Data Mining and Web Algorithms
Course Code: 15B22CI621
Credits: 4 [3+ 1]

Data Mining Techniques

- Descriptive methods
  - Association Mining
  - Clustering
- Predictive Methods
  - Classification/Regression

What is Association Rule Mining

- Association rule mining:
  - Finding frequent patterns, associations, correlations, or causal structures among sets of items or objects in transactional databases, relational databases, and other information repositories
- Motivation (market basket analysis):
  - If customers are buying milk, how likely is that they also buy bread?
  - Such rules help retailers to:
    - plan the shelf space: by placing milk close to bread they may increase the sales
    - provide advertisements/recommendation to customers that are likely to buy some products
    - put items that are likely to be bought together on discount, in order to increase the sales

Association Rules: Basic Concepts

- Given: (1) database of transactions, (2) each transaction is a list of items (purchased by a customer in a visit)
- Find: all rules that correlate the presence of one set of items with that of another set of items
  - E.g., *98% of people who purchase tires and auto accessories also get automotive services done*

Representation of Market Basket data: Format

Let D be database of transactions

- e.g.:

Transaction ID	Items Bought
2000	A,B,C
1000	A,C
4000	A,D
5000	B,E,F

OR

TID	A	B	C	D	E	F
2000	1	1	1	0	0	0
1000	1	0	1	0	0	0
4000	1	0	0	1	0	0
5000	0	1	0	0	1	1

Each item of a transaction is represented as a binary variable.

Components of a Rules

- In data mining, a set of items is referred to as an itemset
- Let D be database of transactions
  - e.g.:

Transaction ID	Items Bought
2000	A,B,C
1000	A,C
4000	A,D
5000	B,E,F

- Let I be the set of items that appear in the database, e.g., $I=\{A,B,C,D,E,F\}$
- A rule is defined by $X \Rightarrow Y$, where $X \subset I$, $Y \subset I$, and $X \cap Y = \emptyset$
  - e.g.: $\{B,C\} \Rightarrow \{E\}$ is a rule

Use of Association Rules

- Association rules do not represent any sort of causality or correlation between the two itemsets.
  - $X \Rightarrow Y$ does not mean X causes Y, so no Causality
  - $X \Rightarrow Y$ can be different from $Y \Rightarrow X$, unlike correlation

- Association rules assist in marketing, targeted advertising, floor planning, inventory control, churning management, homeland security, …

- There could be exponentially many A-rules.

Are all rules are interesting?

- The number of potential rules is huge. We may not be interested in all of them.
- We are interesting in rules that:
  - their items appear frequently in the database
  - they hold with a high probability
- We use the following thresholds:
  - the *support* of a rule indicates how frequently its items appear in the database
  - the *confidence* of a rule indicates the probability that if the left hand side appears in a T, also the right hand side will.

❑Interesting association rules are (for now) those whose Support and Confidence are greater than minSup and minConf (some thresholds set by data miners)

Support & Confidence of a Rule

Find all the rules $X \Rightarrow Y$ with minimum confidence and support

- support, *s*, probability that a transaction contains $\{X \cup Y\}$
- confidence, *c*, conditional probability that a transaction having X also contains Y

Given a dataset *D*, an itemset *X* has a (frequency) *count* in *D*

support of *X* in *D* is *count*(*X*)/|*D*|

For an association rule $X \Rightarrow Y$, we can calculate

support ($X \Rightarrow Y$) = support (XY)

confidence ($X \Rightarrow Y$) = support (XY)/support (X)

Support & Confidence of a Rule

TID	date	items_bought
100	10/10/99	{F,A,D,B}
200	15/10/99	{D,A,C,E,B}
300	19/10/99	{C,A,B,E}
400	20/10/99	{B,A,D}

- What is the support and confidence of the rule: $\{B,D\} \Rightarrow \{A\}$

- support, s, probability that a transaction contains $\{X \cup Y\}$
- confidence, c, conditional probability that a transaction having X also contains Y

Remember:

$$conf(X \Rightarrow Y) = \frac{sup(X \cup Y)}{sup(X)}$$

Example :Support & Confidence

TID	date	items_bought
100	10/10/99	{F,A,D,B}
200	15/10/99	{D,A,C,E,B}
300	19/10/99	{C,A,B,E}
400	20/10/99	{B,A,D}

Remember:
$$conf(X \Rightarrow Y) = \frac{sup(X \cup Y)}{sup(X)}$$

- What is the support and confidence of the rule: $\{B,D\} \Rightarrow \{A\}$
- Support:
  - percentage of tuples that contain {A,B,D} = 75%
- Confidence:

$$\frac{\text{number of tuples that contain } \{A,B,D\}}{\text{number of tuples that contain } \{B,D\}} = 100\%$$

Itemsets and Association Rules

- An *itemset* is a set of items.
  - E.g., {X,Y,Z} is an itemset.
- A *k-itemset* is an itemset with k items.

