Association Rule Mining and Classification

Prof. Jayanand

Contents

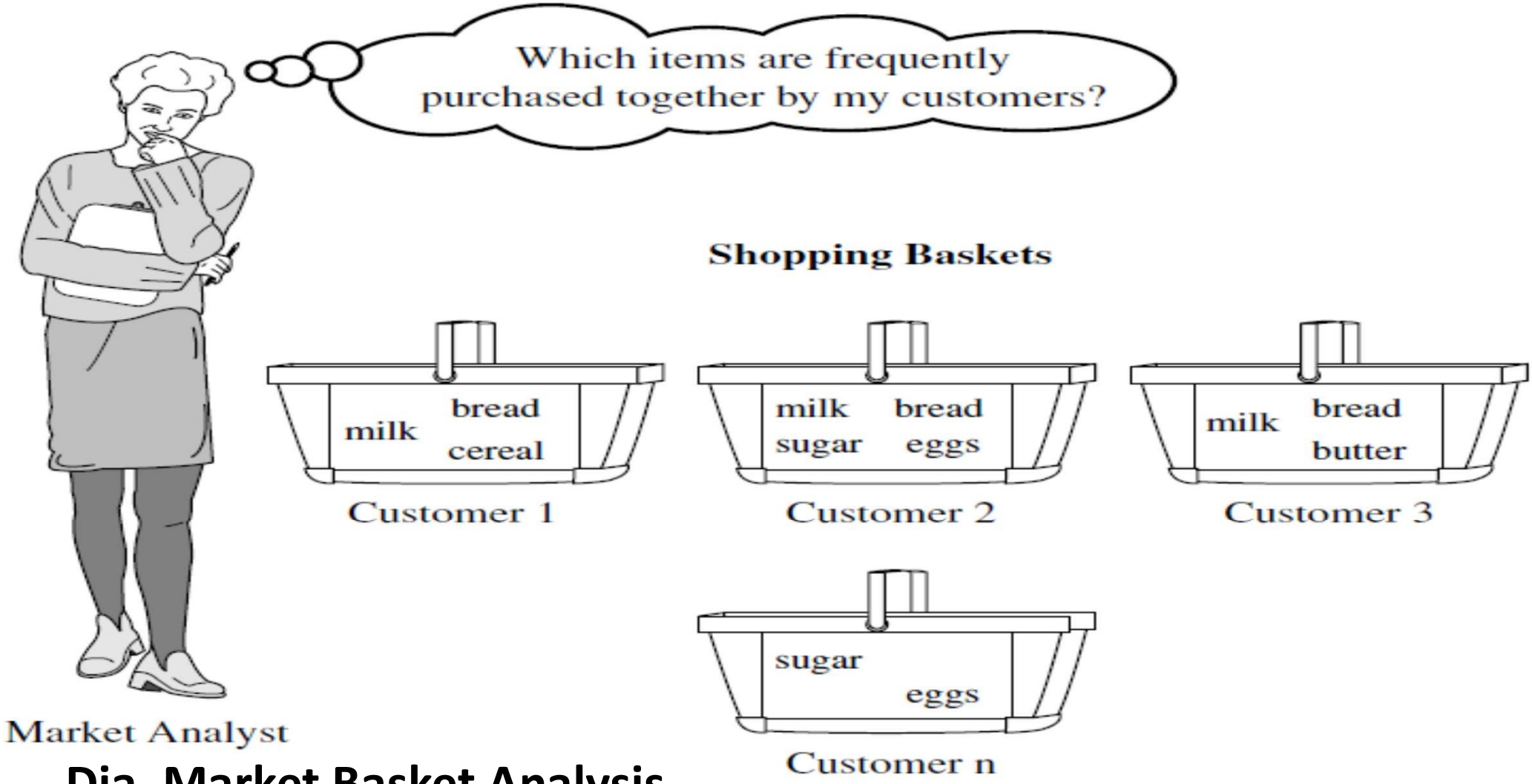
1. Mining Frequent Patterns,
2. Associations and Correlations,
3. Mining Methods,
4. Mining various Kinds of Association Rules,
5. Correlation Analysis,
6. Constraint Based Association Mining,

7. Classification and Prediction,
    1. Basic Concepts,
8. Decision Tree Induction,
9. Bayesian Classification,
10. Rule Based Classification,
11. Support Vector Machines,
12. Regression Models

1. Mining Frequent Patterns

- Frequent patterns are patterns (such as itemsets, subsequences, or substructures) 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.
- A substructure can refer to different structural forms, such as subgraphs, subtrees, or sublattices, which may be combined with itemsets or subsequences. If a substructure occurs frequently, it is called a (frequent) structured pattern.

