



Data Mining

Mining Frequent Patterns, Association and Correlations: Basic Concepts and Methods

Table 6.1. An example of market basket transactions.

Association Analysis

<i>TID</i>	Items
1	{Bread, Milk}
2	{Bread, Diapers, Beer, Eggs}
3	{Milk, Diapers, Beer, Cola}
4	{Bread, Milk, Diapers, Beer}
5	{Bread, Milk, Diapers, Cola}

- Many business enterprises generate large quantities of data from their daily operations.
 - Example: Customer purchase data are collected daily at the checkout counters of grocery stores.
- Each row corresponds to a transaction, which contains a unique identifier labeled **TID** and a **set of items**.
- Retailers are interested in analyzing the data to learn about the **purchasing behavior of their customers**.
- Such valuable information can be used to support a variety of business-related applications such as **marketing promotions, inventory management, catalog design, store layout** and **customer relationship management**.

Association Analysis

- Useful for discovering interesting relationships hidden in large data sets.
- The uncovered relationships can be represented in the form of association rules or sets of frequent items

{Diapers}  {Beer}

- The rule suggests, customers who buy diapers also buy beer.
- Retailers can use this type of rules to help them identify new opportunities for cross-selling their products to the customers.

Key Issues

- Two key issues when applying association analysis to market basket data.
 - Discovering patterns from a large transaction data set can be computationally expensive.
 - Some of the discovered patterns are potentially spurious because they may happen simply by chance.

Table 6.2. A binary 0/1 representation of market basket data.

Basic Terminology

TID	Bread	Milk	Diapers	Beer	Eggs	Cola
1	1	1	0	0	0	0
2	1	0	1	1	1	0
3	0	1	1	1	0	1
4	1	1	1	1	0	0
5	1	1	1	0	0	1

- Each row corresponds to a transaction and each column corresponds to an item.
- An item can be treated as a binary variable whose value is one if the item is present in a transaction and zero otherwise.
- Because the presence of an item in a transaction is often considered more important than its absence, an item is an **asymmetric binary variable**.
- It **ignores** certain important aspects of the data such as the **quantity of items** sold or the **price** paid to purchase them.

Itemset and Support Count

- Let $I = \{i_1, i_2, \dots, i_d\}$ be the set of all items in a market basket data and $T = \{t_1, t_2, \dots, t_N\}$ be the set of all transactions.
- Each transaction t_i contains a subset of items chosen from I .
- In association analysis, a collection of zero or more items is termed an itemset.
- If an itemset contains k items, it is called a k -itemset.
- $\{\text{Beer, Diapers, Milk}\}$ is an example of a 3-itemset.
- The null (or empty) set is an itemset that does not contain any items.
- Support count refers to the number of transactions that contain a particular itemset.
- Support count for $\{\text{Beer, Diapers, Milk}\}$ is equal to two because there are only two transactions that contain all three items
- The transaction width is defined as the number of items present in a transaction.

Association Rule

$$\text{Support, } s(X \longrightarrow Y) = \frac{\sigma(X \cup Y)}{N};$$

$$\text{Confidence, } c(X \longrightarrow Y) = \frac{\sigma(X \cup Y)}{\sigma(X)}.$$

- An association rule is an implication expression of the form $X \rightarrow Y$, where X and Y are disjoint itemsets, i.e., $X \cap Y = \emptyset$.
- The strength of an association rule can be measured in terms of its **support and confidence**.
- Support determines how often a rule is applicable to a given data set
- Confidence determines how frequently items in Y appear in transactions that contain X .

$\{\text{Milk, Diapers}\} \rightarrow \{\text{Beer}\}$

Why Use Support and Confidence?

- A low support rule is uninteresting from a business perspective because it may not be profitable to promote.
- Support is often used to eliminate uninteresting rules
- Confidence measures the reliability of the inference made by a rule
- For a given rule $X \rightarrow Y$, the higher the confidence, the more likely it is for Y to be present in transactions that contain X.
- Confidence also provides an estimate of the conditional probability of Y given X.
- It suggests a strong co-occurrence relationship between items in the antecedent and consequent of the rule.