- An *association rule* is about relationships between two disjoint itemsets X and Y

$$X \Rightarrow Y$$

- It presents the pattern when X occurs, Y also occurs

Steps in association rule mining

- Major steps in association rule mining
  - Frequent itemsets generation
  - Rule derivation
- Use of support(S) and confidence(C) in association mining
  - S for frequent itemsets
  - C for rule derivation

Given a set of transactions T, the goal of association rule mining is to find all rules having

- support ≥ *minsup* threshold
- confidence ≥ *minconf* threshold

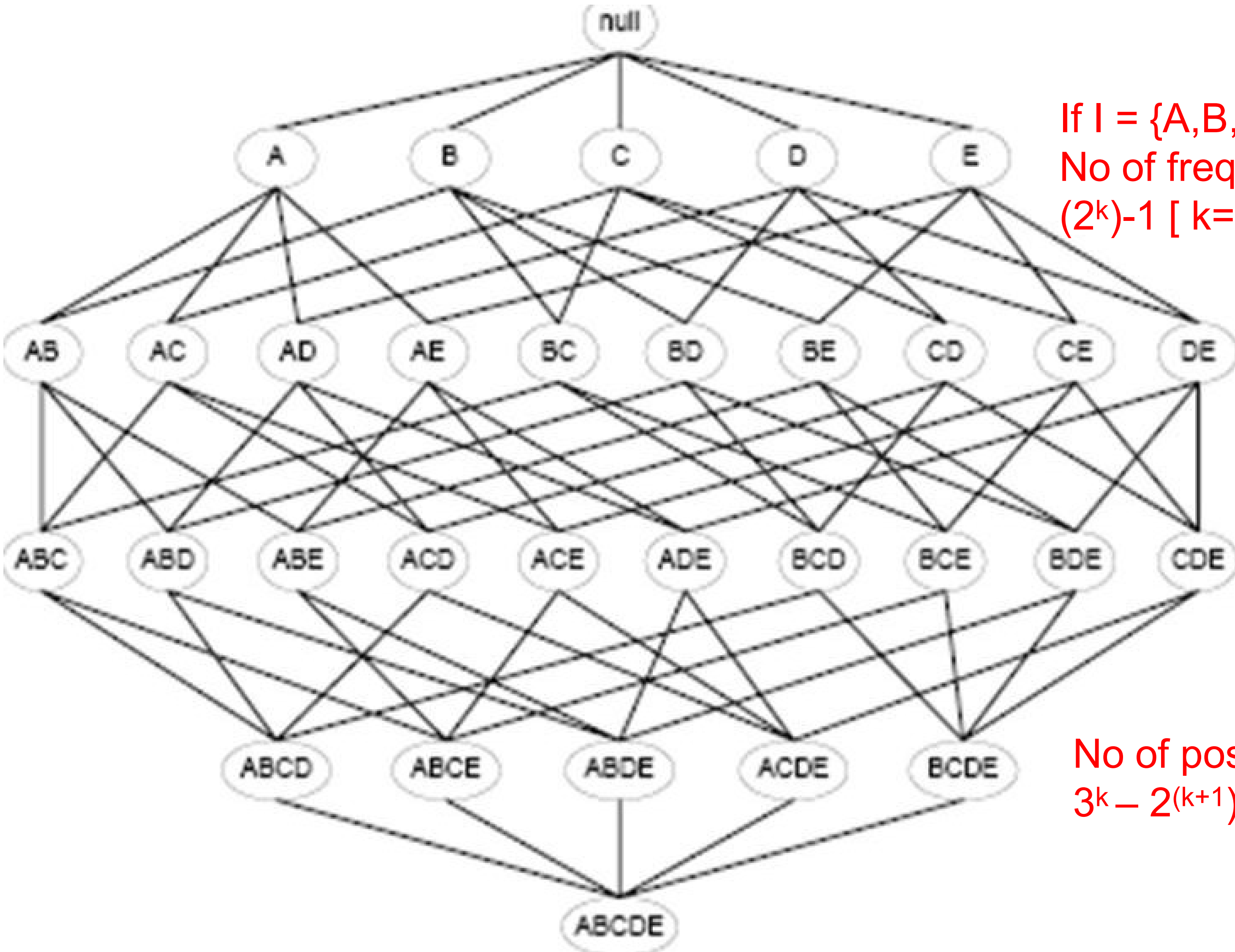
Association Mining Approaches

- 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

  ⇒ Computationally prohibitive!

- **Apriori Approach**
- FP- Growth Approach
- Many More...

Brute Force Approach



If I = {A,B,C,D,E}
No of frequent itemsets = $(2^k)-1$ [k= no. of items]

No of possible rules= $3^k - 2^{(k+1)} + 1$

Figure Source: [2]

The Apriori Algorithm: Basics

The Apriori Algorithm is an influential algorithm for mining frequent itemsets for boolean association rules.

Key Concepts :

- Frequent Itemsets: The sets of item which has minimum support (denoted by L_i for i^{th}-Itemset).
- Apriori Property: Any subset of frequent itemset must be frequent.
- Join Operation: To find L_k , a set of candidate k-itemsets is generated by joining L_{k-1} with itself.

The Apriori Algorithm in a Nutshell

Step (a): Find the *frequent itemsets*: the sets of items that have minimum support

- A subset of a frequent itemset must also be a frequent itemset
  - i.e., if {*AB*} is a frequent itemset, both {*A*} and {*B*} should be a frequent itemset
- Iteratively find frequent itemsets with cardinality from 1 to *k* (*k*-itemset)

Step (b): Use the frequent itemsets to generate association rules.

The Apriori Algorithm -- Example

minSup =2

Database D

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

Scan D

C_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

L_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{5}	3

C_2

itemset
{1 2}
{1 3}
{1 5}
{2 3}
{2 5}
{3 5}

Scan D

C_2

itemset	sup