- Frequent itemset mining leads to the discovery of associations and correlations among items in large transactional or relational data sets.
- A typical example of frequent itemset mining is **market basket analysis**. This process analyzes customer buying habits by finding associations between the different items that customers place in their “shopping baskets”



Dia. Market Basket Analysis

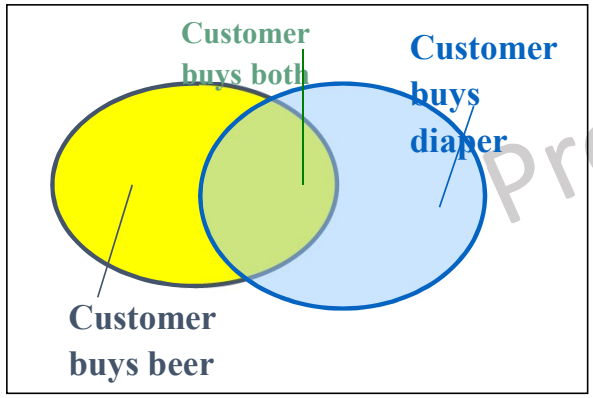
- frequent patterns can be represented in the form of association rules. For example, the information that customers who purchase computers also tend to buy antivirus software at the same time is represented in Association Rule.
  - *Computer* → *antivirus software* [*support* = 2%; *confidence* = 60%]
- Rule 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.
- Typically, association rules are considered interesting if they satisfy both a minimum support threshold and a minimum confidence threshold.

Frequent Itemsets, Closed Itemsets, and Association Rules

- A set of items is referred to as an itemset.
- An itemset that contains *k* items is a *k*-itemset.
  - The set {*computer, antivirus_software}* is a 2-itemset.
- The occurrence frequency of an itemset is the number of transactions that contain the itemset. This is also known, simply, as the frequency, support count, or count of the itemset.

- itemset: A set of one or more items
- k-itemset $X = \{x_1, \ldots, x_k\}$
- *(absolute) support*, or, *support count* of X: Frequency or occurrence of an itemset X
- *(relative) support*, *s*, is the fraction of transactions that contains X (i.e., the probability that a transaction contains X)
- An itemset X is *frequent* if X's support is no less than a *minsup* threshold

Tid	Items bought
10	Beer, Nuts, Diaper
20	Beer, Coffee, Diaper
30	Beer, Diaper, Eggs
40	Nuts, Eggs, Milk
50	Nuts, Coffee, Diaper, Eggs, Milk



- Find all the rules *X* → *Y* with minimum support and confidence
  - support, *s*, probability that a transaction contains X ∪ Y
  - confidence, *c*, conditional probability that a transaction having X also contains *Y*

Let minsup = 50%, minconf = 50%

Freq. Pat.: Beer:3, Nuts:3, Diaper:4, Eggs:3, {Beer, Diaper}:3

- Association rules: (many more!)
  - *Beer* → *Diaper* (60%, 100%)
  - *Diaper* → *Beer* (60%, 75%)

Frequent Pattern Mining: classified in various ways

- Based on the *completeness* of patterns to be mined:
- Based on the *levels of abstraction* involved in the rule set:
  - can find rules at differing levels of abstraction.
  - Computer, Laptop_computer; Printer, HP_Printer
- Based on the *number of data dimensions* involved in the rule:
  - single-dimensional association rule:
    - *buys*(*X*, "*computer*")→*buys*(*X*, "*antivirus software*")
  - A multidimensional association rule:
    - *age*(*X*, "30 . . . 39")^*income*(*X*, "42*K*. . . 48*K*")→*buys*(*X*, "*high resolution TV*")

- Based on the *types of values* handled in the rule:
  - Boolean association rule:
    - If a rule involves associations between the presence or absence of items
  - quantitative association rule:
    - a rule describes associations between quantitative items or attributes
- Based on the *kinds of rules* to be mined
  - Association rules
  - correlation rules: uncover statistical correlations
- Based on the *kinds of patterns* to be mined
  - frequent itemset mining
  - Sequential pattern mining
  - Structured pattern mining

3. Mining Methods:
3.1 The Apriori Algorithm: Finding Frequent Itemsets Using Candidate Generation

- Apriori, the basic algorithm for finding frequent itemsets.
- Apriori is a important algorithm proposed by R. Agrawal and R. Srikant in 1994 for mining frequent itemsets for Boolean association rules.