Formulation of Association Rule Mining Problem

- Definition 6.1 (Association Rule Discovery):

Given a set of transactions T , find all the rules having support $\geq \text{min_sup}$ and confidence $\geq \text{min_conf}$, where min_sup and min_conf are the corresponding support and confidence thresholds.

Brute-force approach

- A brute-force approach for mining association rules is to compute the support and confidence for every possible rule.
- This approach is prohibitively expensive because there are exponentially many rules that can be extracted from a data set.
- Total number of possible rules extracted from a data set that contains d items is

$$R = 3^d - 2^{d+1} + 1$$

- Making most of the computations become wasted
- To avoid performing needless computations, it would be useful to prune the rules early without compute their support and confidence values.

- The support of a rule $X \rightarrow Y$ depends only on the support of its corresponding itemset, $X \cup Y$.
- For example, the following rules have identical support because they involve items from the same itemset, {Beer, Diapers, Milk}:
 - If the itemset is infrequent, then all six candidate rules can be pruned immediately without our having to compute their confidence values

Association rule mining algorithms

- A common strategy adopted by many association rule mining algorithms is to decompose the problem into two major subtasks:
- **1. Frequent Item set Generation** whose objective is to find all the item sets that satisfy the min_sup threshold. These item sets are called frequent item sets.
- **2. Rule Generation** whose objective is to extract all the high-confidence rules from the frequent itemsets found in the previous step. These rules are called strong rules.

