Business Intelligence

G. Mudare

Mining Frequent Patterns,Associations

Model

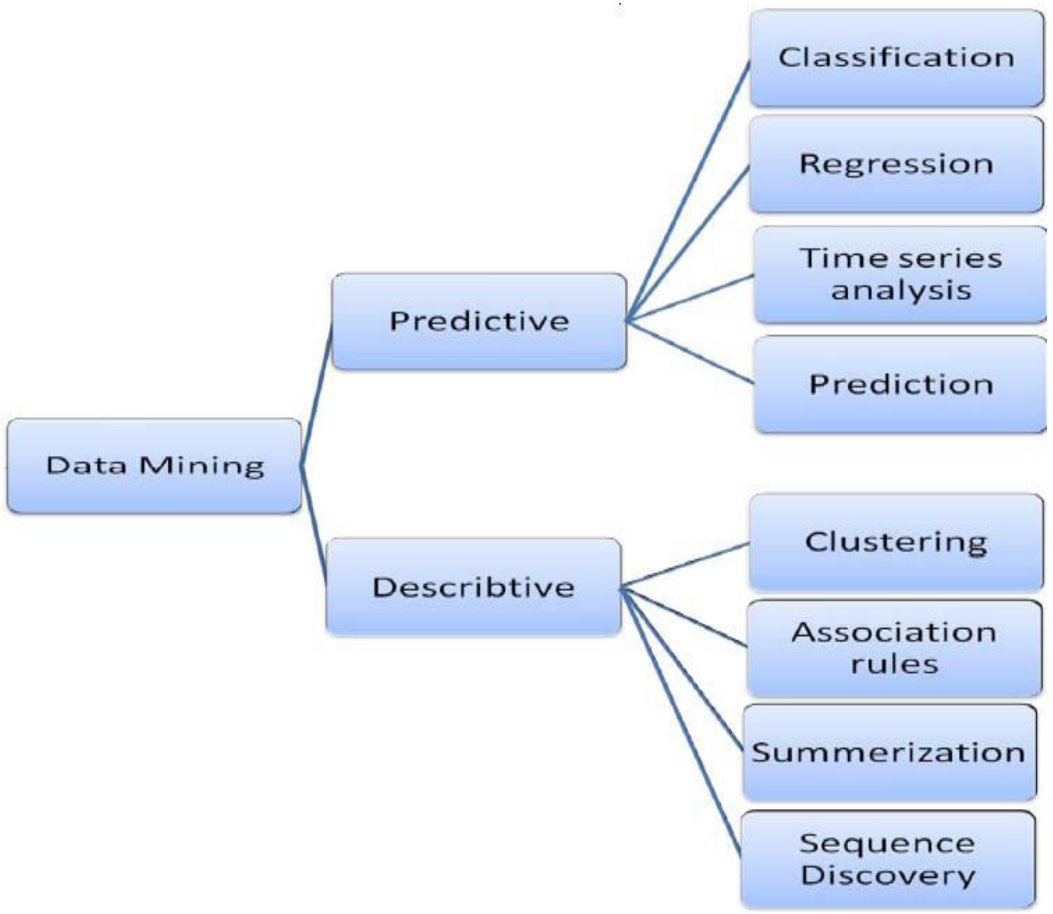
- A model is a high-level, global description of a data set and takes a large sample perspective. It may be :
  1. **descriptive—**summarizing the data in a convenient and concise way

  —or

  2. **inferential (predictive)** - allowing one to make some statement about the population from which the data were drawn or about likely future data values.

In contrast, a pattern is a local feature of the data, perhaps holding for only a few records or a few variables (or both).

Data Mining Models



Models

- ASSOCIATION
- ATTRIBUTE_IMPORTANCE
- CLASSIFICATION
- CLUSTERING
- FEATURE_EXTRACTION
- REGRESSION

ASSOCIATION and Algorithms

- Association is a descriptive mining function.
- An association model identifies relationships and the probability of their occurrence within a data set.
- Association models use the Apriori algorithm.
- The Association model is often associated with "market basket analysis", which is used to discover relationships or correlations in a set of items.
- It is widely used in data analysis for direct marketing, catalog design, and other business decision-making processes
- for example, "70% of the people who buy spaghetti, wine, and sauce also buy garlic bread."

Association models

- Association models capture the co-occurrence of items or events in large volumes of customer transaction data.
- Because of progress in bar-code technology, it is now possible for retail organizations to collect and store massive amounts of sales data.

