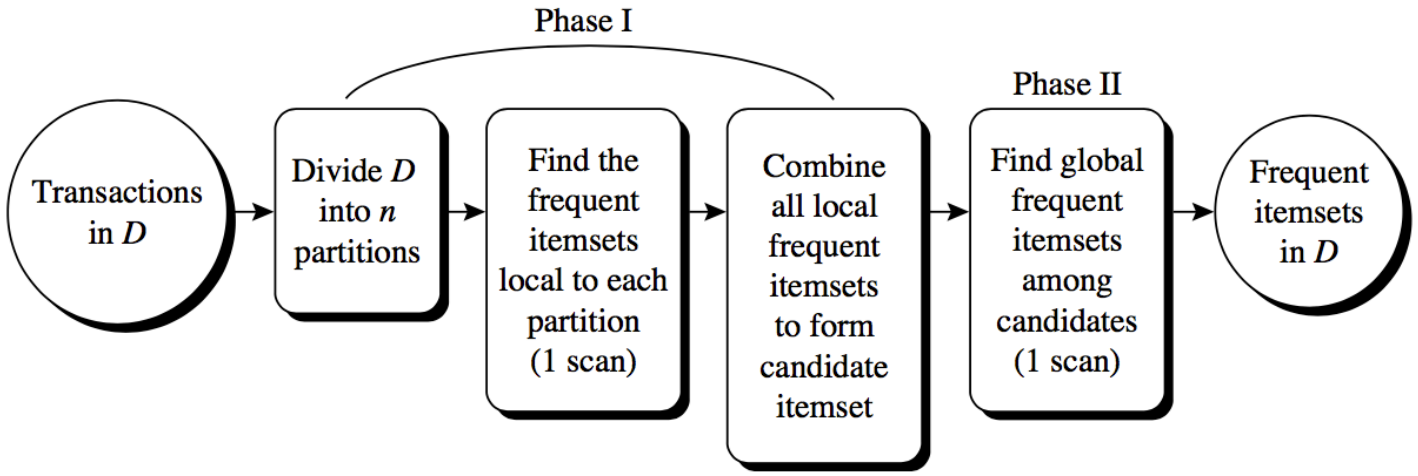
- Hash entries
  - {ab, ad, ae}
  - {bd, be, de}
  - …
- Frequent 1-itemset: a, b, d, e
- ab is not a candidate 2-itemset if the sum of count of {ab, ad, ae} is below support threshold.
- It reduces the number of candidates.

Count	Itemset
35	{ab, ad, ae}
88	{bd, be, de}
. .	. .
102	{yz, qs, wt}

Hash Table

Improve Apriori: Partitioning

- Scan database only twice.
- Any itemset that is frequent in DB must be frequent in at least one of the partitions of DB.



Horizontal Data Format

The same data can be represented by different formats.

TID	Transaction
T_1	A, B, C
T_2	B
T_3	B, C, D
T_4	A, D

Horizontal Format 1

TID	Item
T_1	A
T_1	B
T_1	C
T_2	B
T_3	B
T_3	C
…	…

Horizontal Format 2

TID	Transaction			
	A	B	C	D
T_1	1	1	1	0
T_2	0	1	0	0
T_3	0	1	1	1
T_4	1	0	0	1

Horizontal Format 3

Vertical Data Format

TID	Transaction
T_1	A, B, C
T_2	B
T_3	B, C, D
T_4	A, D

Horizontal Format 1

item	Transaction				
		T_1	T_2	T_3	T_4
A	T_1 , T_4	1	0	0	1
B	T_1 , T_2 , T_3	1	1	1	0
C	T_1 , T_3	1	0	1	0
D	T_3 , T_4	0	0	1	1

Vertical Format

Exp

This format is widely used by search engines (inverted index).
Horizontal: document → words
Vertical: words → document

Eclat Algorithm

- ECLAT (Equivalence Class Transformation) is similar to Apriori algorithm, but it uses the vertical data format.
- First, we transform the horizontal format into the vertical format by scanning the database once.
- The support count of an itemset is simply the length of the TID list of that itemset → Obtain frequent 1-itemsets.