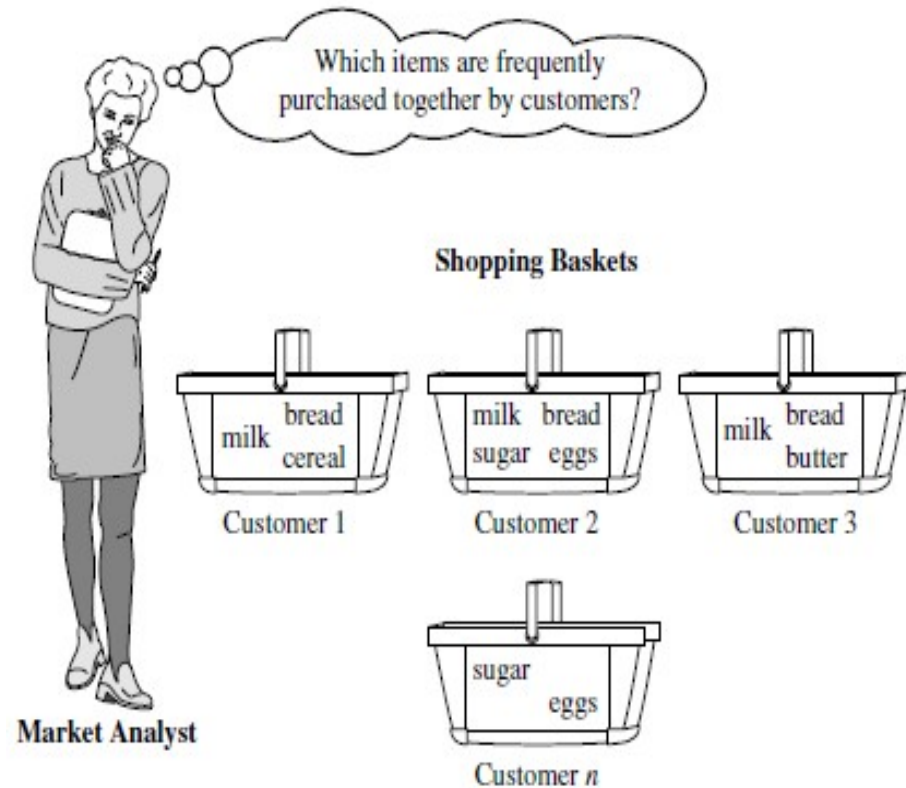
Basic Concepts

- Frequent patterns and Association rules are helpful for making recommendations in business
- Frequent patterns are itemsets, subsequences or substructures that appear frequently in a data set
 - **Frequent itemset** - appear frequently in a transaction dataset (Milk and Bread)
 - **Subsequence** - Buying first PC, then camera, then a memory card, if it occurs frequently in a shopping history DB is a **sequential pattern**
 - **Substructure** - refer to different structural forms like subgraphs, subtrees or sublattices which may be combined with itemsets or subsequences.
- Finding frequent patterns plays an essential role in mining associations and correlation relationships among data stored in transactional and relational data.
- Helps in data classification, clustering and other DM tasks

Market Basket Analysis(MBA)

- Discovery of interesting correlation and association relationships can help in many business **decision making process**

- Helpful for selective marketing and plan for shelf space
- Advertisement strategies or design the new catalogue
- Design of different store layouts
- To plan which items to put on sale at reduced



Cont.,

- Set of items available at store, each item has a Boolean variable, each basket can be represented by a Boolean vector of values assigned to items
- The Boolean vector can be analysed for buying patterns that are frequently ***associated*** or purchased together.
computer $\Rightarrow$ antivirus_software [support = 2%, confidence = 60%]
- Customers who purchase computer also tend to buy antivirus software at the same time
- **Support and Confidence** are two measures of rule interestingness

Cont.,

computer $\Rightarrow$ *antivirus_software*[*sup port* = 2%, *confidence* = 60%]

- Support of 2% means that 2% of all the transactions under analysis show that computer and AV purchased together
- Confidence of 60% means that 60% of customers who purchased a computer also bought the AV
- Association rules are considered interesting if they satisfy both **minimum support threshold** and **minimum confidence threshold**
- Additional analysis can be performed to discover interesting statistical correlations between associated items

Frequent Itemsets, Closed Itemsets and Association Rules

- I is a itemset $I = \{I_1, I_2, \dots, I_m\}$ // set of items – m itemset
- D is task-relevant transactional database where each transaction T is a nonempty itemset $T \subseteq I$
- Let A be a set of items
- A transaction T is said to contain A if $A \subseteq T$
- An association rule is an implication of the form $A \Rightarrow B$
- The rule $A \Rightarrow B$ holds in the transaction set D with **support** s , where s is the percentage of transactions in D that contain $A \cup B$
- This is taken to be the probability $P(A \cup B)$
- The rule $A \Rightarrow B$ holds in the transaction set D with **confidence** c , where c is the percentage of transactions in D containing A that also contain B
- This is taken to be the conditional probability $P(B|A)$

$$\text{sup port}(A \Rightarrow B) = P(A \cup B)$$

$$\text{confidence}(A \Rightarrow B) = P(B|A)$$

- The set {computer, antivirus} is a 2 itemset
- The occurrence frequency of an itemset is the number of transactions that contain the itemset.
- This is also known as **frequency**, **support count** or **count**

$$\text{Confidence}(A \Rightarrow B) = P(B|A) = \frac{\text{sup port}(A \cup B)}{\text{sup port}(A)} = \frac{\text{sup port_count}(A \cup B)}{\text{sup port_count}(A)}$$

- Association rule mining can be viewed as two step process
- i. Find all frequent itemsets – that has to support ***min_sup***
- ii. Generate strong association rules from the frequent itemsets – must satisfy ***min_sup and min_conf***

Challenge:

- Large dataset often generates a huge number of frequent itemsets satisfying min_sup. (when threshold is low)
- A long itemset will contain a combinatorial number of shorter, frequent sub-itemsets
- Ex: A frequent itemset of length 100 $\{a_1, a_2, \dots, a_{100}\}$ contains
 - $\binom{100}{1} = 100$ frequent 1- itemsets: $\{a_1\}, \{a_2\}, \dots, \{a_{100}\}$
 - $\binom{100}{2}$ frequent itemsets: $\{a_1, a_2\}, \{a_1, a_3\}, \dots, \{a_{99}, a_{100}\}$
- Total number of frequent itemsets is
- $2^{100} - 1 = 1.27 \times 10^{30}$ this is huge number to compute or store