Frequent patterns

- **Frequent patterns** are patterns that appear in a data set frequently.
  - For example, a set of items, such as milk and bread, that appear frequently together in a transaction data set is a frequent itemset.
- A **subsequence,** such as buying first a PC, then a digital camera, and then a memory card, if it occurs frequently in a shopping history database, is a **(*frequent*) *sequential pattern*.**
- Finding such frequent patterns plays an essential role in mining associations, correlations, and many other interesting relationships among data.

Frequent pattern Mining

- Market basket analysis is the earliest form of frequent pattern mining for association rules.
- Frequent itemset mining leads to the discovery of associations and correlations among items in large transactional or relational data sets.
- Market basket Analysis analyzes customer buying habits by finding associations between the different items that customers place in their "shopping baskets"

Association Rules

- If we think of the universe as the set of items available at the store, then each item has a Boolean variable representing the presence or absence of that item in a given basket.
- Each basket can then be represented by a Boolean vector of values assigned to these variables.
- The Boolean vectors can be analysed for buying patterns that reflect items that are frequently *associated* or purchased together
- These patterns can be represented in the form of association rules.

Association Rules

- Eg customers who purchase computers also tend to buy antivirus software at the same time is represented as:
  - ***Computer'' → antivirus software [support = 2%; confidence = 60%] Or buys(X, "computer") →buys(X, "antivirus software")***
- The support and confidence are two measures of rule interestingness.
- They respectively reflect the usefulness and certainty of discovered rules.
- A support of 2% for Association Rule means that 2% of all the transactions under analysis show that computer and antivirus software are purchased together.
- A confidence of 60% means that 60% of the customers who purchased a computer also bought the software

Association Rules

- Association rules are considered interesting if they satisfy both a minimum support threshold and a minimum confidence threshold.
- Such thresholds can be set by users or domain experts.
- Additional analysis can be performed to uncover interesting statistical correlations between associated items.
- Thus the problem of mining association rules can be reduced to that of mining frequent item sets.

Association rule

- In general, association rule mining can be viewed as a two-step process:
  1. **Find all frequent itemsets:** By definition, each of these itemsets will occur at least as frequently as a predetermined minimum support count, min sup.
  2. **Generate strong association rules from the frequent itemsets:** By definition, these rules must satisfy minimum support and minimum confidence.

  Additional interestingness measures can be applied for the discovery of correlation relationships between associated items

Challenges

- A major challenge in mining frequent itemsets from a large data set is the fact that such mining often generates a huge number of itemsets satisfying the minimum support (min sup) threshold, especially when min sup is set low.
- If a rule references two or more dimensions, such as the dimensions age, income, and buys, then it is a multidimensional association rule.

$$age(X, \text{“30…39”}) \wedge income(X, \text{“42K…48K”}) \Rightarrow buys(X, \text{“high resolution TV”})$$

Business Intelligence

G. Mudare

Apriori

Apriori

- Apriori is a seminal algorithm proposed by R. Agrawal and R. Srikant in 1994 for mining frequent itemsets for Boolean association rules.
- The name of the algorithm is based on the fact that the algorithm uses prior knowledge of frequent itemset properties.
- Apriori employs an iterative approach known as a 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 L1.Next,
- L1 is used to find L2 ,the set of frequent 2-item sets, 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.

- To improve the efficiency of the level-wise generation of frequent itemsets, an important property called the Apriori property:
  - **Apriori property:** *All nonempty subsets of a frequent itemset must also be frequent.*
    - The Apriori property is based on the following observation.
      - By definition, if an itemset I does not satisfy the minimum support threshold, *min sup*, then I is not frequent; i.e. $P(I) < min\ sup$.
    - 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 **antimonotone** 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 **antimonotone** because the property is monotonic in the context of failing a test.

How is the Apriori property used in the algorithm

A two-step process is followed, consisting

1. **join step** ->To a set of candidate k-itemsets is generated by joining the set with itself.
2. **Prune step-A** scan of the database to determine the count of each candidate in the set that would result in the determination all candidates having a count no less than the minimum support count and remove such from the set.