TID	Items
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

Horizontal

VS

Vertical

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

Eclat Algorithm

- The candidate generation phase is like the Apriori algorithm.
- We intersect the TID lists of the frequent k-itemsets to obtain the TID lists of the corresponding (k+1)-itemsets. There are linear-time algorithm for intersection of the sorted lists.

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

$$L_1 = \{I1, I2, I3, I4, I5\}$$

TID	Items
I1, I2	T1, T4, T8, T9
I1, I3	T5, T7, T8, T9
I1, I4	T4
I1, I5	T1, T8
I2, I3	T3, T6, T8, T9
I2, I4	T2, T4
I2, I5	T1, T8
I3,I4	∅
I3, I5	T8
I4,I5	∅

$$L_2 = \{I_1I_2, I_1I_3, I_1I_5, I_2I_3, I_2I_4, I_2I_5\}$$

$$C_3 = \{I_1I_2I_3, I_1I_2I_5\}$$

TID	Items
I1, I2, I3	T8, T9
I1, I2, I5	T1, T8

$$C_4 = \emptyset$$

Eclat Algorithm

- Besides taking advantage of the Apriori property in the generation of candidate (k+1)-itemsets from frequent k-itemsets, there is no need to scan the database to find the supports of candidate (k+1)-itemsets.

- This is because the TID list of each k-itemset carries the complete information required for counting such supports.

Max-Itemset and Colsed Frequent Itemset

- A long frequent pattern contains an exponential number of frequent sub-patterns, e.g., $\{a_1, \ldots, a_{100}\}$ contains $\binom{100}{1} + \binom{100}{2} + \ldots + \binom{100}{100} = 2^{100} - 1 = 1.27 \times 10^{30}$ sub-patterns!

- Solution: Mine max-itemsets and closed frequent itemsets instead.

- Max-itemsets are lossy compression (lose the support information), while closed frequent itemsets are lossless compression (keep the support information).

Max-Itemset

An itemset X is a max-itemset if X is frequent and there exists no frequent super-pattern Y ↄ X.

L_1

Itemset	Sup.count
I1	6
I2	7
I3	6
I4	2
I5	2

L_2

Itemset	Sup.count
I1, I2	4
I1, I3	4
I1, I5	2
I2, I3	4
I2, I4	2
I2, I5	2

L_3

Itemset	Sup. count
I1, I2, I3	2
I1, I2, I5	2

Max-Itemsets: $\{I_1I_2I_3, I_1I_2I_5, I_2I_4\}$

Closed Frequent Itemset

An itemset X is closed frequent if X is frequent and there exists no super-pattern Y ↄ X, with the same support as X.