Closed frequent itemsets and Maximal frequent itemsets

- An itemset X is **closed** in a dataset D if there exists no proper super-itemset Y s.t Y has the same support count as X in D
- An itemset X is a **closed frequent itemset** in set D if X is both closed and frequent in D
- An itemset X is a **maximal frequent itemset** in a dataset D if X is frequent, and there exists no super-itemset Y s.t $X \subset Y$ and Y is frequent in D
- Let C be the set of closed frequent itemsets for a dataset D satisfying a **min_sup_threshold**
- Let M be the set of maximal frequent itemsets for a dataset D satisfying **min_sup**

Example

- Transaction DB has only two transactions: $\{ \langle a_1, a_2, \dots, a_{100} \rangle; \langle a_1, a_2, \dots, a_{50} \rangle \}$.
- Let the $min_sup = 1$.
- We find two closed frequent itemsets and their support counts, that is, $C = \{ \{a_1, a_2, \dots, a_{100}\} : 1; \{a_1, a_2, \dots, a_{50}\} : 2 \}$.
- There is only one maximal frequent itemset: $M = \{ \{a_1, a_2, \dots, a_{100}\} : 1 \}$.
- **Note:** we cannot include $\{a_1, a_2, \dots, a_{50}\}$ as a maximal frequent itemset because it has a frequent superset, $\{a_1, a_2, \dots, a_{100}\}$.
- Compare this to the preceding where we determined that there are $2^{100} - 1$ frequent itemsets, which are too many to be enumerated!
- The set of closed frequent itemsets contains complete information regarding the frequent itemsets.
- For example, from C , we can derive, say,
 - (1) $\{a_2, a_{45} : 2\}$ since $\{a_2, a_{45}\}$ is a sub-itemset of the itemset $\{a_1, a_2, \dots, a_{50} : 2\}$
 - (2) $\{a_8, a_{55} : 1\}$ since $\{a_8, a_{55}\}$ is not a sub-itemset of the previous itemset but of the itemset $\{a_1, a_2, \dots, a_{100} : 1\}$.
- However, from the maximal frequent itemset, we can only assert that both itemsets ($\{a_2, a_{45}\}$ and $\{a_8, a_{55}\}$) are frequent, but we cannot assert their actual support counts.