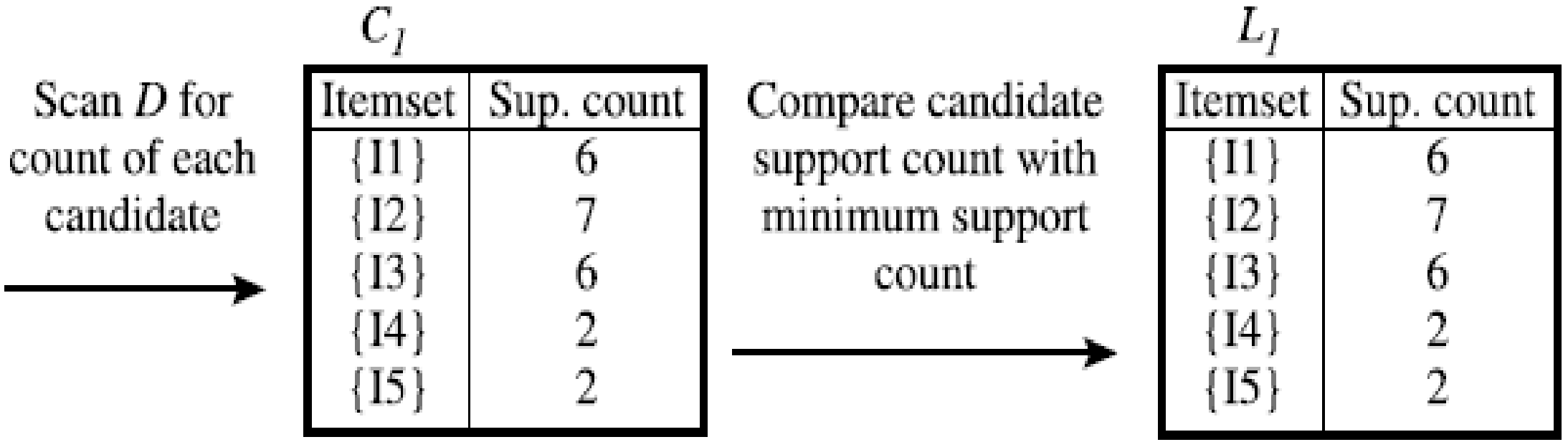
Apriori
(book example)

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 by example

- There are nine transactions in this database, that is, |D|= 9.
- In the first iteration of the algorithm, each item is a member of the set of candidate
- 1-itemsets, C1.
- The algorithm simply scans all of the transactions in order to count the number of occurrences of each item.

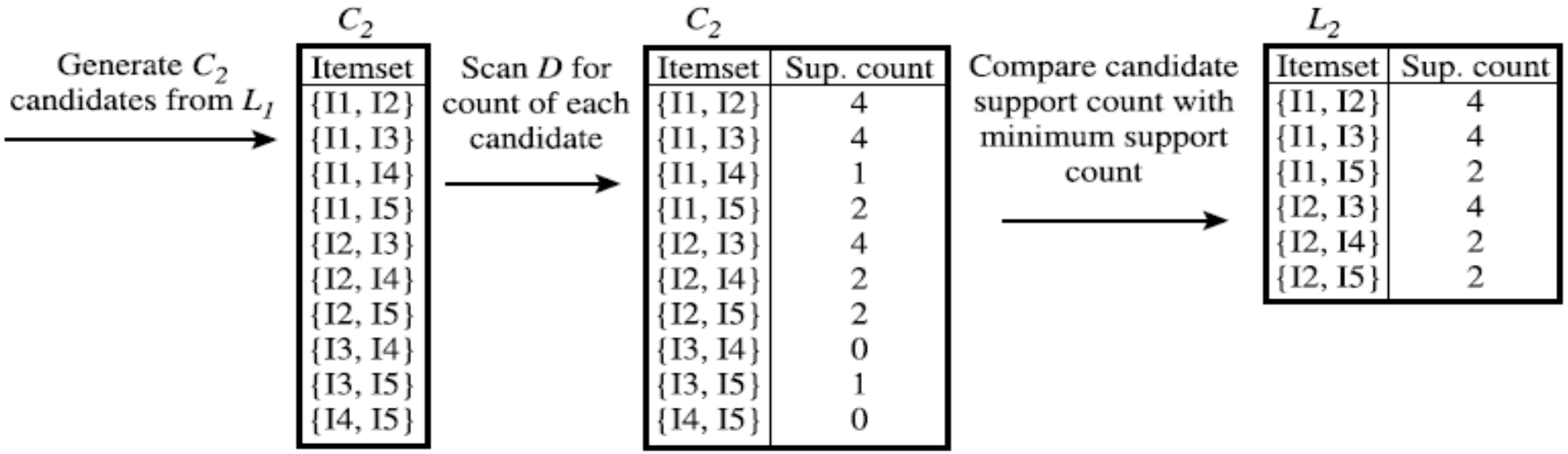
Apriori by example



Suppose that the minimum support count required is 2, that is, min_sup = 2. In our example, all of the Candidates in C1 satisfy minimum support.