L_1

Itemset	Sup.count
I1	6
I2	7
I3	6
I4	2
I5	2

L_2

Itemset	Sup.count
I1, I2	4
I1, I3	4
I1, I5	2
I2, I3	4
I2, I4	2
I2, I5	2

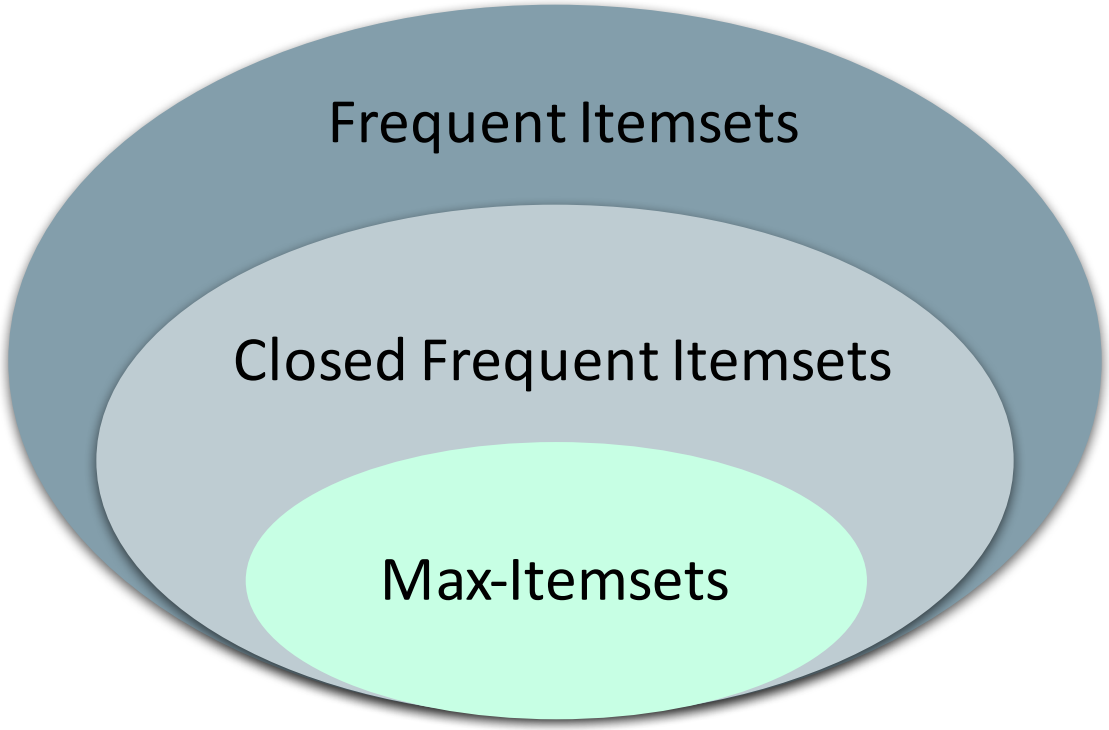
L_3

Itemset	Sup. count
I1, I2, I3	2
I1, I2, I5	2

Closed Frequent Itemsets

$\{I_1I_2I_3: 2, I_1I_2I_5: 2, I_1I_2: 4, I_1I_3: 4, I_2I_3: 4, I_2I_4: 2, I_1: 6, I_2: 7, I_3: 6\}$

Max-Itemsets vs. Closed Frequent Itemsets



Association Rules

- Association Rule
  - An expression of the form X → Y, where X and Y are nonempty disjoint itemsets ($X \neq \emptyset, Y \neq \emptyset, X \cap Y = \emptyset$)
- Given a set of transactions T, the goal of association rule mining is to find all strong rules having
  - support ≥ minsup threshold
  - confidence ≥ minconf threshold

Rule Evaluation Metrics

Support Fraction of transactions that contain both A and B

$$\text{Support}(A \rightarrow B) = \text{Support}\ (A \cup B) = \text{Support}(B \rightarrow A)$$

Confidence How often B appears in transactions that contain A

$$\text{Confidence}(A \rightarrow B) = \frac{\text{Support}\ (A \cup B)}{\text{Support}\ (A)}$$

Support measures the coverage of the rule. Confidence measures the accuracy of the rule.

Example

TID	Items
1	A, B
2	A, C, D, E
3	B, C, D, F
4	A, B, C, D
5	A, B, C, F

{B, C} → {D}	(sup = 0.4, conf = 0.67)
{B, D} → {C}	(sup = 0.4, conf = 1.0)
{C} → {B, D}	(sup = 0.4, conf = 0.5)
{B} → {C}	(sup = 0.6, conf = 0.75)

Support VS Confidence

I. Support and confidence are both high.

I
A, B
A, B, C
A, B, D
A, B
A, B, C, D

$A \rightarrow B$

sup = 1

conf = 1

II. Support and confidence are both low.

II
A
B
A, C
B, C
C, D

$A \rightarrow B$

sup = 0

conf = 0

Support VS Confidence

III. Confidence is high and support is low.

III
A, B, D
A, C, D
A, D, E
B, E, F
B, C, D, E, F
G, A

$$G \rightarrow A$$

$$sup = \frac{1}{6}$$

$$conf = 1$$

IV. Confidence is low and support is high.

It is impossible because:
$Sup \leq Conf$

$$Conf(A \rightarrow B) = \frac{P(A,B)}{P(A)} = \frac{Sup(A \rightarrow B)}{P(A)}$$

$$P(A) \leq 1$$

Association Rule Mining

- Brute-force approach:
  - List all possible association rules
  - Compute the support and confidence for each rule
  - Prune rules that fail the minsup and minconf thresholds

- It is impractical, because we can generate different rules for each itemset, and there are an exponential number of itemsets!