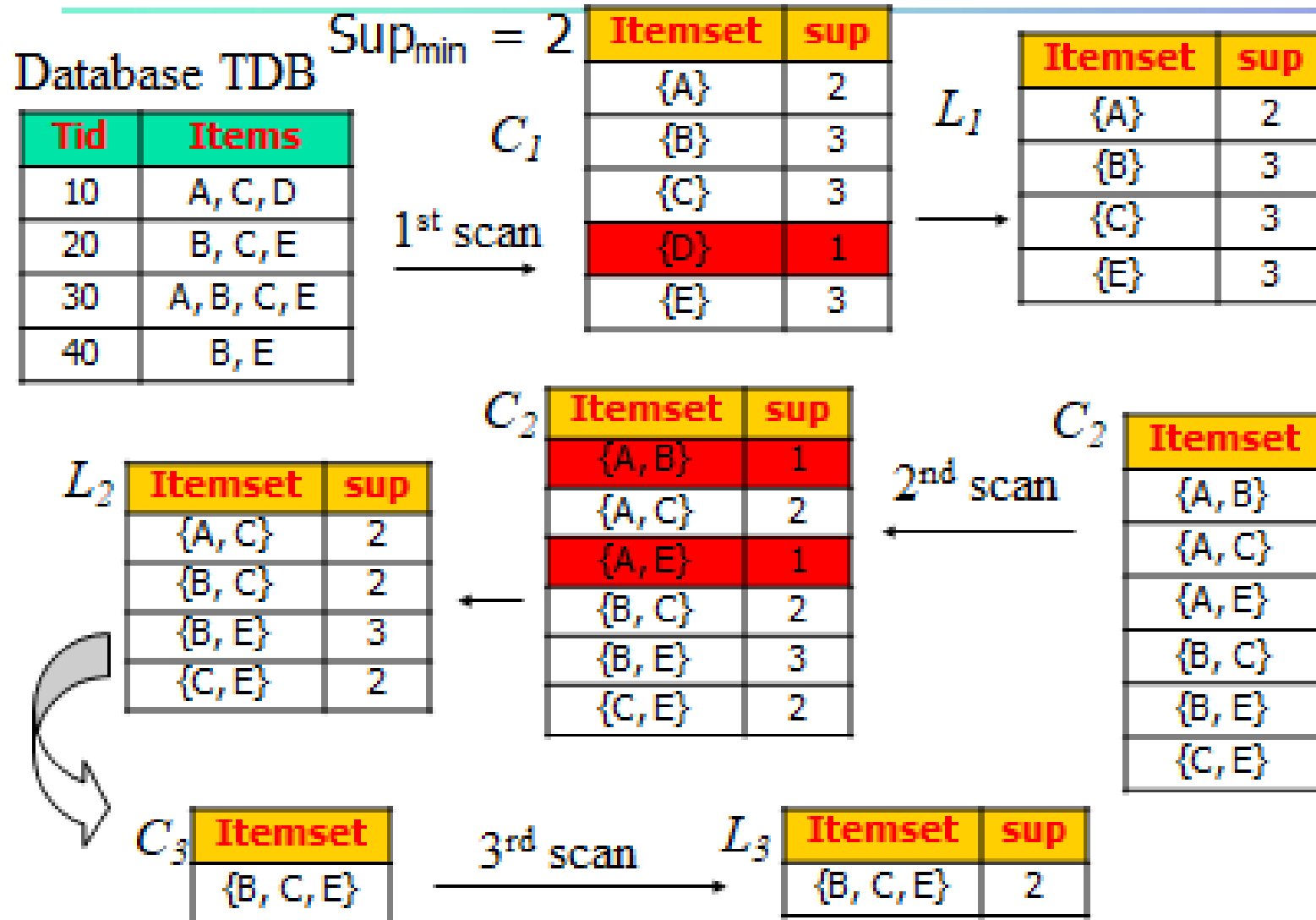
Frequent Itemset Mining Methods

- **Apriori Algorithm** : mining frequent itemsets for Boolean association rules
- Apriori is a candidate generation and test approach
- Name of the algorithm is based on ***Prior Knowledge*** of frequent itemset properties
- It is an **iterative approach** known as ***level-wise*** search, where k-itemsets are used to explore (k+1)-itemsets.

- First, the set of frequent 1-itemsets is found by scanning the database to accumulate the count for each item, and collecting those items that satisfy minimum support. The resulting set is denoted by $L1$.
- Next, $L1$ is used to find $L2$, the set of frequent 2-itemsets, which is used to find $L3$, and so on, until no more frequent k -itemsets can be found. The finding of each Lk requires one full scan of the database.

- **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, $P(I \cup A) < min_sup$.
- This property belongs to a 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.*
- It is called *antimonotonicity* because the property is monotonic in the context of failing a test.

The Apriori Algorithm—An Example



Apriori Algorithm (Pseudo Code)

C_k : Candidate itemset of size k

L_k : frequent itemset of size k

$L_1 = \{\text{frequent items}\};$

for ($k = 1; L_k \neq \emptyset; k++$) **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