- 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 *L*1.Next, *L*1 is used to find *L*2, the set of frequent 2-itemsets, which is used to find *L*3, 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, presented below, is used to reduce the search space.
  - Apriori property: *All nonempty subsets of a frequent itemset must also be frequent.*

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; 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 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?"

- let us look at how L_{k-1} is used to find L_k for *k*>= 2.
- A two-step process is followed, consisting of join and prune actions.

The join step:

- To find L_k, a set of candidate k-itemsets is generated by joining L_{k-1} with itself.
- This set of candidates is denoted C_k. Let l_1 and l_2 be itemsets in L_{k-1}

The prune step

- C_k is a superset of L_k, that is, its members may or may not be frequent, but all of the frequent *k*-itemsets are included in C_k .
- A scan of the database to determine the count of each candidate in C_k would result in the determination of L_k.
- To reduce the size of C_k , the Apriori property is used as follows.
- Any (*k*-1)-itemset that is not frequent cannot be a subset of a frequent *k*-itemset.
- Hence, if any (*k*-1)-subset of a candidate *k*-itemset is not in L_{k-1}, then the candidate cannot be frequent either and so can be removed from C_k.

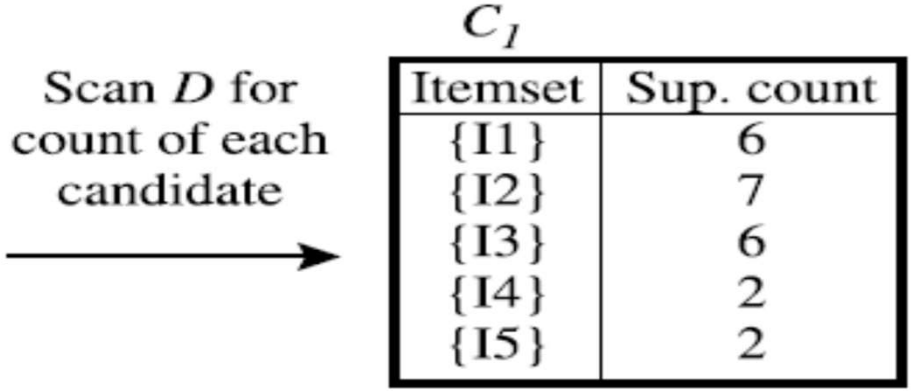
Example: (Table 4.1)

Transactional data for an *AllElectronics* branch.

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

- There are nine transactions in this database, that is, |D|=9.
- Use Apriori algorithm for finding frequent itemsets in *D*.

1. In the first iteration of the algorithm, each item is a member of the set of candidate 1-itemsets, *C*1. The algorithm simply scans all of the transactions in order to count the number of occurrences of each item.