Association Rule Mining

Use the frequent itemsets:

1. Generate all itemset with support $\geq$ minsup

2. Generation high confidence rules from each frequent itemset, where each rule is a binary partitioning of a frequent itemset

Association Rule Mining-Example I

L_1

Itemset	Sup.count
I1	6
I2	7
I3	6
I4	2
I5	2

L_3

Itemset	Sup. count
I1, I2, I3	2
I1, I2, I5	2

TID	Items
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

Minsup = 2, minconf = %70

L_2

Itemset	Sup.count
I1, I2	4
I1, I3	4
I1, I5	2
I2, I3	4
I2, I4	2
I2, I5	2

Association Rule Mining-Example I

I1, I2	I1 → I2	$conf = \frac{4}{6}$ ✗
	I2 → I1	$conf = \frac{4}{7}$ ✗
I1, I3	I1 → I3	$conf = \frac{4}{6}$ ✗
	I3 → I1	$conf = \frac{4}{6}$ ✗
I2, I5	I2 → I5	$conf = \frac{2}{7}$ ✗
	I5 → I2	$conf = 1$ ✓

L_1

Itemset	Sup.count
I1	6
I2	7
I3	6
I4	2
I5	2

L_2

Itemset	Sup.count
I1, I2	4
I1, I3	4
I1, I5	2
I2, I3	4
I2, I4	2
I2, I5	2

Association Rule Mining-Example I

I1, I2, I3	I1 → I2 I3	conf $= \frac{2}{6}$	✗
	I2 → I1 I3	conf $= \frac{2}{7}$	✗
	I3 → I1 I2	conf $= \frac{2}{6}$	✗
	I1 I2 → I3	conf $= \frac{2}{4}$	✗
	I1 I3 → I2	conf $= \frac{2}{4}$	✗
	I2 I3 → I1	conf $= \frac{2}{4}$	✗
I1, I2, I5	I5 → I1 I2	conf = 1	✓
	I1 I5 → I2	conf = 1	✓
	I2 I5 → I1	conf = 1	✓

L_1

Itemset	Sup.count
I1	6
I2	7
I3	6
I4	2
I5	2

L_2

Itemset	Sup.count
I1, I2	4
I1, I3	4
I1, I5	2
I2, I3	4
I2, I4	2
I2, I5	2

Association Rule Mining-Example II

ID	Basketball	Cereal consumption
…	…	…
…	…	…

C \ B	YES	NO	
YES	2000	1750	3750
NO	1000	250	1250
	3000	2000	5000

Basketball → Cereal consumption
$sup = \frac{2000}{5000} = \%\ 40$
$conf = \frac{2000}{3000} = \%\ 66$

Basketball → $\overline{\text{Cereal consumption}}$
$sup = \frac{1000}{5000} = \%\ 20$
$conf = \frac{1000}{3000} = \%\ 33.3$

$P(\text{Cereal consumption}) = \frac{3750}{5000} = \%\ 75$

$P(\overline{\text{Cereal consumption}}) = \%\ 25$

Association Rule Mining-Example III

Is Symptom → Disease a valid rule?

S \ D	YES	NO	
YES	80	40	120
NO	20	10	30
	100	50	150

S → D
$\text{sup} = \frac{80}{15000} = \%\ 53$
$\text{conf} = \frac{80}{120} = \%\ 66$

But S and D are independent!