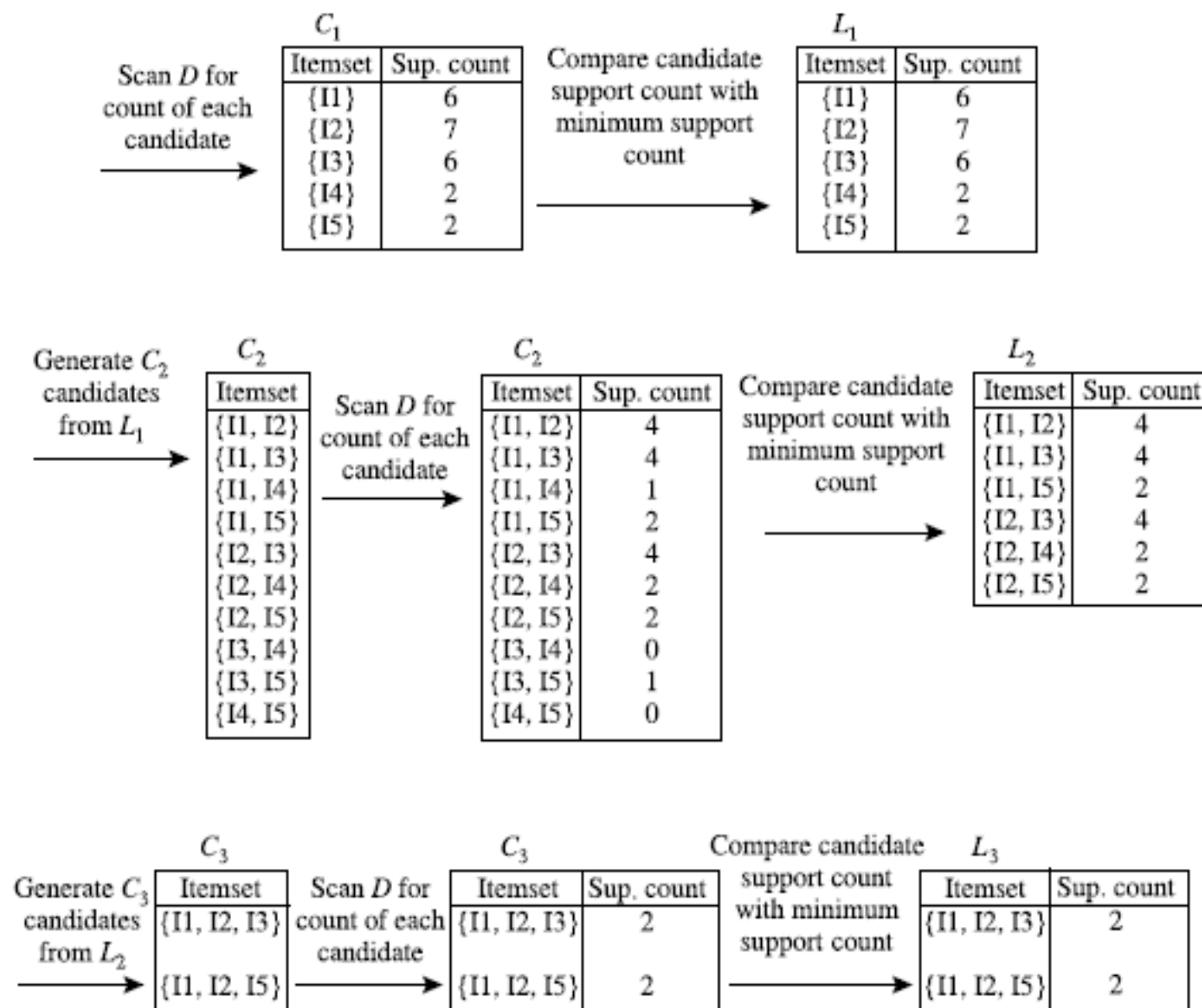
end

return $\bigcup_k L_k$;

Apriori Example

Transactional Data for an *AllElectronics* Branch

<i>TID</i>	<i>List of item_IDs</i>
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



! Generation of the candidate itemsets and frequent itemsets, where the minimum support count is 2.