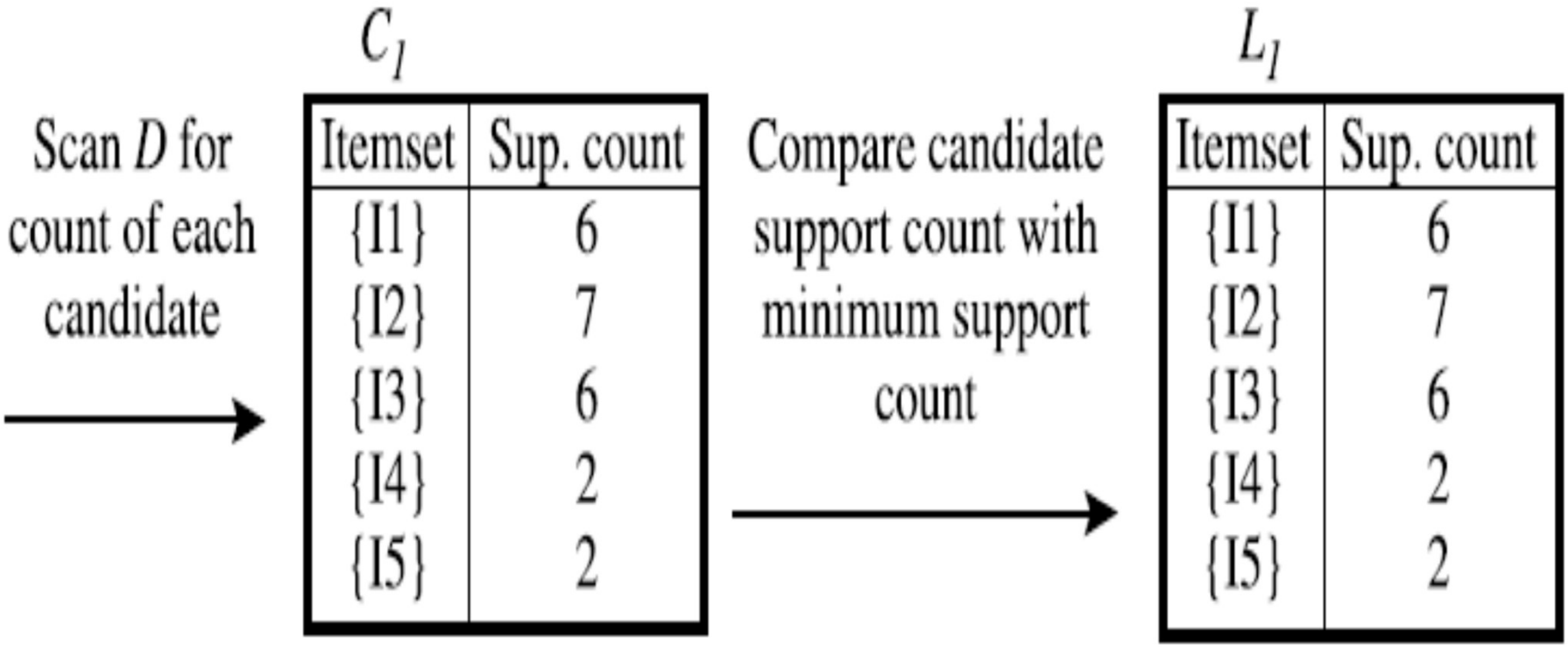
C_1

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

Step:2

2. Suppose that the minimum support count required is 2, that is, *min_sup* = 2. (Here, we are referring to *absolute* support because we are using a support count. The corresponding relative support is 2/9 = 22%). The set of frequent 1-itemsets, *L*1, can then be determined. It consists of the candidate 1-itemsets satisfying minimum support.

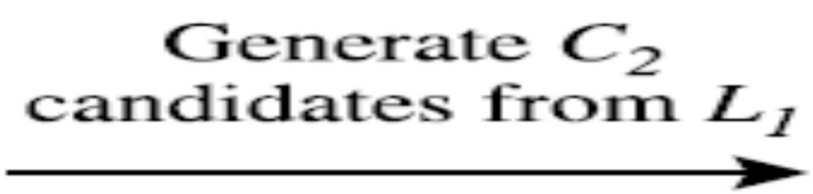
- In this example, all of the candidates in *C*1 satisfy minimum support.


C_1
Scan D for count of each candidate
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

3.

