Association Rule Mining




a presentation by
Dr.B.Sivaselvan
Associate Professor, CSE
IIITD&M Kancheepuram

Img Src: AI on medium mc.ai

Association Rule Mining

- Association Rule – Implication : $X \rightarrow Y$; disjoint item-sets
- Application in Market Basket Analysis (MBA)
- Statistical Measures ;Above rule
  - Support - P(X U Y) – Rule Significance
  - Confidence - P (Y | X) – Certainty Degree
- Min Support / Confidence Strong Association Rules
- Boolean / Multidimensional / Quantitative Rules

Frequent Pattern Mining (FPM)

- Phases of Association Rule Mining
  - Frequent Item-sets Generation
  - Strong Association Rules Generation
- Apriori Algorithm –First Major contribution for FPM.
- Levelwise – Candidate generation based
- Prior Knowledge – Apriori property
- "All nonempty subsets of a frequent itemset must also be frequent."
- Rule Generation – s $\rightarrow$ {l-s}
  - Frequent Item-sets Generation

The Apriori Algorithm (Pseudo-Code)

C_k: Candidate itemset of size k
L_k : frequent itemset of size k

L_1 = {frequent items};
for ($k = 1; L_k \mathrel{!=} \emptyset; k{+}{+}$) **do begin**
    C_{k+1} = candidates generated from L_k;
    for each transaction t in database do
        increment the count of all candidates in C_{k+1} that are contained in t
    L_{k+1} = candidates in C_{k+1} with min_support
    end
return $\cup_k L_k$;

The Join and Prune Trace -

- How to generate candidates?
  - Step 1: self-joining L_k
  - Step 2: pruning
- Example of Candidate-generation
  - 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 - Illustration

TID	*List of item_IDs*
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

Apriori - Illustration

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 count with minimum support count →

L_1

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

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 count with minimum support count →

L_2

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

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 count with minimum support count →

L_3

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

- Apriori property: All nonempty subsets of a frequent itemset must also be frequent.
- By definition, if an itemset I does not satisfy the minimum support threshold, min sup, then I is not frequent, that is, $P(I) <$ min sup. If an item A is added to the itemset I, then the resulting itemset (i.e., $I \cup A$) cannot occur more frequently than I. Therefore, $I \cup A$ is not frequent either, that is, $P(I \cup A) <$ min sup.
- special category of properties called antimonotonicity in the sense that if a set cannot pass a test, all of its supersets will fail the same test as well.
- antimonotonicity because the property is monotonic in the context of failing a test

- confidence(A ⇒ B) = P(B|A)
- = support count(A ∪B) / support count(A) For each frequent itemset l, generate all nonempty subsets of l.
- For every nonempty subset s of l, output the rule "s ⇒ (l − s)" if **support count(l) / support count(s) ≥ min conf**, where min conf is the minimum confidence threshold.
- X = {I1, I2, I5}
- nonempty subsets of X are {I1, I2}, {I1, I5}, {I2, I5}, {I1}, {I2}, and {I5}
- {I1, I2} ⇒ I5, confidence = 2/4 = 50%
- {I1, I5} ⇒ I2, confidence = 2/2 = 100%
- {I2, I5} ⇒ I1, confidence = 2/2 = 100%
- I1 ⇒ {I2, I5}, confidence = 2/6 = 33%
- I2 ⇒ {I1, I5}, confidence = 2/7 = 29%
- I5 ⇒ {I1, I2}, confidence = 2/2 = 100%

- For example, a frequent itemset of length 100,
- such as {a1, a2,..., a100}, contains
- 100 C1 = 100 frequent 1-itemsets: {a1}, {a2}, . . . , {a100};
- 100 C2 frequent 2-itemsets: {a1, a2}, {a1, a3}, . . . ,{a99, a100};