{1 2}	1
{1 3}	2
{1 5}	1
{2 3}	2
{2 5}	3
{3 5}	2

L_2

itemset	sup
{1 3}	2
{2 3}	2
{2 5}	3
{3 5}	2

C_3

itemset
{2 3 5}

Scan D

L_3

itemset	sup
{2 3 5}	2

Note: {1,2,3}{1,2,5} and {1,3,5} not in C_3

Figure Source: [5]

The Apriori Algorithm: Example

TID	List of Items
T100	I1, I2, I5
T101	I2, I4
T102	I2, I3
T103	I1, I2, I4
T104	I1, I3
T105	I2, I3
T106	I1, I3
T107	I1, I2 ,I3, I5
T108	I1, I2, I3

- Consider a database, D , consisting of 9 transactions.
- Suppose min. support count required is 2 (i.e. min_sup = 2/9 = 22 %)
- Let minimum confidence required is 70%.
- We have to first find out the frequent itemset using Apriori algorithm.
- Then, Association rules will be generated using min. support & min. confidence.

(a) Step 1: Generating 1-itemset Frequent Pattern

Scan D for count of each candidate →

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

C_1

Compare candidate support count with minimum support count →

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

L_1

- In the first iteration of the algorithm, each item is a member of the set of candidate.
- The set of frequent 1-itemsets, L_1 , consists of the candidate 1- itemsets satisfying minimum support.

(a) Step 2: Generating 2-itemset Frequent Pattern

Generate C_2 candidates from L_1

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

C_2

Scan D for count of each candidate

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

C_2

Compare candidate support count with minimum support count

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

L_2

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

(a) Step 2: Generating 2-itemset Frequent Pattern [Cont.]

- To discover the set of frequent 2-itemsets, L_2 , the algorithm uses L_1 *Join* L_1 to generate a candidate set of 2- itemsets, C_2.
- Next, the transactions in D are scanned and the support count for each candidate itemset in C_2 is accumulated (as shown in the middle table).
- The set of frequent 2-itemsets, L_2 , is then determined, consisting of those candidate 2-itemsets in C_2 having minimum support.
- Note: We haven't used Apriori Property yet.