Generating Association Rules from Frequent Itemsets

- **strong** association rules satisfy both minimum support and minimum

Generating association rules. Let's try an example based on the transactional data for *AllElectronics* shown before in Table 6.1. The data contain frequent itemset $X = \{I1, I2, I5\}$. What are the association rules that can be generated from X ? The nonempty subsets of X are $\{I1, I2\}$, $\{I1, I5\}$, $\{I2, I5\}$, $\{I1\}$, $\{I2\}$, and $\{I5\}$. The resulting association rules are as shown below, each listed with its confidence:

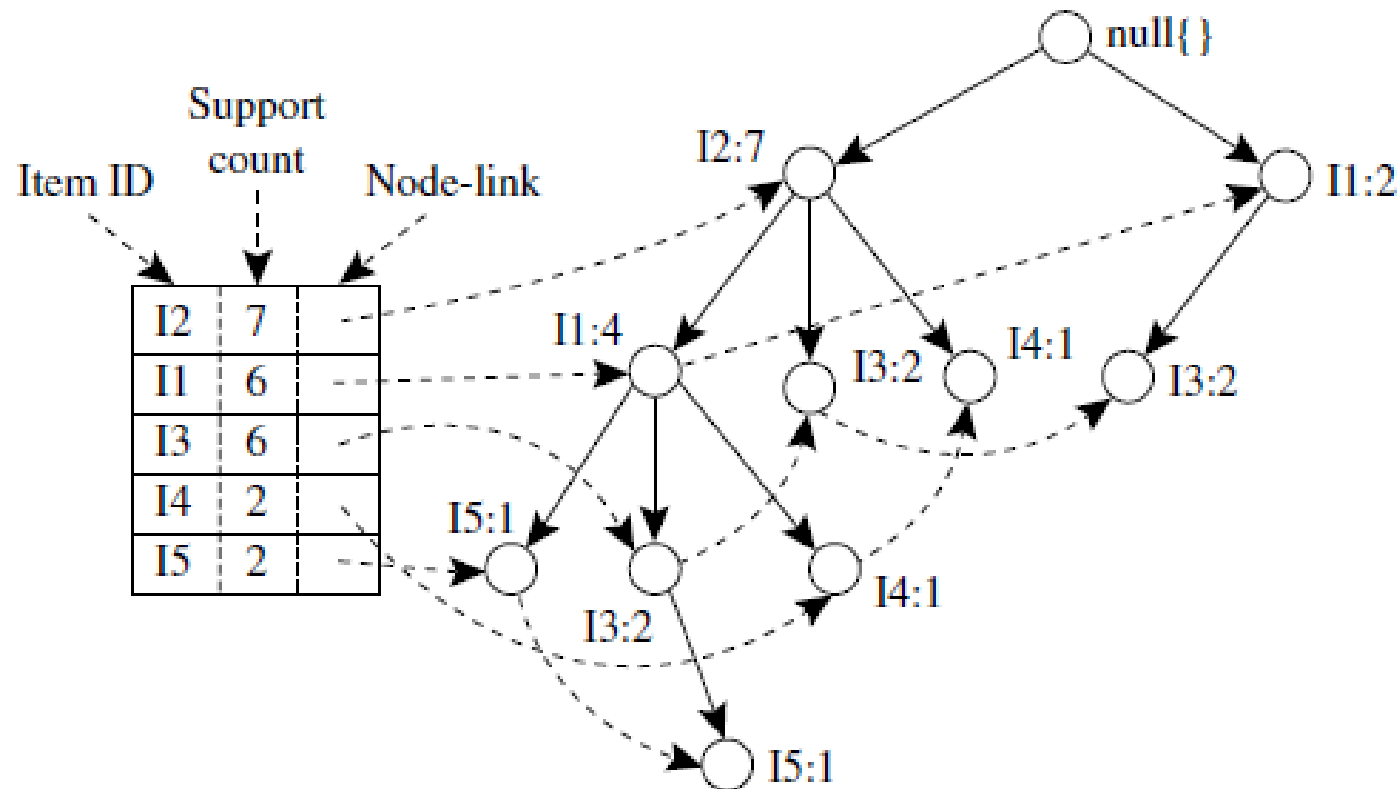
$\{I1, I2\} \Rightarrow I5$,	$confidence = 2/4 = 50\%$
$\{I1, I5\} \Rightarrow I2$,	$confidence = 2/2 = 100\%$
$\{I2, I5\} \Rightarrow I1$,	$confidence = 2/2 = 100\%$
$I1 \Rightarrow \{I2, I5\}$,	$confidence = 2/6 = 33\%$
$I2 \Rightarrow \{I1, I5\}$,	$confidence = 2/7 = 29\%$
$I5 \Rightarrow \{I1, I2\}$,	$confidence = 2/2 = 100\%$

