Association Rule Mining: 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.
   - This step is computationally expensive
2. **Generate strong association rules from the frequent itemsets:**
   - By definition, these rules must satisfy minimum support and minimum confidence.
   - This step is computationally **in**expensive

Because of this, the overall performance is determined by step 1

Generating Association Rules from Frequent Patterns

- Recall That:

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

- Once we mined frequent patterns, 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 $\frac{support\_count(l)}{support\_count(s)} \geq min\_conf$, where min_conf is the minimum confidence threshold.

Because l is a frequent itemset, each rule automatically satisfies the minimum support requirement.

Example: Generating Association Rules

- Items **I1 to I5** in the data set
- There is a 3-itemset X = {I1, I2, I5}
- To generate association rules, list all non-empty subset:
  - {I1}, {I2}, {I5}, {I1, I2}, {I1, I5}, {I2, I5}
- Then for each subset s, output the rule
  - S => X - s
  - {I1} => {I2, I5} // from subset {I1}
  - {I1, I2} => {I5} // from subset {I1, I2}
  - and so on…
- Finally, calculate confidence.
  - For {I1} => {I2, I5}, confidence = 2/6 or 33%
  - For {I1, I2} => {I5}, confidence = 2/4 or 50%
- Therefore, if your minconf is set to 40%, only one of the rules will be considered in the output

Table 4.1 A transactional data set.

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

Pattern Discovery: Basic Concepts

- **Basic Concepts**
  - What Is Pattern Discovery? Why Is It Important?
  - Bäsic Concepts: Frequent Patterns and Association Rules
  - Compressed Representation: Closed Patterns and Max-Patterns

Challenge: There Are Too Many Frequent Patterns!

- A long pattern contains a combinatorial number of sub-patterns
- How many frequent itemsets does the following TDB_1 contain (minsup = 1)?
  - $TDB_{1:}$ T_1: $\{a_1, \ldots, a_{50}\}$; T_2: $\{a_1, \ldots, a_{100}\}$
  - Let's have a try

  1-itemsets: $\{a_1\}$: 2, $\{a_2\}$: 2, …, $\{a_{50}\}$: 2, $\{a_{51}\}$: 1, …, $\{a_{100}\}$: 1,

  2-itemsets: $\{a_1, a_2\}$: 2, …, $\{a_1, a_{50}\}$: 2, $\{a_1, a_{51}\}$: 1 …, …, $\{a_{99}, a_{100}\}$: 1,

  …, …, …, …

  99-itemsets: $\{a_1, a_2, \ldots, a_{99}\}$: 1, …, $\{a_2, a_3, \ldots, a_{100}\}$: 1

  100-itemset: $\{a_1, a_2, \ldots, a_{100}\}$: 1
- The total number of frequent itemsets:

$$\binom{100}{1} + \binom{100}{2} + \binom{100}{3} + \cdots + \binom{100}{100} = 2^{100} - 1$$

A too huge set for any one to compute or store!

Expressing Patterns in Compressed Form: Closed Patterns

- How to handle such a challenge?
- Solution 1: **Closed patterns**: A pattern (itemset) X is closed if X is *frequent*, and there exists *no super-pattern* Y ↄ X, *with the same support* as X
  - Let Transaction DB TDB_1: T_1: $\{a_1, ..., a_{50}\}$; T_2: $\{a_1, ..., a_{100}\}$
  - Suppose *minsup* = 1. How many closed patterns does TDB_1 contain?
    - Two: P_1: "$\{a_1, ..., a_{50}\}$: 2"; P_2: "$\{a_1, ..., a_{100}\}$: 1"
- Closed pattern is a lossless compression of frequent patterns
  - Reduces the # of patterns but does not lose the support information!
  - You will still be able to say: "$\{a_2, ..., a_{40}\}$: 2", "$\{a_5, a_{51}\}$: 1"

Expressing Patterns in Compressed Form: Max-Patterns

- Solution 2: **Max-patterns**: A pattern X is a max-pattern if X is frequent and there exists no frequent super-pattern Y ↄ X
- Difference from close-patterns?
  - **Do not care the real support of the sub-patterns of a max-pattern**
  - Let Transaction DB TDB_1: T_1: $\{a_1, ..., a_{50}\}$; T_2: $\{a_1, ..., a_{100}\}$
  - Suppose *minsup* = 1. How many max-patterns does TDB_1 contain?
    - One: P: "$\{a_1, ..., a_{100}\}$: 1"
- Max-pattern is a lossy compression!
  - We only know $\{a_1, ..., a_{40}\}$ is frequent
  - But we do not know the real support of $\{a_1, ..., a_{40}\}$, ..., any more!
- Thus in many applications, mining close-patterns is more desirable than mining max-patterns

Quiz

Given closed frequent itemsets:

C = { {a1, a2, …, a100}: 1; {a1, a2, …, a50}: 2 }

Is {a10, a20} frequent? Can we know its support?

Ans: Yes, support = 2 because it’s is a sub-pattern of the second pattern whose support is 2

Quiz

Given maximal frequent itemset:

$$M = \{\{a1, a2, \ldots, a100\}: 1\}$$

Is {a10, a20} frequent? Can we know its support?

Ans: No, because we don’t store the support information of sub-patterns in max patterns