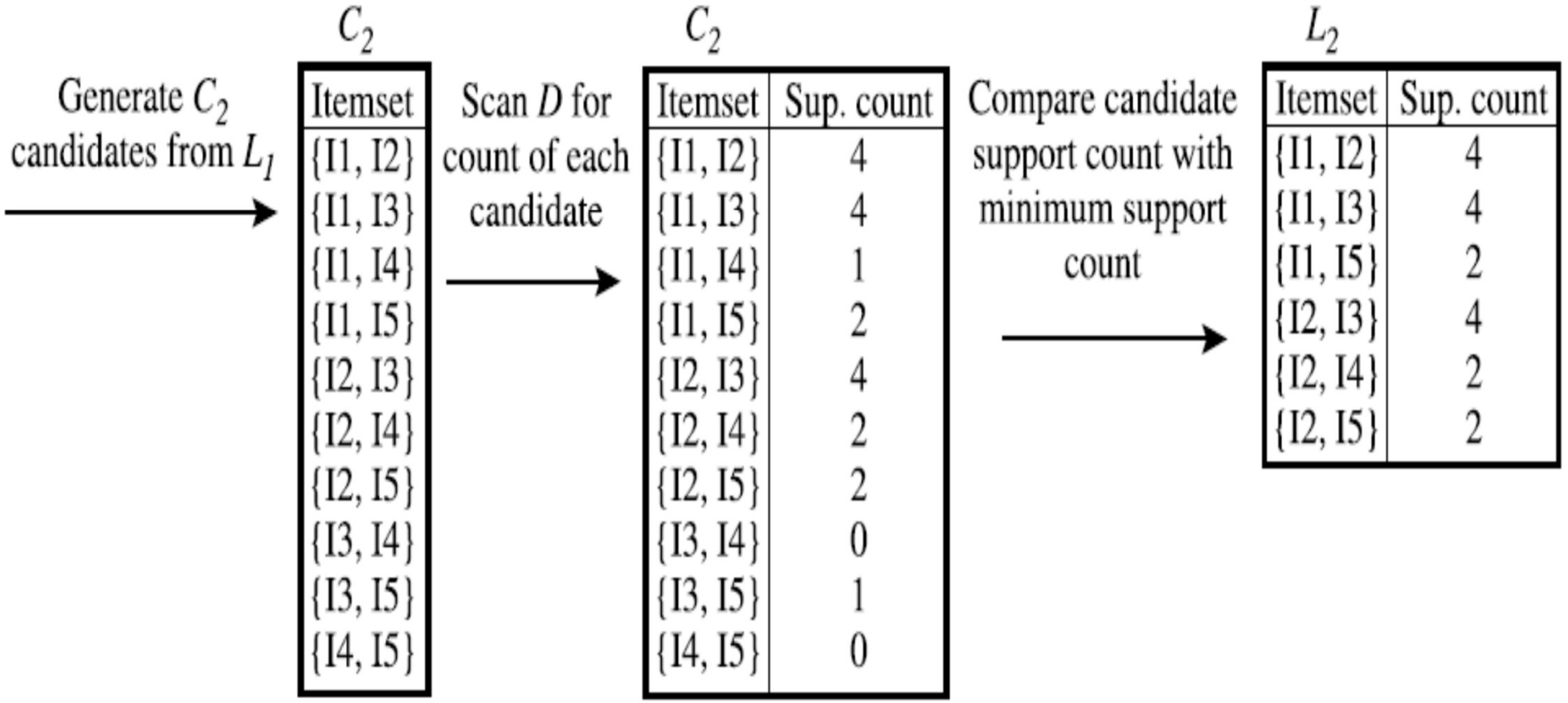
- To discover the set of frequent 2-itemsets, L_2, the algorithm uses the join $L1$ X $L1$ to generate a candidate set of 2-itemsets, $C2$.
- $C2$ consists of 2-itemsets.
- Note that no candidates are removed from$C2$ during the prune step because each subset of the candidates is also frequent.

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}


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

4. Next, the transactions in *D* are scanned and the support count of each candidate itemset in*C*2 is accumulated, as shown in above fig.

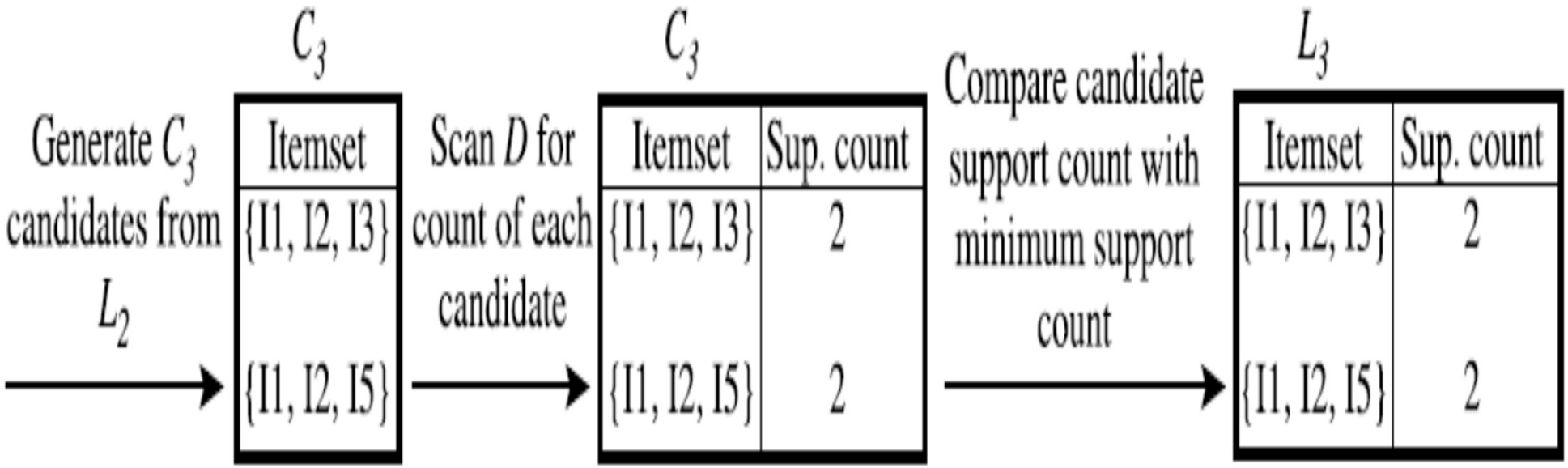
5. The set of frequent 2-itemsets, *L*2, is then determined, consisting of those candidate 2-itemsets in *C*2 having minimum support.