$P(D|S) = P(D) = 0.67$

Lift Measure

Strong Rules are not necessarily interesting.
We need more measures to evaluate rules.

$$\text{Lift}(A \rightarrow B) = \frac{P(A,B)}{P(A)P(B)} = \frac{\text{conf}\,(A \rightarrow B)}{P\,(B)} = \text{Lift}(B \rightarrow A)$$

Lift < 1	P (B | A) < P (B)	Negative Correlation
Lift = 1	P (B | A) = P (B)	Independent
Lift > 1	P (B | A) > P (B)	Positive Correlation

Lift Measure - Example

C \ B	YES	NO	
YES	2000	1750	3750
NO	1000	250	1250
	3000	2000	5000

Basketball → Cereal consumption

$$\text{Lift} = \frac{\frac{2000}{5000}}{\frac{3000}{5000} \times \frac{3750}{5000}} = \frac{100}{3 \times 375} = 0.88$$

Basketball → $\overline{\text{Cereal consumption}}$

$$\text{Lift} = \frac{\frac{1000}{5000}}{\frac{3000}{5000} \times \frac{1250}{5000}} = \frac{500}{3 \times 125} = 1.33$$

Lift Measure

Lift measure is not null-invariant

	B	$\overline{B}$	
C	100	1000	1100
$\overline{C}$	1000	null count	
	1100		

If null count = 100000

$$\text{Lift (B,C)} = \frac{P(B,C)}{P(B)P(C)} = \frac{\frac{100}{102100}}{\frac{1100}{102100} \times \frac{1100}{102100}} = 8.44 \gg 1$$

If null count = 100

$$\text{Lift (B,C)} = \frac{P(B,C)}{P(B)P(C)} = \frac{\frac{100}{2200}}{\frac{1100}{2200} \times \frac{1100}{2200}} = 0.18 \ll 1$$

All-confidence

$$\text{All-confidence}(A,B) = \frac{P(A,B)}{\max(P(A),P(B))}$$

$$0 \leq \text{All-confidence} \leq 1$$

	B	$\overline{B}$
C	100	1000
$\overline{C}$	1000	null count

If null count = 100000: $\text{All-conf}(B,C) = \frac{\frac{100}{102100}}{\max(\frac{1100}{102100}, \frac{1100}{102100})} = \frac{1}{11}$

If null count = 100: $\text{All-conf}(B,C) = \frac{\frac{100}{2200}}{\max(\frac{1100}{2200}, \frac{1100}{2200})} = \frac{1}{11}$