  and so on.
- The total number of frequent itemsets that it contains is thus

  $100C1 + 100C2 + \cdots + 100C100 = 2^{100} - 1 \approx 1.27 \times 10^{3}$

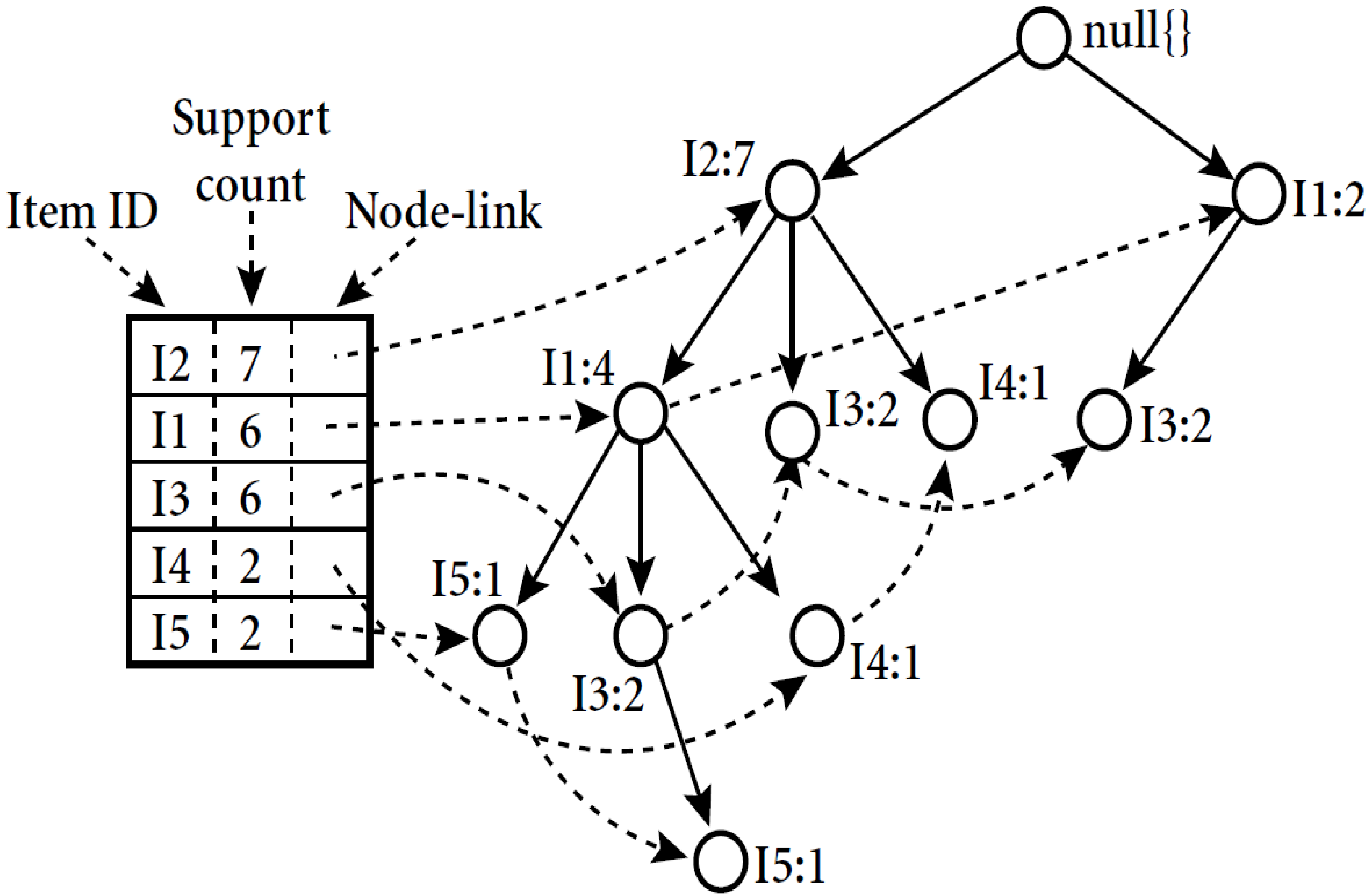
Worst-case time complexity still is exponential in |I| and linear in |D|*|I|, but usual behavior is linear in N=|D|. (detailed average-case analysis is strongly data dependent, thus difficult)

FPM – Literature

- Frequent Pattern (FP) Growth – Next Major contribution
- Improved on Apriori's Limitation – Repeated Scans of Original DB
- Overall Number of Scans – 2
- Reorders Transactions – suits MBA
- Dynamic Itemset Counting (DIC)
  - Reduced Number of Scans
  - Implication Rules – Interest and Conviction
  - Motwani et.al – Google founders

"Today, whenever you use a piece of technology, there is a good chance a little bit of Rajeev Motwani is behind it"

FP Growth - Illustration



Item	Conditional Pattern Base	Conditional FP-tree	Frequent Patterns Generated
I5	{{I2, I1: 1}, {I2, I1, I3: 1}}	⟨I2: 2, I1: 2⟩	{I2, I5: 2}, {I1, I5: 2}, {I2, I1, I5: 2}
I4	{{I2, I1: 1}, {I2: 1}}	⟨I2: 2⟩	{I2, I4: 2}
I3	{{I2, I1: 2}, {I2: 2}, {I1: 2}}	⟨I2: 4, I1: 2⟩, ⟨I1: 2⟩	{I2, I3: 4}, {I1, I3: 4}, {I2, I1, I3: 2}
I1	{{I2: 4}}	⟨I2: 4⟩	{I2, I1: 4}