(a) Step 3: Generating 3-itemset Frequent Pattern

- **Join step** : In order to find C_3, we compute L_2 *Join* L_2.
- C_3 = L2 *Join* L2 = {{I1, I2, I3}, {I1, I2, I5}, {I1, I3, I5}, {I2,I3, I4}, {I2, I3, I5}, {I2, I4, I5}}.

  Now, Join step is complete.

- **Prune step** will be used to reduce the size of C_3. Prune step helps to avoid heavy computation due to large C_k.

- The generation of the set of candidate 3-itemsets, C_3 , involves **use of the Apriori Property.**

L_2

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

TID	**List of Items**
T100	I1, I2, I5
T101	I2, I4
T102	I2, I3
T103	I1, I2, I4
T104	I1, I3
T105	I2, I3
T106	I1, I3
T107	I1, I2 ,I3, I5
T108	I1, I2, I3

(a) Step 3: Generating 3-itemset Frequent Pattern [Cont.]

- Based on the Apriori property that all subsets of a frequent itemset must also be frequent, we can determine that four latter candidates cannot possibly be frequent. How ?
- For example , lets take {I1, I2, I3}. The 2-item subsets of it are {I1, I2}, {I1, I3} & {I2, I3}. Since all 2-item subsets of {I1, I2, I3} are members of L_2, We will keep {I1, I2, I3} in C_3.
- Lets take another example of {I2, I3, I5} which shows how the pruning is performed. The 2-item subsets are {I2, I3}, {I2, I5} & {I3,I5}.
- BUT, {I3, I5} is not a member of L_2 and hence it is not frequent violating Apriori Property. Thus We will have to remove {I2, I3, I5} from C_3.

- Therefore, C_3 = {{I1, I2, I3}, {I1, I2, I5}} after checking for all members of result of Join operation for Pruning.
- Now, the transactions in D are scanned in order to determine L_3, consisting of those candidates 3-itemsets in C_3 having minimum support.

Step 3: Generating 3-itemset Frequent Pattern

Scan D for count of each candidate

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

C_3

Scan D for count of each candidate

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

C_3

Compare candidate support count with min support count

Itemset	Sup Count
{I1, I2, I3}	2
{I1, I2, I5}	2

L_3

(a) Step 4: Generating 4-itemset Frequent Pattern

- The algorithm uses L_3 *Join* L_3 to generate a candidate set of 4-itemsets, C_4. Although the join results in {{I1, I2, I3, I5}}, this itemset is pruned since its subset {{I2, I3, I5}} is not frequent.
- Thus, $C_4 = \varphi$, and algorithm terminates, having found all of the frequent items. This completes our Apriori Algorithm.

What's Next ?

Use frequent itemsets generate strong association rules

Compliance to both minimum support & minimum confidence).

(b)Step 5: Generating Association Rules from Frequent Itemsets

- Procedure:
  - 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 minimum confidence threshold.

- Back To Example:

  We had L = {{I1}, {I2}, {I3}, {I4}, {I5}, {I1,I2}, {I1,I3}, {I1,I5}, {I2,I3}, {I2,I4},
  {I2,I5}, {I1,I2,I3}, {I1,I2,I5}}.
  - Lets take ***l*** = {I1,I2,I5}.
  - Its all nonempty subsets are {I1,I2}, {I1,I5}, {I2,I5}, {I1}, {I2}, {I5}.

(b)Step 5: Generating Association Rules from Frequent Itemsets [Cont.]

- Let minimum confidence threshold is , say 70%.
- The resulting association rules are shown below, each listed with its confidence.
  - R1: I1 ^ I2 → I5
    - Confidence = sc{I1,I2,I5}/sc{I1,I2} = 2/4 = 50%
    - R1 is Rejected.
  - R2: I1 ^ I5 → I2
    - Confidence = sc{I1,I2,I5}/sc{I1,I5} = 2/2 = 100%
    - R2 is Selected.
  - R3: I2 ^ I5 → I1
    - Confidence = sc{I1,I2,I5}/sc{I2,I5} = 2/2 = 100%
    - R3 is Selected.

TID	List of Items
T100	I1, I2, I5
T101	I2, I4
T102	I2, I3
T103	I1, I2, I4
T104	I1, I3
T105	I2, I3
T106	I1, I3
T107	I1, I2 ,I3, I5
T108	I1, I2, I3

(b)Step 5: Generating Association Rules from Frequent Itemsets [Cont.]