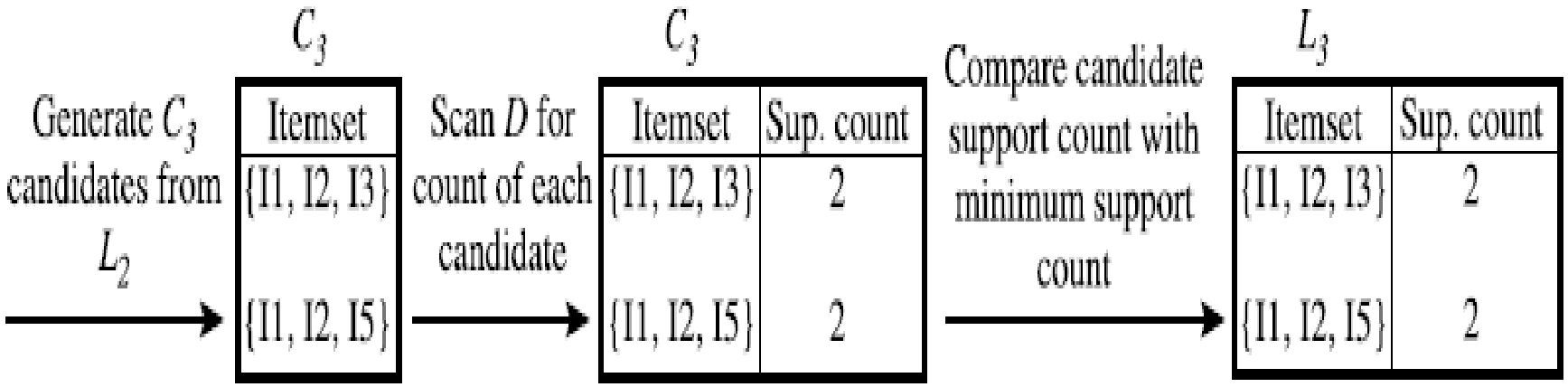
Apriori by example

- To discover the set of frequent 2-itemsets, L2, the algorithm uses the join L1 on L1 to generate a candidate set of 2-itemsets
- Next, the transactions in D are scanned and the support count of each candidate itemset in C2 is accumulated



Apriori by example

- determine L3



procedure apriori

```
procedure apriori_gen(L_{k-1}:frequent (k-1)-itemsets)
(1)     for each itemset l_1 ∈ L_{k-1}
(2)         for each itemset l_2 ∈ L_{k-1}
(3)             if (l_1[1] = l_2[1]) ∧ (l_1[2] = l_2[2]) ∧ ... ∧ (l_1[k-2] = l_2[k-2]) ∧ (l_1[k-1] < l_2[k-1]) then {
(4)                 c = l_1 ⋈ l_2; // join step: generate candidates
(5)                 if has_infrequent_subset(c, L_{k-1}) then
(6)                     delete c; // prune step: remove unfruitful candidate
(7)                 else add c to C_k;
(8)             }
(9)     return C_k;
```

Generating Association Rules from Frequent Itemsets

- Once the frequent itemsets from transactions in a database D have been found, it is straightforward to generate strong association rules from them (where strong association rules satisfy both minimum support and minimum confidence).
- Use the Equation

$$confidence(A \Rightarrow B) = P(B|A) = \frac{support\_count(A \cup B)}{support\_count(A)}.$$

The conditional probability is expressed in terms of itemset support count, where $support\ count(AUB)$ is the number of transactions containing the itemsets AUB, and $support\ count(A)$ is the number of transactions containing the itemset A.

Generating association rules

- For each frequent itemset l, generate all nonempty subsets of l.
- For every nonempty subset s of l, output the rule $(s\text{-}l) \rightarrow s$)"
- If

$$\frac{support\_count(l)}{support\_count(s)} \geq min\_conf,$$

  - where *min_conf* is the minimum confidence threshold.

Example

- Suppose the data contain the frequent itemset
- $l = \{I_1, I_2, I_5\}$.
- What are the association rules that can be generated from l? The nonempty subsets of l are $\{I_1, I_2\}$, $\{I_1, I_5\}$, $\{I_2, I_5\}$, $\{I_1\}$, $\{I_2\}$, and $\{I_5\}$. The resulting association rules are as shown below, each listed with its confidence:

Confidence

$$I1 \wedge I2 \Rightarrow I5,$$
$$I1 \wedge I5 \Rightarrow I2,$$
$$I2 \wedge I5 \Rightarrow I1,$$
$$I1 \Rightarrow I2 \wedge I5,$$
$$I2 \Rightarrow I1 \wedge I5,$$
$$I5 \Rightarrow I1 \wedge I2,$$

Confidence Example

BELGIUM CAMPUS iTversity
It's the way we're wired

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

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

If the minimum confidence threshold is, say, 70%, then only the second, third, and last rules above are output, because these are the only ones generated that are strong.