Other Measures

symbol	measure	range	formula
ϕ	ϕ-coefficient	-1...1	$\frac{P(A,B)-P(A)P(B)}{\sqrt{P(A)P(B)(1-P(A))(1-P(B))}}$
Q	Yule's Q	-1 ...1	$\frac{P(A,B)P(\overline{A},\overline{B})-P(A,\overline{B})P(\overline{A},B)}{P(A,B)P(\overline{A},\overline{B})+P(A,\overline{B})P(\overline{A},B)}$
Y	Yule's Y	-1 ...1	$\frac{\sqrt{P(A,B)P(\overline{A},\overline{B})}-\sqrt{P(A,\overline{B})P(\overline{A},B)}}{\sqrt{P(A,B)P(\overline{A},\overline{B})}+\sqrt{P(A,\overline{B})P(\overline{A},B)}}$
k	Cohen's	-1 ...1	$\frac{P(A,B)+P(\overline{A},\overline{B})-P(A)P(B)-P(\overline{A})P(\overline{B})}{1-P(A)P(B)-P(\overline{A})P(\overline{B})}$
PS	Piatetsky-Shapiro's	-0.25 ...0.25	$P(A,B)-P(A)P(B)$
F	Certainty factor	-1 ...1	$\max(\frac{P(B\|A)-P(B)}{1-P(B)}, \frac{P(A\|B)-P(A)}{1-P(A)})$
AV	added value	-0.5 ...1	$\max(P(B\|A)-P(B), P(A\|B)-P(A))$
K	Klosgen's Q	-0.33 ...0.38	$\sqrt{P(A,B)}\max(P(B\|A)-P(B), P(A\|B)-P(A))$
g	Goodman-kruskal's	0 ...1	$\frac{\Sigma_j \max_k P(A_j,B_k)+\Sigma_k \max_j P(A_j,B_k)-\max_j P(A_j)-\max_k P(B_k)}{2-\max_j P(A_j)-max_k P(B_k)}$
M	Mutual Information	0 ...1	$\frac{\Sigma_i\Sigma_j P(A_i,B_j)\log\frac{P(A_i,B_j)}{P(A_i)P(B_J)}}{\min(-\Sigma_i P(A_i)\log P(A_i)\log P(A_i),-\Sigma_i P(B_i)\log P(B_i)\log P(B_i))}$
J	J-Measure	0 ...1	$\max(P(A,B)\log(\frac{P(B\|A)}{P(B)})+P(A\overline{B})\log(\frac{P(\overline{B}\|A)}{P(\overline{B})}))$ $P(A,B)\log(\frac{P(A\|B)}{P(A)})+P(\overline{A}B)\log(\frac{P(\overline{A}\|B)}{P(\overline{A})})$
G	Gini index	0 ...1	$\max(P(A)[P(B\|A)^2+P(\overline{B}\|A)^2]+P(\overline{A}[P(B\|\overline{A})^2+P(\overline{B}\|\overline{A})^2]-P(B)^2-P(\overline{B})^2,$ $P(B)[P(A\|B)^2+P(\overline{A}\|B)^2]+P(\overline{B}[P(A\|\overline{B})^2+P(\overline{A}\|\overline{B})^2]-P(A)^2-P(\overline{A})^2)$
s	support	0 ...1	$P(A,B)$
c	confidence	0 ...1	$max(P(B\|A), P(A\|B))$
L	Laplace	0 ...1	$\max(\frac{NP(A,B)+1}{NP(A)+2}, \frac{NP(A,B)+1}{NP(B)+2})$
IS	Cosine	0 ...1	$\frac{P(A,B)}{\sqrt{P(A)P(B)}}$
γ	coherence(Jaccard)	0 ...1	$\frac{P(A,B)}{P(A)+P(B)-P(A,B)}$
α	all_confidence	0 ...1	$\frac{P(A,B)}{\max(P(A),P(B))}$
o	odds ratio	0 ...∞	$\frac{P(A,B)P(\overline{A},\overline{B})}{P(\overline{A},B)P(A,\overline{B})}$
V	Conviction	0.5 ...∞	$\max(\frac{P(A)P(\overline{B})}{P(A\overline{B})}, \frac{P(B)P(\overline{A})}{P(B\overline{A})})$
λ	lift	0 ...∞	$\frac{P(A,B)}{P(A)P(B)}$
S	Collective strength	0 ...∞	$\frac{P(A,B)+P(\overline{AB})}{P(A)P(B)+P(\overline{A})P(\overline{B})} \times \frac{1-P(A)P(B)-P(\overline{A})P(\overline{B})}{1-P(A,B)-P(\overline{AB})}$
χ^2	χ^2	0 ...∞	$\Sigma_i \frac{(P(A_i)-E_i)^2}{E_i}$

References

Thanks to Mina Heidari, member of our lab, for her assistance in preparing this slide.

J. Han, M. Kamber, J. Pei, "Data mining, concepts and techniques," 3rd edition, Morgan Kaufmann, 2011.

Some slides are adopted from "Dr. Konstantinos Sagonas Data Mining lectures"

Exercises

1. Find another application of frequent pattern mining or association rule mining from scientific papers (refer to the paper).
2. Chapter 6 of Han's book, third edition: 1, 3, 6, 8, 9, 10, 11, 14
3. Extract the association rules of an arbitrary Persian document (relatively big). Each sentence is a transaction. Use the Hazm module to extract the information. Prepare and submit a *scientific report*.

```
from hazm import sent_tokenize, word_tokenize

print(sent_tokenize('این یک جمله تست است. از ابزار هضم استفاده شده است'))
print(word_tokenize('این یک جمله تست است'))
```

```
['این یک جمله تست است.', 'از ابزار هضم استفاده شده است']
['این', 'یک', 'جمله', 'تست', 'است']
```

http://www.sobhe.ir/hazm/

Deadline: 10 PM, 24th Farvardin 1397

Exam: 26th Farvardin 1397 (exclude from final exam)

The End