- R4: I1 → I2 ^ I5
  - Confidence = sc{I1,I2,I5}/sc{I1} = 2/6 = 33%
  - R4 is Rejected.
- R5: I2 → I1 ^ I5
  - Confidence = sc{I1,I2,I5}/{I2} = 2/7 = 29%
  - R5 is Rejected.
- R6: I5 → I1 ^ I2
  - Confidence = sc{I1,I2,I5}/ {I5} = 2/2 = 100%
  - R6 is Selected.

  In this way, We have found three strong association rules.

TID	List of Items
T100	I1, I2, I5
T101	I2, I4
T102	I2, I3
T103	I1, I2, I4
T104	I1, I3
T105	I2, I3
T106	I1, I3
T107	I1, I2 ,I3, I5
T108	I1, I2, I3

Review of Example

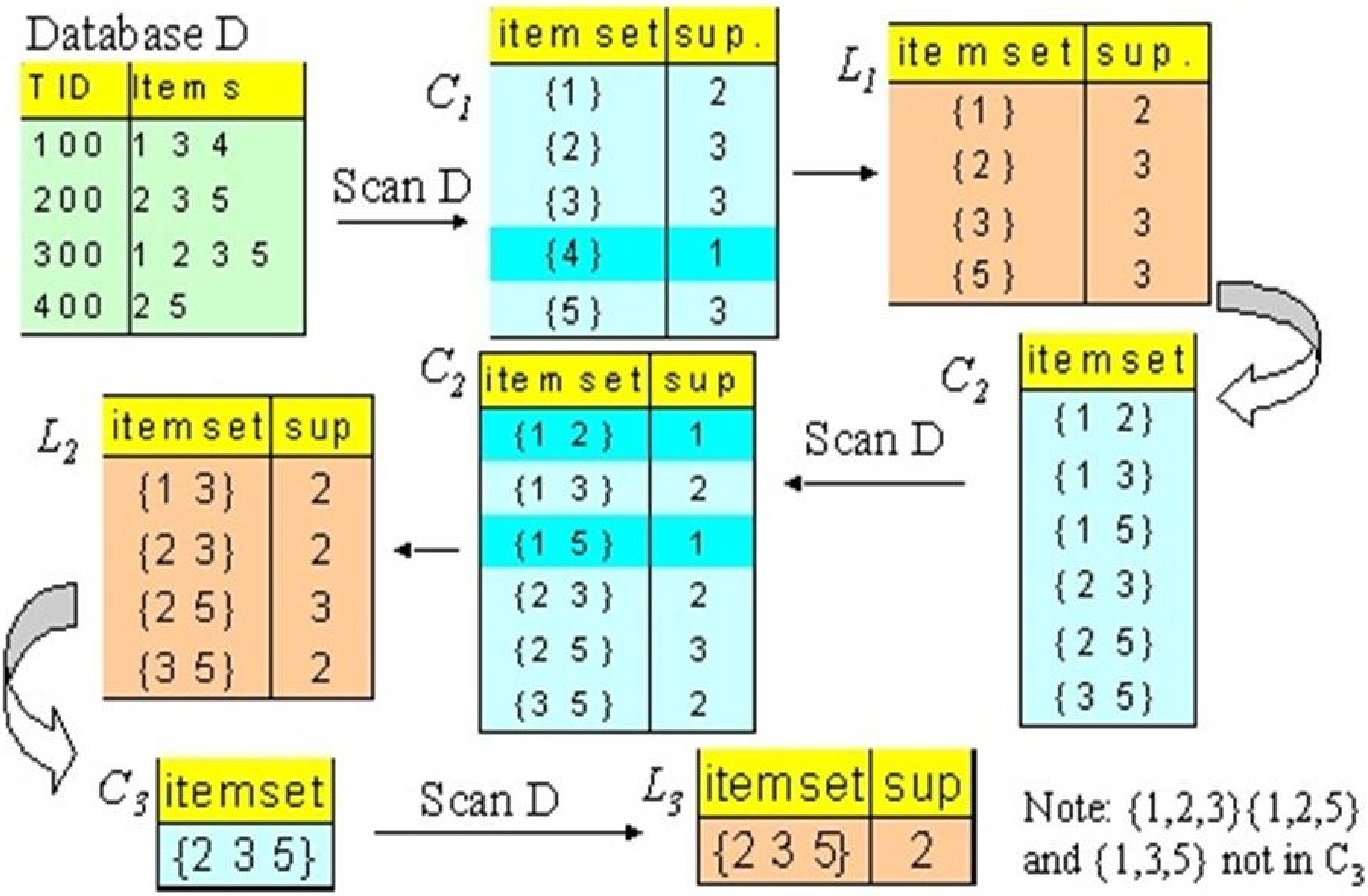


Figure Source: [5]