If the minimum confidence threshold is, say, 70%, then only the second, third, and last rules are output, because these are the only ones generated that are strong. Note that, unlike conventional classification rules, association rules can contain more than one conjunct in the right side of the rule. ■

Pattern - Growth approach for Mining Frequent Patterns

- The FP - Growth Approach
 - Depth-first search
 - Avoid explicit candidate generation
- Adopts divide - and - conquer strategy
- First, it compresses the database representing frequent items into a **frequent pattern tree**, or **FP-tree**, which retains the itemset association information.
- It then divides the compressed database into a set of *conditional databases* (a special kind of projected database), each associated with one frequent item or “pattern fragment,” and mines each database separately.
- For each “pattern fragment,” only its associated data sets need to be examined.
- Therefore, this approach may substantially reduce the size of the data sets to be searched, along with the “growth” of patterns being examined.

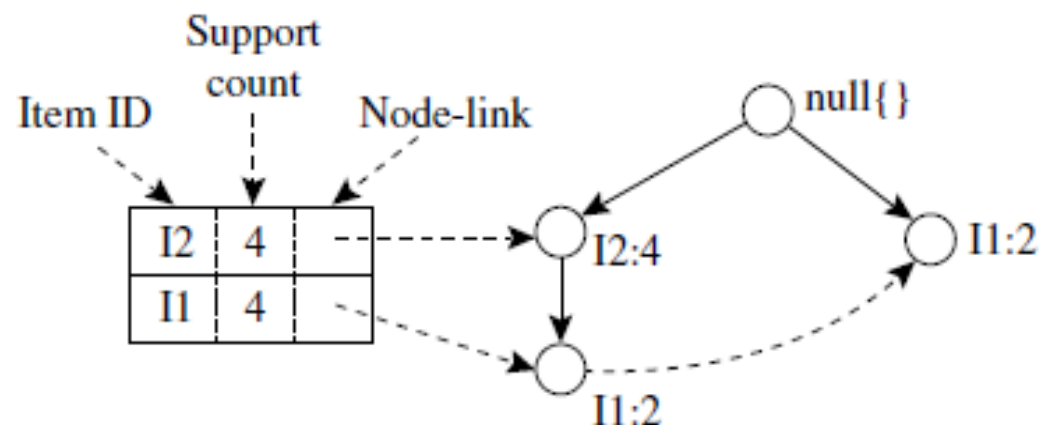
Example



An FP-tree registers compressed, frequent pattern information.

Mining the FP-tree by Creating Conditional (Sub-)Pattern Bases

Item	Conditional Pattern Base	Conditional FP-tree	Frequent Patterns Generated
I5	{I2, I1: 1}, {I2, I1, I3: 1}	$\langle I2: 2, I1: 2 \rangle$	{I2, I5: 2}, {I1, I5: 2}, {I2, I1, I5: 2}
I4	{I2, I1: 1}, {I2: 1}	$\langle I2: 2 \rangle$	{I2, I4: 2}
I3	{I2, I1: 2}, {I2: 2}, {I1: 2}	$\langle I2: 4, I1: 2 \rangle, \langle I1: 2 \rangle$	{I2, I3: 4}, {I1, I3: 4}, {I2, I1, I3: 2}
I1	{I2: 4}	$\langle I2: 4 \rangle$	{I2, I1: 4}



The conditional FP-tree associated with the conditional node I3.

T

...

U