6. The generation of the set of candidate 3-itemsets,*C*3,

- From the join step, we first get $C3 = L2 \text{ X } L2 =$

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

- Based on the Apriori property that all subsets of a frequent item set must also be frequent, we can determine that the four latter candidates cannot possibly be frequent.
- We therefore remove them from $C3$, thereby saving the effort of unnecessarily obtaining their counts during the subsequent scan of D to determine $L3$.


C_3
Generate C_3 candidates from L_2
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

7. The transactions in *D* are scanned in order to determine *L*3, consisting of those candidate 3-itemsets in *C*3 having minimum support.

8. The algorithm uses *L*3 X *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 because its subset {{I2, I3,I5}} is not frequent. Thus, *C*4 = Φ, and the algorithm terminates, having found all of the frequent itemsets.

3.2 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).
- This can be done using Equation for confidence:

$$confidence = (A \rightarrow B) = p(B|A) = \frac{support\_count(A \cup B)}{support\_count(A)}$$

- The conditional probability is expressed in terms of itemset support count, where *support count*($A \cup B$) is the number of transactions containing the itemsets $A \cup B$, and *support count*(A) is the number of transactions containing the itemset A. Based on this equation, association rules can be generated as follows:
- For each frequent itemset l, generate all nonempty subsets of l.
- For every nonempty subset s of l, output the rule "$s \Rightarrow (l - s)$" if

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

Whete min_conf is the minimum confidence threshold.

Example: Generating association rules

- Transactional data for *AllElectronics* shown in Table 4.1
- Suppose the data contain the frequent itemset *l* = {I1, I2, I5}. What are the association rules that can be generated from *l*?
- The nonempty subsets of *l* are {I1, I2}, {I1, I5}, {I2, I5}, {I1}, {I2}, and {I5}. The resulting association rules are as shown below, each listed with its confidence:
    - *I*1^*I*2⇒*I*5, *confidence* = 2/4 = 50%
    - *I*1^*I*5⇒*I*2, *confidence* = 2/2 = 100%
    - *I*2^*I*5⇒*I*1, *confidence* = 2/2 = 100%
    - *I*1⇒*I*2^*I*5, *confidence* = 2/6 = 33%
    - *I*2⇒*I*1^*I*5, *confidence* = 2/7 = 29%
    - *I*5⇒*I*1^*I*2, *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

4. Mining various Kinds of Association Rules

- Mining multilevel association rules, multidimensional association rules, and quantitative association rules.
- *Multilevel association rules* involve concepts at different levels of abstraction. (e.g., rules relating what a customer *buys* as well as the customer's *age*.)
- *Multidimensional association rules* involve more than one dimension or predicate.
- *Quantitative association rules* involve numeric attributes that have an implicit ordering among values (e.g., *age*).

5. Association Mining to Correlation Analysis

- Most association rule mining algorithms employ a support-confidence framework.
- Many rules so generated are still not interesting to the users.
- Some strong association rules can be uninteresting and misleading.