Frequent itemset generation(pseudocode)

Join Step: C_k is generated by joining L_{k-1} with itself

Prune Step: Any (k-1)-itemset that is not frequent cannot be a subset of a frequent k-itemset

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 !=∅; 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 ∪_k L_k;
```

Source: [5]

Problems with the association mining

- Single minsup: It assumes that all items in the data are of the same nature and/or have similar frequencies.

- Not true: In many applications, some items appear very frequently in the data, while others rarely appear.

  E.g., in a supermarket, people buy *food processor* and *cooking pan* much less frequently than they buy *bread* and *milk*.

Rare Item Problem

- If the frequencies of items vary a great deal, we will encounter two problems
  - If minsup is set too high, those rules that involve rare items will not be found.
  - To find rules that involve both frequent and rare items, minsup has to be set very low. This may cause combinatorial explosion because those frequent items will be associated with one another in all possible ways.

Multiple minsups model

- The minimum support of a rule is expressed in terms of *minimum item supports* (MIS) of the items that appear in the rule.
- Each item can have a minimum item support.
- By providing different MIS values for different items, the user effectively expresses different support requirements for different rules.

Minsup of a rule

- Let MIS(*i*) be the MIS value of item *i*. The *minsup* of a rule *R* is the **lowest** MIS value of the items in the rule.
- I.e., a rule *R*: $a_1, a_2, \ldots, a_k \rightarrow a_{k+1}, \ldots, a_r$ satisfies its minimum support if its actual support is $\geq$

  $$\min(\mathrm{MIS}(a_1), \mathrm{MIS}(a_2), \ldots, \mathrm{MIS}(a_r)).$$

An Example

- Consider the following items:

  bread, *shoes*, *clothes*

  The user-specified MIS values are as follows:

  MIS(*bread*) = 2%   MIS(*shoes*) = 0.1%

  MIS(*clothes*) = 0.2%

  The following rule doesn't satisfy its minsup:

  clothes $\rightarrow$ *bread* [sup=0.15%,conf =70%]

  The following rule satisfies its minsup:

  clothes $\rightarrow$ *shoes* [sup=0.15%,conf =70%]

Features of Apriori Algorithm

- Association rules are generated from frequent itemsets.
- Frequent itemsets are mined using Apriori algorithm.
- Apriori property states that all the subsets of frequent itemsets must also be frequent.
- Apriori algorithm uses frequent itemsets, join & prune methods and Apriori property to derive strong association rules.

- It uses breadth first search for candidate set generation.
- Database is scanned multiple times to get support count and candidate set generation.
  - Can we reduce this multiple times????

Yes.....Alternative Methods for Frequent Itemset Generation

References

1. Han, Jiawei, Jian Pei, and Micheline Kamber. *Data mining: concepts and techniques*. Elsevier, 2012(Third Edition).
2. Tan, Pang-Ning, Michael Steinbach, and Vipin Kumar. Introduction to data
3. mining. Pearson Education India, 2016.
4. . Dunham, Margaret H. Data mining: Introductory and advanced topics. Pearson Education India, 2006.
5. https://www.javatpoint.com/data-mining-architecture
6. https://www.sites.google.com/site/getallcodesyouwant/data-mining/apriori- algorithm
7. http://www.cs.sunysb.edu/~cse634/lecture_notes/07apriori.pdf
8. Nasreen, Shamila, et al. "Frequent pattern mining algorithms for finding associated frequent patterns for data streams: A survey." *Procedia Computer Science* 37 (2014): 109-116.
9. Borgelt, Christian. "Efficient implementations of apriori and eclat." *FIMI'03: Proceedings of the IEEE ICDM workshop on frequent itemset mining implementations*. 2003.