Direct Hashing and Pruning (Park-Chen-Yu)

Database D

<table>
<thead>
<tr>
<th>HD</th>
<th>Items</th>
<th>C₁</th>
<th>L₁</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>1 3 4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>2 3 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>300</td>
<td>1 2 3 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>400</td>
<td>2 5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Scan D

Is there some "magic" way to reduce the size of C₂?

Rule Generation

• Given a frequent itemset L, find all non-empty subsets f ⊂ L such that f → L - f satisfies the minimum confidence requirement
  - If {A,B,C,D} is a frequent itemset, candidate rules:
    ABC → D, ABD → C, ACD → B, BCD → A,
    AB → C, AC → BD, AD → BC, BC → AD,
    BD → AC, CD → AB
• If |L| = n, then there are 2^n - 2 candidate association rules (ignoring L → ∅ and ∅ → L)

Rule Pruning

Lattice of rules

Low Confidence Rule

Pruned Rules

Rule Generation

• How to efficiently generate rules from frequent itemsets?
  - In general, confidence does not have an anti-monotone property
    c(ABC → D) can be larger or smaller than c(AB → D)
  - But confidence of rules generated from the same itemset has an anti-monotone property
    e.g., L = {A,B,C,D}:
    c(ABC → D) ≥ c(AB → CD) ≥ c(A → BCD)
  - Confidence is anti-monotone w.r.t. number of items on the RHS of the rule

Rule Generation

• Candidate rule is generated by merging two rules that share the same prefix in the rule consequent
  - join(CD → AB, BD → AC) would produce the candidate rule D → ABC
  - Prune rule D → ABC if its subset AD → BC does not have high confidence
Rule Generation Algorithm

Key fact:

Moving items from the antecedent to the consequent never changes support, and never increases confidence.

Algorithm

- For each itemset \( I \) with \( \text{minsup} \):
  - Find all \( \text{minconf} \) rules with a single consequent of the form \( (I - L_1) \Rightarrow L_1 \)
  - Guess candidate consequents \( C_k \) by appending items from \( I - L_{k-1} \) to \( L_{k-1} \)
  - Verify confidence of each rule \( I - C_k \Rightarrow C_k \) using known itemset support values

Algorithm to Generate Association Rules

Input:
- \( D \) //Database of transactions
- \( I \) //Items
- \( L \) //Large itemsets
- \( \alpha \) //Confidence

Output:
- \( R \) //Association Rules satisfying \( \alpha \) and \( \text{support} \)

ARGen Algorithm:
- \( R = \emptyset \)
- for each \( i \in I \) do
  - for each \( x \subseteq i \) such that \( x \neq \emptyset \) and \( x \neq i \) do
    - if \( \text{support}(x) \geq \alpha \) then
      - \( R = R \cup \{ x \Rightarrow (i - x) \} \)

Factors Affecting Complexity

- Choice of minimum support threshold
  - lowering support threshold results in more frequent itemsets
  - may increase number of candidates and max length of frequent itemsets
- Dimensionality (number of items) of the data set
  - more space is needed to store support count of each item
  - if number of frequent items also increases, both computation and I/O costs may also increase
- Size of database
  - since Apriori makes multiple passes, run time of algorithm may increase with number of transactions
- Average transaction width
  - transaction width increases with denser data sets
  - may increase max length of frequent itemsets

Compact Representation of Frequent Itemsets

- Some items are redundant because they have identical support as their supersets

Closed Itemset

- An itemset is closed if none of its immediate supersets has the same support as the itemset

Maximal Frequent Itemset

An itemset is maximal frequent if none of its immediate supersets is frequent
### Maximal vs. Closed Itemsets

<table>
<thead>
<tr>
<th>TID</th>
<th>Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ABC</td>
</tr>
<tr>
<td>2</td>
<td>ABCD</td>
</tr>
<tr>
<td>3</td>
<td>BCE</td>
</tr>
<tr>
<td>4</td>
<td>ACDE</td>
</tr>
<tr>
<td>5</td>
<td>DE</td>
</tr>
</tbody>
</table>

### Maximal vs. Closed Frequent Itemsets

Minimum support = 2

# Closed = 9

# Maximal = 4

### Subsequent Research on Association Rules

- Mining association rules from sequences
e.g. stocks with similar movements in stock prices,
grocery items bought over a sequence of visits, etc.

- Finding "interesting" rules
  - Low-support, high-correlation mining
- Efficiently handling long itemsets
- Integration with query optimizers
- Adjustments to handle dense/relational databases
- Apply constraints to further filter association rules