5.1 Example:

- Transactions at *AllElectronics* with respect to the purchase of computer games and videos. Let *game* refer to the transactions containing computer games, and *video* refer to those containing videos.
- Of the 10,000 transactions analyzed, the data show that 6,000 of the customer transactions included computer games, while 7,500 included videos, and 4,000 included both computer games and videos.
- Suppose that a data mining program for discovering association rules is run on the data, using a minimum support of, say, 30% and a minimum confidence of 60%
- *buys*(*X*, "*computer games*")→*buys*(*X*, "*videos*") [*support = 40%*, *confidence = 66%*]……rule 4.1

- Rule(4.1) is a strong association rule and would therefore be reported, since its support value of 4,000/10,000 =40% and confidence value of 4,000/6,000 =66% satisfy the minimum support and minimum confidence thresholds, respectively.
- However, Rule 4.1 is misleading because the probability of purchasing videos is 75%, which is even larger than 66%.
- In fact, computer games and videos are negatively associated because the purchase of one of these items actually decreases the likelihood of purchasing the other.
- Without fully understanding this phenomenon, we could easily make unwise business decisions based on Rule.

- The support and confidence measures are insufficient at filtering out uninteresting association rules.
- To tackle this weakness, a correlation measure can be used to supplement the support-confidence framework for association rules

5.2 From Association Analysis to Correlation Analysis

- *correlation rules:*
- $A \rightarrow B$ [*support*, *confidence*. *correlation*]. Rule(4.2)
- That is, a correlation rule is measured not only by its support and confidence but also by the correlation between itemsets *A* and *B*.
- There are many different correlation measures from which to choose.

- Lift is a simple correlation measure.
- The occurrence of itemset *A* is independent of the occurrence of itemset *B* if $P(A \cup B) = P(A)P(B)$; otherwise, itemsets *A* and *B* are dependent and correlated as events.
- This definition can easily be extended to more than two itemsets.
- The lift between the occurrence of *A* and *B* can be measured by computing:
- $lift(A, B) = \frac{P(A \cup B)}{P(A)P(B)}$ rule 4.3

- If the resulting value of Equation (4.3) is less than 1, then the occurrence of *A* is *negatively correlated with* the occurrence of *B*.
- If the resulting value is greater than 1, then *A* and *B* are *positively correlated*, meaning that the occurrence of one implies the occurrence of the other.
- If the resulting value is equal to 1, then *A* and *B* are *independent* and there is no correlation between them.

Lift Example:

- Let $\overline{game}$ refer to the transactions of Example 5.1 that do not contain computer games, and $\overline{video}$ refer to those that do not contain videos.

A 2 × 2 contingency table summarizing the transactions with respect to game and video purchases.

	game	$\overline{game}$	Σ_{row}
video	4,000	3,500	7,500
$\overline{video}$	2,000	500	2,500
Σ_{col}	6,000	4,000	10,000

- The probability of purchasing a computer game is $P(\{game\}) = 0:60$,
- The probability of purchasing a video is $P(\{videog\} = 0:75$,
- and the probability of purchasing both is $P(\{game; video\}) = 0:40$.
- By Equation (4.3), i.e. the lift of Rule (4.2) is:
  - $P(\{game, video\})=(P(\{game\})*P(\{video\})) = 0.40/(0.60*0.75) = 0.89$.
- Because this value is less than 1, there is a negative correlation between the occurrence of {*game}* and {*video}*

The second correlation measure is x^2(chi) measure

	game	$\overline{game}$	$\sum_{row}$
video	4,000 (4,500)	3,500 (3,000)	7,500
$\overline{video}$	2,000 (1,500)	500 (1,000)	2,500
$\sum_{col}$	6,000	4,000	10,000

$$\chi^2 = \Sigma \frac{(observed - expected)^2}{expected} = \frac{(4{,}000 - 4{,}500)^2}{4{,}500} + \frac{(3{,}500 - 3{,}000)^2}{3{,}000} +$$

$$\frac{(2{,}000 - 1{,}500)^2}{1{,}500} + \frac{(500 - 1{,}000)^2}{1{,}000} = 555.6.$$

- Because the x^2 (chi) value is greater than one, and the observed value of the slot (*game*, *video*) = 4,000, which is less than the expected value 4,500, *buying game* and *buying video* are *negatively correlated.*

3. all_confidence $$all\_conf(X) = \frac{sup(X)}{max\_item\_sup(X)} = \frac{sup(X)}{max\{sup(i_j)|\forall i_j \in X\}},$$

4. cosine $$cosine(A, B) = \frac{P(A \cup B)}{\sqrt{P(A) \times P(B)}} = \frac{sup(A \cup B)}{\sqrt{sup(A) \times sup(B)}}.$$

Constraint-Based Association Mining

- Assignment

Reference:

- Jiawei Han and Micheline Kamber, Data Mining: Concepts and Techniques, Second Edition, Elsevier Publication